Organized by

Sponsored by

ISBN: 979-8-3315-2931-4

» December 20-22, 2024
Guangzhou, China

Conference Proceeding

Editor:
Zenghui Wang

ACAI 2024

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899491

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2024)

Compliant PDF Files
IEEE Catalog Number: CFP24ZZ8-ART
ISBN: 979-8-3315-2931-4

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899498

Preface

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2024), organized by Guangdong University of Technology, sponsored by IEEE, IEEE Guangzhou Section, Robotics Society of Singapore, Nanjing University, The Hong Kong University of Science and Technology (Guangzhou), China has been held successfully at Guangdong University of Technology during December 20-22, 2024. ACAI is an annual conference which explores the development and implications in the related fields of Algorithms, Computing and Artificial Intelligence an objective to present the novel and fundamental advancements. It also serves to foster communication among researchers and practitioners working in a wide variety of scientific areas with a common interest in improving Algorithms, Computing and Artificial Intelligence.

We extend our gratitude to all the authors who contributed their research work to this collection. Our thanks also go to the reviewers and committee members, whose expertise and dedication were crucial in upholding the high standard of the conference. Their combined efforts were instrumental in making ACAI 2024 a successful event. ACAI 2024 proceedings centers on collecting the most up-to-date, comprehensive, and worldwide state-of-art knowledge and research in the related fields of Algorithms, Computing and Artificial Intelligence. These papers represent the latest research findings and innovative ideas in the field of Algorithms, Computing and Artificial Intelligence, showcasing the depth and breadth of ongoing research efforts worldwide. The contributions have undergone rigorous peer review to ensure their academic and practical value, and they have not been published elsewhere in academic journals or conferences.

We hope that the proceedings of this conference will serve as a valuable resource for researchers, practitioners, and policymakers in the field of Algorithms, Computing and Artificial Intelligence, inspiring further innovations and collaborations in this exciting and rapidly evolving domain. We believe that the conference has provided an excellent platform for fruitful discussions and networking, offering numerous opportunities for future collaborations. The exchange of ideas and knowledge among participants has undoubtedly contributed to advancing research in these vital fields.

ACAI 2024 Committee

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI)

ACAI 2024 Committee

Conference Chairs

Prof. Lei CHEN (Fellow IEEE and ACM Distinguished Scientist), The Hong Kong University of Science and Technology (Guangzhou), China

Prof. James Tin Yau KWOK (Fellow IEEE), The Hong Kong University of Science and Technology, Hong Kong, China

Prof. Qiu Daowen, Sun Yat-sen University, China

Program Chairs

Prof. Yuriy S. Shmaliy (Fellow IEEE), Universidad de Guanajuato, Mexico

Prof. Maria Pia Fanti (Fellow IEEE), Polytechnic University of Bari, Italy

Prof. Chun-Yi Su, Guangdong University of Technology, China

Prof. Li Chen (Chair, IEEE Information Theory Society Guangzhou Chapter), Sun Yat-sen University, China

Organizing Committee Chair

Distinguished Prof. Wanyang Dai, Nanjing University, China

Assoc. Prof. Xie Ming, Nanyang Technological University, Singapore

Steering Chair

Assoc. Prof. Jing Zhu, Nanjing University of Aeronautics and Astronautics, China

Prof. Fernando G. Tinetti, National university of la plata, Argentina

Assoc. Prof. Qiang (Shawn) Cheng, University of Kentucky, USA

Publication Chair

Prof. Zenghui Wang, University of South Africa, South Africa

Prof. Patrick Siarry, Universite Paris-Est Creteil, France

Prof. Shahadat Hossain, University of Northern British Columbia, Canada

Publicity Chair

Prof. George Magoulas, University of London, Birkbeck College, UK

Assoc. Prof. David Li, University of Glasgow, UK

Assoc. Prof. Sansanee Auephanwiriyakul, Chiang Mai University, Thailand

Assoc. Prof. Te Li, Dalian University of Technology, China

International Technical Program Committee

Prof. Javier Gozalvez, Universidad Miguel Hernandez de Elche, Spain

Prof. Dulani Meedeniya, University of Moratuwa, Sri Lanka

Prof. Madya Dr. Ong Pauline, UTHM Community, Malaysia

Prof. Luisa Maria Arvide Cambra, University of Almeria, Spain

Prof. Badrul Hisham bin Ahmad, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899486

Prof. José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Prof. Fancesco Zirilli Sapienza, Universita Roma, Italy
Prof. Jamaludin Jalani, Universiti Tun Hussein Onn Malaysia, Malaysia
Prof. Ping He, Huazhong Agricultural University, China
Prof. Zeki Ayağ, Piri Reis University, Turkey
Prof. Muhammad Sarfraz, Kuwait University, Kuwait
Assoc. Prof. Pavel Loskot, ZJU-UIUC Institute, China
Assoc. Prof. Sathish Kumar Selvaperumal, Asia Pacific University of Technology and Innovation, Malaysia
Assoc. Prof. Ata Jahangir Moshayedi, Jiangxi University of Science and Technology, China
Assoc. Prof. Bruce Jo, Tennessee Tech University, USA
Assoc. Prof. Teresa A. Oliveira, Universidade Aberta and CEAUL, Portugal
Assoc. Prof. Jinpeng Chen, Beijing University of Posts and Telecommunications, China
Assoc. Prof. Waleed H. Abdulla, The University of Auckland, New Zealand
Assoc.Prof. Mariani Stefano, Politecnico di Milano, Italy
Assoc. Prof. Kasturi Vasudevan, Indian Institute of Technology Kanpur, India
Assoc. Prof. Giuseppe Carbone, University of Calabria, Italy
Assoc. Prof. Yap Hwa Jen, University of Malaya, Malaysia
Assist. Prof. Jayanta Debnath, Marshall University, USA
Assist. Prof. Muhammed Ali Aydin, Istanbul University, Turkey
Assist. Prof. Md Shakhawat Hossain, Independent University Bangladesh, Bangladesh
Assist. Prof. Zahid Akhtar, State University of New York Polytechnic Institute, USA
Assist. Prof. Jianfeng Ren, University of Nottingham, Ningbo, China
Assist. Prof. Rocco Pietrini, Università Politecnica delle Marche, Italy
Dr. Noorlin Mohd Ali, Universiti Malaysia Pahang, Malaysia
Dr. Tan Yi Fei, Multimedia University, Malaysia
Dr. Nachiappan Valliappan, Google, USA
Dr. Yulia Kumar, Kean University, USA
Dr. Guoxin Su, University of Wollongong (UOW), Australia
Dr. Olarik Surinta, Mahasarakham University, Thailand
Dr. Dickson K.W. Chiu, The University of Hong Kong, Hong Kong, China
Dr. Binh P. Nguyen, Victoria University of Wellington, New Zealand
Dr. Chao Ma, University of Science and Technology Beijing, China
Dr. Shuaiby Mohamed Shuaiby Ragab, Assiut University, Egypt
Dr. Nandana Kumara, Kotelawala Defence University, Sri Lanka
Dr. Koorosh Gharehbaghi, RMIT University, Australia
Dr. Mohd Aliff Afira Hj Sani, Universiti Kuala Lumpur, Malaysia

Table of Contents

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899547

WiFi Channel State Information-based Motion Detection Across Time-Domain

1st Qinhong Wang
School of Software
Nanchang Hangkong University
JiangXi, China
0009-0008-0479-7498

2nd Zhenhua Wu
School of Software
Nanchang Hangkong University
JiangXi, China
0009-0002-4096-5483

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899617

***Abstract*—Human presence is a basic requirement for realizing other wireless sensing tasks. With the development of wireless communication technology, WiFi is not only used as a widely used network access method, but also applied in various intelligent environment monitoring and human behavior recognition fields. The motion human detection technique based on WiFi Channel State Information (CSI) is becoming a research hotspot due to its non-invasive and high-precision advantages. However, most of the existing studies are limited to mixing and analyzing data collected within a short period of time, with less research in detection across time-domain. Because CSI data are subject to random changes due to temporal factors, this makes it difficult to accurately predict future data using the used data patterns. For this reason, we propose algorithms based on motion human detection across time-domain. We first perform outlier removal for CSI amplitude and then filter the signal using wavelet denoising. In order to improve the accuracy of motion human detection across time-domain, we use the average variance of sliding time window as the robust feature of motion human and combine it with other statistical features to build the feature set. Finally, an integrated machine learning model is utilized in order to determine the future indoor human states. Experimental results show that our method achieves an average accuracy of more than 98% in detecting motion human across time-domain.**

Keywords—WiFi, Channel State Information, passive sensing, ensemble learning

I. INTRODUCTION

With the rapid development in the fields of smart home[1], smart healthcare[2] and smart security[3], human body detection technology based on wireless signals has gradually become a research hotspot. Traditional human body detection methods usually rely on cameras, infrared sensors, or ultrasonic sensors, but these methods suffer from privacy leakage, high equipment cost, and complex deployment. In contrast, WiFi CSI-based human body detection technology has received widespread attention for its non-invasive, low-cost and easy-to-deploy advantages.

WiFi CSI can show the amplitude and phase changes of wireless signals during propagation, which are closely tied to environmental objects and human body movements. Analyzing CSI changes enables the detection of human body position and behavior.Although there has been notable progress in human detection techniques based on WiFi CSI, most studies have overlooked the issue of human detection across the time-domain. They mainly focus on short-term data analysis by mixing collected data and dividing it into training and test sets according to a certain ratio, without fully considering the impacts of environmental dynamic changes and the passage of time on CSI signals. FIMD[4] first proposed using a correlation matrix of correlation coefficients in the time-domain of each set of CSI within a sliding window for motion detection. Zhu[5] used the correlation variations on different subcarriers to extract the first-order difference of CSI feature vectors across different subcarriers for human body detection. Qian[6] was the first to use both phase and amplitude, extracting the maximum eigenvalue of the covariance matrix of both in the time dimension and combining it with Support Vector Machines (SVM) to detect a moving human body, demonstrating its sensitivity to human activities and robustness even when the target moves slowly. Pilot[7] constructed a fingerprint of CSI packets in the time-domain based on their correlation coefficient to set a threshold and determine if there is someone in the room. FDF-PIHD[8] constructs feature fingerprints by exploiting the correlation between subcarriers, compares the similarity between online and offline fingerprints using Euclidean distance to determine the state of the person in the scene, and introduces multi-antenna voting to improve the accuracy of the method. DeMan[9] inspired by[6], further uses the eigenvalues of the correlation matrix of amplitude and phase of CSI to detect moving targets without the need for tedious scene-specific calibration. MesaCantillo[10] uses time-domain features such as mean, standard deviation, root-mean-square, and the number of times a change occurs in the signal to detect whether a human intrusion has occurred.

All of the above work was studied based on multiple antennas and therefore more robust to noise. In contrast, using the ESP32 chip to achieve the same human detection effect as Intel 5300 with a small amount of data is our challenge. Researchers have also started to study it gradually. Natarajan et al.[11][12] performed a related study, the former performs feature extraction from CSI extracted from a link consisting of ESP32 and a commercial router, and uses Ensemble Learning models to categorize four different human activities (including motion, unoccupied, and stationary). The latter trains the Ensemble Learning model for device-less human motion detection using low-complexity feature sets derived from RSSI and CSI collected by ESP32. In addition to this, they[12] have also looked at how well the model predicts invisible data, i.e. across time domains, but not in depth. This is where we start to delve deeper into the model's effectiveness in detecting "unfamiliar" data.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

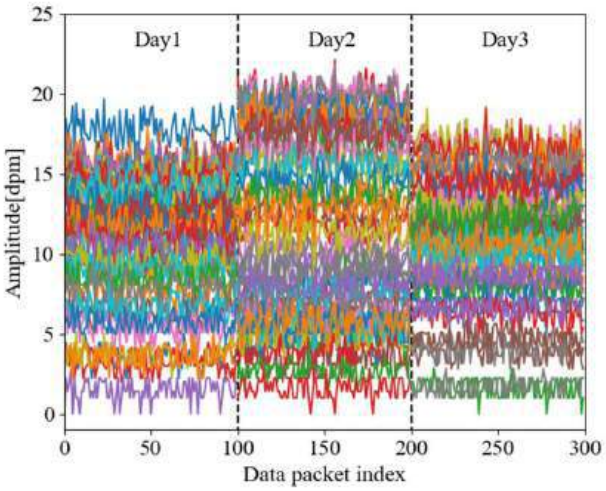


Figure 1 WiFi data in the same state on different days.

Since changes in the wireless environment such as furniture movement, addition or removal of devices, and changes in ambient temperature and humidity can significantly affect the CSI signals over time, as shown in figure 1. WiFi CSI-based human body detection across time-domain faces many challenges, such as the stability of the CSI signals, the robustness of the feature extraction, and the adaptability of the detection algorithms. Therefore, it is of great research significance and application value to explore the WiFi CSI-based motion human detection method across time-domain.

The main contributions of this study are as follows:

1. we applied outlier removal method with wavelet denoising preprocessing to obtain high quality usable CSI amplitude.

2. we design a sliding window approach for motion human feature extraction that overcomes the instability of CSI, ensures robustness and differentiation of features across time-domain, and finally uses an ensemble machine learning model to detect future motion humans.

3. The effectiveness and robustness of the proposed method in practical applications are proved through a large number of experiments, demonstrating its application prospects in the fields of smart home and security monitoring.

II. PRELIMINARY

A. Abbreviations and Acronyms

In wireless communication systems, CSI can be represented by a matrix. For a Multiple Input Multiple Output (MIMO) system with Nt transmitting antennas and Nr receiving antennas, the CSI matrix H describes the channel characteristics from the transmitting end to the receiving end. Assuming that the system has K subcarriers, the CSI can be expressed as a three-dimensional matrix of $N_r \times N_t \times K$. The CSI matrix H_k on each subcarrier can be expressed as:

$$H_k = \begin{bmatrix} h_{11}(k) & h_{12}(k) & \cdots & h_{1N_t}(k) \\ h_{21}(k) & h_{22}(k) & \cdots & h_{2N_t}(k) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1}(k) & h_{N_r2}(k) & \cdots & h_{N_rN_t}(k) \end{bmatrix} \quad (1)$$

where $h_{ij}(k)$ denotes the channel gain from the jth transmitting antenna to the ith receiving antenna on the kth subcarrier, including both amplitude and phase information. The channel gain $h_{ij}(k)$ can be further decomposed as:

$$h_{ij}(k) = |h_{ij}(k)|e^{j\angle h_{ij}(k)} \quad (2)$$

Here, $|h_{ij}(k)|$denotes the amplitude of the channel gain and $\angle h_{ij}(k)$ denotes the phase of the channel gain.In the presence of noise, the received signal Y can be expressed as:

$$Y = H \cdot X + N \quad (3)$$

where X is the transmit signal matrix, H is the CSI matrix, and N is the noise matrix. Since only a single WiFi link exists in the system, the one-dimensional array of CSI from the packet at moment t can be represented as:

$$CSI = [x_{1,t} \quad y_{1,t} \quad x_{2,t} \quad y_{2,t} \quad \cdots \quad x_{k,t} \quad y_{k,t}] \quad (4)$$

where $x_{k,t}$, $y_{k,t}$ are the real and imaginary parts of the complex estimate of the kth subcarrier at the moment t. Thus the amplitude of the kth subcarrier at moment t can be calculated by the following equations:

$$|h(k)| = \sqrt{x_{k,t}^2 + y_{k,t}^2} \quad (5)$$

B. Impact of training across time-domain

In this subsection, we analyze the impact of training across time-domain using three basic machine learning classification algorithms and ensemble machine learning models. To highlight this impact, we only used the variance of the CSI amplitude as the motion body detection feature. We mixed all the collected data without considering the time factor and then proportionally divided it into training and validation sets for prediction. In contrast, we also conducted across time-domain training experiments, where we trained the classifier using only the first day's data and then used future datasets for prediction.

The experimental results in the Table1 show that all evaluation metrics of across time-domain training are lower than those of no across time-domain training. This is because the variance features used are not robust enough to model CSI data of different time periods, resulting in poor prediction results when the model is faced with "unfamiliar" data, such as CSI data of environmental, human movement speed and power. However, this instability feature has better results when not training across time, which indicates that the two different training methods will lead to significant differences in prediction results. Therefore, it is necessary to further study the motion detection of across time-domain training.

TABLE I. IMPACT OF TRAINNING ACROSS TIME-DOMIAN

Classifier	Across time-domain				Not across time-domain			
	ACC	*Prec*	*Rec*	*F1*	*ACC*	*Prec*	*Rec*	*F1*
SVM	76.78%	79.37%	78.79%	76.76%	81.85%	84.43%	83.83%	81.84%
RF	89.28%	88.95%	89.45%	89.13%	98.02%	97.86%	98.14%	97.99%

XG Boost	87.00%	86.71%	87.33%	86.87%	94.74%	94.47%	95.14%	94.68%
Hard vote	86.75%	86.55%	87.26%	86.64%	94.90%	94.64%	95.35%	94.84%
Soft vote	86.85%	86.59%	87.26%	86.73%	96.67%	96.42%	96.93%	96.62%

III. PROPOSED METHOD

In this section, we describe the overall flow of the method, including the pre-processing of the data, the process of feature extraction on the data, and Ensemble Learning. As shown in Figure 2.

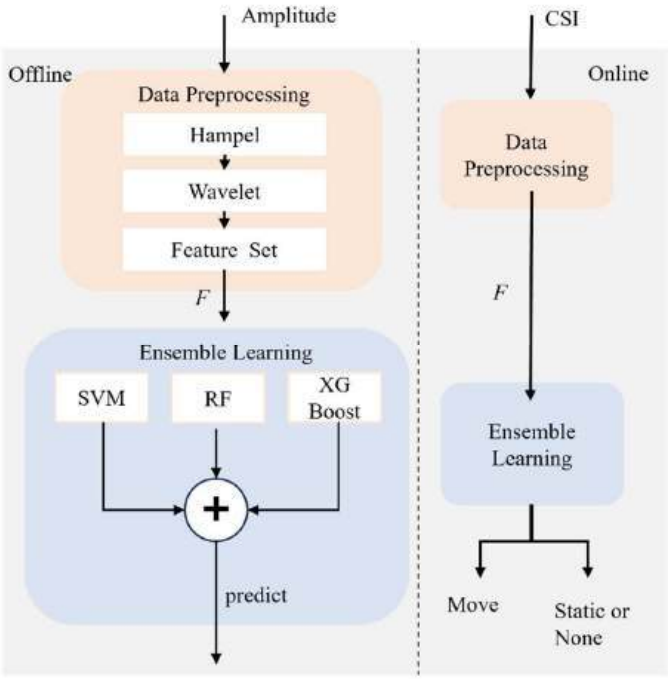


Figure 2 Proposed method.

A. Data preprocessing

Due to the fact that there may be multiple sources of interference in the WiFi environment that affect the quality of CSI data during the data acquisition process, it is necessary to pre-process the raw data before feature extraction. Therefore, the raw CSI data may contain a lot of noise, and it is necessary to preprocess the raw data before feature extraction. We use Hampel and Wavelet denoising to remove the anomalies and reduce the noise, which makes the data cleaner and more reliable.

B. Feature extraction

CSI signals in wireless channels are often non-stationary, i.e., the statistical characteristics of the signal (e.g., mean, variance) vary over time. This variation may be masked if the characteristics are computed directly on the whole signal. Although we have pre-processed the signal using denoising techniques, inevitably there is still some untreated noise. If there are still bursts of noise in the signal, computing features directly on the whole signal will be affected by these noises. In order to obtain robust features that can be used for motion human detection across time-domain, we design a sliding window variance feature extraction method. Specifically, we split the signal into several segments, each of which can be approximated as a smooth process, so that the features of each segment better reflect the fluctuating characteristics of the signal in that time period. At the same time, this method can reduce the influence of sudden noise on the overall feature calculation due to the existence of some segments.

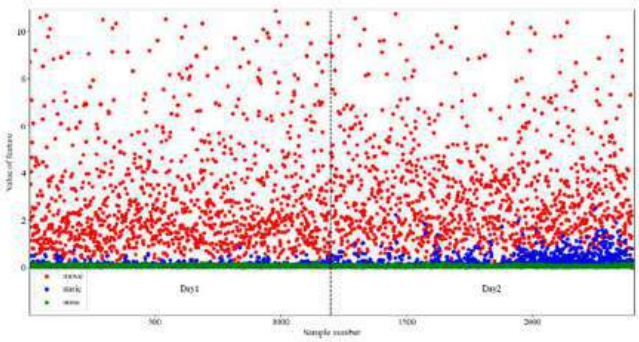


Figure 3 Before and after two-day data distribution without using sliding variance.

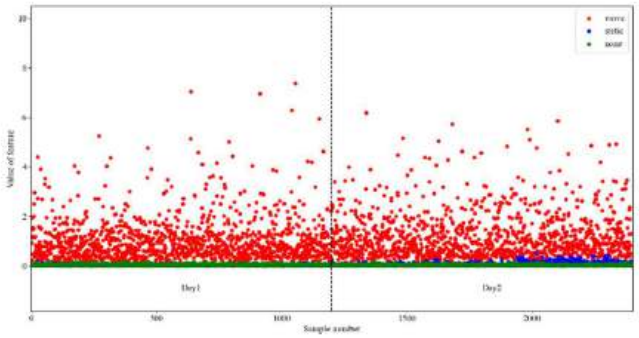


Figure 4 efore and after two-day data distribution with using sliding variance.

As shown in the Figure 3 and Figure 4, each point in the figure represents the variance of the first subcarrier containing 100 packets (i.e., 1 second) of samples. After calculating the variance by using the sliding window method, the data distributions of the two days before and after are relatively close to each other. It is clear that the eigenvalues computed by this method are more robust across time-domain and have smaller errors compared to the first processing method.

We fully utilized all 52 subcarriers and performed a sliding window for each subcarrier to calculate the variance to extract features. To further improve the accuracy of training across time-domain, we also extracted four statistical features, namely, interquartile range of amplitude, coefficient of variation, polar deviation and mean absolute deviation.

C. Ensemble Learning

Ensemble Learning offers significant advantages in machine learning by combining the outputs of multiple base learners to produce classification results with enhanced robustness and accuracy. In this paper, a novel two-stage Ensemble Learning model for detecting future indoor states is proposed.The first stage of the model encompasses the following base classifiers:

(1) SVM: SVM classifies the data by finding the optimal hyperplane to maximize the boundary distances between different classes, thus improving the accuracy of classification.

(2) Random Forest (RF): This model utilizes the integration of multiple decision trees, uses different subsamples to train the data, and averages the predictions of each tree to improve the overall classification performance.

(3) Extreme Gradient Boosting (XG Boost): it is an improved version of the Gradient Boosting (GB) algorithm, which improves computational efficiency and classification accuracy through parallel processing and optimization.

In the second stage, the model uses a weighted voting classifier as a meta-learner to synthesize the prediction results from each base classifier in the first stage. The weights of each base classifier are assigned based on its performance on the training set, ensuring that the top-performing classifiers have a greater impact on the final results.

IV. Experiment and Evaluation

A. Experimental setup

We evaluated the performance of the method in a typical meeting room (3.9m×7.6m) as shown in Figure 5. We selected ESP32 as the receiver. However, considering the possibility of antenna switching when connecting the ESP32 to a dual-antenna wireless router, resulting in unreliable experimental data, we chose a single-antenna commercial wireless router, the TP-Link TL-WDR5620, as the transmitter. To avoid severe signal blockage, we arranged the transmitter and receiver at the two ends of the meeting room table, 5.5 meters apart. We set the transmitter to operate in IEEE 802.11n AP mode with 2.4GHz and 20MHz bandwidth. CSI measurements were received at a rate of approximately 100 packets per second using ESP32-CSI-Tool.

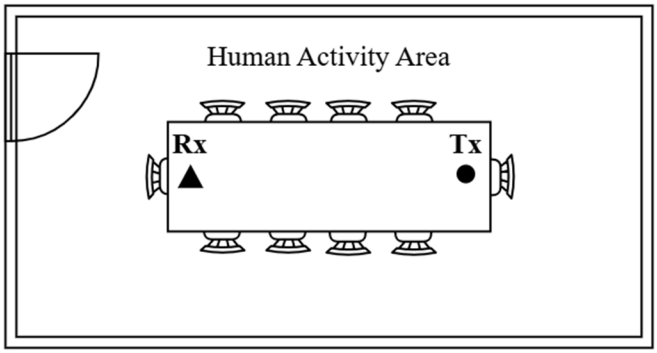


Figure 5 Data collection environment.

B. Dataset details

We collected a total of five datasets (one dataset per day, from D1 to D5), and each dataset contained three kinds of activities: (1) motion state data, i.e., the volunteers walked and moved in the activity area specified in the target room; (2) stationary state data, i.e., the volunteers were asked to stand; (3) empty room data, i.e., the volunteers were not present in the room. Each of the three states was collected for 45 minutes. We divided all the datasets collected on the first day into training and validation sets according to 8:2, and used the data collected on the last four days as a test set for evaluating the performance of this paper's method across time-domain. To verify the generalizability of the model to common situations such as changes in indoor room layouts, we collect datasets in the conference room without recovering environmental changes such as chair positions due to conference room activities, the presence of water cup items on the desktop, etc., in order to simulate the environmental changes that may occur in daily life, and we collect motion data with walking at different speeds of movement.

C. The impact of the number of sliding windows

We verify the validity of the sliding window variance. As shown in figure 6, when the length of the sample packet is 100, the number of sliding windows for calculating the variance reaches 50, and comparing with the results without cross-time-domain training, the accuracy is only reduced by 1.31 percentage points, the accuracy only decreased by 1.31 percentage points, and the accuracy gradually increased with the increase of the sliding window number. This indicates that the sliding window variance can better reflect the fluctuation characteristics of the signal in a specific time period, which makes it more robust in detecting the trained moving human body across time-domain.

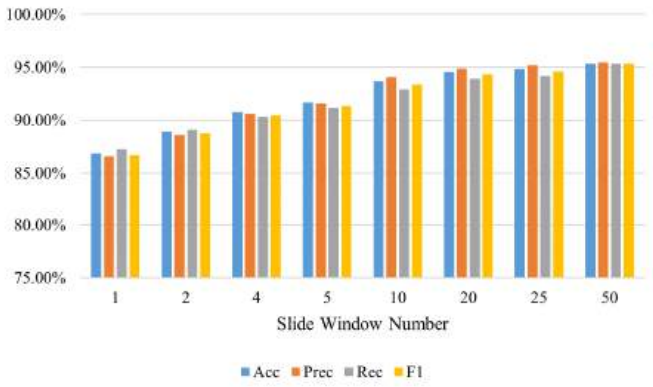


Figure 6 The impact of the number of sliding windows.

D. Overall results of methodology

In this subsection, we investigate and analyze the detection results of the moving human body across time-domain for each dataset, as shown in Table 2. In all the datasets, the accuracy of D3 and D4 is only about 97% compared to D2 and D5 which are close to 100%. This is due to the fact that when the D3 dataset was collected, there was a relevant activity in the meeting room on that day, which led to a change in the table layout of the meeting, while D4 was due to the rotation of the router antenna. The experimental results show that our method is resistant to some factors that impact the CSI data. In addition to that, since we were active at different speeds when we collected motion data (the slower the movement the less impact on CSI), it means that our method is also valid for recognizing the moving human body with slower movement speeds.

TABLE II. OVERALL RESULTS OF METHODOLOGY

	D2	D3	D4	D5	Avg
Acc	99.92%	97.67%	97.53%	99.26%	98.60%
Prec	99.93%	97.61%	97.91%	99.22%	98.67%
Rec	99.91%	97.64%	97.12%	99.27%	98.49%
F1	99.92%	97.62%	97.46%	99.24%	98.56%

E. Comparison experiment

In our method, we are based on a single antenna for motion human detection across time-domain, so we can't use related techniques such as phase difference for experimentation, FIMD, PADS and DeMan methods use the maximum eigenvalue of the CSI correlation matrix for motion detection, so we pick the representative PADS for comparison with our method. Secondly R-TTWD further extracts features from the correlation matrix for motion human detection in through-wall scenarios. The results of the specific comparison experiments are shown in Table 3. Our method achieves better performance than other current motion human detection methods.

TABLE III. COMPARISON EXPERIMENT WITH OTHER

Method	Move	No-move
PADS[6]	96.80%	94.24%
FDF-PIHD[8]	91.73%	88.78%
R-TTWD[5]	78.65%	72.36%
HAR[12]	83.30%	62.44%
Our	99.22%	97.61%

V. CONCLUSIONS

We present a method using WiFi CSI amplitude to infer moving human bodies indoors over time. It gets robust time-domain features for training and prediction. To avoid overfitting of a single Ensemble Learning classifier for future data, we design two-stage Ensemble Learning. Experimental results show accurate prediction (~1 second) for motion human detection. The method's performance exceeds 98% in a classic conference room, resisting factors like device antenna rotation, time variations, and human motion speed. Limitations include detecting only one moving target at a time and not studying multiple targets or transceiver device position changes. Also, the effect of CSI signals passing through walls in presence detection studies will be our next research focus.

REFERENCES

[1] H. Jiang, C. Cai, X. Ma, Y. Yang, and J. Liu, "Smart home based on wifi sensing: A survey," IEEE Access, vol. 6, pp. 13 317–13 325, 2018.

[2] Y. Ge, A. Taha, S. A. Shah, K. Dashtipour, S. Zhu, J. Cooper, Q. H. Abbasi, and M. A. Imran, "Contactless wifi sensing and monitoring for future healthcare-emerging trends, challenges, and opportunities," IEEE Reviews in Biomedical Engineering, vol. 16, pp. 171–191, 2022.

[3] X. Zhu, B. Ding, W. Li, L. Gu, and Y. Yang, "On development of ecurity monitoring system via wireless sensing network," EURASIP Journal on Wireless Communications and Networking, vol. 2018, no. 1, p. 221, 2018.

[4] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, "Fimd: Fine-grained device-free motion detection," in 2012 IEEE 18th International conference on parallel and distributed systems. IEEE, 2012, pp. 229–235.

[5] H. Zhu, F. Xiao, L. Sun, R. Wang, and P. Yang, "R-ttwd: Robust device-free through-the-wall detection of moving human with wifi," IEEE Journal on selected areas in communications, vol. 35, no. 5, pp. 1090–1103, 2017.

[6] K. Qian, C. Wu, Z. Yang, Y. Liu, and Z. Zhou, "Pads: Passive detection of moving targets with dynamic speed using phy layer information," in 2014 20th IEEE international conference on parallel and distributed systems (ICPADS). IEEE, 2014, pp. 1–8.

[7] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, "Pilot: Passive device-free indoor localization using channel state information," in 2013 IEEE 33rd International Conference on Distributed Computing Systems. IEEE, 2013, pp. 236–245.

[8] C. Han, Q. Tan, L. Sun, H. Zhu, and J. Guo, "Csi frequency domain fingerprint-based passive indoor human detection," Information, vol. 9, no. 4, p. 95, 2018.

[9] C. Wu, Z. Yang, Z. Zhou, X. Liu, Y. Liu, and J. Cao, "Non-invasive detection of moving and stationary human with wifi," IEEE Journal on Selected Areas in Communications, vol. 33, no. 11, pp. 2329–2342, 2015.

[10] C. M. Mesa-Cantillo, D. S´anchez-Rodr´ıguez, I. Alonso-Gonz´alez, M. A. Quintana-Su´arez, C. Ley-Bosch, and J. B. Alonso-Hern´andez, "A non intrusive human presence detection methodology based on channel state information of wi-fi networks," Sensors, vol. 23, no. 1, p. 500, 2023.

[11] A. Natarajan, V. Krishnasamy, and M. Singh, "Design of a low cost and device free human activity recognition model for smart led lighting control," IEEE Internet of Things Journal, 2023.

[12] A. Natarajan, V. Krishnasamy, and M. Singh, "A machine learning approach to passive human motion detection using wifi measurements from commodity iot devices," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1–10, 2023.

On the effectiveness of Kolmogorov–Arnold Networks for enhanced oil recovery prediction in polymer flooding

Samson Dawit Bekele
Department of Computer Science
Al-Farabi Kazakh National University
Almaty, Kazakhstan
samsondawitb@gmail.com

Yerzhan Kenzhebek
Department of Computer Science
Al-Farabi Kazakh National University
Almaty, Kazakhstan
kenzhebekyerzhan@gmail.com

Timur Imankulov
Department of Computer Science
Al-Farabi Kazakh National University
Almaty, Kazakhstan
imankulov.timur@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899678

Abstract— **Enhanced oil recovery techniques like polymer flooding are vital for maximizing hydrocarbon extraction, but predicting recovery factors remains challenging due to complex reservoir dynamics. Traditional artificial neural networks offer predictive power but lack interpretability, limiting their practical utility in engineering applications. This study introduces Kolmogorov–Arnold Networks (KANs) as an interpretable alternative for modeling recovery factors in polymer flooding. Using a synthetically generated dataset of over 160,000 samples with key reservoir parameters, we trained KAN models through a series of experiments, varying network architectures, grid sizes, learning rates, and optimizers (Adam and LBFGS) to evaluate performance and interpretability. The best-performing KAN achieved a Test Mean Squared Error (MSE) of 0.000597 and a Test coefficient of determination (R^2) of 0.902 with only 1,885 parameters, closely matching the performance of a previous deep neural network model that used 43,265 parameters (Test R² of 0.908). While KANs did not surpass the DNN in predictive accuracy, they offered comparable results with significantly reduced complexity and enhanced transparency. The modular structure and learnable activation functions of KANs provide insights into the decision-making process. This work contributes to the EOR field by demonstrating that KANs can effectively model complex reservoir behaviors while being transparent in their decision making.**

Keywords— Enhanced Oil Recovery, Kolmogorov–Arnold Networks, Interpretability, Recovery Factor Prediction, Artificial Neural Networks.

I. INTRODUCTION

The global demand for hydrocarbons continues to necessitate innovative methods to maximize recovery from existing reservoirs. Enhanced oil recovery (EOR) techniques, particularly polymer flooding, have demonstrated significant promise in improving oil extraction efficiency by modifying reservoir properties to enhance sweep efficiency and displacement of oil [1].

Despite its effectiveness, accurately predicting the recovery factor in polymer flooding is a complex task due to the nonlinear dynamics of fluid flow, reservoir heterogeneity, and the interplay of multiple physicochemical factors. Traditional numerical simulation methods, such as finite difference or finite element models, often involve computationally expensive calculations and may struggle to handle the variability of real-world reservoir conditions [2-4].

Recent advancements in machine learning (ML) have opened new avenues for data-driven modeling in oil recovery predictions [5]. Unlike conventional simulation methods, ML models can efficiently learn complex relationships from large datasets, offering faster and often more accurate predictions. Techniques such as neural networks [6], XGBoost [7] and support vector machines have shown significant potential in this domain [8, 9].

Polymer flooding has been extensively studied both in experimental and computational frameworks [10]. Simulation-based approaches, like those based on the Buckley–Leverett equations [11], have provided insights into the mechanics of polymer flooding.

Artificial Neural Networks (ANNs) have been extensively applied to EOR problems due to their ability to model complex nonlinear relationships and handle high-dimensional datasets. For instance, Saberi et al utilized ANNs to predict the performance of polymer flooding by considering parameters such as polymer concentration, salinity, and reservoir properties. The model achieved a high coefficient of determination (R^2) of 0.9990 indicating excellent predictive capability [12]. Similarly, Cheraghi et al. proposed a two-stage screening system employing ANNs to predict suitable EOR methods for candidate reservoirs [13]. Le Van and Chon developed ANN models to evaluate critical performances of CO_2-EOR processes and demonstrated the models' effectiveness in capturing complex reservoir behaviors [14]. Vo Thanh et al. applied ANNs to predict the performance of CO_2-EOR and storage in residual oil zones and they effectively highlighted the models' potential in optimizing CO_2 storage strategies [15]. Mohammadi et al. showcased the applicability of ANNs in simulating thermal recovery processes by employing cascade forward neural networks and the group method of data handling to model crude oil pyrolysis during thermal EOR [16].

The heavy usage of ANNs in this research field shows the versatility of the algorithm in addressing various EOR scenarios. However, despite their strengths, ANNs often function as "black-box" models – effectively making their predictions difficult to interpret [17]. Even if an ANN makes highly accurate predictions, it often leaves researchers questioning the

This research was funded by the Committee of Science and Higher Education of the Republic of Kazakhstan, grant number AP23489431.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

underlying decision-making process of the trained model. ANNs are therefore unreliable in scientific and engineering scenarios where transparency is crucial.

Kolmogorov–Arnold Networks (KANs) offer a promising alternative to traditional ANNs by addressing their "black-box" nature. KAN, as implemented by Liu et al., represents a recently introduced and novel approach in machine learning, garnering growing interest for its strong theoretical foundation and potential to enhance interpretability in complex modeling tasks. Rooted in the Kolmogorov–Arnold representation theorem, KANs provide a mathematically robust framework for approximating any continuous multivariate function. The theorem states that any multivariate function can be represented as a finite sum of univariate functions composed with linear transformations [18]. Unlike standard ANNs, KANs have learnable activation functions. Their modular structure decomposes the complex relationships within a dataset into simpler, interpretable components, thereby making them more transparent in their decision-making process.

Recent studies have highlighted the potential of KANs in tabular data modeling. Gao et al. introduced TabKANet, which integrates KANs with Transformer architectures to unify numerical and categorical feature encoding, achieving superior or comparable performance to Gradient Boosted Decision Trees across various datasets [19]. Similarly, Poeta et al. benchmarked KANs against Multi-Layer Perceptrons (traditional neural networks) and found that KANs excel in accuracy and F1 scores, particularly for larger datasets, albeit at a higher computational cost [20]. These studies underscore the growing interest in KANs for handling complex tabular data.

Building on our previous work, where we applied various machine learning algorithms – including polynomial regression, dense neural networks, and cascade-forward neural networks – to predict oil recovery factors in polymer flooding scenarios [21], we now aim to explore the effectiveness of KANs in this context. Our earlier study demonstrated that polynomial regression achieved an R^2 score of 0.909, while dense and cascade-forward neural networks attained R^2 values of 0.908 and 0.906, respectively. These findings highlighted the potential of ML models in enhancing oil recovery predictions.

In this study, we explore the potential of Kolmogorov–Arnold Networks in predicting recovery factors for polymer flooding. By leveraging the inherent interpretability and flexibility of KANs, we aim to overcome the limitations of traditional ANNs, providing both high predictive accuracy and transparent decision-making insights. This novel application of KANs in the EOR domain represents a step toward integrating interpretable machine learning models in oil recovery research.

In this paper, we investigate whether KANs can offer comparable or superior predictive performance, with the added benefit of improved interpretability, thereby addressing some limitations associated with traditional neural network models. By applying KANs in the EOR domain, we aim to advance the integration of interpretable machine learning models into oil recovery research

II. Methodology

A. Dataset

The dataset used in this study consists of several key variables essential for modeling polymer flooding in EOR. These include absolute permeability (k), pressure (P), porosity (m), oil saturation (So), water saturation (Sw), oil viscosity ($visc$), polymer concentration (Cp), and the oil recovery factor (RF), which serves as the target variable. The dataset represents a diverse range of reservoir and operational conditions. Moreover, gaussian noise was added to the dataset to make the learning process challenging for the models and to promote generalization.

To provide a comprehensive evaluation, the dataset contains more than 160,000 samples generated synthetically, simulating realistic EOR scenarios. For detailed information on the dataset, the data generation process, its distribution and statistical properties, we refer readers to our previous work [21].

Normalization was applied using a Standard Scaler, which transforms each feature to have a mean of zero and a standard deviation of one. This step ensures that all features contribute equally to the learning process. The dataset was then split into training and testing sets with an 80:20 ratio to evaluate model performance.

B. Implementation

In this study, we utilized PyKAN, the official and open-source Python library specifically designed for implementing KANs [22]. PyKAN provides a comprehensive framework for constructing and training KAN models, facilitating their application in various machine learning tasks. The library offers functionalities for defining network architectures, managing training processes, and evaluating model performance.

Each KAN architecture is defined by several key parameters: the width, specifying the number of neurons in each layer as a list (e.g., [2, 3, 1] for 2 input neurons, 3 in the first hidden layer, and 1 output neuron); the grid size, determining the number of intervals for spline-based activation functions; and the spline order (K), representing the order of B-splines used in the activations. Activation functions are initialized to SiLU by default unless otherwise specified. The learning rate (lr) varies depending on the optimizer. Regularization parameters include λ (weight decay, set to 0.001 in all experiments) and $\lambda_{entropy}$ (an entropy-based penalty set to 0.1) to encourage sparsity and enhance model interpretability.

The evaluation metrics used are Mean Squared Error (MSE) and the R^2 score. MSE quantifies the average squared difference between observed and predicted values, providing a measure of the model's prediction accuracy; a lower MSE indicates better performance. R^2, on the other hand, represents the proportion of variance in the dependent variable that is predictable from the independent variables, offering insight into the model's explanatory power; values closer to 1 suggest a stronger fit. While both metrics are valuable, R^2 is often more informative in regression analysis, as it can be expressed as a percentage and is more robust in certain scenarios compared to MSE [23].

C. Experiments

To evaluate the performance of KANs in predicting recovery factors for polymer flooding, two sets of experiments were conducted, each employing a different optimizer to train the models. Specifically, the Adam optimizer and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) optimizer were used to compare their effectiveness in minimizing the MSE loss and improving model accuracy. Hyperparameters were chosen via manual tuning and heuristics, balancing model performance and computational efficiency.

All weights were initialized using a normal distribution. Spline parameters were initialized to approximate linear functions for faster convergence. Each experiment involved a fixed number of steps, specified below, to balance computational cost and convergence accuracy. Full-batch optimization was employed for both optimizers.

1) Experiment A: Training with Adam Optimizer

In the first set of experiments, the KAN models were trained using the Adam optimizer, which is well-suited for large datasets and adaptive learning rates. Adam was configured to use full-batch optimization rather than mini-batches, as it allows for consistency with the LBFGS optimizer and better convergence in this context. The experiments ran for 1000 steps because Adam typically requires more iterations to converge compared to LBFGS, which is compensated by its efficiency in handling noisy gradients.

The experimental configurations involved variations in the network width, grid size, K, lr, and training steps. For each configuration, the number of trainable parameters was recorded, and the training time was measured. Model performance was evaluated using both the training and testing MSE as well as the R^2 score. Learning rates were selected heuristically, with values ranging from $10^{-4}\ to\ 2 \times 10^{-3}$ to ensure sufficient updates without overshooting the minimum. The hyperparameter configurations are shown in Table I.

TABLE I. ADAM OPTIMIZER HYPERPARAMETERS – EXPERIMENT A

Case	*Width*	*Grid*	*K*	*lr*
A1	[7, 5, 5, 1]	20	3	0.001
A2	[7, 10, 5, 1]	15	2	0.0005
A3	[7, 5, 5, 5, 1]	10	3	0.0001
A4	[7, 5, 5, 1]	50	4	0.002
A5	[7, 10, 10, 1]	5	3	0.001

2) Experiment B: Training with LBFGS Optimizer

The second set of experiments employed the LBFGS optimizer, a quasi-Newton method that is effective for tasks requiring highly precise convergence, particularly with smooth loss functions. LBFGS was chosen because of its ability to leverage second-order information for rapid convergence in fewer steps, making it ideal for the smooth MSE loss used in this study.

The experiments were limited to 200 steps, as LBFGS generally achieves high convergence efficiency within a smaller number of iterations compared to Adam. Like Adam, LBFGS was also configured for full-batch optimization. This aligns with its inherent design to operate over entire datasets. The same dataset and preprocessing pipeline were used to ensure consistency and comparability between the two optimizers.

Similar to the first experiment, variations in network width, grid size, K, lr, and training steps were explored. Learning rates for LBFGS were also selected heuristically, with values between $10^{-4}\ to\ 2 \times 10^{-3}$, as this range provided stable and consistent convergence during tuning. Training time, MSE, and R^2 were measured and recorded for each configuration. The hyperparameter details can be seen in Table II.

TABLE II. LBFGS OPTIMIZER HYPERPARAMETERS – EXPERIMENT B

Case	*Width*	*Grid*	*K*	*lr*
B1	[7, 5, 5, 1]	30	3	0.001
B2	[7, 10, 5, 1]	20	2	0.0005
B3	[7, 5, 5, 5, 1]	10	3	0.0001
B4	[7, 5, 5, 1]	50	4	0.002
B5	[7, 10, 10, 1]	5	3	0.001

Both experiments utilized a high-performance NVIDIA RTX 4060Ti GPU to expedite training. For regularization, weight decay (λ) and an entropy-based penalty ($\lambda_{entropy}$) were applied to prevent overfitting and improve model robustness. Seeding was used to ensure reproducibility.

III. RESULTS

A. Experiment A: Adam Optimizer

The performance of KAN models trained using the Adam optimizer is summarized in Table III. Training times varied between 124.328 seconds (Case A2) and 429.850 seconds (Case A4). The variation in training time was directly influenced by the number of trainable parameters, with larger configurations requiring more computation. Test MSE values were consistently low, ranging from 0.000597 (Case A1) to 0.001006 (Case A3), while Test R^2 scores spanned from 0.834 (Case A3) to 0.902 (Case A1).

Case A1 achieved the best Test MSE (0.000597) and the highest Test R^2 (0.902), demonstrating that a balanced configuration with moderate grid size and network complexity can yield superior results. Case A5 also performed well, achieving a similar Test R^2 (0.901) with slightly higher computational efficiency compared to Case A4. Larger grid sizes and increased neuron counts, as seen in Cases A4 and A5, generally led to better R^2 scores. This affirms the importance of model complexity in explaining data variability.

Key insights from this experiment suggest that configurations with a grid size of 20-50 and moderate learning rates (~0.001) provide optimal performance, as they balance accuracy with training time.

TABLE III. EXPERIMENT A RESULTS

Case	*Parameters*	*Training time(s)*	*Test MSE*	*Test R^2*
A1	1885	158.693	0.000597	0.902
A2	2875	124.328	0.000609	0.900
A3	1710	147.418	0.001006	0.834
A4	3900	429.850	0.000624	0.897
A5	2520	137.841	0.000598	0.901

B. Experiment B: LBFGS Optimizer

The LBFGS optimizer demonstrated its strengths in rapid convergence with fewer training steps (200), but training times were longer overall compared to Adam due to the computational demands of the quasi-Newton method. As shown in Table IV, training times ranged from 482.973 seconds (Case B3) to 2029.817 seconds (Case B4). Test MSE values ranged from 0.000618 (Case B1) to 0.000692 (Case B4), while Test R^2 values remained high, mostly above 0.89, except for Case B4, which showed slightly degraded performance.

Case B1 exhibited the best overall performance, achieving a Test MSE of 0.000618 and a Test R^2 of 0.898. Case B5 closely followed with a Test MSE of 0.000619 and an identical Test R^2 of 0.898. Interestingly, smaller grid sizes (e.g., Case B3 with a grid of 10) allowed LBFGS to perform efficiently, achieving reasonable accuracy with reduced training times. Conversely, larger grid sizes, as seen in Case B4, suffered from instability and overfitted, leading to lower accuracy despite extended training times.

The results indicate that LBFGS excels with compact configurations, where its precision is leveraged to achieve fast convergence with minimal overfitting.

TABLE IV. EXPERIMENT B RESULTS

Case	*Parameters*	*Training time(s)*	*Test MSE*	*Test R^2*
B1	2535	1029.793	0.000618	0.898
B2	3500	566.849	0.000628	0.897
B3	1710	482.973	0.000664	0.891
B4	3900	2029.817	0.000692	0.886
B5	2520	633.499	0.000619	0.898

Across both experiments, the Adam optimizer emerged as the preferred choice for its ability to achieve lower training times with comparable accuracy. Case A1 demonstrated the best overall performance, with the lowest Test MSE and the highest Test R^2. Interestingly, the best performing model has one of the fewest parameters and balances grid size as well as K. This suggests that compact models are enough to reach a noteworthy predictive performance for KANs.

For tasks requiring relatively high precision with fewer steps, the LBFGS optimizer showed promise, particularly with compact configurations. Case B1 delivered the best performance for this optimizer.

In Fig 1, the predicted versus actual values demonstrate the strong predictive accuracy of the Case A1 model. The plot exhibits a close alignment of the points to the ideal reference line (y = x), which indicates a minimal deviation between the predicted and true recovery factor values. The relatively dense clustering of points along the diagonal in the plot speaks to the potential and reliability of KAN models in predicting recovery factors for polymer flooding scenarios.

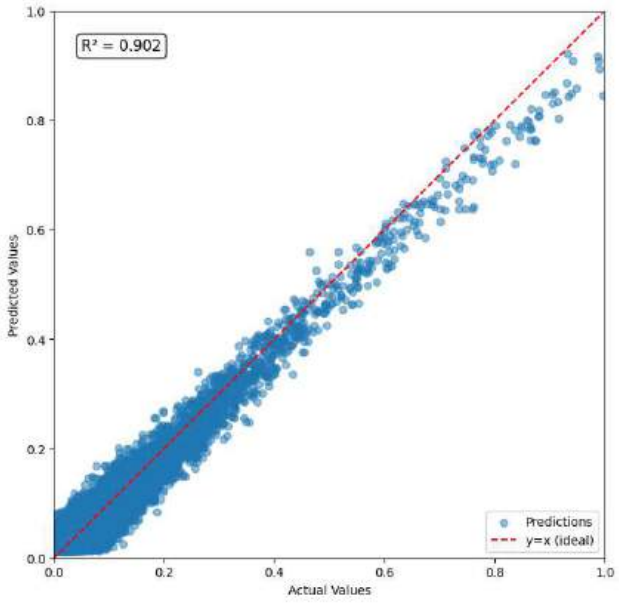


Fig. 1. Predicted vs. Actual Values: Case A1 model

In addition, we can visualize the decision-making process of the best performing KAN model (Case A1), as seen in Fig 2.

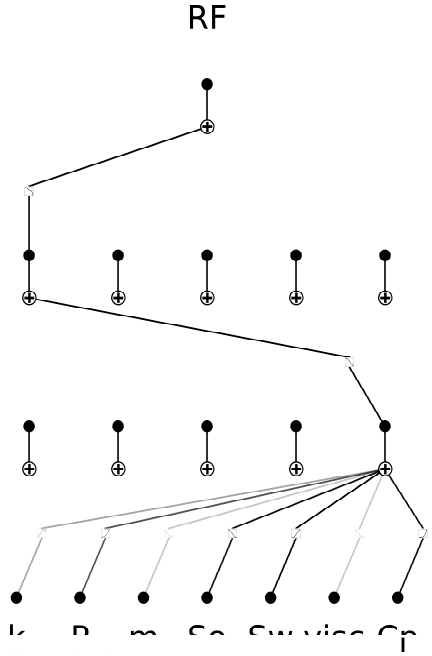


Fig. 2. Interpretable model plot for Case A1

The black lines in Fig 2 indicate active connections, with line thickness corresponding to the significance of the activation. Thinner, lighter lines represent less significant contributions, while the thicker connections highlight the critical pathways for the prediction. This hierarchical and interpretable architecture provides insights into how input features influence the output. This emphasizes the model's ability to integrate both sparse and dense relationships.

Moreover, it is possible to study the activation functions in the model. Fig 3 dives deeper into the individual activation functions of the first layer for three selected connections: (0, 0, 0), (0, 0, 1), and (0, 0, 2). These coordinates indicate the first layer, the first neuron in that layer, and its connections to the first, second, and third input features, respectively. Each subplot demonstrates how the network transforms input values through the activation function of a specific connection. For example, the first activation function exhibits a sinusoidal behavior, while others display distinct nonlinear transformations.

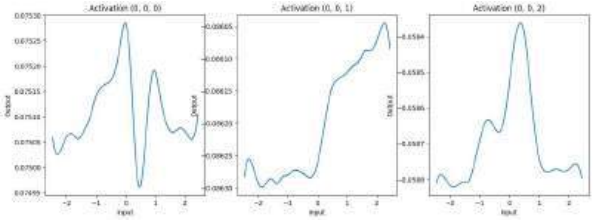


Fig. 3. Visualization of Activation Functions for the First Layer in Case A1 model

The feature attribution analysis, as illustrated in Fig. 4, demonstrates how we can directly extract feature importance from the model itself using the built-in attribution mechanism in KAN. As seen in the visualization, Sw emerges as the most critical feature, with a significantly higher attribution score compared to others. Meanwhile, $Cp, So,$ and P show moderate contributions, while $k, m,$ and $visc$ have minimal impact.

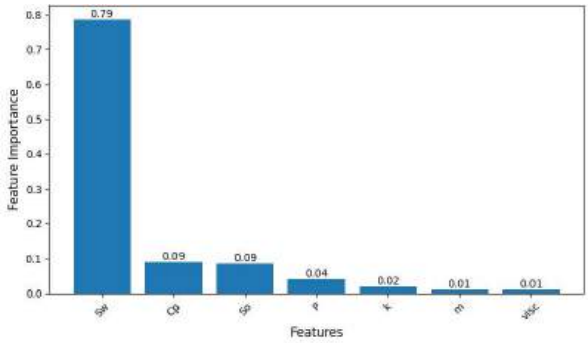


Fig. 4. Feature attribution scores of Case A1 model

It is also possible to extract the specific impact of each feature on individual neurons within its architecture. This is done by quantifying how much each input feature influences the activations of particular neurons in a given layer. For instance, it is possible to examine how features contribute to the activation of specific pathways within the network. This neuron-level feature attribution provides insights into how it interacts with the hierarchical structure of the model. For instance, the feature attributions for neuron 3 in layer 1 are displayed in Fig 5.

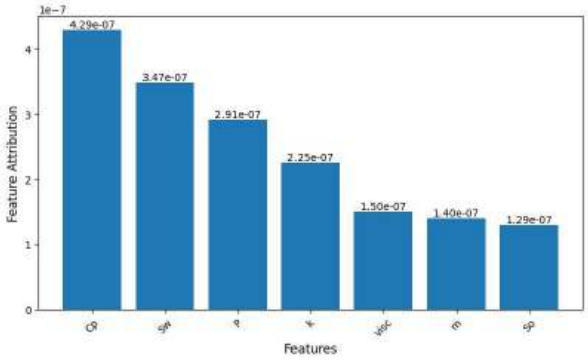


Fig. 5. Feature attribution scores for neuron 3 in layer 1 (Model Case A1)

To visually assess the model's important features and activation functions further, we can prune the model, a process that involves removing less significant input features and connections based on their contribution to the output. This approach simplifies the model and retains only the dominant pathways that are critical to its decision-making process. As shown in Fig. 6, the pruned KAN model focuses on the most influential features, such as Sw, Cp, and $So,$ while eliminating features like m that contribute minimally to the predictions of RF. By removing these weaker connections, pruning enhances the interpretability of the model and makes it easier to understand which features and pathways are most impactful.

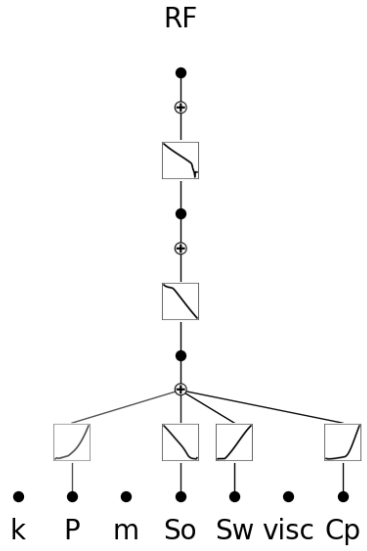


Fig. 6. Pruned model plot for Case A1

IV. Conclusion

In this study, we explored the application of KANs for predicting oil recovery factors in polymer flooding scenarios. Compared to our previous work using a Deep Neural Network (DNN) with 43,265 trainable parameters, the KAN models demonstrated comparable predictive performance with significantly fewer parameters. The best-performing KAN achieved a Test MSE of 0.000597 and a Test R^2 of 0.902, closely matching the DNN's performance (Test R^2 of 0.908) while utilizing only 1,885 parameters.

Although KANs did not surpass the predictive accuracy of the DNN, they offer the distinct advantage of improved interpretability due to their modular structure and learnable activation functions. This interpretability allows for greater insight into the decision-making process of the model, which is particularly valuable in scientific and engineering applications where transparency is crucial. By analyzing activation functions and hierarchical representations, we can better understand how input features influence the output and thereby facilitate more informed decision-making in the EOR field.

This work contributes to the EOR domain by introducing an interpretable variant of ANNs, demonstrating that KANs can effectively model complex reservoir behaviors with reduced model complexity. The ability to achieve high predictive accuracy with fewer parameters while being interpretable (through feature attribution, learnable and observable activation functions, and pruning) highlights the efficiency of KANs and their potential to bridge the gap between predictive performance and model transparency.

Future research will focus on studying the interpretability of KANs to better understand the decision-making process and underlying patterns in predictions. Additionally, integrating domain-specific physics-based constraints and priors into the learning process could further improve model reliability and accuracy in capturing complex reservoir dynamics. To ensure their robustness and adaptability in diverse scenarios, expanding the application of KANs to other EOR techniques and validating their performance on real-world datasets will also be key areas of exploration.

REFERENCES

[1] M. Ahmadi and Z. Chen, "Challenges and future of chemical assisted heavy oil recovery processes," Advances in Colloid and Interface Science, vol. 275, p. 102081, Jan. 2020, doi: 10.1016/j.cis.2019.102081.

[2] Y. Yuan, A. Cheng, D. Yang, C. Li, and Y. Liu, "Theory and application of numerical simulation method of capillary force enhanced oil production," Appl. Math. Mech.-Engl. Ed., vol. 36, no. 3, pp. 379–400, Mar. 2015, doi: 10.1007/s10483-015-1917-6.

[3] K. G. Salem, A. M. Salem, M. A. Tantawy, A. A. Gawish, S. Gomaa, and A. N. El-hoshoudy, "A Comprehensive Investigation of Nanocomposite Polymer Flooding at Reservoir Conditions: New Insights into Enhanced Oil Recovery," J Polym Environ, vol. 32, no. 11, pp. 5915–5935, Nov. 2024, doi: 10.1007/s10924-024-03336-z.

[4] T. Nassan and M. Amro, "Finite Element Modeling of Immiscible Two-Phase Flow in Oil Reservoirs," Jan. 01, 2022, Social Science Research Network, Rochester, NY: 4898872. doi: 10.2139/ssrn.4898872.

[5] W. Ozowe, A. D. Ogbu, and A. H. Ikevuje, "Data science's pivotal role in enhancing oil recovery methods while minimizing environmental footprints: An insightful review," Computer Science & IT Research Journal, vol. 5, no. 7, Art. no. 7, Jul. 2024, doi: 10.51594/csitrj.v5i7.1348.

[6] W. A. Khan, Z. Rui, T. Hu, Y. Liu, F. Zhang, and Y. Zhao, "Application of Machine Learning and Optimization of Oil Recovery and CO2 Sequestration in the Tight Oil Reservoir," SPE Journal, vol. 29, no. 06, pp. 2772–2792, Jun. 2024, doi: 10.2118/219731-PA.

[7] A. Roustazadeh, B. Ghanbarian, F. Male, M. B. Shadmand, V. Taslimitehrani, and L. W. Lake, "Estimating oil recovery factor using machine learning: Applications of XGBoost classification," Oct. 28, 2022, arXiv: arXiv:2210.16345. doi: 10.48550/arXiv.2210.16345.

[8] D. Bui, A.-M. Koray, E. Appiah Kubi, A. Amosu, and W. Ampomah, "Integrating Machine Learning Workflow into Numerical Simulation for Optimizing Oil Recovery in Sand-Shale Sequences and Highly Heterogeneous Reservoir," Geotechnics, vol. 4, no. 4, Art. no. 4, Dec. 2024, doi: 10.3390/geotechnics4040055.

[9] F. Krasnov, N. Glavnov, and A. Sitnikov, "A Machine Learning Approach to Enhanced Oil Recovery Prediction," in Analysis of Images, Social Networks and Texts, W. M. P. van der Aalst, D. I. Ignatov, M. Khachay, S. O. Kuznetsov, V. Lempitsky, I. A. Lomazova, N. Loukachevitch, A. Napoli, A. Panchenko, P. M. Pardalos, A. V. Savchenko, and S. Wasserman, Eds., Cham: Springer International Publishing, 2018, pp. 164–171. doi: 10.1007/978-3-319-73013-4_15.

[10] A. Mohsenatabar Firozjaii and H. R. Saghafi, "Review on chemical enhanced oil recovery using polymer flooding: Fundamentals, experimental and numerical simulation," Petroleum, vol. 6, no. 2, pp. 115–122, Jun. 2020, doi: 10.1016/j.petlm.2019.09.003.

[11] S. G. Dardaganian, "The Application of the Buckley-Leverett Frontal Advance Theory to Petroleum Recovery," Journal of Petroleum Technology, vol. 10, no. 04, pp. 49–52, Apr. 1958, doi: 10.2118/835-G.

[12] H. Saberi, E. Esmaeilnezhad, and H. J. Choi, "Artificial Neural Network to Forecast Enhanced Oil Recovery Using Hydrolyzed Polyacrylamide in Sandstone and Carbonate Reservoirs," Polymers, vol. 13, no. 16, Art. no. 16, Jan. 2021, doi: 10.3390/polym13162606.

[13] Y. Cheraghi, S. Kord, and V. Mashayekhizadeh, "A two-stage screening framework for enhanced oil recovery methods, using artificial neural networks," Neural Comput & Applic, vol. 35, no. 23, pp. 17077–17094, Aug. 2023, doi: 10.1007/s00521-023-08557-2.

[14] S. Le Van and B. H. Chon, "Evaluating the critical performances of a CO2–Enhanced oil recovery process using artificial neural network models," Journal of Petroleum Science and Engineering, vol. 157, pp. 207–222, Aug. 2017, doi: 10.1016/j.petrol.2017.07.034.

[15] H. Vo Thanh, Y. Sugai, and K. Sasaki, "Application of artificial neural network for predicting the performance of CO2 enhanced oil recovery and storage in residual oil zones," Sci Rep, vol. 10, no. 1, Art. no. 1, Oct. 2020, doi: 10.1038/s41598-020-73931-2.

[16] M.-R. Mohammadi, A. Hemmati-Sarapardeh, M. Schaffie, M. M. Husein, and M. Ranjbar, "Application of cascade forward neural network and group method of data handling to modeling crude oil pyrolysis during thermal enhanced oil recovery," Journal of Petroleum Science and Engineering, vol. 205, p. 108836, Oct. 2021, doi: 10.1016/j.petrol.2021.108836.

[17] F.-L. Fan, J. Xiong, M. Li, and G. Wang, "On Interpretability of Artificial Neural Networks: A Survey," IEEE Transactions on Radiation and Plasma Medical Sciences, vol. 5, no. 6, pp. 741–760, Nov. 2021, doi: 10.1109/TRPMS.2021.3066428.

[18] Z. Liu et al., "KAN: Kolmogorov-Arnold Networks," Jun. 16, 2024, arXiv: arXiv:2404.19756. doi: 10.48550/arXiv.2404.19756.

[19] W. Gao, Z. Gong, Z. Deng, F. Rong, C. Chen, and L. Ma, "TabKANet: Tabular Data Modeling with Kolmogorov-Arnold Network and Transformer," Oct. 02, 2024, arXiv: arXiv:2409.08806. doi: 10.48550/arXiv.2409.08806.

[20] E. Poeta, F. Giobergia, E. Pastor, T. Cerquitelli, and E. Baralis, "A Benchmarking Study of Kolmogorov-Arnold Networks on Tabular Data," Jun. 20, 2024, arXiv: arXiv:2406.14529. doi: 10.48550/arXiv.2406.14529.

[21] T. Imankulov, Y. Kenzhebek, S. D. Bekele, and E. Makhmut, "Enhancing Oil Recovery Predictions by Leveraging Polymer Flooding Simulations and Machine Learning Models on a Large-Scale Synthetic Dataset," Energies, vol. 17, no. 14, Art. no. 14, Jan. 2024, doi: 10.3390/en17143397.

[22] Z. Liu, KindXiaoming/pykan. (Nov. 19, 2024). Jupyter Notebook. Accessed: Nov. 20, 2024. [Online]. Available: https://github.com/KindXiaoming/pykan

[23] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in Artificial Intelligence Application in Networks and Systems, R. Silhavy and P. Silhavy, Eds., Cham: Springer International Publishing, 2023, pp. 15–25. doi: 10.1007/978-3-031-35314-7_2.

Terrain image classification based on Vision transformer deep learning algorithm

Shiqi Ren[1*], Yu Zhang[2]
[1]College of information engineering (college of artificial intelligence), Yangzhou university, Yangzhou, Jiangsu, 225000, China
[2]College of computing, Georgia Institute of Technology, North Avenue Atlanta, GA, 30332, USA
[*]Corresponding author: e-mail: 937327378@qq.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899657

***Abstract*—Terrain image classification is an important research direction in the field of remote sensing and computer vision, aiming to realize automatic recognition and classification of different landform features through the analysis and processing of terrain images. In this paper, a deep learning algorithm based on Vision Transformer (ViT) is used to classify terrain images, and the performance of the algorithm in this task is systematically evaluated. In the process of model construction, we first imported the Vision Transformer model and made corresponding parameter Settings to ensure its adaptability and effectiveness. After training, it is observed that the loss function of the training set decreases from the initial value of 2.84 to 0.35, a decrease of 2.49, indicating that the model tends to converge in continuous optimization. At the same time, the accuracy is also significantly improved, from 55.9% to 86.8%, an increase of 30.9%, showing the enhancement of the model's learning ability. For the validation set, loss also decreased from 0.78 to 0.47, a decrease of 0.31, while accuracy increased from 60.1% to 83.5%, an increase of 23.4%. These results further prove the good performance of the model on different data sets and its convergence trend. In addition, through the evaluation of the test set, we get more specific performance indicators: The accuracy of the terrain image classification model based on Vision Transformer on the test set reaches 89.9%, the Precision is 0.9615, the Recall is 0.9494, and the F1-score is 0.9554. These indicators show that the model not only has high classification accuracy, but also performs well in generalization ability. To sum up, this research demonstrates the effectiveness of Vision Transformer deep learning algorithm in terrain image classification, and provides a new solution idea and method for related fields. Through continuous optimization and adjustment, the algorithm is expected to achieve more extensive promotion in practical applications, and bring positive impact on geographic information system, environmental monitoring and other fields..**

Keywords-Terrain image classification; Vision Transformer; Deep learning.

I. INTRODUCTION

Terrain image classification is an important research direction in the field of remote sensing and computer vision. With the development of satellite technology and UAV technology, it is becoming easier and easier to obtain high-resolution terrain images. These images contain rich geographic information, which can be used in many fields such as urban planning, environmental monitoring and agricultural management. However, how to extract useful information from these massive image data for accurate classification and analysis is an important topic of current research.

Traditional terrain image classification methods mainly rely on manual feature extraction and machine learning algorithms. These methods often require expert knowledge to design features and do not perform well when dealing with complex scenarios [1]. For example, methods based on simple features such as color, texture or shape often struggle to achieve satisfactory results in the face of complex terrain. With the development of deep learning technology, especially the emergence of convolutional neural network (CNN) [2], a new solution for terrain image classification has been provided. Deep learning can automatically learn high-level feature representations from raw data, which greatly improves classification accuracy and efficiency.

Deep learning algorithm plays an important role in terrain image classification. First, convolutional neural networks can effectively extract features of different scales and levels through multi-layer structures, which enables them to better capture spatial relationships in terrain images [3]. For example, when identifying different types of land cover such as urban areas, forests, and lakes, CNN can gradually extract edges, textures, and other complex features through hierarchical structure, thus achieving more accurate classification.

Secondly, deep learning algorithms also have good generalization ability. When faced with new, unseen data, well-trained deep learning models can adapt well and make accurate predictions. This advantage makes deep learning particularly suitable for processing frequently changing or diverse terrain image data. In addition, through transfer learning [4] and other technologies, the existing large-scale data sets can be used to pre-train the model and then fine-tune small-scale specific tasks, thus reducing the need for the amount of labeled data and improving the efficiency of model training.

Moreover, in recent years, emerging deep learning technologies such as generative adversarial network (GAN) [5] have also begun to be applied to terrain image classification. Gans can be used to generate high-quality synthetic data to enhance the training set, thereby improving model performance. At the same time, the robustness of the model against noise and interference can be improved through adversarial training, which makes it more reliable in practical applications.

In summary, the application of deep learning algorithms in terrain image classification has greatly promoted the development of this field. By automating feature extraction, efficiently processing massive data and improving prediction accuracy, it provides strong support for various practical

applications. This paper classifies terrain images based on Vision transformer deep learning algorithm, and analyzes the effect of the algorithm in terrain image classification..

II. DATA FROM DATA ANALYSIS

The data set used in this experiment is selected from the open source data set, which contains four types of images, including plain, desert, mountain and forest. Each type of image is divided into two parts: training set and verification set. The training set is used for the training of the model, and the verification set is used for the verification of the model. Four kinds of images are displayed, as shown in Figure 1, namely desert, forest, plain and mountain..

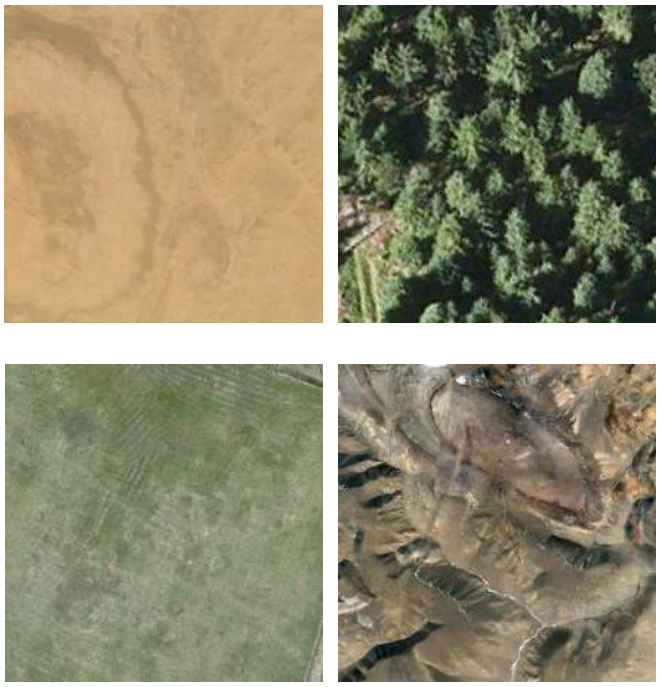


Figure 1. Four kinds of images.

III. METHOD

A. Transformer

Transformer is a deep learning model used for processing sequence data. Its core idea is to capture the dependencies between different positions in the input sequence through the self-attention mechanism, thus overcoming the shortcomings of traditional recurrent neural networks (RNN) in long sequence processing [6]. The structure diagram of Transformer is shown in Figure 2.

.

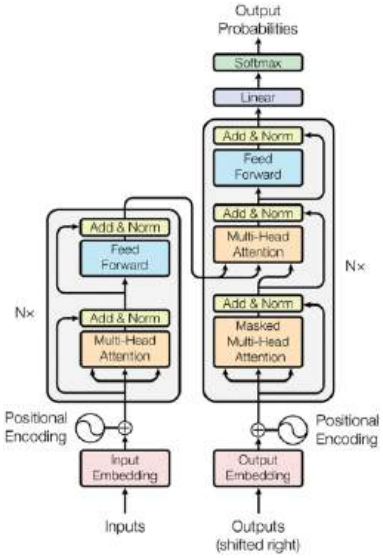


Figure 2. The structure diagram of Transformer.

Transformer is mainly composed of encoder and decoder. The encoder converts the input sequence into a set of context vectors, and the decoder generates an output sequence from these context vectors. Each encoder and decoder contains multiple identical layers, each consisting of two main parts: the self-attention mechanism and the feedforward neural network. In the self-attention mechanism, the model calculates the relevance of each word in the input sequence to other words, thereby dynamically adjusting the importance of each word in the context. The structure of the self-attention mechanism is shown in Figure 3.

In addition, Transformer introduces location coding to preserve the position information of the words in the input sequence, since the self-attention mechanism itself does not have sequential information. This allows Transformer to efficiently process inputs of various lengths while maintaining efficient parallel computing capabilities [7].

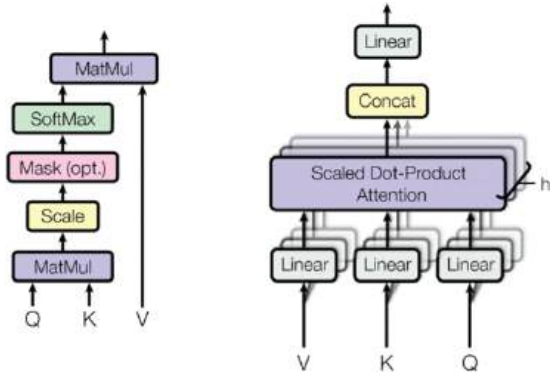


Figure 3. The structure of the self-attention mechanism.

B. Vision Transformer

Vision Transformer (ViT) is a new model that has attracted extensive attention in the field of computer vision in recent years. Its core idea is to transform the image processing of traditional convolutional neural networks (CNNS) into Transformer architecture based on self-attention mechanism.

This innovation not only improves the performance of image classification, but also provides new ideas and methods for computer vision tasks [8].

The rationale for Vision Transformer can be broken down into several steps. First, the input image is divided into several small patches, each of which is treated as a "word". Next, to enable the model to capture location information, the ViT adds location coding to each feature vector. These position codes are generated by sine and cosine functions, allowing the model to identify the relative position relationships between different small pieces. In this way, the ViT converts the image into a series of feature vectors with positional information. The principle structure diagram of Vision Transformer is shown in Figure 4.

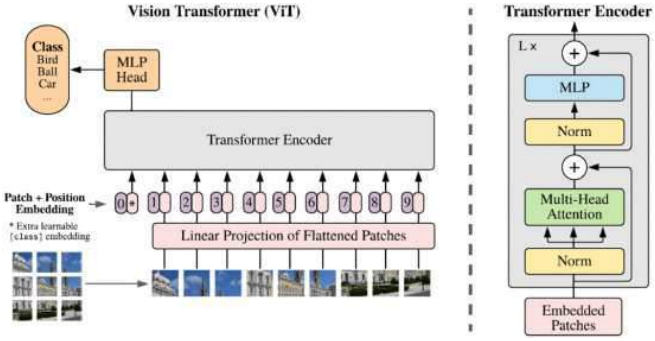


Figure 4. The principle structure diagram of Vision Transformer.

Vision Transformer uses a self-attention mechanism, which is a core component of the Transformer architecture. In the self-attention mechanism, each input feature vector interacts with all other feature vectors to calculate the importance weight of that feature [9].

The Vision Transformer is typically composed of multiple Transformer encoder layers stacked on top of each other. Each encoder layer contains two main parts: a multi-head self-attention mechanism and a feedforward neural network. Multi-head self-attention allows the model to learn information in different subspaces, while the feedforward neural network makes nonlinear transformations for each location independently. The entire structure also includes residual connections and layer normalization to enhance training stability.

After passing through multiple encoder layers, the ViT extracts an information vector representing the entire image from the last layer, which is usually achieved by taking the last Token. Finally, this information vector will be passed to a classifier for the final image classification task [10].

With the development of Vision Transformer, its application scope is gradually expanded, not only limited to image classification, but also includes a variety of visual tasks such as object detection and semantic segmentation. At the same time, some improved versions, such as Swin Transformer, DeiT, etc., have also come out one after another, which combine some advantages of CNN.

IV. RESULT

After the data set is divided, this paper uses Matlab R2023a to conduct experiments. First, the model is imported and parameters are set. The maximum number of iterations is set to 1000, the learning rate is set to 0.01, the genetic algebra is set to 50, the population size is set to 5, and the parameters of the crossover function are set to 2. In terms of hardware parameters, the CPU is 32G, and the graphics card is 3090. loss and accuracy were used to evaluate the prediction effect of the model.

During the training process, the numerical change curves of loss and accuracy are output, as shown in Figure 5. In the verification process, the numerical change curves of loss and accuracy are output, as shown in Figure 6.

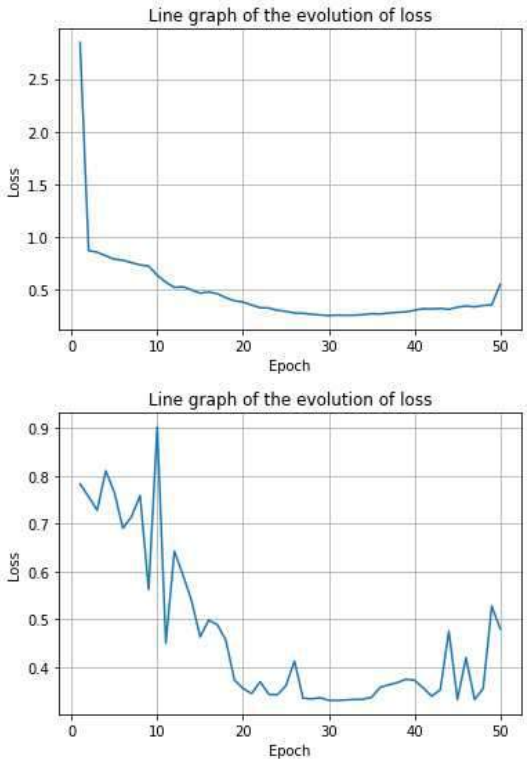


Figure 5. Change curves of loss and accuracy of training process.

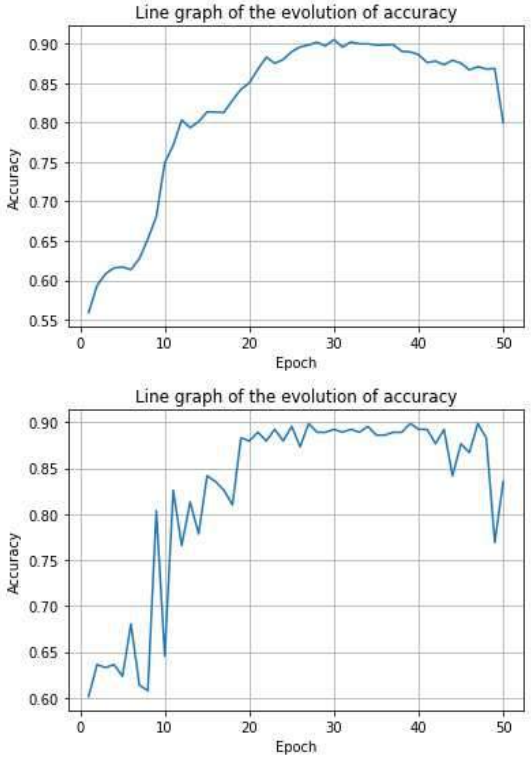


Figure 6. Change curves of loss and accuracy of test process.

It can be seen from the change curves of loss and accuracy in the training set that the value of loss decreases from the initial value of 2.84 to 0.35, the value of loss decreases by 2.49, and the loss curve tends to converge. Accuracy increased by 30.9% from the initial 55.9% to the final 86.8%, and the accuracy curve tended to converge.

It can be seen from the change curves of loss and accuracy in the training set that the loss value decreases from the initial 0.78 to 0.47, and the loss value decreases by 0.31, and the loss curve tends to converge. Accuracy increased by 23.4% from the initial 60.1% to the final 83.5%, and the accuracy curve tended to converge.

The model is tested using a test set to output a confusion matrix of predicted results, as shown in Figure 7.

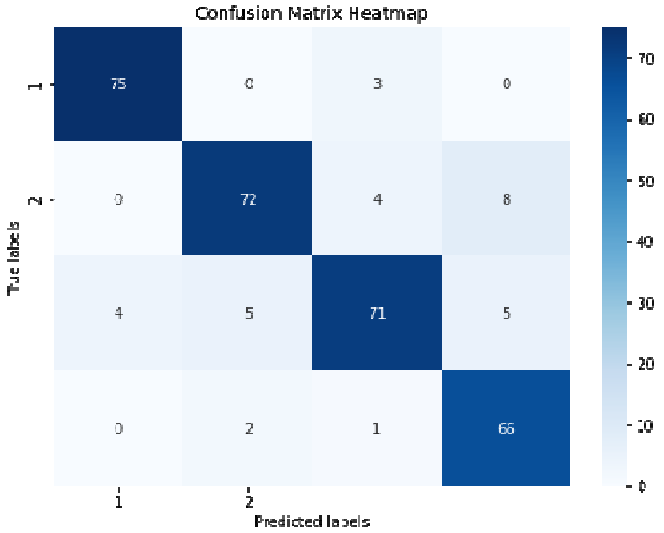


Figure 7. Confusion matrix of predicted results

The evaluation indicators of the forecast results of the output test set are shown in Table 1.

TABLE I. MODEL EVALUATION..

	Precision	Recall	F1-score	Support
0	0.9615	0.9494	0.9554	79
1	0.8571	0.9114	0.8834	79
2	0.8353	0.8987	0.8659	79
3	0.9565	0.8354	0.8919	79
Accuracy			0.8987	316
Macro avg	0.9026	0.8987	0.8991	316
Weighted avg	0.9026	0.8987	0.8991	316

As can be seen from the prediction results of the test set, the Accuracy, Precision, Recall and F1-score of the terrain image classification model based on the Vision transformer deep learning algorithm in this paper are 89.9%, 0.9615, 0.9494 and 0.9554 respectively. The model has good prediction effect and strong generalization ability.

V. CONCLUSION

This paper uses a model based on Vision Transformer (ViT) deep learning algorithm to focus on efficient classification of terrain images. After several rounds of training, it is observed that the loss value of the training set decreases significantly from the initial 2.84 to 0.35, and the reduction amplitude reaches 2.49, indicating that the loss curve tends to converge. This change indicates that the model is effectively learning data features, thereby improving its classification ability. At the same time, the accuracy of the training set also increased from 55.9% to 86.8%, an increase of 30.9%. This result not only reflects the progress of the model in the training process, but also shows that the Vision Transformer architecture has good learning ability when processing terrain images.

Further analysis of the performance of the validation set shows that the loss value decreases from 0.78 to 0.47, with a decrease of 0.31, which once again proves the effectiveness of the model in the optimization process. In addition, the accuracy of the verification set has also increased from 60.1% to 83.5%, an increase of 23.4%. These numerical changes further confirm the stability and reliability of the model, and provide a solid foundation for the subsequent application.

Finally, the prediction results on the test set show that the terrain image classification model built based on Vision Transformer deep learning algorithm has an Accuracy of 89.9%, a Precision of 0.9615, and a Recall of 0.9494. The F1-score was 0.9554. These indicators show that the model not only has strong classification ability, but also shows good generalization performance. This means that it can effectively adapt to new data, making it more practical in practical applications.

In summary, through the application and effect analysis of Vision Transformer deep learning algorithm in terrain image classification tasks, we can conclude that the algorithm has excellent performance in processing complex terrain data, which not only improves classification accuracy, but also enhances model generalization ability.

REFERENCES

[1] McGrory, Shawn, P. Michael Furlong, and Krzysztof Skonieczny. "Characterizing terrain image classification difficulties through reduced-dimension class convex hull analysis." Journal of Terramechanics 96 (2021): 133-145.

[2] Hu, Shimin, et al. "Fast and accurate terrain image classification for ASTER remote sensing by data stream mining and evolutionary-EAC instance-learning-based algorithm." Remote Sensing 13.6 (2021): 1123.

[3] Wang, Wanli, et al. "A visual terrain classification method for mobile robots' navigation based on convolutional neural network and support vector machine." Transactions of the Institute of Measurement and Control 44.4 (2022): 744-753.

[4] Li, Yanyi, et al. "Adoption of machine learning in intelligent terrain classification of Hyperspectral remote sensing images." Computational Intelligence and Neuroscience 2020.1 (2020): 8886932.

[5] Shi, Junfei, Haiyan **, and **aohua Li. "A novel multi-feature joint learning method for fast polarimetric SAR terrain classification." IEEE Access 8 (2020): 30491-30503.

[6] Vulpi, Fabio, et al. "Recurrent and convolutional neural networks for deep terrain classification by autonomous robots." Journal of Terramechanics 96 (2021): 119-131.

[7] Li, Shaojie, et al. "Utilizing the LightGBM Algorithm for Operator User Credit Assessment Research." Applied and Computational Engineering, vol. 75, no. 1, EWA Publishing, July 2024, pp. 36–47, doi:10.54254/2755-2721/75/20240503.

[8] Ma, Danqing, et al. Fostc3net:A Lightweight YOLOv5 Based On the Network Structure Optimization. 2024, https://arxiv.org/abs/2403.13703.

[9] Xiang, Ao, et al. A Multimodal Fusion Network For Student Emotion Recognition Based on Transformer and Tensor Product. 2024, https://arxiv.org/abs/2403.08511.

[10] Dang, Bo, et al. Real-Time Pill Identification for the Visually Impaired Using Deep Learning. 2024, https://arxiv.org/abs/2405.05983.

Weakly Supervised Anomaly Detection by Utilizing Incomplete Anomaly Information

1st Qingqing Fang
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
fangqq3@mail2.sysu.edu.cn

2nd Qinliang Su
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
suqliang@mail.sysu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899497

***Abstract*—Anomaly detection aims to distinguish abnormal samples from normal ones. In this paper, only a limited number and types of anomaly samples are assumed to be available in the training set while most types of anomaly samples cannot be obtained. To effectively utilize this incomplete anomaly information, we introduce GMMAD. In order to get a good representation of samples, we first extract two different scale feature maps from the pre-trained ResNet, then use the fusion module to fuse the maps. Among the fusion module, the projector aims to project the larger maps into the same size as the smaller ones, then the maps are concatenated and fed into an integrator to produce final representations. The Gaussian Mixture Model is employed to characterize the distribution of normal features, whose parameters are updated according to the negative log-likelihood, and then it is used to guide the fusion module to output distinct features to detect anomalies. Through iterative training of the GMM and the fusion module, our approach effectively utilizes incomplete anomaly information to detect anomalies. Experiments on three medical datasets have shown that with only a limited number and types of anomalies, GMMAD outperforms existing methods in detecting anomalies.**

***Index Terms*—anomaly detection, weakly-supervised, incomplete anomalous information**

I. Introduction

Anomaly detection, also known as outlier detection, aims to identify samples that deviate from the normal ones. Among all possible anomalous formats, we mainly focus on image anomaly detection, of which there are many applications in the real world. For instance, in medical diagnostics [1], anomalies play a vital role in disease identification, while in industrial quality assurance [2], the detection of non-conforming samples is crucial for maintaining product certification standards.

To detect anomalous samples, unsupervised methods [3]–[8] are employed under the assumption that anomalous samples come in various forms and types, so it makes manual collection and labeling of large-scale anomaly data impractical and costly. For example, in the medical domain, the wide range of disease images, coupled with the challenge of gathering information on rare conditions, complicates the task of collecting all types of abnormal samples for supervised training. Consequently, many unsupervised approaches overlook anomalous samples, relying solely on normal data. However, without considering anomalous information, these methods are prone to identify anomalies with noise, leading to high detection errors.

While it may not be feasible to collect anomalies of all types, gathering a small amount of available anomaly data is achievable. For example, for medical skin diagnosis, some common skin disease images can be collected for weakly supervised anomaly detection. However, due to the limited number and types of available anomalous samples, directly employing a classifier to differentiate between normal and anomalous samples can lead to inappropriate decision boundaries, resulting in significant false positives. To avoid this issue, some weakly supervised methods [2], [9]–[12] have been developed to enhance anomaly detection performance by leveraging such incomplete anomalous information. Among them, Deep SAD [9] constructs a latent center point by reconstructing normal samples, guiding normal samples to cluster around the center while pushing anomalous samples away. Devnet [10] employs multi-instance learning strategy and deviation loss to differentiate between anomalies and normal samples. Additionally, PReNet [12] learns pairwise anomaly scores by discriminating between three types of instance pairs. Although these weakly supervised methods are useful in detecting anomalies, they use centers to represent normal distribution or just assume they follow a certain distribution like standard Gaussian distribution, ignoring the true distribution of normal data, potentially impacting weakly supervised detection performance.

In this work, we introduce GMMAD, a novel approach aims at effectively utilizing incomplete anomalous information by modeling the feature distribution of normal samples. To extract suitable features for anomaly detection, we begin by extracting two different-scale feature maps from a pre-trained ResNet, which are then input into a trainable fusion module to generate more appropriate anomaly detection features. Given the Gaussian mixture distribution's ability to approximate diverse distributions, we employ it to model the distinct distribution of normal features. As the normal features are dynamically changed during training, GMM parameters are updated by gradient descent via negative log-likelihood loss. To leverage the collected incomplete anomaly data, we design a loss function derived from the GMM to update the feature fusion module so that it can produce distinct features for normal and

anomalous samples respectively. Through iterative training of the GMM and feature fusion module, we ensure that features effectively represent normal and anomalous samples, and the GMM can model the normal feature distribution. Anomalies are subsequently detected based on the GMM-calculated scores. Experimental results on three medical datasets demonstrate GMMAD's effectiveness in anomaly detection, showing better detection performance with increasing labeled anomaly numbers and types.

II. Releated Work

a) Unsupervised methods.: Existing unsupervised methods typically rely solely on normal samples for training. One-class methods [3]–[5] map normal samples to a compact space and identify anomalies based on distances within this space. Storage-based approaches [6], [13] store normal sample features and compare test sample similarity through search mechanisms. Contrastive learning methods [6]–[8] train models using proxy tasks like rotation prediction, with detection performance tied to these designed tasks. Reconstruction-based or Generation-based methods [14], [15] focus on the inability to correctly reconstruct anomalous images. Nevertheless, these unsupervised methods often overlook available anomalies, leading to implicit decisions that may undermine performance on challenging anomalies.

b) Weakly supervised methods.: Recent weakly supervised anomaly detection methods have emerged to improve anomaly detection performance by incorporating observed partial anomalies. Deep SAD [9] learns normal features in a compact space surrounding the normal center while pushing anomalous samples away from the center. DevNet [10] uses deviation loss to differentiate anomalous samples from normal ones. DRA [2] employs multiple normal and anomalous learning heads to assign anomaly scores to images based on possible anomalous patches. AAbiGAN [11] prevents generating anomalous images by special adversarial learning of BiGAN. PReNet [12] trains with three types of instance pairs to detect anomalies based on pairwise scores. While these methods leverage collected anomalies to enhance detection performance, they typically assume the distribution or centers of normal features without fully considering the actual potential distribution capable of distinguishing normal and anomalous samples. This oversight can potentially adversely impact detection performance.

III. Methodology

Previous methods have typically assumed the collection of a set of normal samples

$$\mathcal{X}_n = \{x_{n1}, x_{n2}, ..., x_{n_{M_1}}\}. \tag{1}$$

As previously discussed, while gathering anomalies of various types on a large scale is impractical, it is feasible to obtain a small number of anomalies

$$\mathcal{X}_a = \{x_{a1}, x_{a2}, ..., x_{a_{M_2}}\}. \tag{2}$$

Importantly, the quantity of collected anomalies is significantly less than that of normal samples, that is $M_2 \ll M_1$. Additionally, the available types of anomalies are limited, with certain anomaly types remaining unobtainable. Given this data scenario, we present a novel method for estimating the distinct distribution of normal features by leveraging such incomplete anomalous information.

Fig 1 illustrates the comprehensive framework of our proposed method GMMAD, comprising primarily the fusion module and the Gaussian Mixture Model. Subsequently, we will provide detailed explanations of these components.

A. Extracting Features

To determine whether an image x is anomalous, it is crucial to obtain a distinct representation for each individual sample. To achieve this, we leverage the pre-trained ResNet [16] to extract its feature maps from various levels denoted as $\{F^\ell\}_{\ell=1}^{L}$, where F^ℓ represents the feature map generated by the ℓ-th block of ResNet, with L indicating the total count of blocks in the pre-trained ResNet, typically set to 4. As ResNet blocks tend to produce more abstract features as they progress deeper into the network, the selection of relevant features is crucial for effective anomaly detection based on the specific characteristics of the detection target under consideration. For instance, in scenarios where normal and anomalous samples exhibit distinct dissimilarities (e.g., different categories of images), higher-level features are generally favored, as they are presumed to encapsulate a more semantic-level representation. Conversely, when the disparities between normal and anomalous samples are subtle (e.g., minor tubercles or small defects), lower-level features are more suitable due to the finer details they preserve. In this work, we aim to detect medical images with various diseases, and there are lesions of different sizes, features maps are extracted from blocks 3 and 4 of ResNet18, which contain detailed but also semantic information, and we denote them as $F^3 \in R^{C\times 4H\times 4W}, F^4 \in R^{2C\times 2H\times 2W}$. The channel dimension of F^4 is twice of F^3 while the height and weight are half of F^3.

To effectively combine the two feature maps containing information at different levels, we introduce a feature fusion module $g(\cdot)$ to merge and map these features into a lower-dimensional space for distribution estimation. Specifically, the module $g(\cdot)$ is composed of a projector $P(\cdot)$ and an integrator $I(\cdot)$. Given the two feature maps, we initially utilize the projector to downsample the larger feature map F^3 to match the size of the small one. The projector $P(\cdot)$ is simply implemented by a 3×3 convolutional layer with stride 2, followed by instance normalization and ReLU activation. This process yields $\tilde{F^3} = P(F^3) \in R^{2C\times 2H\times 2W}$. Subsequently, we concatenate $\tilde{F^3}$ and F^4 along the channel dimension to obtain $F = [F^3; F^3] \in R^{4C\times 2H\times 2W}$. Since F is the 2D feature map and the feature obtained by simple concatenation of the two feature maps lacks distinctiveness between normal and anomalous samples, the integrator $I(\cdot)$ is designed to extract more suitable features for anomaly detection. Instead of simple implementation of $P(\cdot)$, $I(\cdot)$ comprises multiple

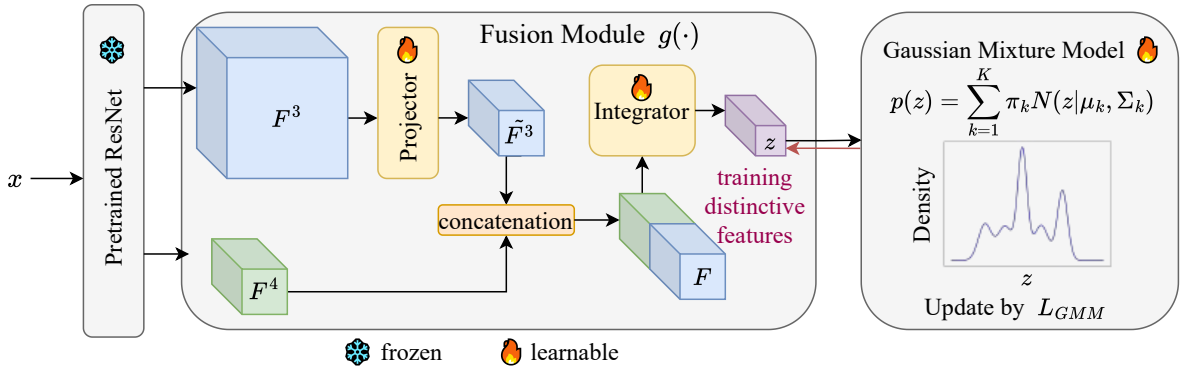


Fig. 1. Framework of proposed GMMAD. Feature maps from layers 3 and 4 of pre-trained ResNet18 are first extracted, then they are fused by the fusion module to produce the sample feature z. Gaussian Mixture Model is used to model the distribution of normal features, then it is used to guide the fusion module to learning distinct features of normal and abnormal samples.

convolutional layers, one with a stride of 2 and another with a stride of 1. Additionally, residual connections are integrated into the convolutional layers to prevent gradient vanishing. By passing F into the integrator $I(\cdot)$, a newly produced feature map $\tilde{F} = I(F) \in R^{8C\times H\times W}$ can be obtained. This output is then aggregated through average pooling to derive a comprehensive representation $z \in R^{8C}$. As depicted in the Fusion Module of Figure 1. For simplicity, we denote the feature extraction process as $z = g(x)$.

B. Estimating the Distribution of Normal Features

After extracting suitable features for each sample, we employ Gaussian Mixture Distribution (GMM) to estimate the distribution of normal features. GMM, which is composed of multiple Gaussian distributions, can fit any type of distribution in theory. It is usually used for clustering or estimating distributions and can be written in the following form:

$$p(z) = \sum_{k=1}^{K} \pi_k N(z|\mu_k, \Sigma_k), \tag{3}$$

where $N(z|\mu_k, \Sigma_k)$ is the Gaussian distribution with mean μ_k and variance matrix Σ_k, can be represented as

$$N(z|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp(-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)), \tag{4}$$

while the π_k is the weight of the k-th distribution and $\sum_{i=1}^{K} \pi_k = 1, 0 \leq \pi_k \leq 1$. The optimal parameters of GMM can be updated by the Expectation Maximization (EM) algorithm, but in our method, we directly minimize the negative log-likelihood of normal feature probability and the loss can be computed by

$$L_{GMM} = -\frac{1}{|\mathcal{X}_n|} \sum_{x\in\mathcal{X}_n} \log p(z). \tag{5}$$

Since the variance matrix Σ_k is required to be semidefinite, $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$, we introduce another parameters $\tilde{\Sigma}_k$ and $\tilde{\pi}_k$, where $\Sigma_k = diag(|\tilde{\Sigma}_k|)$ and $\pi_k = \frac{|\tilde{\pi_k}|}{\sum_{k=1}^{K}|\tilde{\pi_k}|}$ are used to meet these requirements.

After training the parameters of GMM, we can use $p(z)$ to model the distribution of normal features. In our work, the features of normal samples will dynamically change during the training procedure, so the GMM will be iteratively updated to fit the distribution and the K is set to 10 in our method.

C. Distinguishing Anomalies by Estimated Score

After estimating the distribution of normal features, we leverage it to instruct the feature fusion module in extracting distinctive features that exhibit significant disparities between normal and abnormal samples. Initially, we define the score of a sample within the distribution as

$$S(x) = -\log p(z), \tag{6}$$

which equals the negative value of a sample's density within the estimated distribution. In order to enhance the differentiation between normal features and anomalous features, the scores of normal features are encouraged to be small while those of abnormal samples are required to be large. By incorporating these dual objectives, we derive the loss from the GMM model to refine the fusion module, encompassing the projector and integrator, through

$$L = \frac{1}{|\mathcal{X}_n|} \sum_{x_n\in\mathcal{X}_n} S(x_n) - \lambda \frac{1}{|\mathcal{X}_a|} \sum_{x_a\in\mathcal{X}_a} S(x_a) \tag{7}$$

D. Training and Inference

As Algorithm 1 shows, our approach involves iterative training of the GMM and fusion model. Initially, we utilize normal features to train the GMM, which subsequently guides the fusion module in generating distinct features for normal and anomalous samples, facilitating effective anomaly detection.

During the inference stage, the score defined in Equation (6) serves as the anomaly score for anomaly detection. Samples with higher scores are indicative of being more likely to be anomalous.

TABLE I
PERFORMANCE (AVERAGE AUC(%), BEST F1(%)) AND ACC(%)) ON 3 MEDICAL DATASETS. WITH $t = 1$, THE RATIO OF LABELED ANOMALIES r_l IN THE TRAINING IS SET TO 1%, 2% AND 5%.

r_l	Methods	ISIC2018			Chest X-ray			OCT		
		AUROC	F1	ACC	AUROC	F1	ACC	AUROC	F1	ACC
1%	PReNet	73.84	67.02	69.26	86.66	84.83	80.45	60.26	85.88	75.83
	Deep SAD	89.52	77.13	80.83	86.82	83.4	79.33	93.04	90.93	86.50
	AA-BiGAN	87.83	78.96	83.25	90.63	87.52	84.13	96.02	94.41	91.70
	DRA	90.10	78.51	82.04	92.35	88.36	85.59	91.52	91.34	86.83
	DevNet	88.98	77.80	80.83	92.53	**89.41**	85.57	99.38	98.75	98.13
	GMMAD(Ours)	**90.99**	**80.27**	**85.32**	**93.42**	88.44	**85.26**	**99.50**	**98.82**	**98.30**
2%	PReNet	74.28	66.87	68.83	89.69	87.47	84.29	65.25	85.98	75.90
	Deep SAD	89.86	79.18	83.94	90.16	86.56	83.97	95.26	93.18	90.00
	AA-BiGAN	88.20	78.10	83.25	92.13	89.86	87.02	97.15	95.44	93.20
	DRA	90.65	79.70	82.80	92.59	89.43	86.69	94.98	92.94	89.50
	DevNet	89.36	78.80	83.25	94.27	90.48	87.66	99.31	98.56	97.83
	GMMAD(Ours)	**91.29**	**81.04**	**85.49**	**94.89**	**90.57**	**88.62**	**99.70**	**99.12**	**98.44**
5%	PReNet	74.29	67.36	71.24	91.01	87.82	83.81	67.67	86.27	76.47
	Deep SAD	89.81	78.29	82.38	92.87	88.49	85.58	95.74	93.88	91.03
	AA-BiGAN	88.91	78.95	82.30	92.80	89.95	86.86	97.69	96.04	94.10
	DRA	91.05	79.80	83.07	93.61	89.59	86.86	91.91	91.39	86.57
	DevNet	89.93	78.63	82.64	95.77	91.23	88.94	99.60	98.93	98.40
	GMMAD(Ours)	**91.54**	**81.17**	**85.75**	**96.01**	**92.39**	**90.38**	**99.78**	**99.31**	**98.97**

Algorithm 1 Training procedure of GMMAD

Initialize parameters of the fusion module $g(\cdot)$ and GMM.
for each epochs **do**
  for batch samples $X_n \subset \mathcal{X}_n$, $X_a \subset \mathcal{X}_a$ **do**
    $z_n \leftarrow g(x_n)$, $x_n \in X_n$
    $L_{GMM} \leftarrow -\frac{1}{|X_n|}\sum_{x_n \in X_n} \log p(z_n)$.
    update parameters of GMM by L_{GMM}
    $z_n \leftarrow g(x_n)$, $x_n \in X_n$
    $z_a \leftarrow g(x_a)$, $x_a \in X_a$
    $S(x_n) \leftarrow -\log p(z_n)$
    $S(x_a) \leftarrow -\log p(z_a)$
    $L = \frac{1}{|X_n|}\sum_{x_n \in X_n} S(x_n) - \lambda \frac{1}{|X_a|}\sum_{x_a \in X_a} S(x_a)$
    update parameters of $g(\cdot)$ by loss L
  end for
end for

IV. EXPERIMENTS

A. Experimental Settings

1) Datasets: We mainly conduct experiments on 3 real-world medical datasets to detect anomalies: *i)* **Chest X-rays** [17]: The dataset contains Chest X-rays scans of healthy and other unhealthy images. *ii)* **ISIC2018** [18], [19]: The ISIC2018 challenge dataset (task 3) contains 7 categories and the NV (nevus) is classified as normal samples, and the rest 6 categories are regarded as abnormal data. *iii)* **OCT** [17]: Retinal optical coherence tomography (OCT) contains normal OCT scans and 3 other types of scans with diseases.

2) Metrics: We use the AUROC (Area Under Receiver Operating characteristic Curve), and best F1 and Accuracy(ACC) based on produced anomaly scores to evaluate the anomaly detection performance.

3) Parameters: We use Adam to optimize the parameters of GMM and the fusion module, and the learning rate is set to 1e-04. Besides, the mixed distribution number is set to $K = 10$. λ in (7) is set to 1.

4) Experimental Scenarios: Considering the ratio r_l and the types t of possible labeled anomalies in the training set, we conduct experiments under two scenarios:

1) $t = 1$, performance of $r_l \in \{1\%, 2\%, 5\%\}$ is evaluated.
2) $r_l = 5\%$, performance of $t \in 1, 2, 3$ is evaluated.

B. Results

We mainly compare the proposed method with recent weakly-supervised anomaly detection methods Deep SAD [9], DevNet [10], AA-BiGAN [11], DRA [2] and PReNet [20].

1) Detection Results Under Scenarios 1): As shown in TABLE I, with setting $t = 1$, we increase the ratio of labeled anomalies in the training set from 1%, 2% to 5%. The results show that increasing the number of labeled anomalies in the training set can help improve the detection performance. Besides, only utilizing a small number of anomalous samples, our method shows its effectiveness in detecting anomalies. Compared to other weakly-supervised anomaly detection methods, our proposed GMMAD achieves superior performance across all three metrics on the three medical datasets.

2) Detection Results Under Scenarios 2): Since there are 6 anomalous types in ISIC2018, we mainly conduct experiments on this dataset to assess the performance of different labeled anomalous types t in the training set. As shown in TABLE II, with setting $t = 1$, we increase the types of labeled anomalies in the training set to 2 and 3. Combining the results in TABLE I, the results show that increasing the types of labeled anomalies in the training set can also help

TABLE II
PERFORMANCE (AVERAGE AUC(%) AND F1(%)) OF DIFFERENT t ON ISIC2018 $r_l = 5\%$.

t	2			3		
Metric	AURUC	F1	ACC	AURUC	F1	ACC
Deep SAD	91.21	81.12	84.87	92.40	81.90	86.01
AA-BiGAN	88.14	77.73	82.38	90.72	82.00	86.22
DRA	91.21	79.69	83.73	93.66	84.11	87.77
DevNet	92.69	80.54	84.97	94.04	83.01	86.96
GMMAD(Ours)	**93.79**	**82.83**	**86.58**	**95.02**	**84.76**	**89.12**

TABLE III
ABLATION STUDY OF UTILIZING ANOMALIES.

Loss	Chest X-ray			ISIC2018		
	AURUC	F1	ACC	AURUC	F1	ACC
(8)	85.86	83.98	78.36	72.02	63.59	59.06
(7)	**96.01**	**92.39**	**90.38**	**91.54**	**81.17**	**85.75**

improve the detection performance. The AUROC of our proposed method GMMAD is 91.54%, 93.79%, and 95.02% for $t = 1, 2, 3$ respectively. Compared to other weakly-supervised methods, the results show GMMAD's superiority in utilizing such incomplete anomalous information for better detecting anomalies.

C. Abalation Study

To evaluate the effectiveness of GMMAD in utilizing anomalies, we also conduct an ablation study under the setting that $t = 1$ and $r_l = 5\%$. Compared to (7), we conduct ablation experiments using loss

$$L' = \frac{1}{|\mathcal{X}_n|} \sum_{x_n \in \mathcal{X}_n} S(x_n), \tag{8}$$

which is not utilizing anomalous samples. From the results shown in TABLE III, we can see that the proposed loss (7) can effectively leverage the collected incomplete anomalous information to better detect anomalies.

V. CONCLUSION

In this paper, we proposed a new weakly-supervised method GMMAD to detect anomalies. To obtain a good representation, we first extract two feature maps from the pre-trained ResNet and then use the fusion module to fuse the different scale maps. The projector in the fusion module first downsamples the larger maps into the same size as the smaller ones, then they are concatenated in the channel and fed into the integrator to produce the final feature representation. In order to model the distribution of normal features, the Gaussian Mixture Model is trained by negative log-likelihood, and then it is used to guide the fusion module to learn distinct feature representations of samples. After training the GMM and fusion module iteratively, we can detect anomalies according to the anomaly score produced by GMM. The experiments on 3 medical datasets have demonstrated the effectiveness of the proposed GMMAD in leveraging incomplete anomalous information.

REFERENCES

[1] Y. Zhao, Q. Ding, and X. Zhang, "AE-FLOW: Autoencoders with normalizing flows for medical images anomaly detection," in *The Eleventh International Conference on Learning Representations*, 2023.

[2] C. Ding, G. Pang, and C. Shen, "Catching both gray and black swans: Open-set supervised anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7388–7398.

[3] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, 2018, pp. 4393–4402.

[4] J. Yi and S. Yoon, "Patch svdd: Patch-level svdd for anomaly detection and segmentation," in *Proceedings of the Asian conference on computer vision*, 2020.

[5] X. Gui, D. Wu, Y. Chang, and S. Fan, "Constrained adaptive projection with pretrained features for anomaly detection," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2022, pp. 2059–2065.

[6] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: novelty detection via contrastive learning on distributionally shifted instances," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, pp. 11 839–11 852.

[7] H. Cho, J. Seol, and S.-g. Lee, "Masked contrastive learning for anomaly detection," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021, pp. 1434–1441.

[8] T. Reiss and Y. Hoshen, "Mean-shifted contrastive loss for anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, pp. 2155–2162.

[9] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," in *International Conference on Learning Representations*, 2019.

[10] G. Pang, C. Ding, C. Shen, and A. v. d. Hengel, "Explainable deep few-shot anomaly detection with deviation networks," *arXiv preprint arXiv:2108.00462*, 2021.

[11] B. Tian, Q. Su, and J. Yin, "Anomaly detection by leveraging incomplete anomalous knowledge with anomaly-aware bidirectional gans," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2022, pp. 2255–2261.

[12] G. Pang, C. Shen, H. Jin, and A. van den Hengel, "Deep weakly-supervised anomaly detection," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1795–1807.

[13] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 318–14 328.

[14] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *14th Asian Conference on Computer Vision*, 2019, pp. 622–637.

[15] V. Zavrtanik, M. Kristan, and D. Skočaj, "Reconstruction by inpainting for visual anomaly detection," *Pattern Recognition*, vol. 112, p. 107706, 2021.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[18] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.

[19] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti *et al.*, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1902.03368*, 2019.

[20] H. Zhang, Z. Wu, Z. Wang, Z. Chen, and Y.-G. Jiang, "Prototypical residual networks for anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 281–16 291.

A local-mean based pseudo nearest neighbor method

1st Dapeng Li
School of software engineering
Jinling Institute of Technology
Nanjing, China
lidp401@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899653

***Abstract*—The k-nearest neighbor method is a fundamental machine learning algorithm for classification. However, when the sample size is small and outliers exist, the classification performance of this algorithm may be degraded. To solve the issue, a local-mean based pseudo nearest neighbor method (LAPNN) is introduced to reduce the influence of outliers for classification. In the LAPNN method, the local mean categorical points are obtained by computing the mean of the random two points in the first $2k$ nearest neighbors of every class. Then, the first k pseudo neighbors are selected from the local mean points of each category to decide the pattern of the test sample. The local mean points can improve classification performance to some extent. Some experiments are performed on 19 numerical real data sets, comparing LAPNN with the other ten classification rules. The experimental results show the efficacy of LAPNN in the presence of small-scale points with outliers.**

***Index Terms*—local-mean, nearest neighbors, classification, outliers**

I. Introduction

The k-Nearest Neighbor (kNN) method is a relatively mature classification algorithm. It is one of the simplest machine learning algorithms. Its idea is that if the majority of the k nearest points, i.e., the most neighboring points in the feature space around a test point belong to a category, then the test point also belongs to that class. The k-nearest neighbor (kNN) method is an effective yet simple classification algorithm. The algorithm is easy to understand and implement, without requiring a complex training process. Therefore, kNN has received extensive research attention, and many classification methods using kNN idea have been introduced [1]–[12]. For the kNN rule, only a single parameter, k, can be selected. When the too small or too large value of k is chosen, the classification performance may be degraded due to noise points and outliers [13], [14].

To solve the aforesaid issue, several effective approaches have been suggested to mitigate the adverse effects of outliers. One of the methods is the local mean k-nearest neighbor (LMkNN) method proposed by [15]. In the LMkNN rule, the categorical mean points helps to counteract the influence of outliers. Another classic approach is the pseudo nearest neighbor (PNN) rule [16]. The PNN approach builds upon the concept of the weighted kNN (WkNN) method [17]. It utilizes the k nearest neighbor information of each class to decide the category of the test sample. Besides that, an enhanced PNN rule, known as LMPNN,has been reported [18]. The negative impact of outliers on classification is further reduced. In the LMkNN rule, only using one local mean point of each class to make decisions. However, k pseudo nearest neighbors for every class are employed to predict the unclassified sample in the LMPNN rule. Hence, the overall classification performance surpasses that of LMkNN, kNN, and PNN. Besides that, some methods, which are derived from the aforementioned method, have been reported [2], [13], [19]–[27] to improve classification accuracy rate.

Although the above methods are effective in reducing the influence of outliers, how to more effectively mitigate the negative impact of outliers remains a question worthy of investigation. In this paper, $2k(2k-1)/2$ local-averaged pseudo vectors from the first $2k$ nearest neighbors of the unclassified samples in each class are used. Then, the category of the query sample is determined by comparing the distance sum of every class. The first k pseudo neighbors are obtained by averaging pair points from a number of neighbors of every class, the influence of outliers on classification may be further reduced. In this paper, $2k$ nearest neighbors are chosen from every class. Contrary to PNN approach, the LAPNN rule utilizes the k-nearest neighbors derived from the local-mean points in every category, allowing for a more precise determination of the pattern associated with the test sample. The first *k* local-mean points may mitigate the adverse effects of outliers to some degree, thereby improving the effectiveness of classification. The effectiveness and advantages of the LAPNN rule have been confirmed through experimental evaluations on numerical datasets.

The content of this paper is as follows: 1. A classifier utilizing pseudo nearest neighbors based on local averages is introduced. Firstly, $2k(2k-1)/2$ local mean points are calculated by averaging two random samples from the first $2k$ nearest neighbors of the test point for every category. Secondly, the first k pseudo nearest neighbors among the $2k(2k-1)/2$ local-averaged points of each class are chosen in the LAPNN rule. The chosen k pseudo nearest neighbors are somewhat not very sensitive to outliers. Therefore, compared with the traditional kNN, the better classification performance can be obtained by the LAPNN method in the small-sized samples with the presence of outliers. 2. The effectiveness of LAPNN is verified by experiments.

In this paper, the rest of the content is as follows. The LAPNN method is introduced in Section II. In Section III, Some experiments on numerical datasets are conducted. A concise conclusion is presented in Section IV.

II. The proposed LAPNN

A local mean based pseudo nearest neighbor classification method based on the idea of average is introduced in this section. The LAPNN method aims to enhance classification performance, especially when dealing with small-sized training samples that contain outliers.

A. The Basic Thoughts

In the kNN method, outliers tend to impact classification accuracy, particularly when dealing with small scale points. Hence, How to choose the value k is the key to determine the classification performance. With a small k, outliers may be erroneously selected as neighbors, potentially compromising classification result [13]. Conversely, a larger k value, in certain scenarios, may degrade the classification accuracy by incorporating numerous samples from different classes [13].

PNN, LMkNN, and LMPNN algorithms have demonstrated their efficacy in enhancing classification performance, particularly in the situation of small scale points with existing outliers. Specifically, LMkNN has the capacity to mitigate the adverse effects of outliers to a certain extent. However, the uniformity in selecting the number of the nearest neighbors and the weight coefficients for each class may potentially hinder the classification performance [2], [13]. On the other hand, the PNN method assigns a larger weight coefficient to the nearer neighbor, thereby minimizing the detrimental influence of outliers to some degree. Nevertheless, outliers still pose a challenge to classification accuracy when k is excessively high. LMPNN utilizes the k pseudo local mean vectors, which are computed based on the k nearest neighbors of each class, to classify the test points. In comparison to LMkNN, LMPNN gathers more comprehensive class-specific details, ultimately leading to better classification results.

Inspired by the methods mentioned above, a pseudo nearest neighbor classification method, derived from the idea of multi-averages, is presented to further mitigate the negative impact in the case of small-scale samples with existing outliers.

B. LAPNN rule

Let $G = (y_j, c_i)_{j=1}^{n}$ denote a training dataset consisting of n points, each belonging to one of W classes. Here, $y_j \in R^d$ where d denotes the number of feature dimensions, and the integer variable i ranges from 1 to W. The subset $G_i = (y_{ij}, c_i)_{j=1}^{n_i}$ comprises the training samples that belongto class c_i, where n_i denotes the number of training samples in that particular class. The classification of a test sample y is determined in the LAPNN method through the following processes:

1. Compute the Euclidean distance between the training sample y_{ij} and the test sample y for every class:

$$d\left(y, y_{ij}\right) = \sqrt{\left(y - y_{ij}\right)^T \left(y - y_{ij}\right)}, \tag{1}$$

where y_{ij} denotes the jth sample belonging to class ω_i, with $i = 1, 2, 3, ..., W$, and W being the number of categories. Then, by sorting the distances $d(y, y_{ij})$ in ascending order for class ω_j, the first $2k$ nearest distances d_1^j, d_2^j, ..., d_{2k}^j can be obtained.

the first $2k$ nearest distances d_1^j, d_2^j, ..., d_{2k}^j of class ω_j can be obtained by sorting $d(y, y_{ij})$ in ascending order.

2. calculate the local averaged points for each class using the following equation:

$$\begin{aligned} &x_{ip} = (y_{ij} + y_{iv})/2, j = 0, 1, ..., (number - 1), \\ &v = (j+1), ..., number, p = 1, 2, ..., number(number-1)/2, \end{aligned} \tag{2}$$

where x_{ip} represents the pth local mean point of class c_i, $number = 2k$ with $n_i >= 2k$, otherwise, $number = n_i$ with $n_i < 2k$. Additionally, notice that $y_{i1} = x_{ip}$ when $n_i = 1$.

3. Compute the distances between the local mean points of each class c_i and y using the following equation:

$$d(y, x_{ip}) = \sqrt{(y - x_{ip})^T (y - x_{ip})}. \tag{3}$$

Then, the first k distances, denoted as $d\left(y, \overline{x_{i1}}\right)$, $d\left(y, \overline{x_{i2}}\right)$, ... , $d\left(y, \overline{x_{ik}}\right)$, can be obtained by ordering the distance $d(y, x_{ip})$ from smallest to largest.

4. Let x_i^{LAPNN} represent the nearest neighbor of y within a specific category c_i. The weighted sum of distances between the test point y and x_i^{LAPNN} across all classes, is calculated as:

$$d\left(x_i^{LAPNN}, y\right) = \sum_{p=1}^{k} \frac{1}{p} d\left(\overline{x_{ij}}, y\right). \tag{4}$$

5. Assgin y to the class c_i based on the shortest distance:

$$c_i = argmind\left(x_i^{LAPNN}, y\right). \tag{5}$$

As mentioned above, the LAPNN rule is outlined in Algorithm 1.

Overview of the LAPNN Method.

Require:

y : a test sample.

k : the count of nearest neighbors.

$Y = y_1, y_2, ..., y_n$: the training samples.

$Y_i = y_{i1}, y_{i2}, ..., y_{in_i}$: denotes a subset of training data for class c_i, containing n_i samples.

C = c_1, c_2, ... , c_W: the collection of W classes.

Ensure:

The category of the test sample y.

Step1: Compute the distance between the training point y_{ij} and a test point y for every class:.

for i = 1 to W

  for j = 1 to n_i

    $d\left(y, y_{ij}\right)$ = $\sqrt{\left(y - y_{ij}\right)^T \left(y - y_{ij}\right)}$;

  end for

end for

Subsequently, the first $2k$ distances d_1^i, d_2^i, ..., d_{2k}^i of class ω_i are chosen by ordering the distance $d(y, y_{ij})$ from

smallest to largest.

Step2: Obtain the local mean points for every class.

for i = 1 to W **do**
  $number = 0$, $p = 0$;
  if $n_i >= 2k$
    $number = 2k$
  else
    $number = n_i$
  for j = 0 to $(number - 1)$
    for v = j+1 to $number$
      x_{ip} = $(y_{iv} + y_{ij})/2$, $p++$;
    end for
  end for
end for

Step3: Compute the distances between y and the local averaged points for each class c_i.

for p = 1 to $(number(number - 1)/2)$
  $d(x_{ip}, y) = \sqrt{(x_{ip} - y)^T (x_{ip} - y)}$
end for

Then, the first k local mean based nearest samples of class c_i can be found, namely, x_i^{LAPNN} =$\{\overline{x_{i1}}, \overline{x_{i2}}, \ldots, \overline{x_{ik}}\}$.

Step4: Compute the weighted distance sum between the test sample y and the first k local mean based nearest neighbors for each category.

  for i = 1 to W
    for j = 1 to k
      $d\left(x_i^{LAPNN}, y\right)$ += $\frac{1}{j} d\left(y, \overline{x_{ij}}\right)$
    end for
  end for

Step5: The test point y is assigned to the class that has the shortest distance.

c_i^y = $argmind\left(x_i^{LAPNN}, y\right)$.

III. Experimental Evaluation

To evaluate the effectiveness of the LAPNN method, some experiments are performed on the numerical datasets.

A. Datasets

This subsection presents the details of the datasets employed in the experiments.

The nineteen numerical datasets, encompassing diverse domains such as Pima, Steel, Wine(keel), Mammographic, Vehicle, Blood, Balance, Bupa, Ionosphere, Parkinsons, Pima-indians, Haberman-survival, Hill, Sonar, Cardiotocography, Wine, Band, Musk-1, and QSAR are obtained from the KEEL Repository [30] and the UCI Machine Learning Repository [29]. Table I provides a comprehensive overview, encompassing the total number of points, classes, attributes, sources, and indications of any imbalance. It can be observed from Table I that these datasets have various characteristics in terms of their attribute count, sample size, and class distribution. Most of the points in the datasets mentioned above have a relatively small number. Hence, these datasets are appropriate for assessing the LAPNN method in scenarios with the small-size samples.

TABLE I
THE DATASETS UTILIZED FOR THE EXPERIMENTAL PROCESS.

Data	Samples	Attributes	Classes	Source	Imbalance
Pima	768	8	2	KEEL	yes
Steel	1941	27	7	UCI	yes
Wine(keel)	178	13	3	KEEL	yes
Mammographic	830	5	2	KEEL	yes
Vehicle	846	18	4	UCI	yes
Blood	748	4	2	UCI	yes
Balance	625	4	3	UCI	yes
Bupa	345	6	2	UCI	yes
Ionosphere	351	34	2	UCI	yes
Parkinsons	195	22	2	UCI	yes
Pima-indians	768	8	2	UCI	yes
Haberman-survival	306	3	2	UCI	yes
Hill	1210	100	2	UCI	no
Sonar	208	60	2	UCI	yes
Cardiotocography	2126	21	10	UCI	yes
Wine	178	13	3	UCI	yes
Band	365	19	2	KEEL	yes
Musk-1	476	166	2	UCI	yes
QSAR	1055	41	2	UCI	yes

B. Experiments on the numerical datasets

Cross validation is a commonly used technique for assessing the performance of machine learning method. Hence, the experiments utilize a 10-fold cross-validation approach. To ensure the accuracy and reliability of the experimental outcomes, 10 runs are conducted, and determine the mean results for each value of k, which varies from 1 to 20 with an increment of 1. The final classification are derived by averaging the outcomes obtained for k values ranging from 1 to 20.

A comprehensive comparison of LAPNN with various alternative approaches, including kNN [28], WkNN [17], LMkNN [15], PNN [16], LMPNN [18], fuzzy kNN (FkNN) [8], eigenvalue classification method (EigenClass) [11], generalized mean distance based k nearest neighbor method (GMDkNN) [24], bonferroni mean based fuzzy k nearest neighbor method (BMFkNN) [12], and fuzzy parameterized and soft k-nearest neighbor method (FPFS-kNN) [3], is conducted.

The algorithms are evaluated using Accuracy, Recall, Precision, and F1-Score. Table II–Table III provide a succinct listing of the highest-performing results, followed by a comparison of these rankings in a pairwise manner. Notice that the top-ranked outcomes are highlighted in bold within Table III. Upon reviewing Table III, it is evident that the LAPNN method consistently ranks highest among the evaluated algorithms for 19 datasets. Furthermore, Table III reveals that in every pairwise comparison metric, LAPNN is better than other ten algorithms for at least 11 datasets. The results above show that LAPNN method is better than other methods in general.

IV. Conclusion

In this paper, a pseudo nearest neighbor method based on local-mean (LAPNN) is introduced to improve classification performance, particularly when dealing with small scale training points with outliers. The LAPNN method, grounded in the concept of averaging, effectively mitigates the detrimental

TABLE II
RANKING NUMBER OF THE BEST RESULTS BETWEEN TWO ALGORITHMS' COMPARISONS.

Methods	Accuracy	Precision	Recall	F1-Score
LAPNN vs PNN	18	17	18	18
LAPNN vs LMkNN	17	17	17	17
LAPNN vs LMPNN	17	17	17	17
LAPNN vs FkNN	18	19	17	18
LAPNN vs kNN	17	17	17	17
LAPNN vs WkNN	17	17	17	18
LAPNN vs FPFS-kNN	16	16	16	16
LAPNN vs EigenClass	15	17	15	17
LAPNN vs GMDkNN	17	17	17	17
LAPNN vs BMFkNN	18	18	18	18

TABLE III
RANKING NUMBER OF THE BEST RESULTS AMONG ALL ALGORITHM COMPARISONS.

Methods	Accuracy	Precision	Recall	F1-Score	Total Rank
PNN	0	1	0	0	1
LMPNN	0	0	0	0	0
LAPNN	**12**	**11**	**12**	**12**	**47**
FkNN	0	0	0	0	0
kNN	1	0	1	0	2
WkNN	0	0	0	0	0
LMkNN	0	1	0	1	2
FPFS-kNN	2	2	2	2	8
EigenClass	2	2	2	2	8
GMDkNN	2	2	2	2	8
BMFkNN	0	0	0	0	0

effects of these outliers to some degree. The effectiveness of LAPNN is evaluated by comparing with various state-of-the-art classifiers, including kNN, WkNN, LMkNN, PNN, FPFS-kNN, BMFkNN, LMPNN, GMDkNN, EigenClass, and FkNN. The methods underwent training and testing in ten iterations, employing a ten fold cross-validation technique across 19 real datasets. The experimental results reveal that LAPNN has overall superior classification capabilities.

REFERENCES

[1] B. Li, Y.W. Chen, and Y.Q. Chen, "The Nearest Neighbor Algorithm of Local Probability Centers," IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics).2008,38(1):141–154.

[2] Z. Pan, Y. Wang, and W. Ku, "A new k-harmonic nearest neighbor classifier based on the multi-local means," Expert Systems with Applications.2017,67:115–125.

[3] S. Memis, S. Enginoğlu, and U. Erkan, "Fuzzy parameterized fuzzy soft k-nearest neighbor classifier," Neural Computing and Applications.2022,500:351–378.

[4] Y. Zeng, Y. Yang, and L. Zhao, "Nonparametric classification based on local mean and class statistics," Expert Systems with Applications.2009,36(4):8443–8448.

[5] Z. Chai, Y. Li, A. Wang, C. Li, B. Zhang, and H. Gong, "An Efficient Pseudo Nearest Neighbor Classifier," IAENG International Journal of Computer Science.2021,48(4):1075–1086.

[6] S. Memis, S. Enginoğlu, and U. Erkan, "A classification method in machine learning based on soft decision-making via fuzzy parameterized fuzzy soft matrices," Soft Computing.2022,26:1165–1180.

[7] S. Memis, "Picture Fuzzy Soft Matrices and Application of Their Distance Measures to Supervised Learning: Picture Fuzzy Soft k-Nearest Neighbor (PFS-kNN)," Electronics.2023,12(19):4129.

[8] J. M. Keller, M. R. Gray, and J. A. Givens, "A Fuzzy K-Nearest Neighbor Algorithm," IEEE Transactions on Systems Man & Cybernetics.1985,4:580–585.

[9] S. S. Mullick, S. Datta, and S. Das, "Adaptive Learning-Based k-Nearest Neighbor Classifiers With Resilience to Class Imbalance," IEEE Transactions on Neural Networks & Learning Systems.2018,29(11):5713–5725.

[10] S. Memis, S. Enginoğlu, and U. Erkan, "A new classification method using soft decision-making based on an aggregation operator of fuzzy parameterized fuzzy soft matrices," Turkish Journal of Electrical Engineering & Computer Sciences.2022,30(3):871–890.

[11] U. Erkan, "A precise and stable machine learning algorithm: eigenvalue classification (EigenClass)," Neural Computing and Applications.2021,33(10):5381–5392.

[12] M. M. Kumbure, P. Luukka, and M. Collan, "A new fuzzy k-nearest neighbor classifier based on the Bonferroni mean," Pattern Recognition Letters.2018,140:172–178.

[13] J. Gou, W. Qiu, Y. Zhang, Y. Xu, Q. Mao, and Y. Zhan, "A Local Mean Representation-based K-Nearest Neighbor Classifier," ACM Transactions on Intelligent Systems and Technology.2019,10(3):1–25.

[14] S. Zhang, "Challenges in KNN classification," IEEE Transactions on Knowledge and Data Engineering.2022,34(10):4663–4675.

[15] Y. Mitani, and Y. Hamamoto, "A local mean-based nonparametric classifier," Pattern Recognition Letters.2006,27(10):1151–1159.

[16] Y. Zeng, Y. Yang, and L. Zhao, "Pseudo nearest neighbor rule for pattern classification," Expert Systems with Applications.2009,36(2):3587–3595.

[17] Dudani, S. A., "The Distance-Weighted k-Nearest-Neighbor Rule," IEEE Transactions on Systems, Man, and Cybernetics.1976,6(4):325–327.

[18] J. Gou, Y. Zhan, Y. Rao, X. Shen, X. Wang, and W. He, "Improved pseudo nearest neighbor classification," Knowledge-Based Systems.2014,70:361–375.

[19] Y. Xu, Q. Zhu, Z. Fan, M. Qiu, Y. Chen, and H. Liu, "Coarse to fine K nearest neighbor classifier," Pattern Recognition Letters.2013,34(9):980–986.

[20] J. Gou, Y. Zhang, L. Du, and T. Xiong, "A Local Mean-Based k-Nearest Centroid Neighbor Classifier," Computer Journal.2012,55(9):21–27.

[21] J. Gou, W. Qiu, Y. Zhang, X. Shen, Y. Zhan, and W. Ou, "A representation coefficient-based k-nearest centroid neighbor classifier," Expert Systems With Applications.2022,194(15):38–52.

[22] J. Gou, W. Qiu, Y. Zhang, X. Shen, Y. Zhan, and W. Ou, "Locality constrained representation-based K-nearest neighbor classification," Knowledge-Based Systems.2019,167(1):38–52.
[23] Y. Ma, R. Huang, M. Yan, G. Li, and T. Wang, "Attention-based Local Mean K-Nearest Centroid Neighbor Classifier," Expert Systems with Applications.2022,201:117159.
[24] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, "A generalized mean distance-based k-nearest neighbor classifier," Expert Systems with Applications.2019,115:356–372.
[25] Z. Pan, Y. Pan, Y. Wang, and W. Wang, "A new globally adaptive k-nearest neighbor classifier based on local mean optimization," Soft Computing.2021,25:2417–2431.
[26] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN Classification With Different Numbers of Nearest Neighbors," IEEE Transactions on Neural Networks and Learning Systems.2019,29(5):1774–1785.
[27] C. Gong, Z.-G. Su, P.-H. Wang, Q. Wang, and Y. You, "A Sparse Reconstructive Evidential K-Nearest Neighbor Classifier for High-Dimensional Data," IEEE Transactions on Knowledge and Data Engineering.2023,35(6):5563–5576.
[28] T. Cover, and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory.1967,13(1):21–27.
[29] K. Bache, and M. Lichman, "UCI Machine Learning Repository," 2013.
[30] Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., Garála, S., Sánchez, L., and Herrera, F., "Keel Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," Journal of Multiple-Valued Logic and Soft Computing.2011,17(2-3):255–287.

ST-DiffTraj: A Spatiotemporal-Aware Diffusion Model for Trajectory Generation

Bo Wang
Zhejiang Normal University
Zhejiang, China
2022205001071@zjnu.edu.cn

Juan Yu*
Zhejiang Normal University
Zhejiang, China
yujuan@zjnu.edu.cn
* Corresponding author

Jianmin Han
Zhejiang Normal University
Zhejiang, China
hanjm@zjnu.cn

Sheng Qiu
Zhejiang Normal University
Zhejiang, China
qiusheng@zjnu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899629

***Abstract*—Synthetic trajectory generation has significant applications in urban planning, mobility analysis, smart city development, etc. Diffusion models excel at generating high-quality synthetic data and are widely used in trajectory data generation. However, existing diffusion-based trajectory generation models face the following limitations: (1) ignoring temporal sequence information, and (2) inefficient use of conditional guidance. To address these issues, we propose a SpatioTemporal-Aware Diffusion Model for Trajectory Generation, named ST-DiffTraj. Specifically, we design a SpatioTemporal Encoder (STEncoder) to capture correlations between independent and joint spatiotemporal feature distributions by introducing temporal information. Additionally, we develop a noise estimation network by stacking multiple CDAttn-DConv (Condition-Dependent Attention Dilation Convolutional) layers. Each CDAttn-DConv layer integrates attention mechanisms and gated activation units to enhance noise estimation with conditional guidance. Comprehensive experiments on two real-world trajectory datasets demonstrate the superior performance of our model in generating high-quality trajectory data compared to baselines.**

Keywords—trajectory generation, diffusion model, conditional guidance

I. INTRODUCTION

GPS trajectory data reveals human mobility patterns and improve our understanding of human movement regularity [1]. It is essential for various downstream applications such as urban planning, traffic management and epidemic control [2]. However, publicly available GPS trajectory data cannot be directly used due to inaccurate sampling and privacy concerns. To address these issues, trajectory generation has become a popular solution. It aims to generate synthetic trajectories by learning real trajectories' distribution [3].

However, generating high-quality trajectories poses considerable challenges due to the dynamic and high-dimensional nature of trajectories. In the early step, researcher used statistical models to simulate human movement [4] but they failed to capture complex flow patterns inherent in trajectories. Recently, we mainly use deep generation models like variational auto-encoder (VAE) and generative adversarial network (GAN) to generate trajectories [5]. They can capture more complicated distributions. However, they usually discrete trajectories into grid sequences, that makes models cannot capture critical details, reducing the quality of generated trajectories. Nowadays, diffusion model has demonstrated better generation quality and diversity than GAN in other fields. Therefore, it encourage researchers to apply diffusion model to trajectory generation, such as DiffTraj [6] and TrajGDM [7]. However, they still have the some issues. First, the sampling points within a trajectory display strong spatiotemporal correlations. However, they cannot capture spatiotemporal feature distribution due to the lack of temporal information. Secondly, they struggle to accurately extract noise-free distributions with conditional guidance because the process of conditional information is too simple.

To tackle these challenges, we propose ST-DiffTraj, A **S**patio**T**emporal-Aware **Diff**usion Model for **Traj**ectory Generation. And we design a new denoising network CTDNet, **C**ondition **T**rajectory **D**enoising **Net**work. CTDNet takes trajectories, diffusion step, and conditional information as input and output estimated noise. It consists of three main modules: STEncoder (**S**patio**T**emporal **Encoder**), a noise estimation module composed of multiple CDAttn-DConv(**C**onditional **D**ependent **Atten**tion **D**ilation **Conv**olutional) layers, and Wide\&Deep. STEncoder is designed with 1D convolution layers and a self-attention mechanism, which is used to capture the correlation information of spatial and temporal. And inspired by DiffiT [8], we design CDAttn(**C**onditional **D**ependent **Atten**tion) and combined it with gate activate unit (GAU) to build CDAttn-DConv. It can improve the accuracy of estimating noise. Wide&Deep embeds trajectory-related feature information using MLP and embedding layers.

In summary, the contributions of our research are as follows:

- We design STEncoder to capture the spatiotemporal correlations inherent in the data, which helps modeling the spatiotemporal complexity.
- We propose a new noise estimation network, aiming to effective use the guidance of conditional information. The network integrates CDAttn, which utilizes attention mechanism to extract essential features, and improving the noise prediction accuracy.
- We combine the above modules and Wide&Deep to propose a new denoising network CTDNet, and completing trajectory generation task through sampling noise from Gaussian distribution and step-by-step denoising.
- Experiments on two real-world datasets show that ST-DiffTraj demonstrates superior performance over baselines. Additionally, ablation experiments and

This research was funded by the National Natural Science Foundation of China under Grant No. 61702148 and Grant No. 61672648.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

downstream tasks help validate the effectiveness of our model.

II. PRELIMINNARIES

In this section, we introduce the definitions of trajectory and problem and present the fundaments of diffusion model.

A. Problem Definitions

Definition 1 (Trajectory): A Trajectory is a chronologically sequence of sampled location points, which can be denoted as $\boldsymbol{x} = [(p_1, t_1), (p_2, t_2), \ldots, (p_n, t_n)]$, where n is the length of the trajectory. For each i, $1 \leq i \leq n$, t_i represents the sampling timestamp, and $p_i = (lng_i, lat_i)$ is the corresponding GPS coordinate, where lng_i and lat_i denote the longitude and latitude, respectively.

To simplify the problem, continuous timestamps with in each day are often divided into discrete timestamps with a fixed time interval δ. Following most existing work, we set the fixed time interval as 5 minutes, and we define a week as one period. Therefore the sampling timestamp of each sample point is discretized into an integer in the range [0, 2015]. In addition, we assume that trajectories are evenly sampled and have the same length.

Definition 2 (Conditional information): For each trajectory $\boldsymbol{x}$, assuming that the start location p_1, end location p_n, departure time t_1, average speed $\bar{v}_{\boldsymbol{x}}$, travel distance $d_{\boldsymbol{x}}$, and trajectory length n are given, and we define them as conditional information. Formally, the conditional information of a trajectory $\boldsymbol{x}$ is denoted as $\boldsymbol{c}_{\boldsymbol{x}} = (p_1, p_n, t_1, \bar{v}_{\boldsymbol{x}}, d_{\boldsymbol{x}}, n)$.

Definition 3 (Problem definition): Given a real trajectory dataset $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ and its corresponding conditional information $\mathcal{C}$, trajectory generation task is to train a generative model capable of generating a synthetic trajectory dataset $\widehat{\mathcal{X}} = \{\widehat{\boldsymbol{x}}_1, \widehat{\boldsymbol{x}}_2, \ldots, \widehat{\boldsymbol{x}}_M\}$. In our paper, we generate trajectories only contain coordinates because our trajectories' sample interval are equal, which can simplify challenges of trajectory generation.

B. Diffusion Model

Generally, diffusion model consists of two main processes: (1) a forward process introduces noise to perturb data, and (2) a reverse process recovers the original data distribution. The two processes in detail can be seen in [6].

III. METHODLOGY

In this section, we introduce the procedure of ST-DiffTraj. We train the model through the process of adding noise. After training, we generate trajectories by sampling noise from a Gaussian distribution and using CTDNet to remove noise step-by-step. In each denoising step, first, STEncoder extracts the spatiotemporal correlations of trajectories. Then, with the guidance of conditional information and diffusion step, we stack multiple CDAttn-DConv layers to estimate noise. And in each layer, we integrate CDAttn to improve the ability of feature extraction. In the following, we describe the main submodules of CTDNet in detail.

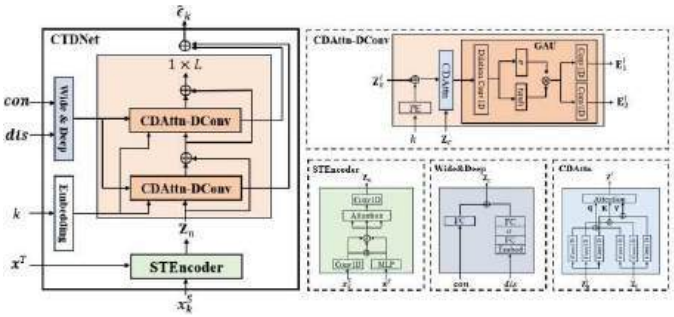


Fig. 1. The Structure of the CTDNet Based Denoising Model.

A. Spatio-temporal encoding

We design a spatio-temporal encoder (STEncoder) to effectively learn spatio-temporal features of trajectories, and the structure of the STEncoder is shown in Figure 1. The STEncoder learns temporal and spatial embeddings of trajectories with different embedding modules, separately.

1) Temporal embedding: We use trigonometric function $\psi = \mathbb{R} \to \mathbb{R}^{d_t}$ [9] to embed temporal information of trajectories. Given a timestamp t, it can be defined as $\psi(t) = (cos(\omega_1 t), sin(\omega_1 t), \ldots, cos(\omega_{d_t/2} t), sin(\omega_{d_t/2} t))$, where $\omega_1, \omega_2, \ldots, \omega_{d_t/2}$ are learnable parameters, and d_t is the dimension of the temporal embedding. Given a trajectory $\boldsymbol{x}$, let the n-dimensional column vector $\boldsymbol{x}^T = [t_1, t_2, \ldots, t_n]^\top$ denote its temporal information. The temporal information embedding $\boldsymbol{Z}_{\boldsymbol{x}}^T$ is:

$$\mathbf{Z}_{\boldsymbol{x}}^{\mathrm{T}} = \mathrm{Concat}\big(\psi(t_1), \psi(t_2), \ldots, \psi(t_n)\big), \quad (1)$$

where $\mathbf{Z}_{\boldsymbol{x}}^{\mathrm{T}} \in \mathbb{R}^{n \times d_t}$, and Concat is the vector concatenation function.

2) Spatial embedding: Given a trajectory $\boldsymbol{x}$, let a two-dimensional matrix $\boldsymbol{x}^S \in \mathbb{R}^{n \times 2}$ denote its spatial information. To capture the spatial features, we empoly a 1D convolution layer for spatial embedding: $\boldsymbol{Z}_{\boldsymbol{x}}^T = Conv1D(\boldsymbol{x}^S)$, where $\boldsymbol{Z}_{\boldsymbol{x}}^S \in \mathbb{R}^{n \times d_s}$, d_s is the embedding dimension of $\boldsymbol{x}^S$, and Conv1D denotes 1D convolution layer.

3) Spatial and temporal embedding fusion: Considering that spatial and temporal information in a trajectory are closely related, it is desirable to capture spatial-temporal correlations for the de-noising procedure. To this end, we firstly combine the spatial and temporal embeddings through element-wise addition resulting in a spatio-temporal fused embedding. After that, we concatenate the three embedding vectors, and then feed it into a self-attention module to capture spatial and temporal correlations. Finally, the calculation of the spatio-temporal encoding vector can be represented as:

$$\mathbf{Z}_{\boldsymbol{x}} = \mathrm{Conv1D}(\mathrm{SelfAttn}(\mathrm{Concat}(\mathbf{Z}_{\boldsymbol{x}}^{\mathrm{S}}, \mathbf{Z}_{\boldsymbol{x}}^{\mathrm{T}}, \mathbf{Z}_{\boldsymbol{x}}^{\mathrm{S}} + \mathbf{Z}_{\boldsymbol{x}}^{\mathrm{T}}))), \quad (2)$$

where $\mathbf{Z}_{\boldsymbol{x}} \in \mathbb{R}^{n \times d}$, d is the dimension of the final embedding vector.

B. Conditional information and diffusion step encoding

In this paper, Wide&Deep network is used to encode conditional information. It categories the conditional information of each trajectory into two types, i.e., continuous information $\boldsymbol{con}$ and discrete information $\boldsymbol{dis}$, and adopts

different methods to encode them. In particular, the embedding vector $\mathbf{Z}_{\boldsymbol{con}} \in \mathbb{R}^{n\times d}$ of $\boldsymbol{con}$ by using the MLP layer: $\mathbf{Z}_{\boldsymbol{con}} = \mathrm{MLP}(\boldsymbol{con})$, and $\boldsymbol{dis}$'s embedding vector $\mathbf{Z}_{\boldsymbol{dis}} \in \mathbb{R}^{n\times d}$ through the Embedding layer: $\mathbf{Z}_{\boldsymbol{dis}} = \mathrm{Embedding}(\boldsymbol{dis})$. Additionally, we add them up, get the output $\mathbf{Z}_{\boldsymbol{c}} \in \mathbb{R}^{n\times d}$ of Wide&Deep: $\mathbf{Z}_{\boldsymbol{c}} = \mathbf{Z}_{\boldsymbol{con}} + \mathbf{Z}_{\boldsymbol{dis}}$. For diffusion step k, we employ Sinusoidal embedding to represent it: $\mathrm{PE}[k] \in \mathbb{R}^{d_k}$, where d_k denotes the embedding dimension of diffusion step.

Finally, the embeddings of conditional information and diffusion step are integrated to each CDAttn-DConv layer.

C. Noise estimation with CDAttn-DConv

The noise estimation network adopts the WaveNet's architecture, which stacks multiple CDAttn-DConv layers, and the structure of the CDAttn-DConv layer is shown in Figure 1.

Considering the l-th CDAttn-DConv layer, it has three inputs at k-th step: (1) $\mathbf{Z}_k^l$, the spatio-temporal encoding vector $\mathbf{Z}_{\boldsymbol{x}}$ resulting from the STEncoder when $l = 1$, or the residual connection of the input vector and one of output vectors of the last CDAttn-DConv layer when $l > 1$; (2) the diffusion step k; (3) the conditional information encoding vector $\mathbf{Z}_{\boldsymbol{c}}$.

To capture the temporal correlation in the trajectory data, we stack L CDAttn-DConv. We set m layers as a block, so there exists L/m blocks. At the l-th layer, the spatiotemporal feature embedding $\mathbf{Z}_k^l$ is first added with transformed $\mathrm{PE}[k]$: $\mathbf{Z}_k^l = \mathbf{Z}_k^{l-1} + \mathrm{Linear}(\mathrm{PE}[k])$.

Subsequently, $\mathbf{Z}_k^l$ is passed through CDAttn(as shown in Figure 1). The query, key and value of CDAttn is computed as follows: $\mathbf{Q} = \mathbf{Z}_k^l W_{qz} + \mathbf{Z}_{\boldsymbol{c}} W_{qc}$, $\mathbf{K} = \mathbf{Z}_k^l W_{kz} + \mathbf{Z}_{\boldsymbol{c}} W_{kc}$, $\mathbf{V} = \mathbf{Z}_k^l W_{vz} + \mathbf{Z}_{\boldsymbol{c}} W_{vc}$, where $W_{qz} \in \mathbb{R}^{\tau\times d\times d}$, $W_{qc} \in \mathbb{R}^{\tau\times d\times d}$, $W_{kz} \in \mathbb{R}^{\tau\times d\times d}$, $W_{kc} \in \mathbb{R}^{\tau\times d\times d}$, $W_{vz} \in \mathbb{R}^{\tau\times d\times d}$, $W_{vc} \in \mathbb{R}^{\tau\times d\times d}$ are convolution projection weights of $\mathbf{Z}_k^l$ and $\mathbf{Z}_{\boldsymbol{c}}$ corresponding their queries, keys, and values respectively. τ is the kernel size. And we can get the output $\mathbf{Z}^l \in \mathbb{R}^{n\times d}$ of CDAttn: $\mathbf{Z}^l = \mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$.

$\mathbf{Z}^l$ is then fed into 1D dilation convolution layer with dilation rate $r = 2^{l\%m}$: $\mathbf{H}^l = \mathrm{DConv1D}(\mathbf{Z}^l)$. Afterward, we put the output $\mathbf{H}^l \in \mathbb{R}^{n\times 2d}$ of dilation convolution into a gate unit to filter useful feature distribution: $\mathbf{E}^l = \sigma(\mathbf{H}^l) \odot \tanh(\mathbf{H}^l)$, where $\odot$ represents the element-by-element multiplication. Then $\mathbf{E}^l$ is split into $\mathbf{E}_1^l \in \mathbb{R}^{n\times d}$ and $\mathbf{E}_2^l \in \mathbb{R}^{n\times d}$ along the embedding dimension.

$\mathbf{E}_1^l$ and the input of current layer through residual connection as next layer's input: $\mathbf{Z}_k^{l+1} = \mathbf{Z}_k^l + \mathbf{E}_1^l$. And $\mathbf{E}_2^l$ of each layer are added to estimate noise $\hat{\boldsymbol{\epsilon}}_k \in \mathbb{R}^{n\times 2}$ of k-th step:

$$\hat{\boldsymbol{\epsilon}}_k = \mathrm{Conv1D}(\textstyle\sum_{i=1}^{L} \mathbf{E}_2^i). \tag{3}$$

IV. Experiments

In this section, we first describe the dataset, baselines, and evaluation metrics. Then, we conduct comprehensive experiments to evaluate our model's effectiveness.

A. Experimental Setups

1) Datasets: We use two real-world datasets: Porto and T-Drive [10], [11]. The Porto dataset contains the trajectories of 442 taxis from January 2013 to June 2014. The T-Drive dataset records 10,357 taxi trajectories during a week. We select 100000 trajectories from each of the two datasets and regularize their GPS trajectory points for model training. Other descriptions are shown in Table I.

TABLE I. Dataset Description.

Dataset	Porto	T-Drive
Latitude	(41.12, 41.19)	(116.17, 116.62)
Longitude	(-8.69, -8.54)	(39.83, 40.05)
Average Speed	24.5km/h	18.4km/h
Time interval	15s	177s
Mean trajectory length	45.7	48.8

TABLE II. Hyperparameter Setting of ST-DiffTraj.

Parameters	Default	Parameters	Default
β	-	Batch size	512
L	12	Embedding dimension	128
m	4	Input length	150
Skip step S	5	Epoch	500
Diffusion step K	500	Learning rate	0.0002

2) Baseline Methods: We compare our model with the following baselines: *VAE*, *GAN*, *SeqGAN*, *MoveSim*, *DiffWave*, *DiffTraj*, *TrajGDM*. Moreover, for ablation study, we set *ours-c* uses addition with conditional information instead of CDAttn, and *ours-t* uses a linear transformation to replace attention mechanism in STEncoder.

3) Evaluation Metrics: In this paper, we analyze the generated trajectory data in terms of both spatial distribution error and transition feature error. The following are the evaluation metrics: (1) Spatial distribution error: Density error [6] measures the difference between real and generated trajectory grid density. Query error [12] assesses the quality of trajectory by the spatial count queries. Diameter Error [13] measures differences of diameter distributions. Frequent Point [6] calculate the similarity of frequent grids. (2) Transition feature error: Trip Error [13] measures the correlation between trip origins and destinations. Distance Error [14] evaluates the distribution of travel distance. Frequent Pattern [14] calculate the pattern similarity between real and generated data.

4) Implement Details: We implement ST-DiffTraj using PyTorch framework, with parameters summarized in Table II. β ranges from 0.0001 to 0.05, scaled linearly.

B. Over Performance

We compare the utility of ST-DiffTraj with baselines on two real datasets (bold indicates the best, underlining indicates the second best).

1) DiffWave and *DiffTraj* fail to fully utilize conditional information, limiting trajectory generate quality. In contrast, *ST-DiffTraj* effectively incorporates temporal correlations and conditional guidance, generating high-quality trajectories. Quantitative Analysis: As shown in Tables III and IV, we evaluate the metrics on the Porto and T-Drive datasets. Both VAE and GAN perform poorly compared to DiffTraj and

TABLE III. Performance Comparison on Porto.

Methods	Porto						
	Spatial				Transition		
	Density↓	Query↓	Diameter↓	FPoint↑	Trip↓	Distance↓	FPattern↑
VAE	0.069617	0.197843	0.006024	0.803	0.454883	0.035964	0.640
GAN	0.044668	0.171360	0.027850	0.837	0.415286	0.534043	0.543
SeqGAN	0.014968	0.154702	0.014217	0.883	0.472539	0.048133	0.727
MoveSim	0.283061	0.396288	0.203284	0.690	0.543207	0.287201	0.300
DiffWave	0.017685	0.078823	0.001625	0.872	0.260376	0.003169	0.760
TrajGDM	0.067559	0.190229	0.011179	0.873	0.507897	0.083464	0.697
DiffTraj	0.024756	0.052592	0.001556	0.882	0.270311	0.002509	0.767
ours-c	0.018448	0.052231	0.001566	0.882	0.271362	**0.002008**	0.770
ours-t	0.012791	0.048807	0.001568	0.903	0.248968	0.002899	0.792
ours	**0.012275**	**0.046892**	**0.001528**	**0.910**	**0.240243**	0.002235	**0.802**

TABLE IV. Performance Comparison on T-Drive.

Methods	T-Drive						
	Spatial				Transition		
	Density↓	Query↓	Diameter↓	FPoint↑	Trip↓	Distance↓	FPattern↑
VAE	0.049067	0.170785	0.054256	0.833	0.523621	0.196775	0.777
GAN	0.066754	0.334576	0.027573	0.817	0.528154	0.109099	0.247
SeqGAN	0.026707	0.104095	0.015743	0.830	0.459698	0.045493	0.800
MoveSim	0.072143	0.071494	0.222849	0.726	0.569479	0.265505	0.750
DiffWave	0.042279	0.050740	0.001433	0.837	0.417706	0.013218	0.793
TrajGDM	0.114344	0.254769	0.026969	0.800	0.589242	0.075831	0.570
DiffTraj	0.031518	0.064723	0.001122	0.863	0.422961	0.012922	0.817
ours-c	0.026268	0.048884	0.001143	0.870	0.407943	0.011564	0.827
ours-t	0.028299	0.061141	0.001765	0.867	0.397972	0.013507	0.827
ours	**0.023859**	**0.045263**	**0.000916**	**0.880**	**0.396582**	**0.008896**	**0.860**

DiffWave, demonstrating the superior performance of diffusion model. Although SeqGAN and MoveSim show some improvement, they still inferior to diffusion model. Compare generation methods based diffusion model, TrajGDM uses grid ids to represent trajectories, reducing accuracy of trajectory points.

2) Heatmap Visualization: We also visualize generated trajectories and calculate the cosine similarity (larger is better). Figures 2 and 3 display the trajectory heat maps for both real and generated datasets on Porto and T-Drive. The results show that *ST-DiffTraj* outperforms the baselines. By contrast, *VAE* and *GAN* capture only approximate movement patterns, while *SeqGAN* produces more accurate heatmap as its strong sequence modeling. *MoveSim* generates sparse trajectory distributions that fail to effectively capture the spatial features. Compared to *DiffTraj* and *DiffWave*, *ST-DiffTraj* effectively guides the diffusion of trajectory points, avoiding invalid positions, and also avoids the excessive constraints seen in *TrajGDM*. This flexibility demonstrates the excellent performance of *ST-DiffTraj*.

3) Downstream Tasks: To assess utility of synthetic data, we apply it to following downstream tasks.

a) Trajectory location prediction: We implement an LSTM-based trajectory prediction model. As shown in Figure 4, the performance of our model closely match the real data, highlighting the effectiveness of the trajectory data generated by our model in location prediction applications.

b) Traffic flow prediction: We compare the prediction metrics of real and generated data using existing traffic flow prediction models (ASTGCN [15], GWNet [16], MTGNN [17], DCRNN [18]). As shown in Table V, the performance differences are within a desirable range, indicating that the synthetic trajectories by our model preserves the spatiotemporal features required for traffic flow tasks.

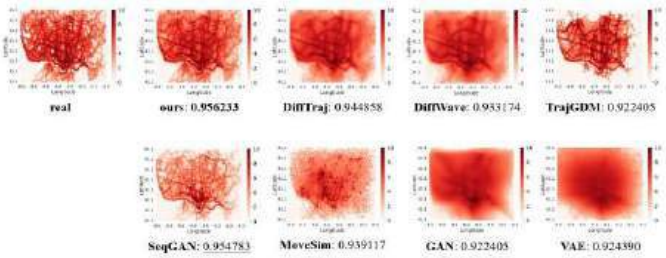


Fig. 2. Comparison of Visualization on Porto.

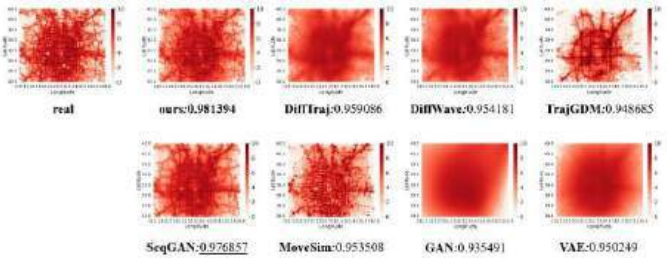


Fig. 3. Comparison of Visualization on T-Drive.

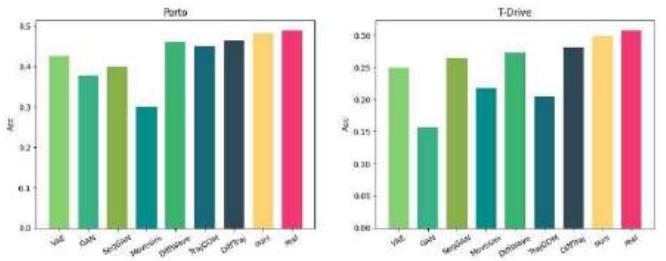


Fig. 4. Accuracy of Trajectory Location Prediction.

C. Ablation Study

We also conduct ablation experiments to evaluate the importance of proposed modules. The results in Table III and Table IV show *ours-c* outperforms the baseline in most metrics, particularly on the Porto dataset, where the distance metric achieves optimal performance. This validates STEncoder's capability to extract spatiotemporal correlations. And *ours-t* retains CDAttn, which effectively utilizes conditional information. This enhances the model's performance across multiple metrics and enables it to generate trajectory distributions that better match the real data.

D. Impacts of Hyperparameters

To optimize model performance, we further explore the impact of hyperparameters of *ST-DiffTraj* on Porto and T-Drive. Specifically, as shown in Figure 5, we focus on the embedding dimension of the trajectory data and the number of stacked layers of CDAttn-DConv.

a) Embedding Dimension Tuning: We experiment with embedding dimensions within {16, 32, 64, 128, 256}. The results indicate that *ST-DiffTraj*'s performance improves with increasing embedding dimension. However, the accuracy decreases when it reaches 256. Therefore, we selected 128 as the optimal embedding dimension.

TABLE V. Accuracy of Traffic Flow Prediction (the Percentage Indicates the Difference between Generated and Original Data).

Methods		Porto			T-Drive		
		DiffTraj	ours	real	DiffTraj	ours	real
ASTGCN	RMSE	58.2(1.69%)	57.72(0.86%)	57.23	61.52(106.44%)	29.63(0.57%)	29.8
	MAE	23.79(4.57%)	22.33(1.85%)	22.75	35.26(97.76%)	18.41(3.25%)	17.83
	MAPE	0.93(17.72%)	0.85(7.59%)	0.79	0.91(44.44%)	0.71(12.69%)	0.63
GWNet	RMSE	37.48(0.37%)	37.34(0.00%)	37.34	29.35(7.43%)	29.01(6.19%)	27.32
	MAE	23.91(11.57%)	23.59(1.01%)	21.43	20.41(7.7%)	19.77(4.33%)	18.95
	MAPE	1.04(25.30%)	0.97(16.86%)	0.83	1.20(34.83%)	0.91(2.25%)	0.89
MTGNN	RMSE	117.88(5.09%)	113.74(1.39%)	112.17	67.45(7.94%)	73.99(0.98%)	73.27
	MAE	69.69(2.21%)	69.29(1.63%)	68.18	59.65(1.43%)	58.86(0.09%)	58.81
	MAPE	6.56(11.38%)	6.08(3.23%)	5.89	8.79(5.65%)	8.09(2.76%)	8.32
DCRNN	RMSE	67.19(22.34%)	55.15(0.42%)	54.92	34.48(37.98%)	25.86(3.48%)	24.99
	MAE	39.45(49.37%)	27.9(5.64%)	26.41	23.48(36.67)	17.73(3.20%)	17.18
	MAPE	1.01(16.09%)	0.88(1.15%)	0.87	1.06(15.22%)	0.95(3.26%)	0.92

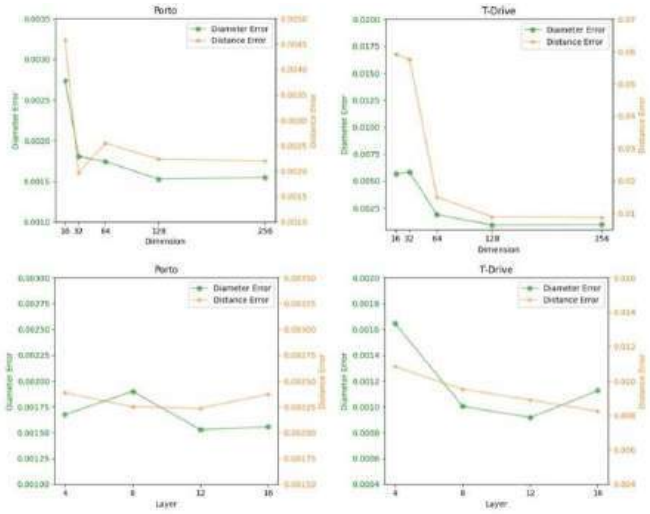


Fig. 5. Hyperparameter Analysis of Embedding Dimension.

b) Stacked Layers Selection: Similarly, we adjust L within {4, 8, 12, 16}. Increasing L improves the model's ability to capture trajectory sequences, leading to better extraction of sequence information. However, when L becomes too large, the performance of *ST-DiffTraj* slightly decreases due to the increased training difficulty caused by excessive layers. Therefore, we set $L = 12$.

V. Conclusion

In this paper, we propose a conditional dependent diffusion trajectory generation method that addresses the limitations of existing trajectory generation methods in spatiotemporal feature extraction and conditional information utilization. First, we introduce STEncoder to capture the correlations between spatiotemporal joint and independent feature distribution of trajectories. Next, we present CDAttn to improve the accuracy of estimating noise. Finally, we complete trajectory generation task based on diffusion model. In future, we plan to investigate the generation task of different sample interval, aiming to better meet the acquire of urban applications.

References

[1] X. Kong, M. Li, K. Ma, K. Tian, M. Wang, Z. Ning, and F. Xia, "Big trajectory data: A survey of applications and services," IEEE access, vol. 6, pp. 58295–58306, 2018.

[2] Y. Zhou, B. P. L. Lau, C. Yuen, B. Tunc¸er, and E. Wilhelm, "Understanding urban human mobility through crowdsensed data," IEEE Communications Magazine, vol. 56, no. 11, pp. 52–59, 2018.

[3] J. Dai, B. Yang, C. Guo, and Z. Ding, "Personalized route recommendation using big trajectory data," in 2015 IEEE 31st international conference on data engineering, pp. 543–554, IEEE, 2015.

[4] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. Gonz´alez, "The timegeo modeling framework for urban mobility without travel surveys," Proceedings of the National Academy of Sciences, vol. 113, no. 37, pp. E5370–E5378, 2016

[5] A. Kapp, J. Hansmeyer, and H. Mihaljevi´c, "Generative models for synthetic urban mobility data: A systematic literature review," ACM Computing Surveys, vol. 56, no. 4, pp. 1–37, 2023.

[6] Y. Zhu, Y. Ye, S. Zhang, X. Zhao, and J. Yu, "Difftraj: Generating gps trajectory with diffusion probabilistic model," Advances in Neural Information Processing Systems, vol. 36, pp. 65168–65188, 2023

[7] C. Chu, H. Zhang, P. Wang, and F. Lu, "Simulating human mobility with a trajectory generation framework based on diffusion model," International Journal of Geographical Information Science, vol. 38, no. 5, pp. 847–878, 2024.

[8] A. Hatamizadeh, J. Song, G. Liu, J. Kautz, and A. Vahdat, "Diffit: Diffusion vision transformers for image generation," 2024.

[9] H. Wan, Y. Lin, S. Guo, and Y. Lin, "Pre-training time-aware location embeddings from spatial-temporal trajectories," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 11, pp. 5510–5523, 2021.

[10] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 316–324, 2011.

[11] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems, pp. 99–108, 2010.

[12] C. Rui, C. Benjamin, M. Fung, and B. C. Desai, "Differentially private trajectory data publication," CoRR, abs/1112.2020, 2011.

[13] X. He, G. Cormode, A. Machanavajjhala, C. Procopiuc, and D. Srivastava, "Dpt: differentially private trajectory synthesis using hierarchical reference systems," Proceedings of the VLDB Endowment, vol. 8, no. 11, pp. 1154–1165, 2015.

[14] M. E. Gursoy, L. Liu, S. Truex, L. Yu, and W. Wei, "Utility-aware synthesis of differentially private and attack-resilient location traces," in Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp. 196–211, 2018.

[15] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in Proceedings of the AAAI conference on artificial intelligence, vol. 33, pp. 922–929, 2019.

[16] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," arXiv preprint arXiv:1906.00121, 2019.

[17] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 753–763, 2020.

[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," arXiv preprint arXiv:1707.01926, 2017.

Prediction of aviation safety using the combination model based on improved immune algorithm

Huayou Wang
School of Mechatronics,
Northwestern Polytechnical University
Xi'an, China
whuayou@163.com

Chengcheng Zhang*
The First Aircraft Institute, Aviation Industry Corporation of China
Xi'an, China
13700207502@163.com*

Guining Zheng
The First Aircraft Institute, Aviation Industry Corporation of China
Xi'an, China
18049472359@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899501

***Abstract*— Flight accidents can cause huge casualties and property damage, and establishing an aviation safety prediction model is of great significance for the design and prevention of aircraft operation safety. Aviation safety incidents have the characteristics of multiple influencing factors and randomness. Existing prediction methods are mainly based on time series, with low explicit expression of causal factors, poor interpretability of models, and weak engineering practicality. This article proposes a method for constructing multiple regression analysis(RA) model functions based on unit regression using the "cumulative addition by cause" method and the "cumulative multiplication by cause" method, which improves the modeling efficiency of the multiple regression model; An improved immune optimization algorithm (IA) has been proposed, which enhances population diversity and global optimization capability; Based on this, this article proposes a "Multiple Regression+Neural Network" (IARA+BPNN) safety combination prediction algorithm based on an improved immune algorithm. The prediction target is decomposed into "rule term+correction term", and the rule term is made explicit through multiple regression model prediction. The global optimization ability of the multiple regression model weight is improved through the improved immune algorithm, and the correction term is predicted through the BP neural network (BPNN) model to enhance the prediction accuracy. Taking the sample data of A-level military support mishaps in the United States as an example, the prediction results show that the "cumulative addition by cause" and "cumulative multiplication by cause" improve the efficiency of constructing multiple regression functions, and the improved immune algorithm improves the prediction accuracy of regular terms. The "IARA+BPNN" prediction model combines the advantages of multiple regression models and BP neural network models, which not only has certain explicit requirements, but also has high prediction accuracy, improving engineering practicality.**

Keywords-aviation safety; multiple regression; improved immune optimization; BP neural network; combination model

I. INTRODUCTION

Safety is the primary quality characteristic of aviation products, and conducting safety design and evaluation is crucial for preventing safety risks in aviation operations. Aviation safety prediction is a prerequisite for effective safety design and evaluation. Therefore, the research on safety prediction algorithms based on operational statistical data and technical data is one of the hot topics of safety prediction research that aviation industry departments, universities, and research institutes in various countries are all concerned about.

The accident/safety prediction algorithm has gone through stages such as time series trend prediction and combination method prediction.

Time series forecasting methods include exponential smoothing, regression analysis, and grey model analysis. Exponential smoothing and regression methods are widely used in the modeling of time series trend prediction for aviation accident symptoms and reliability [1-5]. Among them, multiple regression method is widely used in various traffic accident prediction due to its advantages of explicit causal factors and strong interpretability. However, the modeling of multiple regression models not only requires rich engineering or management experience, but also continuous trial and error, resulting in low modeling efficiency [6]. The grey model takes the trend of small sample "poor information" uncertain systems with "partially known information and partially unknown information" as the research object. The univariate grey model GM (1,1), combined with Markov, grey function or time series analysis models, has been widely used in flight accident prediction [7-10]. The autoregressive integrated moving average model (ARIMA) can be used for long-term macro trend and time series prediction of accidents, failure rates, and reports of civil aviation unsafe events with periodic changes [11].

Markov chain（MK）, support vector machine (SVM) and artificial neural network (ANN) prediction methods are suitable for predicting stochastic fluctuations. Markov chains can predict future trends based on the transition probabilities of historical accidents, but they require data to be homogeneous. The combination of support vector machine and other theories to form support vector machine regression algorithm (SVR), linear support vector machine (LSVM), grey support vector machine (GSVM), genetic support vector machine (GA-SVM) and other algorithms has been applied to the reliability of airborne products, aircraft failure rate, flight accident prediction analysis and evaluation [12]. The adaptive fuzzy neural network, improved BP neural network (BPNN), generalized regression neural network (GRNN) model, and MLP neural network learning algorithm developed on the basis of neural network algorithms have been applied to predict flight accident rates, flight accidents, and failure rates, effectively solving problems such as small samples, nonlinearity, high dimensionality, and local minima, with strong generalization ability. Support vector machines and neural networks have improved the accuracy of nonlinear prediction through nonlinear kernel function training, but they are prone to getting stuck in local minima and

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

overfitting, with low model explicitness and poor interpretability, making them difficult to apply directly in engineering.

In order to improve the prediction accuracy of the model, many scholars have conducted related research by combining various trend algorithms and nonlinear algorithms. For example, combination prediction algorithms such as "GM (1,1) Grey Model+Time Series Model" and "Support Vector Machine+Deep Neural Network" have been developed and applied to aviation safety prediction [13]. So far, these combination prediction models are only applicable to time series prediction or univariate prediction, and are not suitable for practical situations such as multi-variable coupling of accidents/safety. New combination algorithms need to be discovered.

This article proposes a novel "IARA+BPNN" combined prediction algorithm based on improved immune optimization, which decomposes the prediction target into "regular terms" and "correction terms". The regular items are predicted using a multiple regression model to achieve the explicitization of causal factors. which improves the overall engineering practicality of the model. A method for constructing multiple regression functions using the "cumulative addition by cause" method and the "cumulative multiplication by cause" method is proposed to improve modeling efficiency, and an improved immune algorithm with "one father and multiple children" inoculation strategy and adaptive genetic operator is developed to optimize the weights of the multiple regression model. The correction item adopts the BP neural network model prediction to improve the overall prediction accuracy.

II. THEORY OF "IARA+BPNN" COMBINATION MODEL

A. Constructing Method of Multiple Regression Model Functions

It is crucial to use multiple regression models for safety prediction and determine their functional architecture. In both the theoretical and engineering fields, it is generally recommended to use the trial and error method. First, a hypothetical function form is given, and after predicting the weights, the function form or weights are adjusted based on the accuracy performance data. This article proposes two methods based on the unit regression model, namely the "cumulative addition by cause" method and the "cumulative multiplication by cause" method, to derive the form of the multiple regression function based on the unit regression function form between the target variable and each influencing variable. This theoretically achieves a one-time construction of the function form, avoiding the practice of multiple trial and error. The specific derivation process is as follows.

1) Cumulative addition by cause

For the accident target variable $\mathbf{y}$ and m main dependent variables $\mathbf{x}_1,\cdots,\mathbf{x}_m$, a complex unit regression model $\overline{\mathbf{y}}_i = g_i(\mathbf{x}_i)$ is formed. Assuming there is a continuous additive relationship between the target variable $\mathbf{y}$ and the continuous relationship $\overline{\mathbf{y}}_i$, the multiple regression prediction model is formed from the unit regression prediction model as shown in formula (1).

$$\overline{\mathbf{y}} = \sum_{i=1}^{m} \beta_i \overline{\mathbf{y}}_i = \sum_{i=1}^{m} \beta_i g_i(\mathbf{x}_i) \tag{1}$$

Among them, $0 < \beta_j < 1, \sum_{j=1}^{m} \beta_j = 1$.

Thus, the optimization problem formed is as follows.

$$\min(\beta_j) = \min \left\| \mathbf{y} - \overline{\mathbf{y}} \right\| = \min \left\| \mathbf{y} - \sum_{i=1}^{m} \beta_i g_i(\mathbf{x}_i) \right\| \tag{2}$$
$$e.t.\ 0 < \beta_j < 1, \sum_{j=1}^{m} \beta_j = 1$$

2) Cumulative multiplication by cause

If the unit regression model is in the form of a simple function such as an exponential or power function. Assuming there is a continuous multiplication relationship between the target variable A and the unit regression function B, a multiple regression prediction model is formed from the unit regression prediction model as shown in formula (3).

$$\overline{\mathbf{y}} = \sqrt[m]{\prod_{i=1}^{m} \overline{\mathbf{y}}_i} = \sqrt[m]{\prod_{i=1}^{m} g_i(\mathbf{x}_i)} \tag{3}$$

Taking the logarithm of both sides yields equation (4).

$$\ln(\overline{\mathbf{y}}) = \frac{1}{m} \sum_{i=1}^{m} \ln(\overline{\mathbf{y}}_i) = \frac{1}{m} \sum_{i=1}^{m} \ln(g_i(\mathbf{x}_i)) \tag{4}$$

For example, if the unit regression model is exponential, assuming $\overline{\mathbf{y}}_i = a_i e^{b_i \mathbf{x}_i}$, a_i and b_i are the weights of the unit regression model, the above equation can also be transformed into:

$$\ln(\overline{\mathbf{y}}) = \frac{1}{m} \sum_{i=1}^{m} \ln a_i + \frac{1}{m} \sum_{i=1}^{m} b_i \mathbf{x}_i \tag{5}$$

If $\mathbf{z} = \ln(\mathbf{y}), \overline{\mathbf{z}} = \ln(\overline{\mathbf{y}}), \beta_0 = \frac{1}{m} \sum_{i=1}^{m} \ln a_i, \beta_i = \frac{b_i}{m}$ is given, then the predicted value $\overline{\mathbf{z}}$ of $\mathbf{z}$ can be obtained, it is:

$$\overline{\mathbf{z}} = \beta_0 + \sum_{i=1}^{m} \beta_i \mathbf{x}_i = \sum_{i=0}^{m} \beta_i \mathbf{x}_i \tag{6}$$

According to equation (6), the continuous multiplication form can be transformed into the continuous addition form. Referring to (1), to solve the extremum problem with known initial values, the least squares method or other optimization

methods can be used to obtain the optimal value of the fitting parameter β_i.

B. Improved immunization algorithm

The standard immune algorithm has two drawbacks, one is that it is easy to enter a local optimal equilibrium state; The second is that it is prone to stagnation in the later stages of evolution.

In order to further enhance the global optimization capability of the immune algorithm and improve the accuracy of accident prediction, this paper has made three improvements on the basis of the standard immune genetic algorithm: firstly, the optimization performance function combining the mean variance of accident prediction and the average absolute percentage error is constructed as the affinity function, so that the optimization search can consider both the mean variance and the relative error; secondly,introducing an adaptive mutation operator to increase the convergence speed by increasing the mutation probability in the early stage of iteration, and automatically reducing the mutation probability in the later stage of iteration to improve the prediction accuracy; The third approach is to adopt multiple immune selection strategies simultaneously, using different incentive operators, mutation operators, and population refreshing to clone, mutate, and suppress the formed multiple samples. By merging multiple different samples, the diversity of the immune population can be improved, thereby enhancing the global optimization ability.

The specific application steps for improving the immune algorithm are as follows.

1) Perform antigen recognition.

That is, understand the problem to be optimized. Set initial parameters such as various immune dimensions, number of immune individuals, mutation probability, similarity threshold, immune individual boundary, incentive coefficient, mutation operator, etc.

2) Generate initial antibody population.

By encoding, the feasible solution of the problem is represented as antibodies in the solution space, and an initial population is randomly generated in the solution space.

3) Build affinity function

According to formula (7) to construct the performance function combining the mean variance and relative error of accident prediction as the affinity function of the immunization algorithm

$$\begin{aligned} aff(ab_j) &= MSE(ab_j) + k \times MAPE(ab_j) \\ &= \frac{1}{L}\sum_{i=1}^{L}(M_i - \hat{M}_i(ab_j))^2 + \frac{k}{L}\sum_{i=1}^{L}\left|\frac{M_i - \hat{M}_i(ab_j)}{M_i}\right| \end{aligned} \tag{7}$$

In the formula, ab_j represents the jth antibody in the population, which is the corresponding weight to be fitted for each influencing variable in the accident prediction model, and $j = 1, 2, \cdots, m$ represents the number of weight values; L is the total dimension of antibody encoding, which is the number of accident observations; M_i is the observed value of the accident, and $\hat{M}_i$ is the predicted value; k is weighted coefficient (default value is 10000), which can be adjusted according to actual situation; MSE is mean variance for accident prediction; MAPE is the average absolute percentage error of accident prediction.

4) Evaluate the affinity of each feasible solution in the population.

Determine whether the algorithm termination condition is met. If the conditions are met, terminate the algorithm optimization process and output the calculation result; Otherwise, continue with the optimization operation.

5) Calculate antibody concentration and stimulation degree

Antibody concentration is usually defined as:

$$den(ab_i) = \frac{1}{NP}\sum_{j=1}^{NP} S(ab_i - ab_j) \tag{8}$$

In the formula, NP represents the population size; $S(ab_i - ab_j)$ represents the similarity between antibodies.

The calculation of antibody activation degree can usually be done using (9), and different activation calculation parameters can be used for different immune selection strategies. Among them, α and β are excitation calculation parameters,

$$sim(ab_i) = \alpha \bullet aff(ab_i) - \beta \bullet den(ab_i) \tag{9}$$

6) Perform immune processing, including immune selection, cloning, mutation, and clone suppression

Mutation operator T_m is an important operator in immune algorithms for generating promising new antibodies and implementing region search, which has a significant impact on the performance of the algorithm.

$$T_m(ab_{i,j,m}) = \begin{cases} ab_{i,j,m} + (rand - 0.5)\cdot\delta, & rand < P_m \\ ab_{i,j,m}, & otherwise \end{cases} \tag{10}$$

In the formula, $ab_{i,j,m}$ is the collection of the jth dimension of the mth clone of antibody ab_i ; The range of the neighborhood defined by δ can be predetermined or adaptively adjusted according to the evolutionary process; *rand* is a random number function that generates a range of (0,1); P_m is the probability of mutation.

P_m can be set adaptive as (11). At the beginning of optimization, it is changed with a large probability while it is changed with the local optimal when it is close to the global optimal.

$$p_m = p_0 - \Delta p \times (G - i) / G \tag{11}$$

Where, G is the maximum cycle limit, i denotes the number of cycles, p_0 is the initial mutation probability, and Δp is the gradient of the mutation probability.

7) Population refresh

A novel refresh strategy with "one father,multiple children" characteristics for improving the immune optimization algorithm is to generate new populations from each immune population, and merge and recombine the old population with each new population according to formulas (12) and (13). Step 3) Calculate affinity.

$$f = [f_a, f_b, f_c, \cdots] \tag{12}$$

$$sim = [sim_a, sim_b, sim_c, \cdots] \tag{13}$$

Among them, f_a、f_b、f_c、sim_a、sim_b、sim_c respectively represent multiple offspring immune populations and their motivation levels.

C. Establishment of IARA +BPNN Combination Prediction Algorithm

The multiple regression model is used as a trend item prediction model, and the immune algorithm is improved to optimize the weights of the multiple regression model, minimizing the prediction error. Then, the prediction error of trend item is corrected by the BP neural network model, using the rule term to predict residuals as the target value, N variable observations as input values, extracting some observations as training samples, and a small portion of observations as validation and testing samples. By setting the number of training layers, the training accuracy is continuously adjusted until the training accuracy, validation accuracy, and testing accuracy reach the expected accuracy before stopping training and adjusting parameters. Add the BP neural network prediction and IARA model prediction output data to obtain the final prediction output. The specific steps are shown in Figure 1.

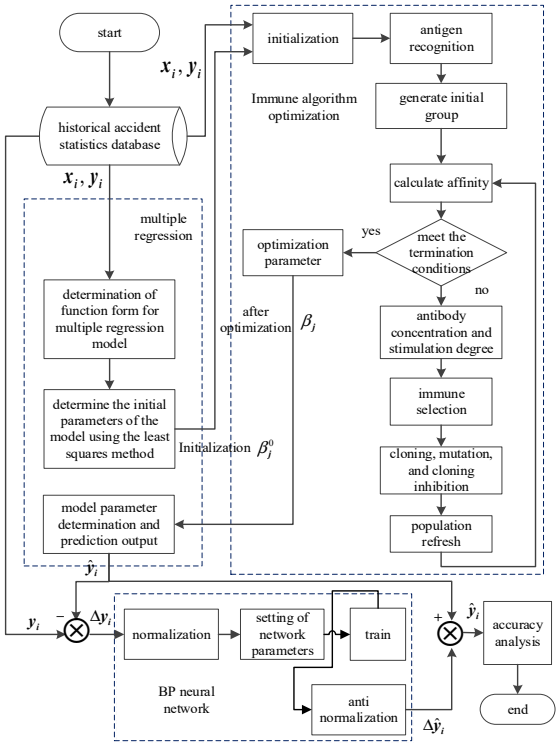


Figure 1. Flow Chart of "IARE+BPNN" Combination Prediction Based on Improved Immune Algorithm Optimization

III. CASE STUDY

A. Historical data organization

According to the historical statistics of A-class accidents of various types of aircraft released by the US air force safety center, the accumulated number of A-class mishaps(M), technical maturity level(TML), first flight year(FFY), equivalent use years(UY), number of engines(EN), and accumulated flight hours(AFH) of various types of support aircraft of the US Air Force can be obtained. The goal is to predict the number of A-level accidents based on known information. The details are shown in Table 1.

B. IARA model prediction

1) Determination of Multiple Regression Model Architecture

Through unit regression analysis, it can be concluded that there is an exponential function relationship between the cumulative number of A-level mishaps and the technology maturity level of the first flight , the first flight year, and the number of engines. There is also a power function relationship between the cumulative number of A-level mishaps and the equivalent use years and accumulated flight hours.

TABLE I. TABLE TYPE STYLES

No.	TML	FFY	UY	EN	AFH	M
1	9.5	1981	38	3	19.16	21
2	9.5	1981	26	3	11.57	12

3	9.5	1981	25	3	10.97	11
4	9.5	1981	21	3	8.31	6
5	9.5	1981	20	3	7.86	5
6	9.5	1981	16	3	5.85	4
7	9.5	1981	15	3	5.33	2
8	7.5	1957	62	4	159.62	87
9	7.5	1957	50	4	133.32	81
10	7.5	1957	49	4	131.07	81
11	7.5	1957	48	4	128.51	81
12	7.5	1957	45	4	121.18	79
13	7.5	1957	44	4	119.33	79
14	7.5	1957	40	4	111.25	77
15	7.5	1957	39	4	109.10	77
16	8	1955	64	4	196.80	162
17	8	1955	52	4	170.80	151
18	8	1955	51	4	168.26	151
19	8	1955	50	4	164.96	149
20	8	1955	47	4	155.22	145
21	8	1955	46	4	152.49	143
22	8	1955	42	4	141.27	140
23	8	1955	41	4	138.33	139
24	8	1964	43	4	106.52	34
25	8	1964	42	4	106.51	34
26	8	1964	41	4	106.31	34
27	8	1964	38	4	105.11	34
28	8	1964	37	4	104.59	34
29	8	1964	33	4	100.82	32
30	8	1964	33	4	100.82	32
31	8	1964	32	4	102.67	32
32	7	1991	31	4	36.67	36
33	7	1991	28	4	32.71	32
34	7	1991	16	4	10.48	17
35	7	1991	15	4	8.91	15
36	7	1991	11	4	3.06	3
37	7	1991	10	4	2.24	3
38	7	1991	6	4	0.40	1
39	7	1991	5	4	0.19	0
40	8	1968	51	4	26.66	27
41	8	1968	39	4	22.80	23
42	8	1968	38	4	22.37	21
43	8	1968	37	4	21.62	20
44	8	1968	34	4	18.89	17
45	8	1968	33	4	18.31	16
46	8	1968	29	4	15.94	15
47	8	1968	28	4	15.27	15
48	8	1975	44	2	7.77	3
49	8	1975	32	2	4.22	2
50	8	1975	31	2	4.17	2
51	8	1975	30	2	4.13	2
52	8	1975	27	2	4.01	2
53	8	1975	26	2	3.97	2
54	8	1975	22	2	3.78	2
55	8	1975	21	2	3.74	2
56	10	1984	35	2	12.86	4
57	10	1984	23	2	10.84	3
58	10	1984	22	2	10.39	3
59	10	1984	21	2	0.92	3
60	10	1984	18	2	8.48	2
61	10	1984	17	2	7.99	2
62	10	1984	13	2	6.14	2
63	10	1984	12	2	5.67	2
64	9.5	1968	44	2	9.02	3
65	9.5	1968	39	2	8.99	3
66	9.5	1968	38	2	8.98	3
67	9.5	1968	37	2	8.95	3
68	9.5	1968	34	2	8.53	3
69	9.5	1968	33	2	8.31	3
70	9.5	1968	29	2	7.47	2
71	9.5	1968	28	2	7.22	2

Based on the data in Table 1 and following the method of constructing the multiple regression model function in Section 1.1, the multiple regression model style as shown in Equation (14) is determined using the factor wise cumulative method.

$$M = e^{m_0 + m_1 \times TML + m_2 \times FFY + m_3 \times \log(UY) + m_4 \times EN + m_5 \times \log(AFH)} + \varepsilon \quad (14)$$

Among them, $M_i (i = 0,1,2,3,4)$ represents the cumulative predicted value sequence of A-level accidents, *TML* represents the maturity sequence of first flight technology, *FFY* represents the time sequence of first flight, *UY* represents the equivalent usage year sequence, *EN* represents the number of single engine units, and *AFH* represents the cumulative flight time

(100000 hours). ε is the prediction residual and $m_i (i = 0,1,2,3,4,5)$ is the multiple regression coefficient.

2) Determination of initial weights for multiple regression models

Select the historical statistical data of the US Air Force's support aircraft from Group 1-68 in Table 1 as training data, and from Group 69-71 as testing data. The cumulative number of A-level accidents is the target variable data for the training model, and the rest are the influencing variable data for the training model. According to the model function architecture (15) and the combination prediction method flow in Section 1.3, m_0=-10.4213, m_1=0.0033, m_2=0.042, m_3=0.614, m_4=0.7794, m_5=0.5375 can be solved using the least squares method. So the initial model for multiple regression is:

$$\hat{M} = e^{-10.4213} \times e^{0.0033 \times TML} \times e^{0.042 \times FFY} \times UY^{0.614} \times e^{0.7794 \times EN} \times AFH^{0.5375} \quad (15)$$

3) Multiple regression weight optimization

Adopting an improved immune optimization algorithm process, setting the average absolute percentage error function as the affinity function, and setting the adaptive mutation probability. Set the initial constraint conditions for antigens or weights, and continuously adjust and narrow down the constraint range. Set various parameters and operators of the algorithm as shown in Table 2.

TABLE II. IMPROVE THE PARAMETER VALUES RELATED TO IMMUNE ALGORITHM

Parameter	abbreviation	value
Variable dimension	D	6
Maximum immune generation	G	800
Number of immune individuals	NP	200
mutation probability	P_m	adaptive
Similarity threshold	δ_{S1}	0.2
	δ_{S2}	0.18
	δ_{S3}	0.15
Number of clones	N_{C1}	10
Incentive coefficient	α	2
	β_1	1
	β_2	0.5
	β_3	1.5

The final multiple regression model was optimized as follows:

$$\hat{\boldsymbol{M}} = \boldsymbol{e}^{2.8812} \times \boldsymbol{e}^{-0.0021 \times \boldsymbol{TML}} \times \boldsymbol{e}^{-0.1513 \times \boldsymbol{FFY}} \times \boldsymbol{UY}^{0.4003} \times \boldsymbol{e}^{0.554 \boldsymbol{EN}} \times \boldsymbol{AFH}^{0.5993} \quad (16)$$

C. Prediction of improved IARA+BPNN Combination Model

On the basis of improving the immune algorithm optimized multiple regression model (IARA) prediction, the residual between it and the observed values is used as the prediction output, and each influencing variable is used as the prediction input. The BP neural network model is used for residual prediction.

1) Normalization

Among the processed 71 sets of data, the first 63 sets were used as training data, the 64th to 67th sets were used as validation data, and the 68th to 71st sets were used as test data. Perform normalization processing.

2) Build BP neural network and train it

Build a network architecture with 5 input layers, 1 output layer, and ultimately 3 hidden layers. The training algorithm chooses Bayesian regularization for training.

3) Output prediction model

After optimizing the multiple regression model using the improved IA algorithm, the residual values obtained from the prediction were fitted with a BP neural network and output as predicted values. These predicted values were then added to the original IARA fitted prediction values to obtain the "IARA+BPNN" combination method prediction values.

D. Comparison of prediction accuracy among various models

Summarize the calculations of different models, as shown in Table 3 and Figure 2.

TABLE III. PREDICTION ERROR OF MAPE COMPARISON

Model name	Training error	Test error	Composite error
RA	0.2909	0.2779	0.2902
IARA	0.2817	0.147	0.2862
Improved	0.2805	0.1436	0.2566
IARA	0.1108	0.1365	0.1122
IARA+BPNN	0.0921	0.1351	0.0982

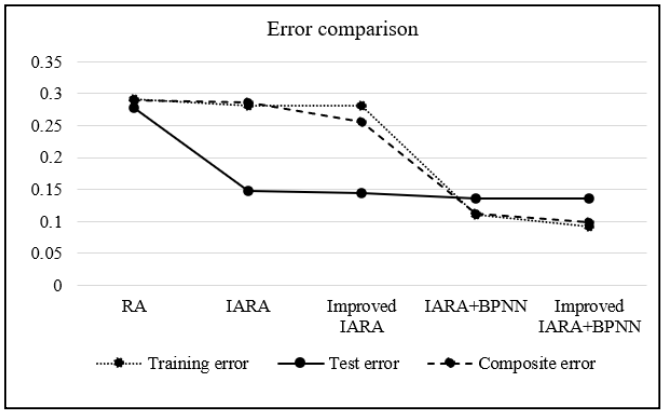


Figure 2. Accuracy Comparison Chart

According to Table 3 and Figure 2, the following conclusions can be drawn:

- The accident prediction based on multiple regression, IARA, and improved IARA has average absolute percentage errors of 29.09%, 28.17%, and 28.05%, respectively; The average absolute percentage errors of the tests were 27.79%, 14.7%, and 14.36%,

respectively; The comprehensive average absolute percentage errors are 29.02%, 28.62%, and 25.66%, respectively. Immune algorithm optimization can improve the prediction accuracy of multiple regression models, and the effect of improved immune algorithm on global optimization improvement is more significant compared to standard immune algorithm.

- On the basis of using IARA and improved IARA models to predict the regular terms, the BP neural network model was continued to be used for correction, and the average absolute percentage errors in training were reduced to 11.09% and 9.21%, respectively; The average absolute percentage error of the test was reduced to 13.65% and 13.51%, respectively; The comprehensive average absolute percentage errors were reduced to 11.22% and 9.82% respectively, indicating that the BP neural network model modification can significantly improve the overall prediction accuracy.

IV. CONCLUSION

This article proposes a combined prediction algorithm of multiple regression and BP neural network based on improved immune optimization (improved IARA+BPNN) for aviation safety prediction modeling. Using publicly available statistical data on the cumulative number of A-class accidents of US military support aircraft, aviation safety modeling, prediction, and accuracy comparison were carried out, and the following conclusions were drawn:

- Based on the unit regression model, multiple regression model functions can be constructed once successfully by the "cumulative addition by cause" method or "cumulative multiplicaiton by cause "method, avoiding the complex calculations of trial and error methods and improving modeling efficiency.

- Immune algorithm optimization can slightly improve the prediction accuracy of multiple regression models, and improving immune algorithm further enhances its global optimization effect.

- The "Improved IARA+BPNN" combination model significantly improves the engineering utility and can be generalized. By using the "Improved IARA+BPNN" combination model for prediction, the multiple regression model has achieved the explicitization of the causal factors of the regular term. The improved immune algorithm can slightly reduce the testing error and comprehensive error of the multiple regression model, while further modification of the neural network model significantly improves the overall prediction accuracy of the model.

ACKNOWLEDGMENT

Thank you to the teachers, classmates, and colleagues who provided assistance during the completion of the paper. Without your support, this paper would not have been presented.

REFERENCES

[1] Park D. Ashley.Operational Risk Management and Military Aviation Safety[D].Wright-Patterson AFB OH:the US Air Force Institute of Technology Air University, 1999:1-101.

[2] Mathew G. Cho. Air Force Operational Risk Management Program and Aviation Safety[D].Air Force Institute of Technology Air University, 2003,(2):1-159.

[3] Light T.,Hamilton T.,Pfeifer S..Trends in U.S. Air Force Aircraft Mishap Rates (1950-2018) [R].RAND Report,2022:1-6.From https://www.rand.org/t/RRA254-1.

[4] DU Y. Regression Analysis of the 10000 Hour Rate of Civil Aviation Accident Symptoms[J]. Journal of Civil Aviation Flight University of China, 2010(in Chinese).

[5] HUO Z Q, LUO F. Statistical analysis of civil aviation accidents and accident symptoms in the last decade in China Civil Aviation[J]. China Safety Science Journal, 2006, 16(12):7(in Chinese).

[6] Zheng Xiaoping, Gao Jinji, Liu Mengting, et al., theory and method of accident prediction [M].Beijing: Tsinghua University Press, 2009.

[7] WANG R Q, LU C. Fault Rate Prediction Based on EMD and Grey Model[J]. Journal of Ordnance Equipment Engineering, 2018, 39(7):4(in Chinese).

[8] MA Y C, PI D C. Research on Fault Rate Prediction Model Based on Grey Model and Neural Network[C]// 2010 International Forum on Computer Science, Technology and Applications. 0(in Chinese).

[9] GAN X S, DUANMU J S, WANG Q. Grey time series combination prediction model for aviation equipment accidents[J]. China Safety Science Journal, 2012, 22(4):6(in Chinese).

[10] LIU G, ZHU J F. Flight accident prediction based on grey metabolic Markov model[J]. China Safety Science Journal, 2007, 17(5):4(in Chinese).

[11] LI R Y, KANG R. Research on failure rate forecasting method based on ARMA model[J]. Systems Engineering and Electronics, 2008, 30(8):4(in Chinese).

[12] Zhang Xian.A se direct support vector regression machine method for failure rate prediction [J].AAP, 2010,25 (11): 2594-2599.

[13] GAN X S, DUANMU J S, CONG W, et al. Flight Accident Prediction Method Based on the Combination of ARIMA and SVM[J]. China Safety Science Journal, 2011, 21(7):6(in Chinese).

Robust Tracking Based on Improved YOLOv5s and Dynamic Neighborhood Target Association

Zifu Wei
School of Electronic Science and Engineering
Hunan University of Information Technology
Changsha, China
weizifu614@126.com

Jian Yang(Corresponding author)
School of Electronic Science and Engineering
Hunan University of Information Technology
Changsha, China
yangjian@hnuit.edu.cn

Chen Peng
School of Electronic Science and Engineering
Hunan University of Information Technology
Changsha, China
moon511@163.com

Guangyao Zhao
School of Electronic Science and Engineering
Hunan University of Information Technology
Changsha, China
securityzgy@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899463

***Abstract*—In order to enhance the video single-target tracking task's ability to cope with complex scenarios such as, rotation, rapid scale change, background interference, and occlusion, this paper proposes a tracking method based on improved YOLO detection and dynamic neighboring region target association. First, the YOLOv5s model is improved by introducing different attention mechanisms such as CBAM, CA, and PA, and retraining to select the optimal model so as to enhance the target detection effect. Second, adaptive tracking search region adjustment is performed within the neighborhood of the tracked target, and an IOU-based similarity measure and filtering mechanism is used to reduce the number of detected target boxes, thus reducing the number of false matches and computation. Finally, the appearance features of the retained target boxes are extracted by a re-identification network, and the final tracking target box is determined by performing cosine matching followed by IOU matching to enhance the tracking stability in complex scenes. The experimental results show that the algorithm of this paper achieves an average success rate of 0.816 and an average accuracy rate of 0.946 on 30 vehicle video sequences, and has good robustness in a variety of complex scenarios.**

Keywords—Target Tracking, YOLOv5s, Attention Mechanism, Dynamic Neighborhood Target Matching

I. INTRODUCTION

Target tracking techniques are widely used in intelligent video surveillance, autonomous driving, and UAV reconnaissance [1, 2]. Although great achievements have been made in the above fields, due to the complexity and diversity of the tracking scenarios, such as the deformation, occlusion, motion blur, illumination change, scale change, and fast motion, the study of high-performance robust target tracking algorithms is still a challenging task. Single target tracking methods are mainly categorized into: correlation filtering methods[3, 4], twin network based methods[5, 6, 7, 8] and other deep learning based methods[9, 10, 11]. Correlation filter tracking methods perform efficient target localization by training filters and utilizing the correlation between the target template and the search region, which has the advantages of high computational efficiency and certain robustness, but has the shortcomings of limited feature representation capability, complex model updating strategy and boundary effects. The twin network tracking method extracts the features of the template and the search region through the structure of the two-branch network with shared parameters, and uses the mutual correlation operation to achieve efficient target matching, which has the advantages of simplicity, efficiency and strong feature representation capability, but has the shortcomings of poor adaptability to changes in the target appearance due to the template fixation, as well as the disadvantages of limited generalization capability.

Detection-based target tracking methods are often used in multi-target tracking tasks, such as Deepsort[12] and Bytetrack[13], etc. Detection-based target tracking methods using deep neural networks as feature extractors have stronger feature representation capabilities, target detection models trained on a large amount of data have better generalization capability, and detection-based methods re-detect the target at every frame, which is equivalent to implicitly updating the template, and therefore are more flexible and better able to cope with drastic changes of the target's appearance. Based on this, this paper adopts a detection-based target tracking method and improves the matching and association of targets for single-target tracking tasks. The main work done in this paper:(1) The detection-based tracking method relies heavily on the performance of the detector, for which two aspects are done: first, a small number of vehicle target detection datasets are constructed and labeled for the nighttime and UAV aerial vehicle tracking task scenarios, to improve the model's performance of target detection in these two scenarios; second, different attention mechanisms, such as CBAM, CA, and PA, are used to improve the YOLOv5s model, and the optimal model is selected by comparison after retraining; (2) target matching and association based on dynamic neighboring region and re-identification are used to improve tracking performance.

II. ALGORITHM FRAMEWORK

A. Improved YOLOv5s target detection

1) Related work

In the field of target detection, YOLO (You Only Look Once) is a breakthrough algorithm. YOLO algorithm has evolved several versions since its inception, and two of the most popular versions are YOLOv5 and YOLOv8. YOLOv5 is highly optimized for real-time applications, achieving a good balance between speed and accuracy.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

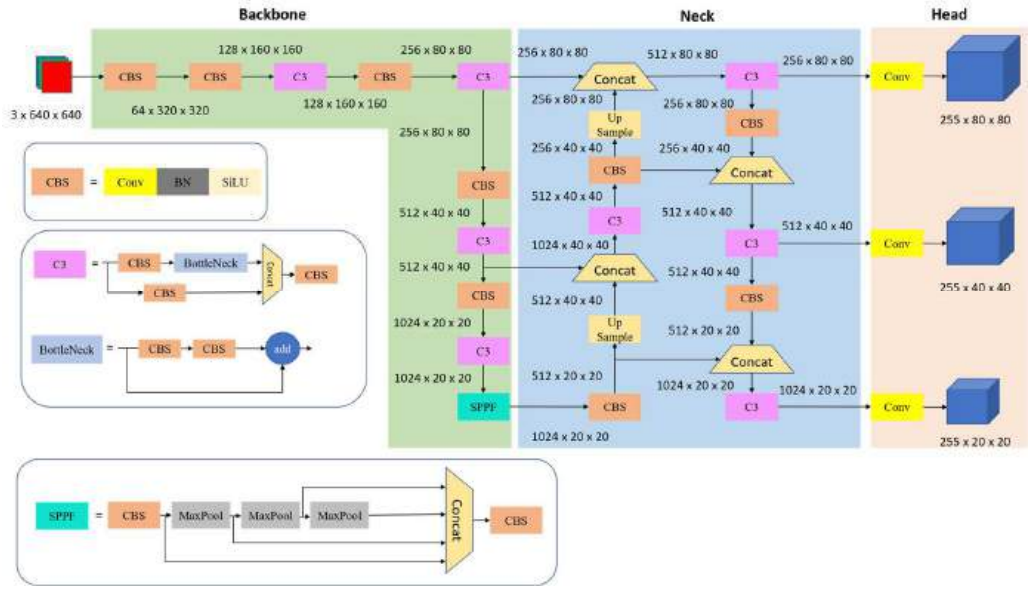


Fig. 1. YOLOv5s network structure

In this paper, the YOLOv5s model is selected as the basis for the dual requirements of detection accuracy and speed for the target tracking task, and its performance is further improved by fusing the attention mechanism. As shown in Fig. 1, the network structure of YOLOv5s is mainly composed of three core parts: backbone, neck and head[14]. The backbone network extracts multilevel features from the image through deep convolutional operations, and mainly applies the C3 module and the SPPF module; the former improves the efficiency of feature extraction while maintaining feature richness, while the latter realizes multi-scale feature extraction for the same feature map, which helps to improve the accuracy of detection. The neck network contains a feature pyramid FPN and a path aggregation network PAN. The FPN is responsible for delivering semantic information top-down in the network, while the PAN delivers localization information bottom-up, fusing feature maps from different layers of the backbone network, which further enhances the detection capability. The detection head focuses on predicting targets of different sizes on feature maps of different sizes.

2) Incorporating Attention Mechanisms for YOLOv5s

In computer vision applications, the effectiveness of attention mechanisms has been demonstrated, and they can assist the model to focus more on the key features, which in turn improves the model's performance. In this paper, the following three attention mechanisms, i.e., CBAM, CA, and PA attention mechanisms are used to improve the YOLOv5s model and the optimum is selected based on the experimental results.

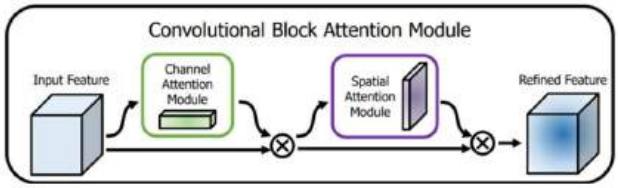


Fig. 2. CBAM attention module structure

CBAM (Convolutional Block Attention Module) is a combination of channel and spatial attention mechanism module[15], its structure is shown in Fig.2. The channel attention mechanism is responsible for optimizing the allocation relationship of feature map channels, and the spatial attention mechanism motivates the neural network to focus more on those pixel regions that have a significant impact on the classification results during image processing, while ignoring irrelevant regions. Simultaneous allocation of attention to these two dimensions enhances the model's performance.

CA(Coordinate Attention) achieves multi-directional aggregation of features by dividing the channel attention into two one-dimensional feature encoding processes[16], the structure is shown in Fig. 3. The advantage is that it can capture long-range dependencies in one spatial dimension while retaining precise location information in the other. The generated feature maps are then encoded separately to form a pair of direction-aware and location-sensitive feature maps, which can complement each other and act on the input feature maps to enhance the feature expression of the interested target.

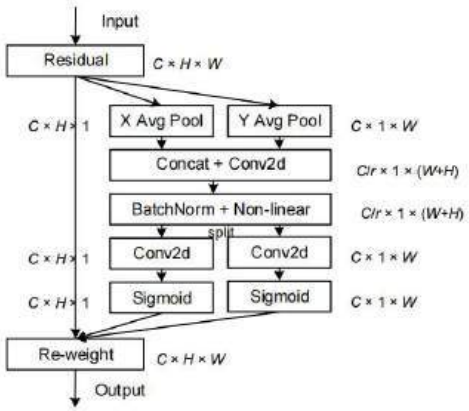


Fig. 3. CA attention module structure

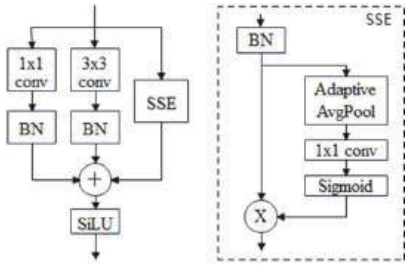


Fig. 4. ParNet attention module structure

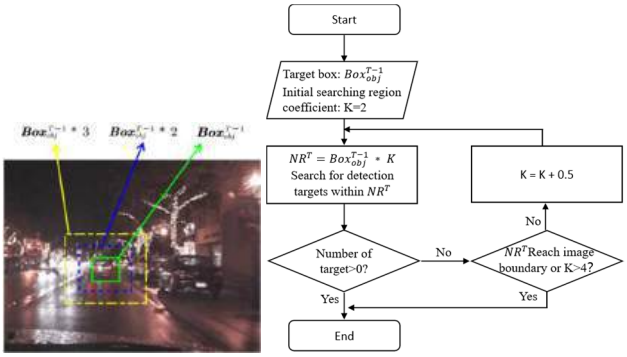
Non-Deep Network (ParNet, Parallel Network) is an innovative neural network architecture that significantly reduces the depth of the network while maintaining high performance by using parallel sub-networks instead of the traditional layer-by-layer stacking structure[17]. The ParNet Block is the core component of ParNet, whose main purpose is to enhance the feature expression capability through the use of parallel sub-networks and Skip- Squeeze-and-Excitation (SSE) module to enhance the feature expression capability, the structure of which is shown in Fig. 4. The parallel sub-network includes 1×1 convolution, 3×3 convolution, and SSE modules, which are branches that allow the network to capture features from different perspectives and thus enhance feature diversity. The SSE module enhances feature representation capability through the channel attention mechanism, which increases the receptive field and allows the model to better focus on important features.

In vehicle target tracking, most of the vehicle scales are medium or large scales, so in this paper, two attention blocks are added to the YOLOv5s neck part. One attention block is added after the C3 module of the P4 feature layer in the neck part to deal with the P4/16 scale features, at which the model has been sampled from the P3 feature layer and fused with the P4 feature layer, the main role of attention block here is to further enhance and refine these fused features for medium scale target detection head. Another attention block is added to the neck part of the P5 feature layer after the C3 module, to process the P5/32 scale features for the large scale target detection head.

B. *Dynamic Neighborhood Region Target Search and Association*

1) Dynamic neighborhood target search

In most of the tracking scenarios, especially the tracking scenarios where the camera does not have drastic jitter, the tracked target position does not change abruptly. Based on this characteristic, the candidate targets are identified by adaptively adjusting the search range within the neighborhood of the tracking target box of the previous frame. The schematic diagram of the dynamic neighborhood target search is shown in Fig. 5, and its basic process is to take the target tracking box $\boldsymbol{Box}_{obj}^{T-1}$ at the moment of T-1 as the reference (e.g., the green solid line box in the interior of Fig. 5(a)), and determine a neighboring region $NR^T = \boldsymbol{Box}_{obj}^{T-1} * K$ (e.g., the blue dashed line box in the middle of Fig. 5(a)) in the image at the moment of T with the searching region coefficient K=2, and search for a target detected by the detector in the region, and stop the searching if there is a target. If no target exists, increase the search region coefficient by 0.5, and search for the detected target again in the expanded region until at least 1 target is found or the search region coefficient K>4 or the image boundary is reached, then the search is stopped.



(a) Schematic diagram (b) Flowchart

Fig. 5. Dynamic neighborhood target search

2) Target filtering based on IOU shape similarity metrics

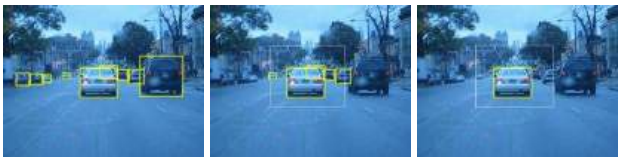
During the tracking process, the scale and aspect ratio of the target will not change dramatically in a short period of time, based on this, this paper adopts the IOU similarity metric based on the dynamic neighborhood to do further filtering of the detected targets, the specific steps are as follows:

- Create and maintain a list of tracking boxes for the 20 frames prior to the current time T, i.e $\boldsymbol{Box}_{obj}^{t},\ t \in \{T-1, T-2, ..., T-20\}$
- Iterate over all the candidate target boxes $\boldsymbol{Box}_{k}^{T}$, $k \in \{0, 1, 2, ..., N-1\}$ that detected in the neighborhood at the current moment T, where N is the total number of detected targets, and calculate the IOU values between the k-th detected box and the above 20 tracked boxes according to Eq. (1), respectively, Eq. (1) is calculated without considering the position of the target boxes, and only calculates the similarity in the shape of the target boxes, and obtains 20 IOU values.

$$IOU_{obj}^{t}(k) = \frac{\boldsymbol{Box}_{obj}^{t} \cap \boldsymbol{Box}_{k}^{T}}{\boldsymbol{Box}_{obj}^{t} \cup \boldsymbol{Box}_{k}^{T}} \tag{1}$$

- When the maximum of these 20 IOU values is greater than the set threshold (set to 0.6 in this paper), i.e. $\max(IOU_{obj}^{t}(k)) > 0.6, t \in \{T-1, T-2, ..., T-20\}$, the detection box is retained, otherwise it is discarded.

Fig. 6 gives the process of candidate target determination in the dynamic neighborhood region, Fig. 6(a) shows the detected targets in the current frame, the white box in Fig. 6(b) is the neighborhood region determined based on the target box of the previous frame, and the yellow box is the retained detected targets in the dynamic neighborhood region, and Fig. 6(c) shows the final candidate targets retained after IOU shape similarity filtering, and only one candidate target box is retained in the current frame after filtering, greatly reducing interference from similar targets or backgrounds.



(a) Detected targets (b) Targets in neighborhood (c) Retained targets

Fig. 6. Determining candidate targets in the dynamic neighborhood

From the figure, it can be seen that the dynamic neighboring area target search and filtering proposed in this paper has two effects: firstly, it reduces the interference of the background and other similar targets by excluding the candidate targets outside the target's region of interest, which enhances the anti-interference ability, secondly, it reduces the number of candidate targets, which reduces the computation of the subsequent feature extraction and target association.

3) Target association

In this paper, target association is mainly realized through two steps, firstly, Re-ID network is used for appearance feature extraction for the retained detection targets, and then cosine distance and IOU are used for target association.

The feature extraction part uses a simple Re-ID network to extract the features of each target, the structure of the Re-ID network is shown in Table 1, the input image size of the network is 64(H)*128(W), and a feature vector of 512 dimensions is obtained through the network, and the Veri-wild dataset is used to train this Re-ID network.

TABLE I. RE-ID MODEL STRUCTURE

Block	Kernel_size/Stride	Output Size
Conv 1	3 x 3/1	64 x 64 x 128
Max Pool 3	3 x 3/2	64 x 32 x 64
Residual 4	3 x 3/1	64 x 32 x 64
Residual 5	3 x 3/1	64 x 32 x 64
Residual 6	3 x 3/1	64 x 32 x 64
Residual 7	3 x 3/2	128 x 16 x 32
Residual 8	3 x 3/1	128 x 16 x 32
Residual 9	3 x 3/2	256 x 8 x 16
Residual 10	3 x 3/1	256 x 8 x 16
Residual 11	3 x 3/2	512 x 4 x 8
Avg Pool 8	4 x 8/1	512

The target association part establishes and maintains a list of the appearance feature vectors of the 100 most recent tracked targets, which is used to dynamically update appearance features of the target, performs feature extraction on the retained candidate detection boxes and performs appearance similarity computation with the 100 existing target features, and associates the targets finally. The specific steps are as follows:

- Establish and maintain a list of appearance feature vectors of the existing targets obtained from the Re-ID network by tracking the target box 100 frames before the current moment *T*, i.e. $f_j^{obj},\ j \in \{0,1,...,99\}$
- Based on the candidate target boxes $\boldsymbol{Box}_k^T,k \in \{0,1,2,...,M-1\}$ that retained in the neighborhood at the current moment *T*, the Re-ID network is used to obtain a list of candidate target appearance feature vectors, i.e. $f_i^{\det},\ i \in \{0,1,...,M-1\}$.
- Calculate the cosine distance between the feature vector of the *i*-th candidate target and the 100 existing target feature vectors according to Eq. (2), and take the value with the smallest cosine distance as the final distance value of the *i*-th candidate target, i.e $D_i^T = \min(D_{i,j}^T),\ j \in \{0,1,...,99\}$. Traverse all candidate targets and select the candidate target with the smallest value and less than the threshold of 0.2 as the matching target at the current time, i.e $\min(D_i^T),\ i \in \{0,1,...,M-1\}$. If $\min(D_i^T) > 0.2$, the target may undergo occlusion, motion blur, etc., then IOU matching is used, which calculates the IOU values of all candidate boxes and the tracking box of the previous frame, and selects the candidate box with the highest IOU value and greater than the threshold of 0.3 as the matching target at the current time. If the IOU matching still cannot find a matching target, the target is considered lost.

$$D_{i,j}^T = \frac{\langle f_i^{\det}, f_j^{obj} \rangle}{\| f_i^{\det} \| \| f_j^{obj} \|} \tag{1}$$

C. Algorithm Process

The overall framework of the algorithm is shown in Fig. 7, which is mainly divided into two modules: detector and tracker. The detector uses the improved YOLOv5s for target detection to improve the performance of target detection. The tracker performs data correlation and tracking. Firstly, dynamic neighboring region target search is used to eliminate candidate targets outside the target's region of interest, then target filtering

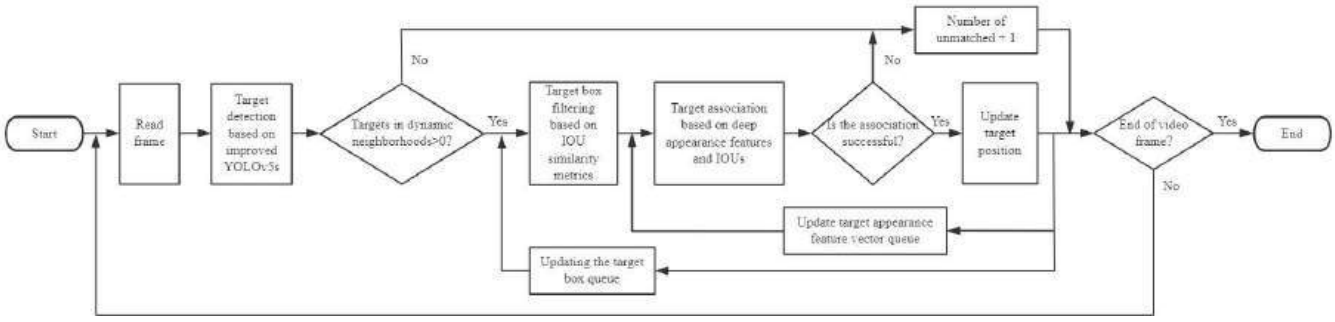


Fig. 7. General framework of tracking algorithm

based on the IOU similarity metric is used to further reduce the number of candidate targets in order to reduce the amount of computation and to reduce the interference from the similar targets and the background. Secondly, the appearance features of retained target boxes are extracted by a Re-ID network, then using cosine distance and IOU for target association. If the association is successful, the target box queue and appearance feature vector queue will be updated. If no target is detected in the dynamic neighborhood or the target is not associated successfully, the target position of the previous frame is retained.

III. Experimental results and analysis

A. Experimental setup

This paper mainly focuses on vehicle target tracking, using a total of 16,921 pictures of the three categories of car, bus, and truck in the COCO dataset, and in order to enhance the model's ability to detect vehicles in the nighttime scene and drone aerial scenes, 658 nighttime vehicle images and drone aerial images are obtained from the network, which constitutes a total of 17,579 pictures of vehicle detection dataset. The dataset is divided into 12284 training sets and 5295 validation sets according to a 7:3 ratio.

The hardware configuration for model training is Intel(R) Core(TM) i5-13400F 2.50 GHz processor, 16G RAM, NVIDIA GeForce RTX 3060 GPU, and the experimental software environment is Win11 system, cuda11.3, python3.8.13, pytorch1.10.0.

Tracking tests were performed using a laptop with hardware configuration of Intel(R) Core(TM) i5-10200H 2.40 GHz processor, 16G RAM, NVIDIA GeForce GTX 1650 Ti, and software environment of Win10 system, cuda10.1, python3.8.13, pytorch1.7.1 .

B. Analysis of model training results

In order to verify the performance of the models YOLOv5s-PA, YOLOv5s-CBAM, YOLOv5s-CA after adding PA, CBAM, CA attention, this paper adopts the above datasets to train the improved model and the original YOLOv5s model separately, the main parameters of model training are set as epochs=300, batch size=16, and the training results are shown in Table 2. The training results show that YOLOv5s-PA performs the best on both mAP@0.5and mAP@0.5:0.95 metrics, reaching 0.716 and 0.512 respectively, but at the same time, its number of parameters is relatively large. YOLOv5s-CBAM and YOLOv5s-CA models show little change in performance compared to YOLOv5s. Overall, the introduction of PA attention mechanism improved the performance of the model, so the algorithm in this paper uses the YOLOv5s-PA model.

TABLE II. Comparison of model training results

Model	mAP@0.5	mAP@0.5:0.95	Parameters
YOLOv5s-PA	**0.716**	**0.512**	**10628072**
YOLOv5s-CBAM	0.712	0.507	7059372
YOLOv5s-CA	0.713	0.505	7050544
YOLOv5s	0.712	0.506	7018216

C. Analysis of tracking results

This paper focuses on vehicle target tracking, and in order to verify the effectiveness of the algorithm, experiments are conducted on the vehicle sequences of the OTB100 dataset and LaSOT dataset, including BlurCar1~BlurCar4, Car1, Car2, Car4, Car24, CarDark, and CarScale in the OTB100, and the car-1~car-20 in the LaSOT, a total of 30 sequences. These sequences contain a variety of challenging factors such as motion blur, occlusion, rapid scale changes, rotation, low-resolution images, illumination changes, background clutter, etc.

Tracking experiments on the 30 sequences is performed using the YOLOv5s-PA model, and the average tracking speeds is 18.5fps. Considering that 16 of the 30 sequences have a resolution of 1280*720, the real-time performance of the algorithm is acceptable.

The tracking results of this paper's algorithm on some of the representative sequences are given in Fig. 8, where the white box region is the neighboring region determined by the algorithm, the yellow box is the detection boxes retained in the neighboring region, and the green box is the tracking result box. The car-7 sequence in Fig. 8(a) and the car-10 sequence in Fig. 8(b) both show large scale changes and target rotation, and this paper's algorithm is able to achieve stable tracking. Especially, there are more similar target interference near the target in frame #381 of the car-10 sequence, the algorithm proposed in this paper can greatly reduce the interference from similar targets in the background by restricting the adjacent areas, and can distinguish the similar targets efficiently by the Re-ID network feature extraction. In addition, when the target in the car-10 sequence is rotated, the appearance characteristics also change greatly, and the algorithm can effectively adapt to the change of the target's appearance by updating the Re-ID feature queue to ensure the stability of tracking. Frames #1315, #1325, and #1334 of the car-14 sequence in Fig. 8(c) show the process of the target being completely occluded, and the algorithm correctly tracked on the target by appearance matching after the target reappeared in frame #1334, and the target entered the shadow zone of the truck in frame #1703, where the illumination changed dramatically, and the algorithm tracked stably. The car-16 sequence in Figure 8 (d) has a low resolution, complex and chaotic background, and there are many similar target interference near the target in frames # 269 and # 1644. The appearance and scale of the target in frames # 426 and # 752 vary greatly, The algorithm proposed in this paper can track stably.

A comparison of the tracking results of different algorithms in the above video sequences is shown in Fig. 9 In order to ensure the objectivity of the results, the tracking results data of other algorithms are obtained from LaSOT_Evaluation_Toolkit_V2[18], in which some algorithms' tracking results are missing, so this paper adopts the algorithms with complete and representative tracking results, such as ECO, CFNet, MDNet, SaimFC, etc., for the comparison. As seen in the figure, the tracking results of this paper's algorithms are significantly better than the comparison algorithms in a variety of complex scenarios, such as occlusion, scale change, rotation change, illumination change, and target appearance change.

(a) car-7

(b) car-10

(c) car-14

(d) car-16

Fig. 8. Tracking results of ours algorithm

(a) Comparison of car-7 tracking results

(b) Comparison of car-10 tracking results

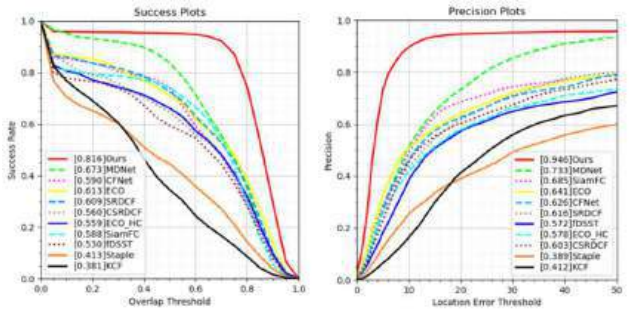
(c) Comparison of car-14 tracking results

(d) Comparison of car-16 tracking results

Groundtruth Ours ECO CFNet MDNet SaimFC

Fig. 9. Comparison of tracking results of different algorithms

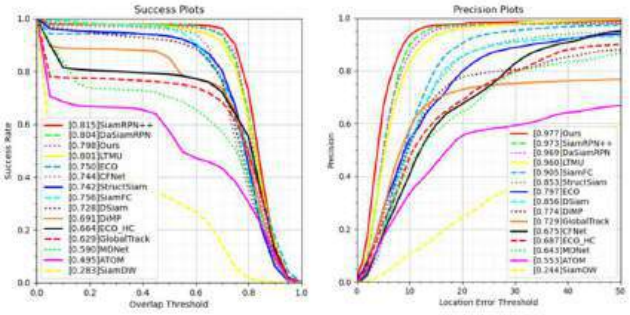
This paper evaluates tracker performance using OPE (Overlap Precision Evaluation) for accuracy assessment, and calculates success rate and precision rate of the tracking results of different algorithms on 30 vehicle sequences respectively, and the results are shown in Fig. 10. Fig. 10 shows that the proposed algorithm ranks first in both success rate and accuracy rate compared with MDNet, CFNet, ECO, SiamFC, etc., the success rate and accuracy rate have been significantly improved.



(a) success plots of OPE (b) precision plots of OPE

Fig. 10. Results of different algorithms on 30 vehicle sequences

LaSOT Evaluate_Toolkit_V2 only provides tracking result data on car-2, car-6, car-9, car-17 video sequences for algorithms such as SiamRPN++, DaSiamRPN, SiamDW, LTMU. This paper also compares these tracking results with OPE, and the results of different algorithms are shown in Fig. 11. The accuracy of the algorithm proposed in the paper ranks first, and its success rate is slightly lower than SiamRPN++, DaSiamRPN, and LTMU algorithms. This is mainly due to the annotation of the training set data during YOLOv5s-PA model training, and the overall performance of the model remains good.



(a) success plots of OPE (b) precision plots of OPE

Fig. 11. Results of different algorithms on car-2, car-6, car-9, and car-17

IV. Conclusion

In this paper, an improved method for visual single-target tracking is proposed, which is based on an improved YOLOv5s detection model and dynamic neighborhood target matching strategy. The performance of YOLOv5s is significantly improved by introducing the PA attention mechanism in the neck part of YOLOv5s. Combined with adaptive search region adjustment and IOU-based similarity metric, the mis-matching and computation are effectively reduced, and the tracking stability is enhanced by re-identification network feature extraction. The average success and accuracy of the algorithm on 30 vehicle video sequences containing multiple challenging factors are 0.816 and 0.946, respectively, which validates the algorithm's robustness. Future work will focus on further optimizing the model structure and expanding the diversity of the dataset to improve tracking accuracy in specific scenarios.

References

[1] R. Z. Han, W. Feng, Q. Guo, and Q. H. Hu, "Single Object Tracking Research: A Survey," Chinese Journal of Computers, vol. 45, no. 93, pp. 1877-1907, Sep. 2022.

[2] L. Chen and Y. G. Liu, "UAV Visual Target Tracking Algorithms: Review and Future Prospect," Information and Control, vol. 51, no. 1, pp. 23-40, Feb. 2022.

[3] J. Henriques, R. Caseiro, P. Martins, and J. Batista," High speed tracking with kernelized correlation filters," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no. 3, pp. 583-596, Mar 2015.

[4] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: efficient convolution operators for tracking," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 6931-6939.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, " Fully-convolutional siamese networks for object tracking," in Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops, 2016, pp. 850−865.

[6] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 8971-8980.

[7] Z. Zhu, Q. Wang, B. Li, W. Wu, Y. Yan and W. Hu, "Distractor-aware siamese networks for visual object tracking," in Proc. Eur. Conf. Comput. Vis. (ECCV), Cham: Springer, 2018, pp.103-119.

[8] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: evolution of siamese visual tracking with very deep networks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 4277-4286.

[9] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking " in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 4293-4302.

[10] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and H. S. Philip, "End-to-end representation learning for correlation filter based tracking," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 2805-2813.

[11] M. Danelljan, G. Bhat, F. S. Khan, et al., "ATOM: Accurate tracking by overlap maximization," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 4655–4664.

[12] N. Wojke, A. Bewley, and V. Ramesh, "Simple online and realtime tracking with a deep association metric," in Proc. IEEE Conf. on Image Processing (ICIP), Beijing, China, 2017, pp. 3645-3649.

[13] Y. Zou, P. Sun, Y. Jiang, D. Yang, F. Wang, Z. Yu, P. Li, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," arXiv:2110.06864 [cs.CV], Oct. 2021. [Online]. Available: https://arxiv.org/abs/2110.06864.

[14] Ultralytics. 2022. "YOLOv5, ". GitHub repository, https://github.com/ultralytics/yolov5.

[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," arXiv:1807.06521 [cs.CV], Jul. 2018. [Online]. Available: https://arxiv.org/pdf/1807.06521.pdf

[16] Q. Hou, D. Zhou, and J. Feng, "Coordinate Attention for Efficient Mobile Network Design," arXiv:2103.02907 [cs.CV], Mar. 2021. [Online]. Available: https://arxiv.org/abs/2103.02907.

[17] A. Goyal, A. Bochkovskiy, J. Deng, and V. Koltun, "Non-Deep Networks," arXiv:2110.07641 [cs.CV], Oct. 2021. [Online]. Available: https://arxiv.org/pdf/2110.07641

[18] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "LaSOT: a high-quality benchmark for large-scale single object tracking," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 5369-5378.

Layered Mixing Miner: A process Mining Algorithm for Complex Programming Debugging Data

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899496

Ruiqi Chen
Beijing University of Posts and Telecommunications
Beijing, China
crq@bupt.edu.cn

Jiangli Kong
Beijing University of Posts and Telecommunications
Beijing, China
kongjl@bupt.edu.cn

***Abstract*—In the context of increasingly complex application scenarios and unstructured debugging data in software engineering education, analyzing students' programming and debugging processes presents significant challenges. This study introduces the Layered Mixed Miner (LMM), a novel process mining algorithm designed to address these challenges by generating interpretable models from complex and unstructured student programming and debugging data. The algorithm integrates various process mining techniques by constructing an initial process tree, calculating filtering thresholds, and employing diverse methods to construct process models based on different filtering results. Consistency tests demonstrate that LMM achieves higher accuracy and precision compared to established benchmarks, providing a reliable and insightful representation of debugging behaviors. Moreover, expert feedback further confirms the algorithm's superior interpretability and usability over baseline methods.**

Keywords—Process mining, programming education, Educational Data Mining

I. INTRODUCTION

In today's era of rapid advancements in information technology, software engineering education faces unprecedented challenges. As application scenarios grow increasingly complex, analyzing the procedural data of students engaged in programming and debugging within large-scale or real-world projects—whether through project-based courses, advanced-level curricula, or structured short-term projects—has become critical[1].

Previously, researchers have attempted to analyze student behavior sequences using existing process mining algorithms. [2-4] However, due to the complexity of debugging behavior sequences, academic programming debugging data often exhibits an unstructured nature. When applied to such unstructured data, these algorithms tend to generate overly complex process diagrams, making it difficult to extract meaningful information.[5] To address this issue, past studies have typically preprocessed data to ensure the resulting models include as many activities and paths as possible while avoiding excessive complexity.[6-8] Researchers constantly strive to balance information richness with model complexity, aiming to produce analyzable models without overlooking key behaviors. Techniques such as clustering and classification are employed to adjust data granularity, and low-frequency data are often filtered out to simplify the models.[9,10] However, these methods risk ignoring critical information, limiting a comprehensive understanding of each step in students' debugging processes and hindering deeper insights into their learning states.

To tackle these challenges, this study proposes a Layered Mixing Miner specifically designed for complex and unstructured programming debugging data. The algorithm seeks to generate models of students' debugging behaviors from multiple perspectives.

The structure of this paper is as follows: Section 2 provides a detailed description of the methodology, including the design and implementation of the algorithm. Section 3 presents experimental results that validate the algorithm's effectiveness. Section 4 concludes the paper and discusses potential directions for future work..

II. METHOD

Layered Mixing Miner specifically uses two process mining techniques: causal matrix[11], process tree[12], with two filtering methods: activity activation count, activity dependency count. The following are the relevant definitions and algorithmic framework for Layered Mixing Miner.

A. Basic definitions

Case: Assuming that the executor or the object of execution of each piece of data in the data is case c, then the set of all cases is C, where $c \in C$.

Activity: Assuming that the behavior in the data is set to a, the set of all behaviors is A, where $a \in A$.

Time: Assuming the time in the data is set to t, the set of all times is T, where $t \in T$.

Additional attributes: If there are other attributes in each piece of data, set them to d_i as additional attributes ($i = 1,2,3$ ...), the set of all additional attributes is D_i, where $d_i \in D_i$.

Event: we treat each piece of data as an event e, where e is a tuple containing other contents, $e = c, a, t, d_1, \dots, d_i, c \in , a \in, t \in, d_i \in D.$ And $e \in E$.

Trace: Let σ be a trace, which is a collection of multiple events $\sigma = \{e_1, e_2, e_3, \dots, e_n\}, e_i \in E$. The trace is typically ordered by the timestamp of each event.

Event log: Assume that L represents the set of all given traces.

Directly-follows graph: A directly-follows graph is a way of describing a process that consists of nodes and edges, and

for each log L, a directly-follows graph G(L) of it can be generated. As shown in Figure 1, squares represent nodes, with each node indicating an activity. Arrows represent the control flow; if activity a is immediately followed by activity b, there is an arrow pointing from b to a. The numbers on the arrows indicate the frequency of each transition in the process.

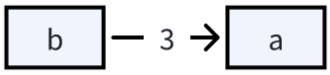


Figure 1: Example of Directly-follows graph

B. Definitions related to causal matrix

$>_L$:In the event log L if there exists a trace $\sigma_i = \{e_1, e_2, e_3, \dots, e_n\}, e_i \in E$, there are events e_i, e_j in the trace and e_j occurs immediately after the occurrence of the event e_i, and they are $a_i, a_j, a_i \in e_i, a_j \in e_j$, it can be decided that a_i pair a_j exists in log L as $a_i >_L a_j$.

$\rightarrow_L$: $a_i \rightarrow_L a_j$ in event log L if $a_i >_L a_j$ and $a_j \not>_L a_i$

$\#_L$: $a_i \#_L a_j$ in event log L if $a_i \not>_L a_j$ and $a_j \not>_L a_i$

$|_L$: $a_i |_L a_j$in event log L if $a_i >_L a_j$and $a_j >_L a_i$

$\Rightarrow_L$:Dependency measure notation, $a \Rightarrow_L b$ means to compute the degree of b's dependency on a. The formula is as follows, and it takes values between -1 and 1:

$$a \Rightarrow_L b = \left(\frac{|a>_L b| - |b>_L a|}{|a>_L b| + |b>_L a| + 1}\right) \quad (1)$$

C. Definitions related to process tree

Silent activity: Silent activity means that the current activity is empty, and setting the silent activity as the τ.

Process Trees: A process tree is a representation of a workflow net in the specific form of a rooted tree in which the leaves are labeled with activities and all other nodes are labeled with operators. Let the process tree be P, which contains a set of activities and operators. Set:

- the set of individual activities is a process tree, and $\{a\}, a \in A \cup \{\tau\}$ is a process tree.
- A process tree combined with operators forms a process tree structure, denoted as $P = \oplus (P_1, P_2, \dots)$ where $P_1 = \oplus (P_i, P_{i+1} \dots), P_2 = \cdots$,and so on. Here, $P_1, P_2, \dots P_n$ are all process trees .

Operators set: The operator collection $\oplus$ contains a series of operators. The operators are as follows

- $\rightarrow$: a process tree collection exists with multiple process trees $P_1, P_2, \dots, P_n$, and if the process trees in the collection are executed sequentially, there exists $\rightarrow (P_1, P_2, \dots, P_n)$.
- $\times$: a process tree collection exists with multiple process trees $P_1, P_2, \dots, P_n$, and if only a certain process tree is executed in the collection, there exists $\times (P_1, P_2, \dots, P_n)$.
- $\circlearrowleft$: a process tree collection exists with multiple process trees $P_1, P_2, \dots, P_n$, if all process trees in the collection are executed in a loop, there exists $\circlearrowleft (P_1, P_2, \dots, P_n)$
- $\wedge$: a process tree collection exists with multiple process trees $P_1, P_2, \dots, P_n$, and if all process trees in the collection are executed interleaved, there exists $\wedge (P_1, P_2, \dots, P_n)$. For the difference between interleaved execution and other execution methods, here is an attempt to explain through an example, for the process tree $P_1 = \oplus (\{a\}, \{b\}), P_2 = \oplus (\{c\}, \{d\}), P_3 = \oplus (\{e\}, \{f\})$, if there exists a $\wedge (P_1, P_2, P_3)$, then it may actually be represented in the log as $\{a, c, e, b, d, f\}$.

Cut: The operation of partitioning a direct successor graph G(L) into several mutually exclusive subsets is referred to as cutting. Each operator is associated with a specific cutting method, as detailed below.

- Sequence cut: Given a log L and its sub-logs $L_1, L_2, \dots, L_n$, consider the direct-follow graph $G(L)$ of L. If there exist two nodes a and b such that there is a path $a <_L b$, where $a \in L_i, b \in L_j, i < j$, removing this path constitutes a sequence cut. This corresponds to the $\rightarrow$operator..
- Exclusive choice cut: Given a log L and its sub-logs $L_1, L_2, \dots, L_n$, consider the direct-follow graph $G(L)$ of L. If there are no edges between $L_i, L_j, i \neq j$, this indicates the presence of an exclusive choice cut. This corresponds to the $\times$ operator.
- Parallel cut: Given a log L and its sub-logs $L_1, L_2, \dots, L_n$, consider the direct-follow graph $G(L)$ of L. If there exist two nodes a and b such that paths $a <_L b$ and $b <_L a$ both exist, where $a \in L_i, b \in L_j$, removing these paths constitutes a parallel cut. This corresponds to the $\wedge$ operator.
- Loop cut: Given a log L and its sub-logs $L_1, L_2, \dots, L_n$, consider the direct-follow graph $G(L)$ of L. If a and b are the start and end nodes of L_i, and there exists another sub-log $L_j, i \neq j$, such that there are nodes $c_i, i = 1,2, \dots, n$ with paths between cic_ici and LiL_iLi, where every path is either $c_i <_L a$ or $b <_L c_i$, removing these paths constitutes a loop cut. This corresponds to the $\circlearrowleft$ operator.
- Nontrivial cut: For n-ary cuts, the cut is nontrivial when n>1.

D. Definitions related to filtration

Count Function COUNT(): Assume a function COUNT() that calculates the number of elements within a given set.

Set of Activity Occurrence Counts O: In the log L, the set O represents the occurrence counts of various activities. For each activity a_i, there is a corresponding count $o_i = COUNT(\{e(e| \in\}E, a_i \in e, a_i \in A\})$, where $o_i \in O$.

Set of Activity Dependency Counts ADC: In the log L, the set ADC includes counts for activity dependencies. For each dependency $a_i >_L a_{i+1}$, there is a corresponding dependency count $adc_i = COUNT(\{(e_i, e_{i+1})(a_i| >_L\}), a_{i+1}, a_i \in e_i, a_{i+1} \in e_{i+1}, a_i, a_{i+1} \in A\}$, with $adc_i \in ADC$.

E. algorithmic architecture

The LLM(Layered Mixing Miner) architecture is shown in the Figure 2, with the core idea being to calculate the filtering level using an initially generated process tree and the original log. Based on the results after filtering, different methods are then applied to generate various process models. The process follows these steps:

- Input the log.
- Generate an initial Directly-Follows Graph based on the log.
- Identify different cuts in the Directly-Follows Graph and recursively partition sub-logs to create an initial process tree.
- Calculate the activity occurrence set O and activity dependency set ADC using the log and the initial process tree.
- Based on O and ADC, different levels of filtering are applied to the logs and process trees to generate data at various levels. The resulting data is then used to compute causal matrices or sub-log cuts of process trees, generating corresponding process models. Process trees are recursively divided through sub-log cuts, segmenting the overall log and analyzing relationships between activities. In contrast, causal matrices explore the relationships among all activities as a whole. Tools for generating process models using high-level filtering differ depending on the focus: causal matrices are used when prioritizing processes with higher information content and involve cutting the model at a more comprehensive level. Conversely, process tree sub-log cuts are used to identify processes with smaller amounts of information.
- Return the result.

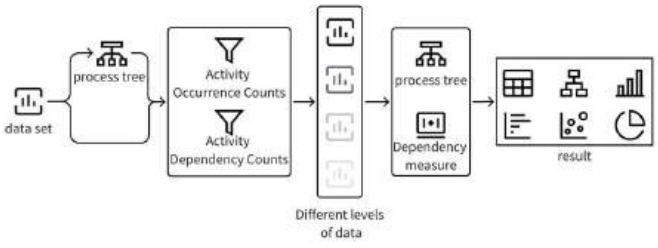


Figure 2: The LMM(Layered Mixing Miner) architecture

III. EXPERIMENTS AND ANALYSIS OF RESULTS

A. Dataset

The dataset originates from a programming debugging dataset collected from students at Beijing University of Posts and Telecommunications. Students were tasked with correcting code in a WeChat mini-program that contained 14 bugs, each covering different knowledge points and varying levels of difficulty, including tasks such as code completion and correction. The bugs were classified into four levels based on their comprehensive difficulty. Students needed to debug the code to ensure the mini-program operated correctly and met assignment requirements. The final dataset contains each student's unique ID, the specific issue they debugged, debugging time, and whether debugging was successful. In total, the dataset comprises 14,192 records, involving 480 students across 14 categories.

General process model metrics aim to evaluate the descriptive accuracy of a model, with the expectation that a single diagram can represent most of the information contained in the logs. However, the results of this algorithm include the generation of multiple process model diagrams that vary in the amount of information displayed and their alignment with the logs. Therefore, the evaluation approach not only involves consistency checks to assess the fitness of the generated models but also incorporates expert user experience reports on the models.

B. the descriptive accuracy of a model

To ensure that the models generated by the algorithm accurately reflect portions of the complete log, we performed conformance checking between the model diagrams and the log. The algorithm produces various types of model diagrams, including disconnected graphs, decision trees, Petri nets, and direct-follow graphs, each containing varying amounts of information. For instance, Figure 3 compares a disconnected graph and a Petri net, with the disconnected graph containing less information. These model diagrams were compared against those produced by other process mining algorithms to evaluate their conformance checking result. The test results are in Table 1. In this context, fitness assesses the model's ability to reproduce behaviors observed in the log, while precision evaluates the model's ability to fit the behaviors in the log without overgeneralization.[13]

The results indicate that both fitness and precision of the models are maintained above a certain threshold. Models with higher information content tend to demonstrate greater fitness, while more concise models, with distilled information, exhibit higher precision. This suggests that all results serve effectively as partial representations of information extracted from the complete log, reflecting real-world conditions. Compared to inductive mining algorithms[11] and heuristic mining algorithms[14], these models show a certain degree of improvement.

C. Expert Experience Report

This algorithm, designed for student programming debugging sequences, generates process model diagrams that highlight different aspects of the data. Its improvement objectives include enhanced information extraction capabilities, enabling analysts to obtain more insights in a simplified manner.

The overall effectiveness of the algorithm can be assessed through analysts' experience reports. For this purpose, we provided experts with results from the layered hybrid miner and

TABLE 1: CONFORMANCE CHECKING RESULTS BETWEEN LLM AND OTHER PROCESS MINING MODELS

	Unconnected graph(LMM)	process tree(LMM)	Patri net(LMM)	Directly-follows graph (LMM)	Induction Mining	Heuristic Mining	Fuzzy Mining[15]
Fitness	**0.77**	**0.77**	0.62	0.52	0.76	0.72	0.66
Precision:	0.61	0.64	**0.98**	**0.99**	0.56	0.63	0.88

TABLE 2: EXPERT EXPERIENCE REPORT

	complexity	accuracy	interpretability
Induction Mining	**8.5**	6.9	3.2
fuzzy Mining	3.8	3.2	6.5
Heuristic Mining	7.7	**7.6**	4.5
Layered Mixing Miner	2.3	7.5	**6.9**

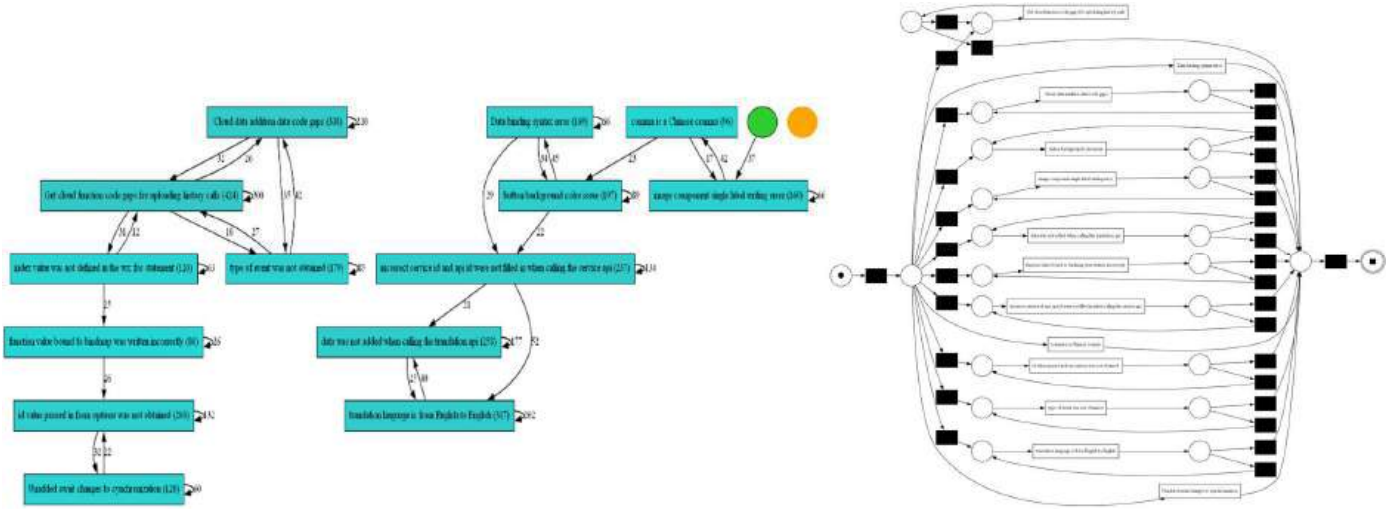


Figure 3: The LMM result(The left image is a disconnected graph, and the right image is a Petri net)

compared these with outputs from other mining algorithms using student debugging data. A survey was then conducted to collect user experience reports from all experts, thereby gauging the algorithm's effectiveness improvements.

We conducted a survey to collect user experience reports from 10 experts, including instructors and teaching assistants from the introductory courses at Beijing University of Posts and Telecommunications. These experts evaluated the outputs generated by different algorithms applied to programming debugging sequence data. They analyzed the results and provided guidance for both students and instructors, which was then implemented in practical teaching. Based on their user experience, the experts rated the results generated by each algorithm. Specifically, "complexity" indicates the level of disorder in the process model diagrams, "accuracy" reflects how closely the model aligns with the real-world scenarios observed by the experts, and "interpretability" measures how easily the experts could understand the process model diagrams. The results, shown in the Table 2, reflect scores from 1 to 10 across different aspects of user experience.

Through the survey, we observed that, while maintaining a high level of accuracy and low complexity, the experts found the layered hybrid miner to be more interpretable and easier to understand.

IV. CONCLUSION

This study successfully proposes an innovative process mining algorithm tailored for handling complex programming debugging data. The algorithm integrates various process mining techniques by constructing an initial process tree, calculating filtering thresholds, and employing diverse methods to construct process models based on different filtering results. Consistency tests demonstrate that the accuracy and precision of the algorithm's mining results exceed established benchmarks, confirming its effectiveness in extracting critical information from comprehensive log data. These results provide an accurate and reliable reflection of actual debugging behaviors.

Furthermore, through expert application and feedback collection, we found that the mining results based on the Layered Mixed Miner (LMM) algorithm exhibit excellent interpretability while maintaining high accuracy and low

confusion. This finding is significant for enhancing the usability and user-friendliness of process mining outcomes.

Despite these achievements, certain limitations remain. For instance, the algorithm's performance needs further optimization when handling particularly large datasets, and its generalizability across different programming languages and debugging environments requires additional validation. Future research could focus on enhancing algorithm performance, improving generalization capabilities, and exploring broader application scenarios.

In conclusion, this study not only advances the theoretical development of process mining technologies but also provides a powerful tool for software engineering education and debugging behavior analysis. We hope that the outcomes of this research will inspire further studies on process mining techniques and foster their application in a wider range of fields.

REFERENCES

[1] Qi Zhou. Design and implementation of data acquisition and analysis tool for programming debugging process[D]. Beijing University of Posts and Telecommunications, 2023. DOI:10.26969/d.cnki.gbydu. 2023. 002323.

[2] Nammakhunt A, Sukkri M, Porouhan P, et al. Applying Process Mining Techniques for Data-Driven Self-Learning Behavior Analysis in E-Learning Systems[C]//2023 21st International Conference on ICT and Knowledge Engineering (ICT&KE). IEEE, 2023, 21: 1-12.

[3] Domínguez C, Jaime A, Pérez B, et al. Using Process Mining to Analyze Tasks Involvement and Collaboration in a Student Generated Questions Activity[C]//Proceedings of the 16th International Conference on Computer-Supported Collaborative Learning-CSCL 2023, pp. 19-26. International Society of the Learning Sciences, 2023.

[4] Thiyagarajan G, Prasanna S. Process Mining-Based Behavioral Modeling of Learners in Self-paced Learning Environment[C]//International Conference on Signal & Data Processing. Singapore: Springer Nature Singapore, 2022: 121-132.

[5] Koschmider A, Aleknonytė-Resch M, Fonger F, et al. Process Mining for Unstructured Data: Challenges and Research Directions[J]. arXiv preprint arXiv:2401.13677, 2023.

[6] Fluxicon**.** Disco user's guide [EB/OL]. Fluxicon, 2019 [2023-01-07]. Available: https://fuxicon.com/disco/fles/Disco-User-Guide.pdf.

[7] Marin-Castro H M, Tello-Leal E. Event log preprocessing for process mining: a review[J]. Applied Sciences, 2021, 11(22): 10556.

[8] dos Santos Garcia C, Meincheim A, Junior E R F, et al. Process mining techniques and applications–A systematic mapping study[J]. Expert Systems with Applications, 2019, 133: 260-295.

[9] Liu F, Zhao L, Zhao J, et al. Educational process mining for discovering students' problem-solving ability in computer programming education[J]. IEEE Transactions on Learning Technologies, 2022, 15(6): 709-719.

[10] Zhang F, Liu D, Liu C. Mooc video personalized classification based on cluster analysis and process mining[J]. Sustainability, 2020, 12(7): 3066.

[11] Weijters A J M M, van Der Aalst W M P, De Medeiros A K A. Process mining with the HeuristicsMiner algorithm[J]. 2006.

[12] Leemans S J J, Tax N, ter Hofstede A H M. Indulpet miner: Combining discovery algorithms[C]//On the Move to Meaningful Internet Systems. OTM 2018 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part I. Springer International Publishing, 2018: 97-115.

[13] Van der Aalst W, Adriansyah A, Van Dongen B. Replaying history on process models for conformance checking and performance analysis[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2012, 2(2): 182-192.

[14] Weijters A J M M, van Der Aalst W M P, De Medeiros A K A. Process mining with the HeuristicsMiner algorithm[J]. 2006.

[15] Günther C W, Van Der Aalst W M P. Fuzzy mining–adaptive process simplification based on multi-perspective metrics[C]//International conference on business process management. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 328-343.

Incomplete Multi-view Clustering Based on Dual Aggregation Strategy and Dual Contrastive Completion

Yu Wang
School of Information Science and Engineering
Yunnan University
Kunming, China
972850912@qq.com

Lihua Zhou*
School of Information Science and Engineering
Yunnan University
Kunming, China
lhzhou@ynu.edu.cn
(Corresponding author)

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899505

***Abstract*—In real-world applications, data often comes from multiple views, and learning from multi-view data can improve clustering accuracy. While existing incomplete multi-view clustering methods have made progress in utilizing representation information, they often overlook global information, particularly raw data, which is seldom considered. To address this, we propose an incomplete multi-view clustering method based on dual aggregation and dual contrastive completion. Our approach aggregates both raw data and representations to capture global information, which is then combined with local view-specific information for contrastive learning and missing representation completion. The completed representations are used for clustering, with clustering results fed back to guide representation learning, enabling mutual optimization. Extensive experiments on three multi-view datasets demonstrate the effectiveness and robustness of our approach.**

Keywords—dual aggregation, dual contrastive completion, incomplete multi-view learning clustering

I. INTRODUCTION

Clustering is a key machine learning task that groups data based on similarity, with high similarity within clusters and low similarity between them. As data acquisition methods evolve, multi-view data, where the same object is described from different perspectives, has become more prevalent, making multi-view clustering (MVC)[1] an important area of research. While most MVC methods assume complete data[2], real-world scenarios often involve missing data due to issues like sensor failures or network problems, leading to incomplete multi-view data.

The main challenge in incomplete multi-view clustering (IMVC) is handling missing data. A simple approach is to exclude samples with missing views[3], but this performs poorly when the missing rate is high. Most methods first complete the missing data and then apply traditional clustering techniques. Common completion strategies include zero filling[4], using the mean of observed values[5], or leveraging across-view neighbor information[6][7][8]. While these methods, including contrastive learning, improve clustering, they often overlook global feature information across all views. Unlike local information, which focuses on pairwise relationships, global views provide a more comprehensive understanding and enhance clustering performance[9]. Additionally, many methods focus on encoded representations while neglecting the raw data' s valuable features[10]. Both raw data and encoded representations contribute to robust feature learning, suggesting that using both can improve clustering, especially in IMVC.

To leverage both global information from multi-view representations and raw data in IMVC, we propose a dual-aggregation and dual-contrastive-completion (DA-DCC) method. Dual-aggregation includes representation aggregation and raw data aggregation. In representation aggregation, each view's representation is learned through an autoencoder, and these are aggregated to learn global information across views. Raw data aggregation directly aggregates the raw data of each view to capture global information. These two types of global information improve consistency learning and the completion of missing representations. Dual-contrastive completion includes dual-contrastive learning and dual-information completion. Dual-contrastive learning contrasts global feature information with view-specific features and across views to improve positive and negative sample pair selection and strengthen consensus representation learning. Dual-information completion uses both global and local information to fill missing representations, improving completion quality and clustering performance.

The main contributions of this paper are as follows:

a) We propose a dual-aggregation and dual-contrastive completion(DA-DCC) method. Dual aggregation captures latent feature information from both raw data and representations. This information, combined with view-specific local information, is used for contrastive learning and missing representation completion. Dual aggregation effectively increases the information available for learning consistency and completing missing representations, while dual-contrastive completion fully utilizes all information, optimizing representation learning and completion.

b) We validate the model through clustering comparisons, parameter sensitivity analysis, and ablation studies on three real-world multi-view datasets with varying missing rates.

II. RELATED WORK

Zhou et al.[1] categorize existing IMVC methods into six types: non-negative matrix factorization, multiple kernel learning, graph learning, subspace learning, deep learning, and contrastive learning. Matrix factorization uncovers data structure and learns consensus representations; multiple kernel learning targets view-specific or consensus representations;

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

graph learning focuses on consistency or consensus representations; subspace learning uses local descriptions to avoid high-dimensional computations; autoencoders infer missing features; and contrastive learning seeks highly consistent and complementary representations.

These methods mainly focus on intra- and inter-view information, lacking the ability to capture and utilize view information from a global perspective. They also prioritize data aggregation at the low-dimensional representation stage, neglecting the original data. Global aggregation captures high-level features across views. For instance, GACFAgg[9] enhances similarity via structural contrastive learning; DCGP[11] uses global similarity to guide sample selection; RecFormer[12] aggregates representations to reconstruct missing samples. While low-dimensional representations reduce noise, they may discard valuable discriminative information[10]. In contrast, we extract latent features from both raw data and representations to enhance consistency learning and complete missing representations.

III. Method

DA-DCC uses both global and local information from all views to improve clustering performance. The model consists of three main modules: global information capture, dual-information contrastive completion, and clustering (Figure 1). The global information capture module extracts global information from both raw data and representations. The dual-information contrastive completion module contrasts global and local information to complete missing representations. The clustering module performs clustering on the completed data and uses pseudo-labels to guide representation learning, enhancing the quality of the completed views.

A. Global Information Capture

a) Global Information Capture in Raw Data: To capture global information from the raw data, DA-DCC aggregates the raw data from all views, as shown in equation (1):

$$X^{(A)} = \sum_{i=1}^{V} \omega_i X^{(i)}, \tag{1}$$

where ω represents the weight matrix learned through the attention network to aggregate all representations X, ω_i denotes the weight of the representation from the the i-th view, and $X^{(i)}$ represents the representation of the i-th view.

b) Global Information Capture in Representation Data: We use an autoencoder to learn the representation of each view, with encoding and decoding shown in equation (2):

$$Z^{(v)} = f^{(v)}(X^{(v)}), \quad \hat{X}^{(v)} = g^{(v)}(Z^{(v)}), \tag{2}$$

where $f^{(v)}(\cdot)$ is the encoder for the v-th view, $g^{(v)}(\cdot)$ is the decoder for the v-th view, and $Z^{(v)}$ represents the representation of $X^{(v)}$. The loss function of the autoencoder is shown in equation (3):

$$\mathcal{L}_{re} = \sum_{v=1}^{V} \sum_{t=1}^{n} \left\| X_t^{(v)} - g^{(v)}(f^{(v)}(X_t^{(v)})) \right\|_2^2, \tag{3}$$

After learning the representations, the data from all views are aggregated to capture global information, as shown in equation (4):

$$Z^{(\bar{A})} = \sum_{i=1}^{V} \omega_{Z^{(i)}} Z^{(i)}, \tag{4}$$

where ω represents the weight matrix learned through the attention network to aggregate all representations Z, $\omega_{Z^{(i)}}$ denotes the weight of the representation from the i-th view, and $Z^{(i)}$ represents the representation of the i-th view.

B. Dual-Information Contrastive Completion

a) Dual-Information Contrastive Learning: To capture high consistency across views, we use dual-information contrastive learning, contrasting global and view-specific local information, as well as local information across views. The global and local contrastive losses, shown in equations (5) and (6), guide the optimization by evaluating feature information through their probability distributions.

$$\mathcal{L}_{gcl} = \sum_{i=1}^{l_1} \sum_{j=1}^{l_2} p'_{i,j} log(\frac{p'_{i,j}}{(p'_i \times p'_j)^b}), \tag{5}$$

$$\mathcal{L}_{lcl} = \sum_{i=1}^{l_1} \sum_{j=1}^{l_2} p''_{i,j} log(\frac{p''_{i,j}}{(p''_i \times p''_j)^b}), \tag{6}$$

Here, l_1 denotes the dimension of the first feature, and l_2 denotes the dimension of the second feature. $p'_{i,j}$ denotes the joint probability distribution of global and local information, while p'_i and p'_j represent the marginal probability distributions of global and local information, respectively. $p''_{i,j}$ is the joint probability distribution of the local information from two views, and p''_i and p''_j are the marginal probability distributions of the local information from the two views, respectively. b is a constant, determined based on the study by Lin et al[6].

The two losses are combined to form the overall loss in the contrastive learning phase, as shown in equation (7).

$$\mathcal{L}_{cl} = \alpha \mathcal{L}_{gcl} + \mathcal{L}_{lcl}, \tag{7}$$

where is α weighting parameter.

b) Dual-Information Completion: We project the global raw data into the same low-dimensional space as the representations for further operations, as shown in equation (8).

$$Z^{(A)} = \sum_{i=1}^{V} \omega_{(i)} X^{(i)}, \tag{8}$$

In multi-view data, different views of the same object provide useful information for completing missing representations, including both global and view-specific local information. Therefore, during the dual-information completion phase, both global and local information are used to complete the missing representations, as shown in equations

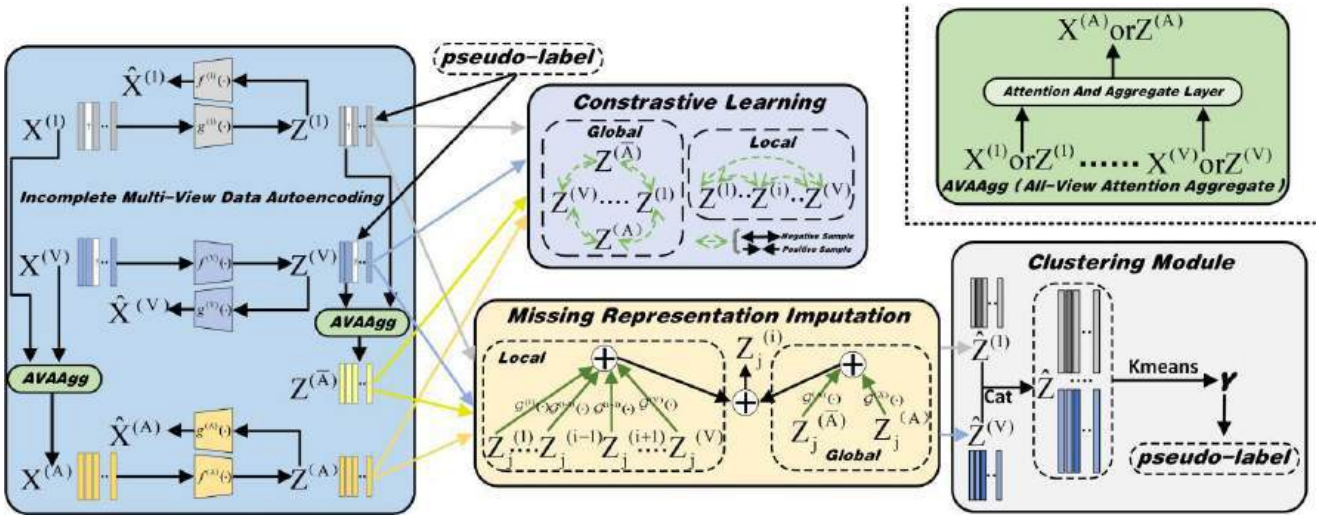


Fig. 1. The framework of DA-DCC. As shown in the figuredenotes, $\{X^{(v)} \in R^{n\times m_v}\}_{v=1}^{V}$ denotes the multi-view data composed of n samples from V views. m_v denotes the feature dimension of the v-th view, $\{Z^{(v)}\}_{v=1}^{V}$ represents the encoded information, and $X^{(A)}$ represents the global information obtained from the raw data. $Z^{(A)}$ is the representation of $X^{(A)}$, and $Z^{(\bar{A})}$ represents the global information obtained from the representations. $Z_j^{(i)}$ denotes the representation of the j-th sample from the i-th view, and $\hat{Z}^{(i)}$ represents the completed representation. The function $\mathcal{G}^{(i)}(\cdot)$ in the missing representation completion stage is a learnable parameterized model. The top-right corner's AVAAgg represents the attention aggregation module. In the encoding phase, $f^{(v)}(\cdot)$ denotes the encoder, and $g^{(v)}(\cdot)$ denotes the decoder.

(9) and (10).

$$Z^{(i)}{}_g = \mathcal{G}^{(A)}(Z^{(A)}) + \mathcal{G}^{(\bar{A})}(Z^{(\bar{A})}), \tag{9}$$

$$Z^{(i)}{}_l = \sum_{j=1, j\neq i}^{V} \mathcal{G}^{(j)}(Z^{(j)}), \tag{10}$$

$Z^{(i)}{}_g$ denotes the result after completing the i-th view using global information, $Z^{(i)}{}_l$ denotes the result after completing the i-th view using local information, and $\mathcal{G}^{(\cdot)}(\cdot)$ represents the learnable parameterized model.

The combination of both information completions forms the overall completion, as shown in equation (11).

$$\hat{Z}^{(i)} = \alpha Z^{(i)}{}_g + Z^{(i)}{}_l, \tag{11}$$

$\hat{Z}^{(i)}$ denotes the completed representation of the i-th view.

We designs loss functions for the two types of completion to optimize the parameterized model. The losses for local and global information completion are shown in equations (12) and (13), respectively.

$$\mathcal{L}_{lmri} = \sum_{i=1}^{V}\sum_{j=1, j\neq i}^{V} \left\| \mathcal{G}^{(j)}(Z^{(j)}) - Z^{(i)} \right\|_2^2, \tag{12}$$

$$\mathcal{L}_{gmri} = \sum_{i=1}^{V} \left\| \mathcal{G}^{(A)}(Z^{(A)}) + \mathcal{G}^{(\bar{A})}(Z^{(\bar{A})}) - Z^{(i)} \right\|_2^2, \tag{13}$$

The two losses are combined to form the overall loss in the completion phase, as shown in equation (14):

$$\mathcal{L}_{mri} = \alpha \mathcal{L}_{gmri} + \mathcal{L}_{lmri}, \tag{14}$$

C. Clustering Module

The completed representations are clustered using the clustering module. In DA-DCC, k-means is chosen as the clustering method, as shown in equation (15).

$$y = kmeans(Cat(Z^{(1)}, ..., Z^{(V)})) \tag{15}$$

where $Cat(\cdot)$ denotes the concatenation of $\{Z^{(1)}, ..., Z^{(V)}\}$.

To integrate representation learning and clustering, we use clustering results y as pseudo-labels to guide representation learning. The loss for this is shown in equation (16).

$$\mathcal{L}_{pl} = -\frac{V}{n}\sum_{j=1}^{n} y_j + \sum_{i=1}^{V} \log(\sum_{j=1}^{n} Z^{(i)}{}_j) \tag{16}$$

Here, y_i denotes the pseudo-label of the i-th sample, n is the number of samples, and $\sum_{i=1}^{V} \log(\sum_{j=1}^{n} Z^{(i)}{}_j)$ represents the concatenation of the representations from all views.

In summary, the objective function of DA-DCC consists of four loss functions: the autoencoder network reconstruction loss, dual-information contrastive loss, dual-information completion loss, and pseudo-label guidance loss. The objective function is shown in equation (17).

$$\mathcal{L} = \mathcal{L}_{re} + \mathcal{L}_{cl} + \mathcal{L}_{mri} + \mathcal{L}_{pl} \tag{17}$$

IV. Experiments

A. Experiments Settings

a) Datasets: We conduct experiments using three real-

TABLE I. RESULTS OF ALL METHODS UNDER DIFFERENT MISSING RATES

	MR	Caltech101-20			LandUse-21			CUB		
Metrics		**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**
CVCL	0	32.54	55.76	24.54	23.78	25.44	10.48	64.86	62.36	47.52
ProImp		36.26	55.94	26.02	25.14	29.76	12.12	79.27	76.25	66.14
DCP		70.95	69.18	75.59	25.74	30.51	11.23	52.24	61.88	38.34
COMPLETER		66.40	68.37	70.74	25.11	31.93	13.12	56.70	65.09	47.35
Ours		**72.02**	**70.79**	**76.61**	**29.00**	**32.89**	**13.66**	**83.56**	**76.97**	**69.31**
CVCL	0.5	29.86	43.10	14.11	16.90	16.89	4.37	47.90	48.74	27.69
ProImp		33.68	53.99	24.33	23.79	26.73	10.65	71.73	66.02	54.14
DCP		68.63	64.28	74.36	25.17	29.81	9.63	23.23	23.93	3.29
COMPLETER		71.24	70.25	81.72	21.65	27.97	11.21	55.27	61.49	45.35
Ours		**76.42**	**72.11**	**88.93**	**28.59**	**33.09**	**11.60**	**73.53**	**68.62**	**57.93**

[a.] Due to space limitations, only the clustering results with missing rates of 0, and 0.5 are presented in the table.

TABLE II. RESULTS OF ALL METHODS UNDER DIFFERENT MISSING RATES

Components				MR	Caltech101-20			LandUse-21			CUB		
$\mathcal{L}_{cl}$	$\mathcal{L}_{re}$	$\mathcal{L}_{mri}$	$\mathcal{L}_{pl}$		**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**
√				0	61.90	68.86	63.77	28.83	32.21	13.62	77.70	75.84	65.61
	√				65.16	59.31	60.76	23.23	27.28	8.25	54.70	54.20	38.22
		√			55.72	47.70	47.05	21.48	27.63	8.13	72.00	67.55	55.04
			√		52.71	46.60	53.08	23.15	26.00	9.45	65.07	64.22	49.56
√	√	√			73.21	70.75	80.60	29.01	32.86	14.06	80.70	76.51	66.87
√	√		√		75.92	72.86	85.65	29.02	33.11	14.14	82.03	74.70	66.21
√		√	√		55.72	66.19	55.34	29.06	33.16	13.75	78.90	75.49	64.88
	√	√	√		61.79	56.97	56.61	23.67	28.26	8.67	45.37	47.64	28.51
√	√	√	√		**76.03**	**72.92**	**85.85**	**29.50**	**33.19**	**14.19**	**83.56**	**76.97**	**69.31**
√				0.5	43.28	49.74	32.09	22.76	26.67	3.96	68.97	65.00	51.88
	√				54.72	34.57	31.71	17.87	21.06	4.98	39.40	35.58	17.69
		√			49.88	32.14	38.27	14.99	16.22	3.66	37.57	35.66	15.55
			√		33.14	36.10	21.52	17.69	17.87	4.11	35.07	35.95	15.21
√	√	√			76.21	71.48	88.49	27.41	31.80	10.13	68.40	63.36	49.53
√	√		√		65.88	61.93	74.94	22.19	26.26	4.05	71.00	66.53	53.16
√		√	√		55.68	63.21	56.14	27.49	31.89	10.30	69.10	67.35	55.09
	√	√	√		50.69	32.86	25.27	22.12	27.33	8.36	33.23	32.37	14.21
√	√	√	√		**76.42**	**72.11**	**88.93**	**28.59**	**33.09**	**11.60**	**73.53**	**68.62**	**57.93**

[b.] Due to space limitations, only partial results for the four loss combinations are presented in the table.

world multi-view datasets: Caltech101-20[6], Landuse21[6], and CUB[13]. Table 1 summarizes the key statistics of these three datasets. Imcomplete multi-view data is simulated by randomly removing m instances from n instances, where the missing rate is defined as m/n.

TABLE III. THE INFORMATION OF THE DATASETS IN OUR EXPERIMENTS

Datasets	Samples	Views	Features	Classes
Caltech101-20	2386	3	1984/512/928	20
LandUse-21	2100	3	20/59/40	21
CUB	600	2	1024/300	10

b) Comparison Methods and Evaluation Metrics: We compare four baseline methods: one complete multi-view and three IMVC approaches. The experiment evaluates whether adding global information and using dual-contrastive completion for contrastive learning and missing representation completion improve clustering performance.

CVCL[14]: A complete multi-view method that obtains co--nsistent semantic labels from a single view and uses these labels to measure the intrinsic relationships between data samples.

COMPLETER[15]: An IMVC that utilizes information theory to link and mutually promote cross-view consistency th--eory to link and mutually promote cross-view consistency learning and the recovery of missing views.

DCP[6]: An IMVC method that learns consistent representations through contrastive learning and then uses dual contrastive prediction to generate missing views of samples.

ProImp[13]: An IMVC method that introduces a dual-attention layer to learn mutual representations of samples and prototypes, enhancing the commonality of instances, and uses dual contrastive learning to maintain the generality of views.

we use clustering accuracy (ACC), normalized mutual information (NMI), and adjusted Rand index (ARI)[6] as evaluation metrics. Higher values of these metrics indicate better clustering performance.

B. Experimental Results

Table 2 presents the results of five methods on three datasets. The best and second-best values are highlighted in bold and underlined, respectively. The proposed method outperforms all baselines, showing that dual aggregation of global information from raw data and representations captures consistent, latent features. The results confirm that global information benefits both complete and incomplete methods, improving representation learning and completion.

C. Model Analysis

a) Ablation Study: Equation (17) shows that the proposed model includes four loss functions. Table 3 presents clustering results under missing rates of 0 and 0.5 with different loss function combinations. A ' √ ' indicates the loss function is used, otherwise, it is not used.

As observed from Tables 3 , the model achieves the best performance when all four loss functions are used. For example, on the Caltech101-20 dataset with a missing rate of 0.5, the ACC value when using all four losses improves by approximately 33%, 22%, 27%, and 43%, respectively, compared to using each loss function individually. Additionally, it can be noted that with a missing rate of 0.5, the optimal loss combination varies across datasets. For instance, At a missing rate of 0.5, on the LandUse-21 dataset, the highest performance is achieved when the combination of $\mathcal{L}_{cl}$, $\mathcal{L}_{mri}$, and $\mathcal{L}_{pl}$ is used, whereas for Caltech101-20, the optimal combination includes $\mathcal{L}_{cl}$, $\mathcal{L}_{re}$, and $\mathcal{L}_{mri}$. This variability is likely due to the differing sensitivities of each dataset to the various loss functions.

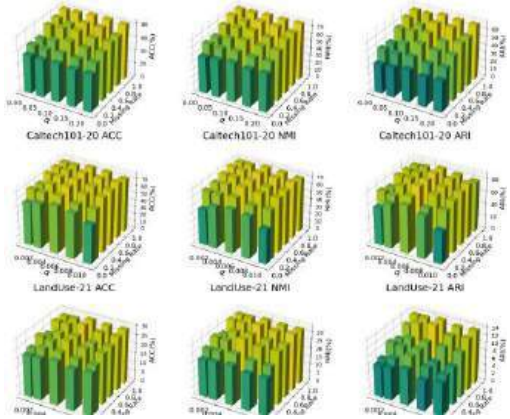


Fig. 2. Parameter Sensitivity Analysis of Three Datasets under Different Missing Rates

b) Parameter Sensitivity Analysis: We defines each loss module with a single weighting parameter α . Figure 2 shows how the three evaluation metrics change across three datasets as α varies, with the missing rate ranging from [0, 0.9], while the range of α is [0.01, 0.2] for the CUB dataset and [0.001, 0.01] for the other two datasets. The value of α varies for different datasets. Figure 2 demonstrates that the model's performance remains stable across the parameter range, indicating low sensitivity to parameter selection.

D. Conclusion

We propose an IMVC method based on dual-aggregation and dual-contrastive completion. Experimental results show its effectiveness and robustness. As shown in Table 2, the method outperforms existing approaches at different missing rates, highlighting the value of global information from both raw data and representations in improving both complete and incomplete methods. Table 3 validates the effectiveness of different loss functions at varying missing rates. Additionally, Figure 2 demonstrates that the model is insensitive to parameter selection across different missing rates.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (62062066, 62266050 and 62276227), Yunnan Fundamental Research Projects (202201AS070015); Yunnan Key Laboratory of Intelligent Systems and Computing (202205AG070003), the Block-chain and Data Security Governance Engineering Research Center of Yunnan Provincial Department of Education.

REFERENCES

[1] Zhou, L., Du, G., Lü, K., Wang, L., Du, J. (2024). A survey and an empirical evaluation of multi-view clustering approaches. ACM Comput. Surv., 56(7), 1–38.

[2] Wen, J., et al. (2023). A Survey on Incomplete Multiview Clustering. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 53(2), 1136-1149.

[3] Wen, J., Liu, C., Deng, S., Liu, Y., Fei, L., Yan, K., Xu, Y. (2023). Deep double incomplete multi-view multi-label learning with incomplete labels and missing views. IEEE Transactions on Neural Networks and Learning Systems. 35(8), 11396-11408

[4] Zhao, H., Liu, H., Fu, Y. (2016). Incomplete multi-modal visual data grouping. IJCAI'16, 2392–2398.

[5] Li, S.-Y., Jiang, Y., Zhou, Z.-H. (2014). Partial multi-view clustering. AAAI'14, 1968–1974.

[6] Lin, Y., Gou, Y., Liu, X., Bai, J., Lv, J., Peng, X. (2023). Dual Contrastive Prediction for Incomplete Multi-View Representation Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(4), 4447-4461.

[7] Fang, S., Yang, Z., Chen, J. (2024). Incomplete multi-view clustering via diffusion completion. Multimedia Tools and Applications, 83(18), 55889-55902.

[8] Chao, G., Jiang, Y., Chu, D. (2024). Incomplete contrastive multi-view clustering with high-confidence guiding. In Proceedings of the AAAI Conference on Artificial Intelligence, 38(10), 11221-11229.

[9] Yan, W., Zhang, Y., Lv, C., Tang, C., Yue, G., Liao, L., Lin, W. (2023). Gcfagg: Global and cross-view feature aggregation for multi-view clustering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 19863-19872.

[10] Du, G., Zhou, L., Lü, K., Wu, H., Xu, Z. (2023). Multiview Subspace Clustering With Multilevel Representations and Adversarial Regularization. IEEE Transactions on Neural Networks and Learning Systems, 34(12), 10279-10293.

[11] Yin, Z., Zhou, L., Wang, L., Chen, H. (2024). Dual-Contrastive Multi-view Clustering Under the Guidance of Global Similarity and Pseudo-label. In Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, 35-49.

[12] Liu, C., Wen, J., Wu, Z., Luo, X., Huang, C., Xu, Y. (2024). Information Recovery-Driven Deep Incomplete Multiview Clustering Network. IEEE Transactions on Neural Networks and Learning Systems, 35(11), 15442-15452.

[13] Li, H., Li, Y., Yang, M., Hu, P., Peng, D., Peng, X. (2023). Incomplete multi-view clustering via prototype-based imputation. IJCAI '23, 3911–3919.

[14] Chen, J., Mao, H., Woo, W. L., Peng, X. (2023). Deep multiview clustering by contrasting cluster assignments. In Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), 16752-16761.

[15] Lin, Y., Gou, Z., Liu, B., Li, J., Lv, J., Peng, X. (2021). COMPLETER: Incomplete Multi-view Clustering via Contrastive Prediction. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11169-11178.

Research on the "Chinese+Vocational Skills" Competency Iceberg Model Design of Thai Students Majoring in Industrial Robot

Qingwei Meng
School of Foreign languages
Shanghai Technical Institute of Electronics & Information
Shanghai, China
2842885032@qq.com

Changyu Xu
School of Foreign languages
Shanghai Technical Institute of Electronics & Information
Shanghai, China
Xcy122@126.com(corresponding author)

Xiaowen Ruan
School of Business and Trade Nanjing
Vocational University of Industry Technology
Nanjing, China
ruanxw@niit.edu.cn

Abstract—This study uses the Analytic Hierarchy Process (AHP) to allocate the evaluation weights for the construction of the competency model of Thai students majoring in robotics in China. On the foundation of extensive collection of expert opinions, the Iceberg model is developed and integrated to refine 25 competency trait factors to evaluate high-quality and high-level international vocational education talents based on language literacy, vocational skill and cross-cultural competence. After testing the coefficient design of this model is reasonable, the first batch of international professional talents are basically competent.

Keywords— "Chinese+ vocational skills" competence model, Iceberg model, Analytic Hierarchy Process (AHP)

I. INTRODUCTION

This study follows the design principle of Iceberg model proposed by McClelland [1] , and concentrates on the interior traits of students' social role. The "Chinese+vocational skills" competence model should achieve a balance between professional skills and students' social role as a culture spreader in Southeast Asia. The newly-built model may accelerate the spread of traditional Chinese culture, promote the development of vocational education and the adjustment of industrial structure layout in ASEAN countries, and change the original situation where the single industry development is lacking in training of vocational education talents. Meanwhile, the Analytic Hierarchy Process (AHP) adopted in this study verifies the reliability and validity of the weight of the three-dimension index. Members of the expert group are invited to compare and score the three-level indicator system in the model. In this round, the experts adopt the "back-to-back" scoring method, and individuals evaluate the two indicators according to their own experience, knowledge level and cognition. After Southeast Asian students come to China, professional teachers, curriculum allocation and teaching resources should not only serve the career development of overseas students, but also adapt to the historical, political and economic development of Southeast Asia. Only by integrating into the local vocational education development system can it serve the "Belt and Road" Initiative while promoting cultural exchanges and mutual learning between China.

II. LITERATURE REVIEW

A. Related research

Based on the data base of Web of Science dating from 1993 shown in Figure1, the world's leading platform for scientific research and citation data, this study utilizes Citespace supporting the major sources of bibliographic data to track back the comprehensive research related to "competence model", the Analytic Hierarchy Process (AHP) is utilized by some scholars to assign data weights to the model. And, as seen from the keyword co-appearance network diagram, competence model is widely used in the domain of health care, education, business and human resource management to assess talents' professional development.

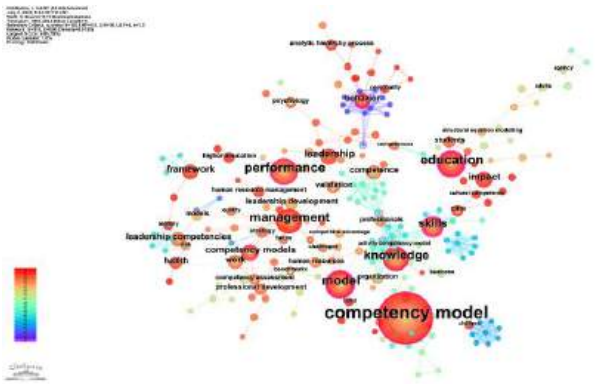


Figure1: Keyword co-appearance network diagram

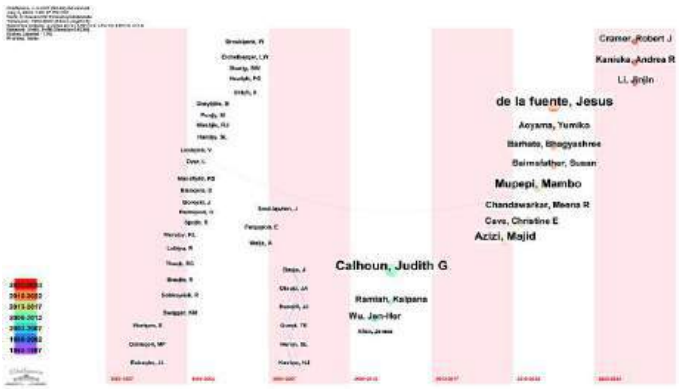


Figure2: Author time zone chart

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899650

As shown in Figure2, 1683 authors offered opinions concerning the design and application of competence model. Foreign competency model research is mainly inclusive of medical treatment, education and other fields. From the perspective of the entrepreneurship education model design, Sateesh V. Shet et al. [2] proposed a model based on an educational model for Industry 4.0 based on e-CF which defined the European e-Competence Framework (e-CF) 40 competences that were classified according to five main ICT business areas (plan, build, run, enable, manage). For the medical management personnel, Mica Rutschke [3] used a phenomenological qualitative design to investigate leadership competencies and mentoring needs of physical and occupational therapy leaders. Veeraporn Siddoo [4] utilized a computer-aided analysis to identify 14 distinct themes and 35 MTF competencies. These findings build up an innovative MTF competency model. Cramer, Robert J, Kaniuka, Andréa R et al. [5] outlined a stepwise model of suicide prevention training ranging from gatekeeper approaches to advance suicide-specific assessment and intervention skills.

At present, domestic model research mainly involves the observation and evaluation of students' ability in the field of education and teaching, and also includes the evaluation of teachers' teaching ability. From the perspective of six dimensions of competency model, four first-level indicators of talent quantity, talent quality, talent ability and talent contribution of science and technology innovation think tank and 16 subordinate second-level indicators are constructed for talent evaluation of think tank [6]. Tianshu Xu, Xiaopeng Wu et al. [7] formed a mathematical competency model composed of seven cognitive attributes. Sixty-seven released mathematical items for the fourth grade in TIMSS-2011 were used as assessment tools in this study. Xiaojing Weng [8] suggested a competency model through the analysis of the synthesized literature on VBL learning outcomes. The model provided insights into how VBL activities should be designed and assessed for PSTs' PD.

Overall, a multitude of studies at home and abroad have used the competency model as the measurement basis, which provides an intuitive assessment basis for the fields of physical therapy, management, education, electronic communication and other fields. These measurement models supply the theoretical and practical basis to this study.

B. Hypotheses Development

McClelland [1] proposed formal introduction of the concept of "competency" refers to a deep characteristic that can distinguish the good from the bad at work. He suggested that better measures of competence might be derived by analysis of successful life outcomes and the competencies involved in them, criterion sampling, and assessment of communication skills. Richard E. Boyatzis [9] further complemented McClelland's competency model and makes the core elements of competency more hierarchical.

Most studies utilized several dimensions to assess talents' competence via a scientific analytical method while the researches on the competency model reached a peak in recent years, climbing the climax in 2022 (55 publications), shown in Figure2. This study stands on the foundation of previous researches and developed the Iceberg model with full respect for the social role the Thai students. The newly-developed competence model takes students' language literacy, vocational skill and cross-cultural competency into account and verifies the reliability and validity of the weight within Competency trait factors through the Analytic Hierarchy Process (AHP). Given these theory bases, this study proposes the following research hypotheses:

1. Choosing Language literacy, vocational skill and cross-cultural competency as Level 1 dimension corresponds to the rationale of the Iceberg model.

2. The weight index distribution of 25 competency trait factors in newly-built model is valid and practical through the application of the Analytic Hierarchy Process (AHP).

3. The newly-built "Chinese+vocational skills" competency model effectively assesses international students' competence and give constructive suggestions.

III. Indicator weighting and measurement of "Chinese+vocational skills" competence iceberg model

Domestic vocational colleges can seek the corresponding state to cooperate according to the high level professionals they owned, contributing to the globalization of China's vocational education model and education standards [10]. The research model of the project is based on the results of the research and interviews of many colleges and universities in China and Thailand, combining with the talent cultivation program of many schools, together with comprehensive multi-level analysis of the elements, so we try to use the Analytic Hierarchy Process (AHP) to allocate the evaluation weights for the construction of the competency model of "Chinese +vocational skills" talents.

A. Participants

The study tracks 23 Thai students in China-Thailand Shengu-Institute from Shanghai Technical Institute of Electronics and Information, which is committed to carrying out the joint training project of overseas talents specializing in industrial robot technology. The model is designed after the opinions of experts are investigated by evaluating their grades of various courses and internships during their two-year study in China. Thailand is affected by economic and social development, language, culture and customs, and there are a large number of local speakers of Thai and English. Therefore, international students in China who master regular English expressions on the basis of Thai can also play a positive role in the teaching of vocational courses. Influenced by geographical and historical reasons, English is one of the mainstream languages in Southeast Asia and plays an important role in international exchanges and cooperation. As far as Chinese learning is concerned, Thai students have two semesters to learn industrial Chinese, which requires special learning of words and phrases in specialized fields.

B. *Model Factors Design*

McClelland, American well-known psychologist, proposed the Iceberg model in 1973, dividing talent qualities into explicit qualities such as knowledge and skills and recessive qualities such as personality traits, motivation and values. Among these explicit qualities, knowledge and skills are usually easier to understand and measure, while recessive qualities are not easy to observe and measure, but they play a pivotal role in model design. Obeying with the designing norm of the Iceberg Model developed by McClelland, this study develops and defines "Chinese+vocational skills" Iceberg competency model within competency trait elements shown in Figure3, including language literacy, vocational skill and cross-cultural competence. There are 25 competency trait factors, ranging from F1 to F25, dominated by Level3 Dimension (competency trait factors). In additional, this model highlights students' social role as "culture spreader", which is incarnated in familiarizing with the Chinese traditional culture and international cooperation project.

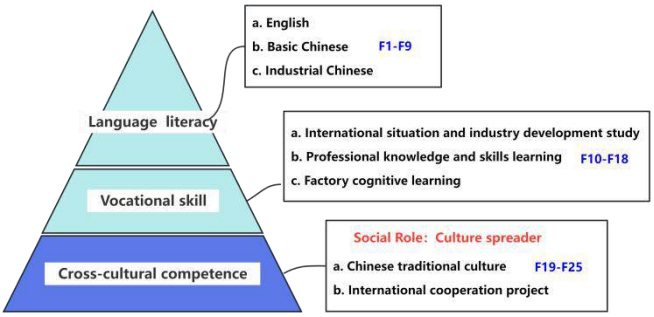


Figure3 "Chinese+ vocational skills" iceberg competency model

C. *Index Explanation*

As is shown in Table1, "Language literacy" level necessitates students to communicate smoothly in English without obvious grammatical errors, complete sentences and semantics, and use technical terms and vocabulary with the goal of tackling relevant practical problems. Meanwhile, students are expected to achieve HSK Level 3 examination level, master about 100 high-frequency vocabulary words which are identified for each post and used as a basis for combining common phrases and short sentences. What's more, students are able to use specialist nouns and verbs from relevant robotics applications and collocate gerunds into phrases, identify and draw mechanical and electrical engineering drawings, and use Chinese software to write professional documents and specialist nouns and verbs from relevant robotics applications and collocate gerunds into phrases. Meanwhile, "Vocational skill" level focuses on the current development of new industries, technologies and business models in Thailand, understand the international development situation of each node position and equip strong work site management skills and adherence to "5S" site management practices. What's more, "Professional knowledge and skills learning" mode encourages students to read and understand mechanical structure drawings, hydraulic, pneumatic, and electrical system diagrams of industrial robotic systems, maintain and repair the robot according to the robot operating instructions and use common tools and instruments for electricians and electronics, and to install and debug mechanical and electrical systems for industrial robots. In addition, students are urged to learn from Fortis Group (Thailand) Communication Technology Co, complete off-campus top-up internships at off-campus practice bases such as ABB (China) Engineering Co. and explain the installation, debugging, maintenance and repair of industrial robots and automatic production lines, system integration applications and other comprehensive knowledge of electromechanics. "Cross-cultural competence" level is divided into "Chinese traditional culture" and "International cooperation project". Firstly, students are provided courses on traditional Chinese culture, such as martial arts, shadow culture and tea ceremony, Shanghai's special intangible cultural heritage, such as Fengxian Rolling Lanterns, Nanqiao Torn Paper. Additionally, students are able to make cultural exchange and integration in Southeast Asia. Subsequently, students partake in competitions or exchange programs such as the World Skills Competition, RoboMasters Competition and Belt & Road and BRICS Skills Development and Technology innovation competition where Chinese and Thai students have an opportunity to cooperate with each other. The Belt and Road International Skills

TABLE 1: EVALUATION ANALYSIS OF THE WEIGHT INDEX OF THE "CHINESE+VOCATIONAL SKILLS" TALENT COMPETENCY MODEL

Level 1 Dimension Criterion level	Weight	Level 2 Dimension Scheme level	Weight	Level 3 Dimension Competency trait factors	Weight
Language Literacy	0.35	English	0.55	F1 Basic English Communication Skills	0.3300
				F2 Workplace English Literacy	0.3200
				F3 General Industrial English literacy	0.3500
		Basic Chinese	0.20	F4 HSK Chinese Level 3 and above	0.3990
				F5 Practical Chinese Listening and Speaking	0.2610
				F6 Practical Chinese reading and writing skills	0.3400
		Industrial Chinese	0.25	F7 Industrial Chinese communication skills	0.5880
				F8 Chinese engineering drawings, Chinese equipment manual reading ability	0.2810
				F9 Ability to use Chinese software	0.1310
Vocational skill	0.35	International situation and industry development study	0.35	F10 Career planning skills	0.1783
				F11 Professional sensitivity	0.3014
				F12 Knowledge of certain information management techniques	0.5203
		Professional knowledge and skills learning	0.35	F13 Ability to describe industrial robot applications	0.4435
				F14 Ability to understand the cutting edge of industrial robotics applications	0.3632
				F15 Ability to clearly identify vocational skills and professionalism in industrial robotics	0.1933
		Factory cognitive learning	0.30	F16 Organizational and coordination skills	0.3881
				F17 Critical incident management capacity	0.4824
				F18 Creative entrepreneurship in a cross-cultural context	0.1295
Cross-cultural competence	0.3	Chinese traditional culture	0.20	F19 Chinese cultural export capacity	0.4217
				F20 Resilience to the "clash of cultures"	0.2487
				F21 Capacity development in multicultural contexts	0.3296
		International cooperation project	0.80	F22 Creative and co-operative awareness in an intercultural context	0.1593
				F23 Awareness of teamwork in a cross-cultural context	
				F24 Other intercultural applied competence practices	
				F25 Awareness of international laws and regulations in a cross-cultural context	

Competition is also promoted. Last, they possess the capability of solving the language and cultural barriers, achieving a seamless transition from domestic to overseas markets and complying to laws and regulations under various international trade agreements.

D. Weight Index Design

This study is on the basis of the preliminary expert research, while experts uniformly identified the first-level indicators of the weight of each weight using qualitative values with language literacy, vocational skill, cross-cultural competence weight of 0.35, 0.35 and 0.3 respectively. Through questionnaires and various interviews, experts set the weights of multiple secondary dimensions such as the proportion of cognitive learning in factories, as shown in Table 1. Therefore, in this study, the weight setting of the third-level dimensions is highlighted, i.e. programme level indicators, which have been proved by experts.

(a) A judgement matrix is constructed by comparing each element two by two based on expert opinion and assigning values 1,3,5,7,9 respectively. The questionnaire is scored on a 1-5 scale, with larger values indicating greater indicator importance. By establishing the judgement matrix A, this study uses the sum method of determining the relative weight vectors to normalize the weight values to obtain the matrix A_{std} and averages the normalized matrix by rows to obtain the weights ω.

(b) This study calculates the maximum characteristic root λ_{max} by using the software (Equation 1).

(c) The consistency CR test is performed on the judgement matrix using Equation 2 and Equation 3, and only when $CR < 0.1$, the consistency test is regarded as passed, and the inconsistency is within the acceptable range, and the calculated weight vector is the index weight. RI denotes the stochastic consistency index, which can be obtained by consulting the reference cross-reference table of consistency indexes.

(d) The overall ranking and one-time test CR_k for layer-by-layer indicators using Equation 4, and the consistency test of the layer passes when $CR_k < 0.1$.

(e) The final composite score can be obtained by weighting the weighting matrix with the planned data.

① $\lambda_{max} = \frac{1}{n}\sum_{i=1}^{n}\frac{(A\omega)_i}{\omega_i}$ Formula1 (the maximum characteristic root of the judgment matrix)

② $CR = \frac{CI}{RI}$ Formula2 (consistency test)

③ $CI = \frac{\lambda_{max} - n}{n-1}$ …… Formula3 (Judging the consistency index of the matrix, the greater the CI value, the more serious the inconsistency)

④ $CR^{(k)} = CR^{(k-1)} + \frac{CI^{(k)}}{RI^{(k)}}$, k≥3 Formula 4 (Consistency test of order "k" relative to the total target)

IV. DISCUSSION

A. Identification of the Panel of Experts

In this study, 10 experts and scholars were invited to participate in the research, questionnaires, and the scoring of the measured at a later stage. The expert survey respondents mainly consisted of experts engaged in digital application research, vocational education teaching managers, and representatives of first-line teachers of secondary colleges, to whom consultation questionnaires, indicator weighting questionnaires, and empirical evaluation forms of the tested indicators were distributed in batches according to the progress of the project. The recovery rate and validity rate of the questionnaires were 100 per cent. At the same time, in order to ensure the degree of expert authority, the expert authority coefficient (Cr) was set up, and the direct authority was composed of the expert's familiarity with the problem (Cs) and judgement of the problem (Ca), and the degree of familiarity, Cs, was assigned a value of 0.2-1, with a total of five grades, ranging from "very unfamiliar" to "very familiar". The basis of judgement was based on "very familiar". Judging basis (Ca) values 0.2-0.8 within 4 levels, including "intuition, peer understanding, theoretical analysis, practical experience". Cr = (Cs + Ca)/2, generally believe that Cr ⩾ 0.7, that is, the results of the study is considered reliable. Cs adopts 5 levels of quantification, Ca adopts 4 levels of quantification, after calculating the average value of Cr of the test results of this expert consultation is 0.712, and it is concluded that the consultation results are reliable and can be adopted.

B. Construction of judgement matrix and indicator scores

Members of the expert group were invited to compare and score the three-level indicator system in the model. In this round, the experts adopted the "back-to-back" scoring method, in which individuals judged the two indicators based on their own experience, knowledge and perception. Diverse representatives from schools, industries, enterprises and other organizations focused on different aspects of the evaluation, e.g. Industry experts concentrated more on the long-term planning and development of digital technology, and university experts focused more on the evaluation of classroom teaching and implementation. By collecting the scores from 10 experts, and taking the evaluation of the three level 3 indicators under "Industrial Chinese" as an example, we obtained the judgement matrix and the process of weights and eigenvectors as follows:

$$C = \begin{bmatrix} 1.000 & 2.633 & 3.578 \\ 0.380 & 1.000 & 2.712 \\ 0.279 & 0.369 & 1.000 \end{bmatrix} \rightarrow \omega_i \begin{bmatrix} 0.588 \\ 0.281 \\ 0.131 \end{bmatrix} \rightarrow A\omega_i \begin{bmatrix} 1.796 \\ 0.858 \\ 0.399 \end{bmatrix}$$

As is shown in Table2, this study takes the three-dimension index module judgment of "Industrial Chinese" as an example, 10 qualified expert group members are invited to score the weight indicators in the module with the 5-level

evaluation method, and finally take the mean value for matrix calculation. The normalized matrix is obtained after normalization processing, and the final weight vector ω_i is obtained after normalization processing according to the geometric average value of the normalized matrix, and then the feature vector A ω_i is obtained after ω_i, and the maximum feature root λ_{max} can be obtained (realized by using $\lambda_{max}=\frac{1}{n}\sum_{i=1}^{n}\frac{(A_w)_i}{w_i}$ Matlab software).

TABLE2 JUDGMENT MATRIX AND WEIGHT OF THE THREE-DIMENSION INDEX OF "INDUSTRIAL CHINESE"

Industrial Chinese Level 3 Dimension Competency trait factors	Industrial Chinese communication skills	Chinese engineering drawings, Chinese equipment manual reading ability	Ability to use Chinese software	ωi
F7 Industrial Chinese communication skills	1.000	2.633	3.578	0.588
F8 Chinese engineering drawings, Chinese equipment manual reading ability	0.380	1.000	2.712	0.281
F9 Ability to use Chinese software	0.279	0.369	1.000	0.131

C. Computing Eigenvectors, Eigenroots and One-time Test

Weight vector ω_i= (0.588, 0.281, 0.131), according to Equation 1 to get λ_{max} = 3.05, CI = 0.027, RI = 0.52 (find the RI random consistency index), that is, consistency test CR = 0.051 < 0.1 (Formula 2), the judgement matrix passes the consistency test. Therefore, F5 weight is 0.588, F6 weight is 0.281, F7 weight is 0.131, and the weight evaluation results are valid. This to verify hypothesis1 that the weight index distribution of 25 competency trait factors in newly-built model is valid and practical through the application of the Analytic Hierarchy Process (AHP).

Due to the complexity of the system, the diversity of cognition, the subjective one-sidedness and the instability of the evaluator, in order to ensure the effectiveness of hierarchical ranking, the consistency test of the given judgment matrix must be carried out. Generally, the consistency test usually uses the consistency ratio CR as the test standard (Formula 4). When CR<0.1, it is considered that the consistency of the judgment matrix is acceptable. When CR>=0.1, the judgment matrix should be adjusted, and then the weight vector should be recalculated and the consistency test should be carried out until the test is passed. CI=$\frac{\lambda_{max}-n}{n-1}$,where CI is a consistency indicator. The larger the CI, the more serious the inconsistency. According to the order of the matrix, it can be obtained by looking up the following table that the consistency index RI is related to the order "k" of the judgment matrix, and CR is finally calculated shown in Table3.

$$CR^{(k)} = CR^{(k-1)} + \frac{CI^{(k)}}{RI^{(k)}}$$

TABLE3 CONSISTENCY INDICATOR REFERENCE VALUE

K(order)	1	2	3	4	5	6	7
RI	0	0	0.58	0.89	1.12	1.26	1.32
K(order)	8	9	10	11	.....		
RI	1.41	1.45	1.49	1.51	.....		

D. Hierarchical Total Ranking and One-time Test

After testing and adjusting, the CR_k is less than 0.1, and the weighting indicators at all levels are available, so that we can get the updated three-level weighting version of the "Chinese+vocational skills" Competency model.

E. Empirical Analysis of Model Evaluation

Nine Thai students were randomly selected to take the test. To make the evaluation presentation intuitive, the module was scored on a five-point scale, with 1 being the lowest, 3 being the mean, and 5 being the highest. A virtual student 0 with all scores at the mean was used as a comparison to participate in all subjects, and the rest of the sampled 9 students participated in the test, for a total of 10 subjects, and the test results were lower than the baseline test for 2 students and higher than the baseline test results for the combined results of 7 students.

V. A CASE OF "CHINESE + VOCATIONAL SKILLS" COMPETENCY MODEL ASSESSMENT

Completion of the competency elements is reflected through the assessment of the relevant courses. Under each competency element, different courses are opened to meet the needs of more professional training, and the course assessment adopts the percentage system, the average score of the corresponding course of each competency element is taken, multiplied by the weight value of the competency element, and the final score of the subject of the second-level indicator is obtained. The study takes the second-level dimension module of industrial Chinese of a student A as an example, as is seen in Table4.

TABLE4 INDUSTRIAL CHINESE MODULE COMPETENCY TRAIT FACTOR ASSESSMENT MODULE (TAKING STUDENT A AS AN EXAMPLE)

Level 2 Dimension Scheme level	Level 2 Dimension Competency trait factors	Curriculum	Course remark	Weight	Assessment score	Final score
Industrial Chinese	F7 Industrial Chinese communication skills	Project Overview	100	0.588	78	45.864
	F8 Chinese engineering drawings, Chinese equipment manual reading ability	Chinese Engineering Drawing Recognition	100	0.281	95	26.695
	F9 Ability to use Chinese software	Mechanical Drawing (Chinese Version)	100	0.131	90	11.790
Grade						84.35
Completion rate						84.35%

VI. CONCLUSION AND IMPLICATION

According to the survey, the development of various schools and Chinese education in Thailand, Cambodia, Singapore and other countries is fierce, and the scale and level of Chinese education are constantly expanding and improving, which helps Southeast Asian students to improve their language foundation for studying in China. By the end of 2023, more than 170 vocational schools in Thailand have started Chinese language and culture courses, and there are 844 kinds of Chinese teaching materials in Thailand. A large multitude of high-quality teaching materials such as "Applied Chinese", "Chinese Communication" and "Basic Chinese" have been published.

In the year of 2019, 134 universities in Thailand have offered Chinese language and culture courses, and eight Chinese universities in Thailand use Chinese teaching materials, including Oriental University, Royal Chiang Mai University, Ramkang Heng University, Khon Kaen University, and so on. More than 170 vocational schools offer Chinese courses. With the implementation of the "Belt and Road"education initiative, international students in China can learn Chinese through HSK Academy, SuperChinese and other Chinese learning apps. Overcome the language level early. In 2020, the "Chinese+Vocational Skills" industrial Chinese textbook series was published, aiming to solve the problem of training local technical and technical talents of Chinese enterprises "going out" under the "Belt and Road" initiative, better support the local development of railway, power, aviation and other fields related industries, and the "Chinese+vocational" model usher in a turning point of Chinese vocational education characteristics.After Southeast Asian students came to China, professional teachers, curriculum allocation and teaching resources should not only serve the career development of overseas students, but also adapt to the historical, political and economic development of Southeast Asia. Only by integrating into the local vocational education development system can it serve the "Belt and Road" Initiative while promoting cultural exchanges and mutual learning between China. This study mainly focuses on the competency training of Thai international students. The language environment of Southeast Asian countries is complex and the requirements for language literacy are quite different, so there is still room for improvement and supplement of this model for different countries. At the same time, the model refers to tracking the study data of Thai students in China for two years, and insufficient monitoring of their study process data in their home country. According to the current data, Thai students participated in fewer vocational skills competitions and did not achieve outstanding results. Vocational colleges should strengthen cooperation between Chinese and Thai students, break down language barriers, and encourage in-depth exchanges and cooperation between students of the two countries.

ACKNOWLEDGMENT

This work was supported by International Chinese Language Education Research Program (Grant No.23YH30D), China and Association of Fundamental Computing Education in Chinese Universities Program (2023-AFCEC-365).

REFERENCES

[1] McClelland, D. C. (1973). Testing for competence rather than for "intelligence". American Psychologist, 28(1), 1–14. https://doi.org/10.1037/h0034092

[2] Sateesh V. Shet, Vijay Pereira,Proposed managerial competencies for Industry 4.0–Implications for social sustainability,Technological Forecasting and Social Change,Volume 173,2021,121080,ISSN 0040-1625,https://doi.org/10.1016/j.techfore.2021.121080.

[3] Rutschke, Mica & Fick, John. (2023). Exploring Leadership Competencies and Mentoring Needs of Physical and Occupational Therapy Leaders in the United States. Journal of Health and Allied Sciences NU. 14. 10.1055/s-0043-1764356.

[4] Veeraporn Siddoo, Worawit Janchai, Orawit Thinnukool,Understanding the multidimensional role of medical travel facilitators: A study on competencies and a proposed model,Heliyon,Volume 10, Issue 9,2024,e30479,ISSN 2405-8440,https://doi.org/10.1016/j.heliyon.2024.e30479.

[5] Cramer, R. J., Hawgood, J., Kaniuka, A. R., Brooks, B., & Baker, J. C. (2023). Updated suicide prevention core competencies for mental health professionals: Implications for training, research, and practice. Clinical Psychology: Science and Practice. Advance online publication. https://doi.org/10.1037/cps0000172

[6] Qi, Shaobo. (2024). Evaluation index system of science and technology innovation think tank talents based on competency model. Journal of Computational Methods in Sciences and Engineering. 24. 1101-1117. 10.3233/JCM-247315.

[7] Xu, Tianshu & Wu, Xiaopeng & Sun, Siyu & Kong, Qiping. (2023). Cognitive Diagnostic Analysis of Students' Mathematical Competency Based on DINA Model. Psychology in the Schools. 60. 10.1002/pits.22916.

[8] Xiaojing Weng, Oi-Lam Ng, Thomas K.F. Chiu,Competency development of pre-service teachers during video-based learning: A systematic literature review and meta-analysis,Computers & Education,Volume199,2023,104790,ISSN0360-1315,https://doi.org/10.1016/j.compedu.2023.104790.

[9] Boyatzis, Richard E.. "The Competent Manager: A Model for Effective Performance." (1982).

[10] Du, Zenghui. (2020). Based on research on the demand for "Chinese + vocational skills" project by Confucius institutes in Asia and Africa, discussing on Chinese professional standards going global. Lifelong Education. 9. 17. 10.18282/le.v9i5.1194.

Strong Co-location Pattern Mining Incorporating Multi-path and Distance Decay Effects

Yonggui He
School of Information Science and Engineering, Yunnan University
Kunming, China
iqiliaohe@163.com

Peizhong Yang✉
School of Information Science and Engineering, Yunnan University
Yunnan Key Laboratory of Intelligent Systems and Computing
Kunming, China
ypz@ynu.edu.cn
(Corresponding author)

Lizhen Wang
School of Information Science and Engineering, Yunnan University
Yunnan Key Laboratory of Intelligent Systems and Computing
Kunming, China
lzhwang@ynu.edu.cn

Hongmei Chen
School of Information Science and Engineering, Yunnan University
Yunnan Key Laboratory of Intelligent Systems and Computing
Kunming, China
hmchen@ynu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899493

***Abstract*—Spatial co-location pattern mining (SCPM) is a sub-field of data mining, which aims to discover the subset of spatial features whose instances are frequently located in proximate areas. SCPM has broad prospects in many applications, such as ecology, public health, smart cities, etc. In recent years, improving and applying SCPM technology with the constraints of road networks has emerged as a prominent research focus. However, existing studies solely focus on the shortest path between instances when assessing the proximity relationships, neglecting other proximity paths, and thus may overlook co-location patterns with strong associations. To address this issue, this paper introduces a novel metric called Strong Proximity Score, which integrates both multi-path proximity and distance decay effects to measure the strength of proximity relationships. Additionally, the Minimum First Search (MFS) algorithm is presented, which utilizes the strategy of minimum instance pair search to accelerate the calculation of Strong Proximity Score. Extensive experiments conducted on real datasets of points of interest demonstrate the superiority of the method based on Strong Proximity Score over traditional SCPM methods and confirm the efficiency of MFS algorithm.**

Keywords—co-location pattern, road network, multi-path, distance decay

I. INTRODUCTION

With the widespread use of spatial databases and GPS devices, vast and continuously growing volumes of spatial data are being generated. Extracting valuable insights from spatial data has become a major focus of research. Spatial co-location pattern mining (SCPM) is a key area of knowledge discovery in spatial data, often used to identify associations between spatial features. The goal of SCPM is to find subsets of spatial features whose instances frequently occur in close geographic proximity, known as co-location patterns [1][2]. For instance, {Hospital, Flower Store} is a co-location pattern, since hospitals and flower stores are often found near each other in the city. Co-location patterns provide significant insights across diverse fields, such as ecological protection [3] and public safety [4].

SCPM was initially proposed by Shekhar and Huang [1], who introduced a general approach for SCPM in their subsequent work [2]. They used Euclidean distance to assess the proximity between instances and defined a metric called the participation index to measure the association strength of features in a co-location pattern. After that, many researchers have improved SCPM from various research perspectives. Among these, SCPM tailored for urban spaces is a noteworthy example. Road networks constrain the movement of people and goods in urban areas. If two instances are not connected by a road, people or goods cannot effectively move between them, even if they are geographically close via Euclidean distance. Considering this, Yu [5] first adopted network distance over Euclidean distance to assess proximity between instances in urban areas and proposed a novel framework of SCPM in urban areas. Then, some studies further developed SCPM in urban areas by using road network distance as the basic component [6][7]. However, when measuring the strength of proximity relationships, existing studies only consider the shortest path between instances, neglecting other proximity paths whose distance no large than the user-defined threshold. As a result, they may miss some co-location patterns with strong associations.

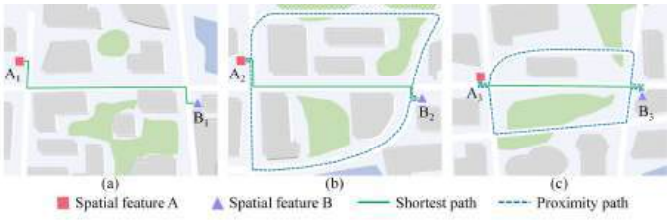


Fig. 1. The example of different proximity relationships.

Considering Fig.1, instances A_1 and B_1 maintain proximity by a single path, while instances A_2 and B_2 maintain proximity through three paths. Since the length of shortest path is the same in Fig.1(a) and Fig.1(b), existing methods regard their proximity relationships as identical. However, the proximity relationship in Fig.1(b) is more robust. If the shortest path is disrupted due to traffic accidents, natural disasters, or other causes, instances A_2 and B_2 in Fig.1(b) can still maintain proximity via other proximity paths, while the instances in Fig.1(a) would be isolated and no longer proximate. By focusing solely on the shortest path, existing methods fail to capture the differences in

robustness between two types of proximity relationships. Furthermore, when applying the first law of geography (the closer two instances are, the stronger their association), existing methods overlook paths other than the shortest one. This limitation prevents them from properly assessing the strength of proximity relationships when instances maintain proximity via multiple paths. Considering Fig.1(b) and Fig.1(c), although the shortest path length is the same, other proximity paths in Fig.1(c) are shorter than those in Fig.1(b), i.e., instances in Fig.1(c) have stronger proximity relationships. Instances connected frequently by shorter and more paths have stronger proximity relationships, and the corresponding features of these instances possess stronger associations with one another, such features can form the strong co-location patterns. Strong co-location patterns are more valuable in various application fields, but existing SCPM methods are ineffective at mining such patterns in urban areas.

To address above issue, we propose a framework to effectively discover strong co-location patterns in urban areas. Specifically, we utilize all paths that meet the distance threshold limitation to model proximity relationships between instances within the constraints of urban road networks. Then, we design a novel prevalence measure called the Strong Proximity Score (SPS). SPS considers all proximity paths between instances, incorporating both the number of paths and the effects of distance decay. Furthermore, we analyze the calculation of SPS and introduce a computational acceleration method based on minimum instance pairs, called Minimum First Search (MFS). MFS can avoid the operation of fully generating row instances for candidate patterns. Lastly, extensive experiments are conducted on the point of interest (POI) datasets, and the results demonstrate that SPS can capture the differences in strength between proximity relationships and appropriately measure the association strength of co-location patterns. The proposed method is effective in identifying strong co-location patterns overlooked by existing methods. Additionally, MFS is several times more efficient than its competitors on real datasets.

The principal contributions of this paper are as follows.

- A novel measurement named Strong Proximity Score is proposed, which incorporates the effects of both multi-path proximity and distance decay in evaluating the strength of proximity relationships between instances.
- The algorithm Minimum First Search is designed to avoid the time-consuming operation of fully generating row instances for candidate patterns and greatly improve the mining efficiency of strong co-location patterns.
- Extensive experiments are conducted on POI datasets to demonstrate the effectiveness of Strong Proximity Score and the efficiency of Minimum First Search.

II. RELATED WORK

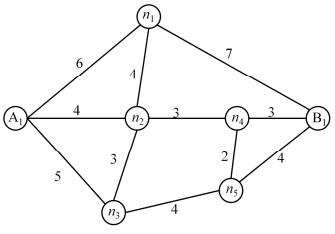
Shekhar and Huang [1][2] introduced the participation index (PI) to quantify the association strength of features within a co-location pattern and suggested a general join-based SCPM algorithm. This algorithm generates row instances of co-location patterns (a set of instances neighboring each other and covering all features in a co-location pattern only once) for the calculation of PI through the instance join operation, but the instance join operation is time-consuming. Subsequently, many studies have been devoted to enhancing the efficiency of SCPM. Yoo et al. [8] introduced a join-less method by utilizing star neighborhoods to represent spatial proximity relationships, leveraging instance lookup with clique verification to identify all row instances of co-location patterns. Wang et al. [9] proposed the iCPI-tree structure to represent these spatial proximities, enabling faster generation of row instances for co-location patterns. Yang et al. [10] developed a column-based approach, which directly identifies participating instances of features without creating all row instances, significantly enhancing the efficiency of SCPM. In addition to improving efficiency, many researchers have also worked on extending the participation index to address the challenges brought by unique characteristics of spatial data and achieve targeted mining goals. Chan et al. [11] introduced Fraction-Score, an interest measure for co-location patterns, to address the prevalence overestimation caused by overlapping instances. Yang et al. [12] proposed a mixed prevalence index which considers both feature-level and instance-level heterogeneity, covering the variations in feature instance counts and distribution. Yao et al. [13] integrated distance decay effects into SCPM in Euclidean space using a kernel-density model. Yu et al [5] proposed an SCPM framework with road network constraints, modeling urban areas as a graph and using network distance rather than Euclidean distance to identify proximity between instances. Moreover, they incorporated distance decay effects into the prevalence measure of co-location patterns, introducing new interest metrics [6]. Yao et al. [7] expanded this framework by considering traffic direction, improving the accuracy of SCPM in urban settings.

III. METHODOLOGY

In this section, we introduce the key technology of Strong Proximity Score and Minimum First Search algorithm.

A. Strong Proximity Score and strong co-location patterns

We adopt the spatial model proposed in [5], which abstracts urban areas as a weighted graph by relating spatial instances to the road network. Spatial instances and road intersections are represented as vertices, while road segments serve as edges, with their lengths as the weight of edges. The right part of Fig.2 illustrates an example of such a weighted graph, where vertices A_1 and B_1 are instances and other vertices are road intersections.



path	length
p_1={(A_1, n_2), (n_2, n_4), (n_4, B_1)}	10
p_2={(A_1, n_1), (n_1, B_1)}	13
p_3={(A_1, n_3), (n_3, n_4), (n_4, n_5), (n_5, B_1)}	13
p_4={(A_1, n_3), (n_3, n_5), (n_5, B_1)}	13

Fig. 2. A weighted graph and the proximity paths in which the distance from instance A_1 to B_1 is less than 14.

Once urban space is modeled as a weighted graph, the proximity relationships between instances can be modeled. We model the proximity relationship between instances using all proximity paths with the distance no larger than a threshold, e.g., p_1, p_2, p_3 and p_4 for A_1 and B_1 in Fig.2. Based on the multi-path

proximity, the strength of proximity relationships between instances is defined.

Definition 1. (Strength of proximity relationships between instances) Given two instances o_i and o_j, the strength of proximity relationships between them is defined as:

$$SD(o_i, o_j) = \sum_{p_i \in PS(o_i, o_j)} Core(p_i) \quad (1)$$

Where $PS(o_i, o_j)$ is the set of proximity paths from o_i to o_j, The $Core(\cdot)$ function is a kernel function to model the distance decay effect for a single path [13], defined as $Core(p_i)=\exp\left(-\frac{l(p_i)^2}{(2*R)^2}\right)$, where $l(p_i)$ is the length of path p_i, R is the distance threshold.

Definition 1 uses the kernel function to quantify the distance decay effect for each proximity path between instances, and the values are accumulated. Therefore, the more paths that maintain the proximity of two instances and the shorter these paths are, the stronger the proximity relationship between them, resulting in a larger value of $SD(o_i, o_j)$.

Example 1. In Fig.2, given the distance threshold R=13, then we have $PS(A_1, B_1)$ is $\{p_1, p_2, p_3, p_4\}$, and $SD(A_1, B_1)$=3.2.

Searching for all proximity paths between two instances in a weighted graph is an active research problem, with various applicable methods [14], thus, given a weighted graph G abstracted from urban areas, proximity paths between each pair of instances can be identified easily. Then, by calculating the strength of proximity relationships for each pair based on Definition 1, we can obtain a proximity relationships for all instances. The right of Fig.3 shows an example of proximity relationships between instances, and the red values on the edges indicate the strength of proximity relationships between the connected instances, e.g., $SD(A_1, B_1)$=3.2. In the SCPM filed, a co-location pattern C is a subset of spatial features, and the table instance of C consists of all row instances of C. A participating instance of feature f in C is an instance that has feature f and is included in any row instance of C. These concepts will be carried forward in the subsequent. Next, we define the weight of an instance within a co-location pattern.

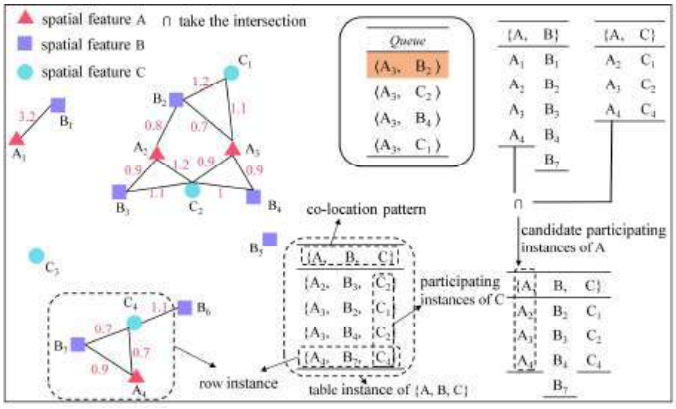


Fig. 3. Illustration of proximity relations and related concepts.

Definition 2. (The weight of an instance within a co-location pattern) Given an instance o, a co-location pattern C and its table instance $TI(C)$, we have the weight of o within C is:

$$OCP(o, C) = \min_{RI_i \in TI(C), o \in RI_i} \{ORI(o, RI_i)\} \quad (2)$$

Where RI is a row instance of C, $ORI(o, RI)$ represents the weight of o in a row instance and $ORI(o, RI) = \min_{o_i \in RI, o_i \neq o} \{SD(o, o_i)\}$. For each RI in $TI(C)$ that includes o, Equation 2 calculates the weight of o in RI and then takes the minimum value of $ORI(o, RI)$ as the result of $OCP(o, C)$. This process sequentially calculates the proximity strength between o and other instances in RI, ultimately using the minimum value for $ORI(o, RI)$. By applying two minimum functions, Definition 2 rigorously assesses the weight of an instance within a pattern, i.e., the value of $OCP(o, C)$ is large only if o maintains a strong proximity relationship with each instance in per row instance.

Example 2. In Fig.3, considering the pattern {A, B, C} and instance A_3, the table instance of {A, B, C} has marked with the dotted line, and the row instances that include A_3 are $\{A_3, B_2, C_1\}$ and $\{A_3, B_4, C_2\}$. Because $OCP(A_3, \{A, B, C\}) = \min\{ORI(A_3, \{A_3, B_2, C_1\}), ORI(A_3, \{A_3, B_4, C_2\})\)=0.7$, so the weight of A_3 within {A, B, C} is 0.7.

Based on Definitions 1 and 2, we define the Strong Proximity Score.

Definition 3. (Strong Proximity Score) Given a co-location pattern C, the Strong Proximity Score of C is defined as follows.

$$SPS(C) = \min_{f_i \in C} \{FCP(f_i, C)\} \quad (3)$$

Where f is a spatial feature, $FCP(f, C)$ is the weight of f in C, defined as:

$$FCP(f, C) = \frac{\sum_{o_i \in Spi(f, C)} OCP(o_i, C)}{|N(f)|} \quad (4)$$

Where $Spi(f, C)$ is the set of participating instances of f in C and $N(f)$ is the set of all instances with feature f. The more instances of feature f participating in C, and the stronger the proximity relationships between instances (according to Definition 2), the greater the value of $FCP(f, C)$. Then, SPS calculates the weight of each feature within the pattern and uses the minimum value as the metric to reflect the strength of association among features in the co-location pattern.

Example 3. In Fig.3, for the pattern {A, B, C}, we have the participating instances of feature A are A_2, A_3 and A_4, and it is easy to know $FCP(A, \{A, B, C\})$=0.575. Similarly, $FCP(B, \{A, B, C\})$=0.457, $FCP(C, \{A, B, C\})$=0.675, and thus $SPS(\{A, B, C\})$=0.475.

Strong Spatial Co-location Pattern Mining Problem (SSCPM)*:* Given a set of spatial instances O, a set of spatial features F, a road network RN, a distance threshold R, and a Strong Proximity Score threshold *min-SPS*, a strong spatial co-location pattern (SSCP) is a pattern with a Strong Proximity Score no less than *min-SPS*, and strong spatial co-location pattern mining task aims to identify all SSCP from the given dataset. Next, an efficient algorithm is introduced for SSCPM.

B. An efficient algorithm for SSCPM

Algorithm 1 presents the basic approach for solving the SSCPM problem. Since SPS possesses anti-monotonicity, we employ an Apriori-like mining framework. This framework

generates size-k candidate patterns only from size-(k−1) strong co-location patterns, effectively narrowing the search space.

Algorithm 1: the general framework for SSCPM.

Input: spatial instances set O; spatial features set F; road network RN; distance threshold R; Strong Proximity Score threshold *min-SPS*.

Output: a set of strong co-location patterns.

Variables: G: weight graph; *PRS*: the proximity relationships for instance pairs; k: size of co-location patterns; C_k: set of size-k candidates; S_k: set of size-k strong co-location patterns.

1. G=Abstract(O, RN)
2. *PRS*=GetProximityRelationship(O, G, R)
3. initialize S_1=F, k=2
4. **While** S_{k-1} is not empty **do**
5. C_k=Apriori-gen(S_{k-1})
6. Initialize S_k=Ø
7. **For each** C in C_k
8. generate table instance *TI*(C)
9. calculate *SPS*(C)
10. **If** *SPS*(C)≥*min-SPS* **then**
11. S_k=S_k∪{C}
12. k=k+1
13. **Return** union (S_2,… ,S_{k-1})

Algorithm 1 first abstracts the study area as a weighted graph in Step 1. Step 2 then identifies proximity paths between instances within this graph, calculates the strength of proximity relationships between them via Definition 1 and stores the results in *PRS*. In Step 3, each feature is initialized as a size-1 strong co-location pattern, serving as the start of iteration. Steps 4-12 continue by searching for all size-k ($k \geq 2$) patterns in ascending order of size. In Step 5, size-k candidates are generated with size-(k-1) strong co-location patterns, and then each candidate is checked in Steps 7-12. Step 8 generates the table instance for a candidate C, and Step 9 calculates the Strong Proximity Score of C. If *SPS*(C) is no less than *min-SPS*, Step 12 stores C into the result set. Even though Algorithm 1 prunes a significant number of candidate patterns, its efficiency remains constrained due to the well-known time-consuming operation of generating table instances for each remaining candidate (Step 8). However, this operation is unnecessary for computing SPS.

When calculating SPS, row instances are used to calculate the weight of an instance within a pattern (Definition 2). Considering the calculation of *OCP*(A_3, {A, B, C}) in Fig.3, because there are two row instances included A_3, we have *OCP*(A_3, {A, B, C})=min{*ORI*(A_3, {A_3, B_2, C_1}), *ORI*(A_3, {A_3, B_4, C_2})}, which is mathematically equivalent to min{*SD*(A_3, B_2), *SD*(A_3, C_1), *SD*(A_3, B_4), *SD*(A_3, C_2)}, since the instance pair ⟨A_3, B_2⟩ has the minimum strength of proximity relationships, i.e., *SD*(A_3, B_2)=0.7, so *OCP*(A_3, {A, B, C})=0.7. From the above procedure, we can find that instance pairs are needed instead of row instances when computing *OCP*(•). Thus, the operation of generating row instances can be replaced by identifying instance pairs. Based on this, we calculate *OCP*(o, C) by first using o and the participating instances of other features in C to form instance pairs, then verifying whether these instance pairs are participating instance pair of C, at last obtaining *OCP*(o, C) by selecting the participating instance pair with the minimum strength of proximity relationships. Since the participating instances of feature are unknown in advance, we introduce the set of candidate participating instances. The set of candidate participating instances of feature f in a size-k (where k > 2) co-location pattern C is the intersection of all participating instance sets of f in each size-(k-1) sub-pattern of C that contains f, denoted as *CSpi*(f, C), e.g., *CSpi*(A, {A, B, C}) is {A_2, A_3, A_4}, as shown in the right side of Fig.3. Using instances in such set to form instance pairs, we can obtain *OCP*(o, C) with the above procedure.

For the calculation of *OCP*(A_3, {A, B, C}), the result is the minimum strength of proximity relationships (strength for short) among all instance pairs. Therefore, we can check instance pairs in ascending order of their strength. If the participating instance pair with the minimum strength is identified, we can immediately stop the check of the remaining instance pair, thus speeding up the calculation of *OCP*.

Example 4. Calculate *OCP*(A_3, {A, B, C}) using the above method. Considering Fig.3, we first obtain the set of candidate participating instances for each feature in {A, B, C}. Then, we form instance pairs with A_3 and these candidate participating instances and store such pairs into the minimum priority queue *Queue*. Note that we do not form an instance pair if the candidate participating instance is not proximate to A_3 (e.g., B_3) since it cannot be a participating instance pair, i.e., being included in a row instance of {A, B, C}. Next, we sort all instance pairs in *Queue* in ascending order by their strength and retrieve the first pair to check, i.e., ⟨A_3, B_2⟩. Since ⟨A_3, B_2⟩ is a participating instance pair, *OCP*(A_3, {A, B, C})=*SD*(⟨A_3, B_2⟩), then computation is done. We call this Algorithm *Minimum First Search* (MFS) since it preferentially checks the instance pair with minimum strength (minimum instance pair for short). Algorithm 2 shows the details of MFS.

Algorithm 2: Minimum First Search

Input: a candidate C

Output: *SPS*(C)

Variables: f: a feature; o: an instance; Q: a minimum priority queue;

1. **For each** f in C **do**
2. Generate *CSpi*(f, C)
3. **For each** f_i in C do
4. **For each** o in *CSpi*(f_i, C) **do**
5. Initialize Q=Ø
6. **For each** o_j in other *CSpi*(f, C) **do**
7. **If** o_j is in proximity to o **do**
8. Q.put(⟨o, o_j⟩)
9. **While** Q is not empty **do**
10. ⟨o, o_j⟩=Q.pop()
11. **If** *isParticipatingInstancePair*(⟨o, o_j⟩) **do**
12. *OCP*(o, C)=*SD*(o, o_j)
13. **break**
14. Calculate *FCP*(f_i, C)
15. Calculate *SPS*(C)
16. Return *SPS*(C)

In Steps 1-2, the set of candidate participating instances for each feature f in C is generated. Steps 3-13 then sequentially calculate the weight of f_i in C. For each candidate participating instance o of f_i, a minimum priority queue Q is initialized with empty (Step 5), then each instance o_j in the set of candidate participating instances of other features is checked in Step 7. If o_j is in proximity to o, we form an instance pair ⟨o, o_j⟩ and store it in Q (Step 8). Continue, we iteratively fetch the current

minimum instance pair $\langle o, o_j \rangle$ from Q and then use the function *isParticipatingInstancePair*(•) to verify if $\langle o, o_j \rangle$ is a participating instance pair (Steps 9 and 11). If true, the value of $OCP(o, C)$ is set to $SD(o, o_j)$ and stop iteration (Steps 12 and 13); otherwise, we continue to check the next minimum instance pair until the Q is empty. Based on $OCP(o, C)$, we calculate $FCP(f_i, C)$ and $SPS(C)$, and finally, Step 16 returns the Strong Proximity Score for C. By replacing Steps 8 and 9 of Algorithm 1 with MFS, we achieve an efficient SSCPM method.

IV. Experiment

In this section, we conduct experiments on real datasets to evaluate the effectiveness of the Strong Proximity Score (SPS) and the efficiency of Minimum First Search (MFS).

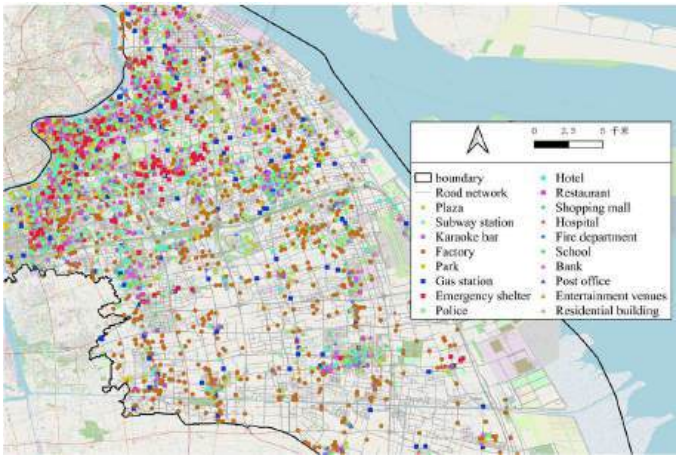


Fig. 4. POI data distribution diagram of Pudong.

We conduct experiments on POI data from Pudong, Shanghai. This dataset contains 7,329 instances and 18 features, with their spatial distribution illustrated in Fig.4. For comparison, we introduce two comparison methods: CMN [5] and NDB [6]. CMN utilizes the weighted PI measurement that incorporates distance decay effects, while NDB applies traditional PI measurement. Both approaches are tailored for SCPM in urban settings and employ the model of shortest path proximity. Table 1 presents the Top-10 prevalent patterns mined by each method with the distance threshold to 600m. SSCPM is our approach, which utilizes the SPS measurement.

TABLE I. Top-10 prevalent patterns mined by three methods

Rank	SSCPM	CMN	NDB
1	{SS, BK}	{HT, BK}	{HT, BK}
2	{SS, PLZ}	{EV, KB}	{RB, BK}
3	{PLZ, ES}	{KB, HT}	{KB, HT}
4	{PK, PLZ}	{RB, BK}	{EV, BK}
5	{KB, HT}	{ES, BK}	{EV, KB}
6	{HT, BK}	{EV, BK}	{ES, BK}
7	{EV, BK}	{KB, REST}	{RB, EV}
8	{SS, EV}	{KB, BK}	{KB, BK}
9	{EV, KB}	{EV, HT}	{RB, HT}
10	{RB, BK}	{RB, HT}	{PLC, BK}

*SS: Subway station; BK: Bank; HT: Hotel; PLZ: Plaza; EV: Entertainment venues; KB: Karaoke bar; RB: Residential building; ES: Emergency shelter; PK: Park; REST: Restaurant; PLC: Police;

The Top-3 prevalent patterns mined by SSCPM are {SS, BK}, {SS, PLZ}, and {PLZ, ES}. Notably, 58.6%, 54.4%, and 53.2% of the row instances for these patterns maintain proximity relationships through more than one proximity path, which indicates a strong proximity relationship between instances. However, because CMN and NDB focus solely on the shortest paths when measuring proximity relationships and neglect other proximity paths, they fail to capture this strength of proximity relationship, resulting in these patterns not including in their Top-10 mining results. Additionally, while both CMN and NDB include the pattern {KB, BK} in their Top-10 results, 77% of its row instances maintain proximity through only one path, thus {KB, BK} is absent in the results of SSCPM. Furthermore, we employ the Pearson correlation coefficient to analyze the correlation between the ranks of patterns in the mining results and the proportion of multipath proximity instances in the instances of patterns.

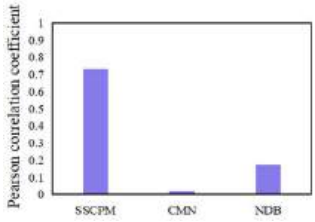


Fig. 5. The Pearson correlation coefficient between the rank of patterns and the proportion of multipath proximity in their instances.

As Fig.5 shown, the correlation between the ranks of patterns mined by SSCPM and the proportion of multipath proximity among pattern instances exceeds 0.7. This indicates that patterns with a higher proportion of instances connected by multiple paths tend to rank higher. In contrast, the coefficients for CMN and NDB are both below 0.2, indicating that the correlation between the ranks of mined patterns and this proportion is weak, which highlights their oversight of the impact of multipath proximity in SCPM.

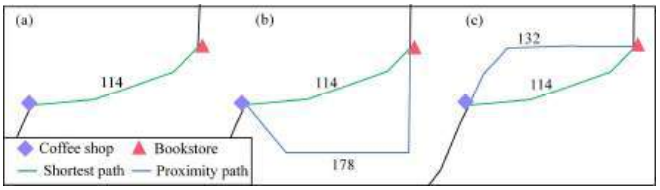


Fig. 6. Three different proximity relationships.

TABLE II. The strength of the proximity relationship calculated by different methods at a distance threshold of 200m.

Methods / Relationships	SSCPM	CMN	NDB
Fig.6(a)	0.92	0.67	1
Fig.6(b)	1.74	0.67	1
Fig.6(c)	1.82	0.67	1

In addition to considering the number of proximity paths, the SPS also considers the path distance. Fig.6 illustrates three different proximity relationships. We apply the three methods to calculate the strength of these proximity relationships, with the results presented in Table 2. NDB only considers whether there is a proximity between the coffee shop and the bookstore, but ignores the number and distance of proximity paths, resulting in

the same strength assigned to all three proximity relationships. CMN considers path length but focuses solely on the shortest path, which remains the same across the three proximity relationships, thus it also assigns equal strength for all proximity relationships. In contrast, SSCPM assigns different strengths. Since the proximity relationship shown in Fig.6(b) includes an additional proximity path compared to Fig.6(a), it receives a higher strength than Fig.6(a). Additionally, since Fig.6(c) has shorter proximity paths than Fig.6(b) while they both have two paths, Fig.6(c) is assigned a higher strength than Fig.6(b).

Next, we conduct the efficiency experiments on the POI dataset of Wuhou, Chengdu. This dataset consists of 3,119 instances and 16 features. The competitors include join-base [2], which generates table instances for candidate patterns through row instance join operations; the join-less [8], which utilizes star-neighborhood queries to create table instances for candidate patterns; and the FV algorithm adapted from [11], which employs a filter-and-verify mechanism to search for participating instances of patterns.

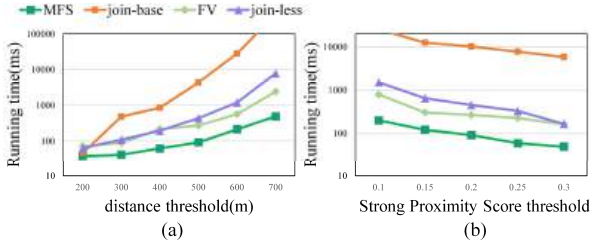


Fig. 7. The running time of algorithms under different parameters.

Fig.7(a) shows the running time of algorithms while varying the distance threshold *R* with a fixed Proximity Score threshold *min-SPS* of 0.1. When the distance threshold increases, more instances become proximate, thus the number of row instances and participating instances of a candidate pattern increases. For this reason, the time consumption of all algorithms is increasing. However, MFS consistently achieves the highest efficiency. Because MFS searches minimum instance pairs directly instead of generating row instances for candidate patterns, and MFS checks instance pairs in ascending order in their strength, allowing it to quickly terminate the search once the minimum instance pair is found. On the contrary, Fig.7(b) shows the running time while varying the *min-SPS* with a fixed *R* of 600m. When *min-SPS* increases, more candidate patterns are discarded, and the number of patterns that need to be checked reduces, so the time consumption of all algorithms decreases too. MFS is still several times more efficient than other algorithms because of its minimum instance pair search strategy.

The above experiments show that the SPS measurement can effectively measure the strength of proximity relationships, and MFS can hold its efficiency advantages under different thresholds.

V. CONCLUSION

In this paper, we mainly solve two problems. First, to address the issue in existing studies that focus solely on the shortest path between instances, which may overlook strong co-location patterns, we propose a novel prevalence measure named the Strong Proximity Score. This measure incorporates both multi-path proximity and distance decay effects to evaluate the association strength of patterns, thus effectively identifying strong spatial co-location patterns under road network constraints. Second, to improve the efficiency of Strong Spatial Co-location Pattern Mining(SSCPM), we introduce the Minimum First Search (MFS) algorithm, which uses a minimum instance pair search strategy to avoid generating table instances of candidate patterns. Extensive experiments are conducted on real datasets. Experiment results demonstrate that SSCPM can effectively identify strong co-location patterns overlooked by existing methods, and the MFS algorithm can enhance the efficiency of SSCPM.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (62306266, 62276227, 62266050), the Program of Yunnan Key Laboratory of Intelligent Systems and Computing (202405AV340009), the Project of Yunnan Province "Xingdian Talent Support Plan".

REFERENCES

[1] Shekhar. S, and Huang. Y, "Discovering Spatial Co-location Patterns: A Summary of Results," Advances in Spatial and Temporal Databases, 2001, vol.2121, pp.236-256.

[2] Huang. Y, Shekhar. S, and Xiong. H, "Discovering colocation patterns from spatial data sets: a general approach," IEEE Trans. Knowl. Data Eng, 2004, vol.16, pp.1472-1485.

[3] Lu. J, Wang. L, Fang. Y, and Zhao. J, "Mining strong symbiotic patterns hidden in spatial prevalent co-location patterns," Knowledge-Based Syst, 2018, vol.146, pp.190-202.

[4] He. Z, Deng. M, Xie. Z, Wu. L, Chen. Z, and Pei. T, "Discovering the joint influence of urban facilities on crime occurrence using spatial co-location pattern mining," Cities, 2020, vol.99, article.102612.

[5] Yu. W, "Spatial co-location pattern mining for location-based services in road networks," Expert Syst. Appl, 2016, vol.46, pp.324-335.

[6] Yu. W, Ai. T, He. Y, and Shao. S, "Spatial co-location pattern mining of facility points-of-interest improved by network neighborhood and distance decay effects," Int. J. Geogr. Inf. Sci, 2017, vol.31, pp.280-296.

[7] Yao. X, Jiang. X, Wang. D, Yang. L, Peng. L, and Chi. T, "Efficiently mining maximal co-locations in a spatial continuous field under directed road networks," Inf. Sci, 2021, vol.542, pp.357-379.

[8] Yoo. J. S, & Shekhar. S, "A Joinless Approach for Mining Spatial Colocation Patterns," IEEE Trans. Knowl. Data Eng, 2006, vol.18, pp.1323-1337.

[9] Wang. L, Bao. Y, and Lu. Z, "Efficient Discovery of Spatial Co-Location Patterns Using the iCPI-tree," Open Inf Syst J, 2009, vol.3, pp.69-80.

[10] Yang. P, Wang. L, Wang. X, and Zhou. L, "A spatial co-location pattern mining approach based on column calculation," Sci Sin Inform, 2022, vol.52, pp.1053-1068.

[11] Chan. H. K. H, Long. C, Yan. D, Raymond C. W. Wong, and Lu. H, "Fraction-Score: A Generalized Support Measure for Weighted and Maximal Co-Location Pattern Mining," IEEE Trans. Knowl. Data Eng, 2024, vol.36, pp.1582-96.

[12] Yang. P, Wang. L, Zhou. L, and Chen. H, "Mining Spatial Co-Location Patterns With a Mixed Prevalence Measure," IEEE Trans. Neural Netw. Learn. Syst, 2024, vol.35, pp.7845-7859.

[13] Yao. X, Chen. L, Peng. L, and Chi. T, "A co-location pattern-mining algorithm with a density-weighted distance thresholding consideration," Inf. Sci, 2017, vol.396, pp.144-161.

[14] Chondrogiannis. T, Bouros. P, Gamper. J, Leser. U, and Blumenthal. D. B, "Finding k-shortest paths with limited overlap," VLDB J, 2020, vol.29, pp.1023-1047.

Enhancing NMF-based Community Detection via A Higher-order Proximity-Incorporated Graph Attention Autoencoder

Hao Yan
School of Computer Science and Technology
Dongguan University of Technology
Dongguan, China
yanhao@dgut.edu.cn

Zhigang Liu
School of Computer Science and Technology
Dongguan University of Technology
Dongguan, China
liuzhigangx@gmail.com

Yurong Zhong
School of Computer Science and Technology
Dongguan University of Technology
Dongguan, China
zhongyurong91@gmail.com

Weiling Li
School of Computer Science and Technology
Dongguan University of Technology
Dongguan, China
weilinglicq@outlook.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899563

***Abstract*—Community detection reveals the organizational patterns of complex networks. With good interpretability, non-negative matrix factorization (NMF) approaches are often used for this task. However, their linearity limits the performance of community detection on networks with complex, sparse, and diverse structures. As graph neural networks (GNNs) have strong nonlinear representation power, combining NMF with GNNs can overcome this limitation. However, a GNN model generally faces the issue of over-smoothing and struggles to explicitly use higher-order proximity (HOP) among nodes. To address this issue, this paper proposes HNG, a HOP-enhanced nonlinear NMF approach for community detection, with three modules: HOP-incorporated network enhancement, graph attention autoencoder (GAAE), and NMF. It first leverages HOP explicitly to better characterize the community structure for the facility of community detection, and then applies a nonlinear NMF improved via a GAAE for accurate community detection. Experimental results on six real networks show that HNG significantly outperforms state-of-the-art methods in achieving community detection accuracy gain.**

Keywords—Community Detection, Symmetric Non-negative Matrix Factorization, Network Enhancement, Graph Neural Networks, Network Representation Learning

I. INTRODUCTION

Networks describe sophisticated interaction relationships among entities in real complex systems [1], such as cyber-physical systems and biological networks. The underlying community structure that reveals organizational patterns is a fundamental aspect of network analysis. Community detection aiming to identify community structure is a long-standing issue [2]. Accurately identified communities further facilitate the study of various real applications, for example, in biological networks [3], identifying communities formed by proteins or genes with similar functions aids in understanding biological processes and disease mechanisms.

To date, a variety of community detection methods have been developed, e.g., traditional heuristic methods [4], optimization methods [5], network dynamics [6], and network representation learning [7]. Among them, NMF has notable suitability for graph clustering owing to its inherent clustering capabilities [8]. While NMF-based models are effective, they rely on linear representation principles, which makes capturing non-linear features from irregular, non-Euclidean networks challenging [9], limiting their performance when dealing with networks containing diverse structural information [10].

To break this limitation, existing methods try to improve an NMF-based community detection model by enhancing its representation learning ability. Some works design detection models based on deep NMF [10] and nonlinear NMF [11]. By utilizing a deep NMF's multilayer factorization framework, Ye *et al*. [12] introduce a deep auto-encoder-like NMF (DANMF) model for community detection. Nonetheless, the inherent linear factorization at each layer means that the community detection remains fundamentally linear. Maekawa *et al*. [11] present an approach for graph clustering, i.e., nonlinear attribute graph clustering (NAGC), which adopts symmetric NMF (SNMF) with positive sample labeling learning. It employs non-linear mapping to discern the connections across various communities learned from the topology and attributes. Despite its superior results compared to existing NMF-type methods, NAGC's application is limited to networks without attributes.

In recent years, graph neural networks (GNNs) have gained attention for their ability to learn nonlinear features from graph data. Variants like graph convolutional networks (GCNs) and graph attention networks (GATs) have emerged, showing potential in tasks like community detection. Inspired by this, He *et al*. [13] propose to enhance NMF-based community detection using GNN's strong nonlinear feature learning ability. In general, a GNN model aggregates high-order neighborhood information via the message-passing mechanism. Nevertheless, it takes an original network topology describing the first-order proximity (FOP) as input, which may lead to accuracy loss since it fails to fully capture the importance of higher-order proximity (HOP) in the original network.

This work is supported in part by the Guangdong Basic and Applied Basic Research Foundation under grant 2023A1515110689, in part by the National Natural Science Foundation of China under grant 62102086, and in part by the Guangdong Province Universities and College Pearl River Scholar Funded Scheme (2019). *Corresponding Author: Zhigang Liu*.

Motivated by the above findings, this study aims to explicitly utilize HOP from the perspective of data enhancement, thus better characterizing the community structure for the facility of community detection. To do this, the paper proposes a novel HOP-enhanced nonlinear NMF method by incorporating a GNN module, i.e., HNG, for community detection. It consists of three parts, i.e., HOP-incorporated network enhancement, graph attention autoencoder (GAAE), and SNMF. Our HNG method includes three-fold ideas:

- Measuring HOP among nodes in a network with a pointwise mutual information (PMI)-based method;
- Encoding the captured HOP into the target network via an iterative enhancement scheme to explicitly strengthen its topological structure; and
- Implementing a nonlinear SNMF approach for highly accurate representation learning and community detection of the enhanced network by integrating a GAAE unit.

The contributions of this paper are as follows:

a) *An HOP-integrated network enhancement scheme.* It first adopts a carefully designed random walk-based neighbor sampling scheme to acquire a certain proportion of underlying HOP node pairs from the target network, with these node pairs having different proximity orders, and then calculates the correlation weight between each HOP node pair with the PMI based on their co-occurrence frequencies. By doing so, the overall proximity strength between specific node pairs can be accurately represented. Finally, the scheme encodes the acquired HOP indices into the network to clarify its community structure and make it more informative.

b) *An HNG-based community detection model.* Accounting for GNN's representation power for complicated features from graph data, this work implements a nonlinear SNMF-based method that integrates SNMF and GAAE into a unified framework for community detection. By doing so, the HNG-based community detection method owns the merits of NMF's well-interpretable clustering characteristics and GNN's nonlinear and non-European representation power. In addition, taking the HOP-enhanced network as input, the model achieves highly accurate community detection.

Experimental results on six real networks indicate that our HNG model significantly outperforms state-of-the-art methods in achieving high community detection accuracy gain.

The rest of this paper is organized as follows. Section II presents the preliminaries. Section III describes the proposed methods and Section IV analyzes experimental results in detail. Section V concludes the paper.

II. Notations and Problem Statement

This work considers an undirected, unweighted network $G=(V, E)$, where $V=\{v_1, v_2, \ldots, v_n\}$ represents the set of n nodes and $E=\{e_1, e_2, \ldots, e_m\}$ represents the set of m edges. G's structural information is represented by an adjacency matrix $\mathbf{A}\in\mathbb{R}^{n\times n}$, where $\mathbf{A}$ is a symmetric, non-negative binary matrix. If $e_{ij}\in E$, there is an edge between nodes v_i and v_j with $\mathbf{A}_{ij}=1$, and otherwise $\mathbf{A}_{ij}=0$. Given a target network G, assuming it contains K non-overlapping communities. A community detection model aims to partition the node set V into a collection of communities, i.e., $C=\{C_k \mid C_k\neq\emptyset, \cup_K^{k=1}=V, C_k\neq C_t, 1\le k\le K, 1\le t\le K\}$, where C_k is the k-th community in C, and $\cup$ denotes the union set.

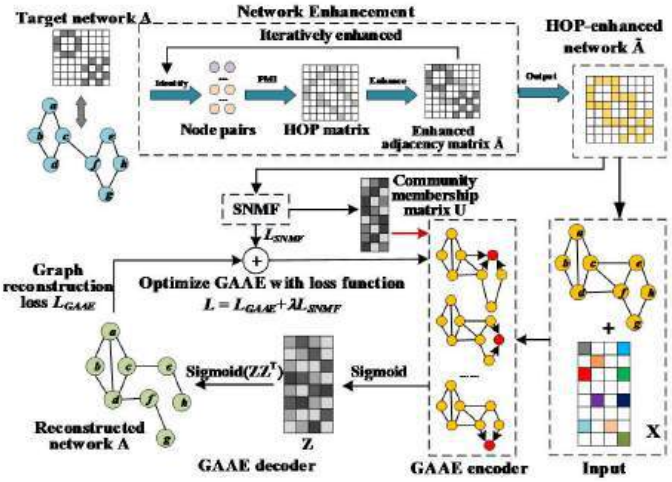


Fig. 1. The framework of the HNG-based community detection method

III. Methods

A. An Overview of an HNG-based Community Detector

Fig. 1 illustrates the framework of the proposed HNG-based community detection method. HNG is composed of three primary modules, i.e., HOP-incorporated network enhancement, GAAE, and SNMF. In detail, the network enhancement module first adopts a random walk-based search strategy with various step sizes to acquire the HOP node pairs within a target network G, and utilizes a PMI-based method to measure HOP indices among nodes, and then explicitly encodes the captured HOP indices into the adjacency matrix $\mathbf{A}$ via a well-designed iterative augmentation scheme to strengthen the topological information, thus achieving $\breve{\mathbf{A}}$. By doing so, the network becomes more informative, thus strengthening community structure within the network. Based on the enhanced $\breve{\mathbf{A}}$, the GAAE and SNMF modules work compatibly to perform network representation learning and community detection together. Specifically, the GAAE module aims to learn network embedding $\mathbf{Z}$ by taking $\breve{\mathbf{A}}$ and the attribute matrix $\mathbf{X}$ (It can be substituted with a simple identity matrix or derived from other network embedding algorithms like LINE [14]) as input. More specifically, we use the SNMF module to obtain the community indicator matrix $\mathbf{U}$ and leverage it to guide the GAAE module's attention mechanism. Moreover, we adopt graph regularization in the SNMF module to maintain the local network geometrical characteristic. By doing so, a unified framework for community detection, i.e., the HNG model, is achieved.

B. HOP-integrated Network Enhancement

HOP is critical for a network's structural representation and helpful for improving a learning-based community detection method. To quantify HOP, we adopt PMI as a measurement, which compares the frequency at which two entities are observed simultaneously and the frequency at which they are observed separately, formulated as

$$PMI(w_1, w_2) = \log\frac{p(w_1, w_2)}{p(w_1)\,p(w_2)}, \tag{1}$$

where $p(w_1)$ and $p(w_2)$ are observing frequencies of w_1 and w_2, and $p(w_1, w_2)$ is the co-occurrence frequency.

Given a node pair (v_i, v_j), the observing frequencies can be calculated as follows

$$p(v_i)=\frac{N(v_i)}{|M|}=\xi_1\frac{N_1(v_i)}{|M_1|}+\ldots,+\xi_r\frac{N_r(v_i)}{|M_r|}, \quad (2)$$

$$p(v_j)=\frac{N(v_j)}{|M|}=\xi_1\frac{N_1(v_j)}{|M_1|}+\ldots,+\xi_r\frac{N_r(v_j)}{|M_r|}, \quad (3)$$

$$p(v_i,v_j)=\frac{N(v_i,v_j)}{|M|}=\xi_1\frac{N(v_i,v_j)}{|M_1|}+\ldots,+\xi_r\frac{N(v_i,v_j)}{|M_r|}, \quad (4)$$

where $p(v_i)$ and $p(v_j)$ denote the observing frequencies of all node pairs that contain v_i and v_j in the searched neighbor set, and $p(v_i, v_j)$ denotes the observing frequency of node pair $\{v_i, v_j\}$. $N_r(v_i)$ and $N_r(v_j)$ denote the observing numbers of the r-order node pairs that contain v_i and v_j, and $N_r(v_i, v_j)$ denotes the numbers of the r-order node pair $\{v_i, v_j\}$. M_r denotes a set of the r-order node pairs, and $|M_r|$ is the total number of node pairs in M_r. ξ_r is the weight for the r-order HOP. Considering that the HOP's importance could be weakened as the proximity order increases, in this study, we set it with the reciprocal of the proximity order. By submitting (2)-(4) into (1), we achieve the following computational formula for HOP index

$$PMI(v_i,v_j)=\log\frac{\sum_{y=1}^{r}\frac{N_y(v_i,v_j)}{y|M_y|}}{\sum_{y=1}^{r}\frac{N_y(v_i)}{y|M_y|}\cdot\sum_{y=1}^{r}\frac{N_y(v_j)}{y|M_y|}}. \quad (5)$$

To acquire HOP node pairs, we adopt the random walk search strategy with various step sizes on the target network. Initially, we empirically set the percentage of nodes to be searched to 75%, and then specify the number of random walks (n_W) starting from each node and the length of each random walk (n_L), i.e., the number of nodes included in each walk. It is worth noting that the settings of n_W and n_L are crucial. For instance, too many random walks will gather redundant information, increasing computational complexity and time. Conversely, a longer walk length may shift the focus to the global structure of the network, neglecting the local community characteristics and introducing noise, which degrades the performance.

With the collected HOP node pairs, based on (5), we first calculate the HOP indices among node pairs, and then take the HOP indices as valuable supplementary information and encode them into the original network G, resulting in the enhanced adjacency matrix $\breve{\mathbf{A}}=[\breve{\mathbf{A}}_{ij}]$. The network enhancement scheme is formulated as:

$$\breve{\mathbf{A}}_{ij}\begin{cases}1, & \mathbf{A}_{ij}=1,\\ \begin{cases}1, & PMI(v_i,v_j)\geq\log\varepsilon,\\ 0, & PMI(v_i,v_j)<\log\varepsilon,\end{cases} & \text{otherwise;}\end{cases} \quad (6)$$

where ε is a threshold coefficient that determines whether the weight between the node pair $\{v_i, v_j\}$ can be added to $\breve{\mathbf{A}}$ based on their HOP index.

C. A GAAE Module

The GAAE module comprises an encoder and a decoder. The encoder initially takes the enhanced adjacency matrix $\breve{\mathbf{A}}$ and the attribute matrix $\mathbf{X}$ as input to learn the node embeddings $\mathbf{Z}$ of the HOP-enhanced network by adopting a multiple-layer attention-based graph convolutional network model. To make the GAAE module better fit the overall goal of community detection, we adopt a specific attention mechanism guided by the community membership knowledge achieved from the SNMF module (we will introduce it later). Thus, the attention coefficient between two nodes is defined as follows:

$$\vartheta_{ij}=\alpha\left(\mathbf{U}_i\mathbf{W}\,\|\,\mathbf{U}_j\mathbf{W}\right), \quad (7)$$

where α is an attention mechanism, $\mathbf{W}\in\mathbb{R}^{K\times K}$ is a shared weight matrix. Notably, $\mathbf{U}$ is the community indicator matrix instead of the enhanced adjacency matrix $\breve{\mathbf{A}}$ to learn attention coefficients.

By integrating the community membership knowledge into the learning of attention coefficients, the GAAE can be directed by the SNMF-based community detection module, making HNG obtain the merits of SNMF's well-interpretable clustering characteristics and GNN's nonlinear representation capability. To calculate attention coefficients comparable, we adopt the softmax function and the LeakyReLU activation function to obtain the final expression of the attention coefficients, i.e.,

$$\alpha_{ij}=\frac{\exp\left(\delta\left(\boldsymbol{\alpha}\left[\mathbf{U}_i\mathbf{W}\,\|\,\mathbf{U}_j\mathbf{W}\right]\right)\right)}{\sum_{q\in N_i}\exp\left(\delta\left(\boldsymbol{\alpha}\left[\mathbf{U}_i\mathbf{W}\,\|\,\mathbf{U}_q\mathbf{W}\right]\right)\right)}, \quad (8)$$

where $\boldsymbol{\alpha}\in\mathbb{R}^{2K}$ is a weight vector, δ is the LeakyReLU activation function.

For stacking L graph attention layers, we use ReLU as the nonlinear activation function. The layer-wise representation of each node is obtained through the following aggregation scheme:

$$\mathbf{Z}_i^{(l+1)}=\sigma\left(\sum_{j\in N_i}\alpha_{ij}\mathbf{W}^{(l)}\mathbf{Z}_j^{(l)}\right), \quad (9)$$

where σ denotes the ReLU activation function, $\mathbf{W}\in\mathbb{R}^{K\times K}$ is a shared weight matrix, $\mathbf{Z}_i^{(l)}$ and $\mathbf{W}^{(l)}$ denote the embedding of v_i and the weight matrix at the l-th layer. It is worth noting that in the first layer, we initialize $\mathbf{Z}^0=\mathbf{X}$, and we use the sigmoid function to learn the final network embedding $\mathbf{Z}$. Thus, we have

$$\mathbf{Z}=sigmod\left(\mathbf{Z}^{(L-1)}\right). \quad (10)$$

Note that $\mathbf{Z}$ is non-negative, and its values fail in the range of [0, 1]. The multiple attention mechanism is introduced to allow the transformation in (9) to be executed independently. Thus, the hierarchical embeddings after the transformation are averaged again, which is formulated as

$$\mathbf{Z}_i^{(l+1)}=\sigma\left(\frac{1}{H}\sum_{t=1}^{H}\sum_{j\in N_i}\alpha_{ij}^t\mathbf{W}_t^{(l)}\mathbf{Z}_j^{(l)}\right), \quad (11)$$

where H is the number of attention heads, and α_{ij}^t and $\mathbf{W}_t^{(l)}$ denote the attention coefficient and the corresponding weight matrix related to the t-th attention mechanism.

Based on $\mathbf{Z}$, the decoder aims to reconstruct the network G. By adopting the inner product as the decoder operation, we have

$$\hat{\mathbf{A}} = sigmod\left(\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\right). \quad (12)$$

where $\hat{\mathbf{A}}$ is the reconstructed adjacency matrix. Here, we use the binary cross-entropy function to calculate the reconstruction loss of GAAE, i.e.,

$$\mathrm{L}_{GAAE} = -\frac{1}{n^2}\sum_{i,j}\left(\breve{\mathbf{A}}_{ij}\log\hat{\mathbf{A}}_{ij} + \left(1-\breve{\mathbf{A}}_{ij}\right)\log\left(1-\hat{\mathbf{A}}_{ij}\right)\right). \quad (13)$$

D. A SNMF Module

The SNMF module takes the enhanced adjacency matrix $\breve{\mathbf{A}}$ as input and decomposes it to obtain the community membership indicator $\mathbf{U}$. To do this, we build the following loss for the SNMF module

$$\mathrm{L}_{SNMF} = \left\|\breve{\mathbf{A}} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\right\|_F^2. \quad (14)$$

To maintain the local invariance of the target network, we further incorporate graph regularization into (14) to achieve the following extended learning objective:

$$\mathrm{L}_{SNMF} = \left\|\mathbf{A} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\right\|_F^2 + \gamma\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{L}\mathbf{U}),\ \ s.t.\ \ \mathbf{U}\geq 0. \quad (15)$$

where $\mathbf{L}=\mathbf{D}-\mathbf{A}$, $\gamma>0$ is a tunable parameter to adjust the effect of graph regularization, $\mathbf{L}$ is the Laplace matrix of $\mathbf{A}$, and $\mathbf{D}$ is the degree matrix computed by $\mathbf{D}_{ii}=\sum_l \mathbf{W}_{il}$.

E. A Unified Optimization Scheme

To implement HNG, we combine the reconstruction loss of GAAE and the loss of SNMF to construct a unified loss function so that the GAAE module and the SNMF module benefit from each other. The constructed unified loss function is given as:

$$\mathrm{L} = \mathrm{L}_{GAAE} + \lambda\mathrm{L}_{SNMF}, \quad (16)$$

where λ is a weighting parameter that balances the two loss components. Optimizing (16) is a non-convex problem, making a closed-form solution difficult. Nevertheless, it can be solved using an alternating optimization strategy.

We first optimize GAAE by fixing SNMF's variables, i.e., $\mathbf{U}$, as constants. Standard gradient descent is then used to update GAAE's network parameters according to the learning rules:

$$\mathbf{W} = \mathbf{W} - \eta\frac{\partial\mathrm{L}}{\partial\mathbf{W}}, \mathbf{W}_t^{(l)} = \mathbf{W}_t^{(l)} - \eta\frac{\partial\mathrm{L}}{\partial\mathbf{W}_t^{(l)}}, \quad (17)$$

where η is the learning rate. The gradients, i.e., $\partial\mathbf{\Lambda}/\partial\mathbf{W}$ and $\partial\mathbf{\Lambda}/\partial\mathbf{W}_t^{(l)}$, are calculated via the back-propagation algorithm. Actually, (17) is optimized by an Adam optimizer.

With the variables in GAAE fixed, the SNMF module can be solved. By fixing GAAE, (15) is reformed as

$$\mathbf{U} \leftarrow \arg\min \mathrm{L}_{SNMF},\ \ s.t.\ \ \mathbf{U}\geq 0. \quad (18)$$

Following Lee and Seung *et al.* [15], the optimization problem in (18) can be solved by the non-negative multiplicative update (NMU) learning scheme, i.e.,

$$\mathbf{U}_{ik} = \mathbf{U}_{ik}\frac{(1+\gamma)(\mathbf{A}\mathbf{U})_{ik}}{\left(\mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{U}+\gamma\mathbf{D}\mathbf{Y}\right)_{ik}}. \quad (19)$$

With the learning rules in (17) and (19), an HNG-based community detection model is obtained. It leverages a standard k-means algorithm to cluster node representation vectors in $\mathbf{Z}$ to determine the final affiliation that an arbitrary node belongs to a specific community.

IV. Experimental Results and Analysis

A. General Settings

To obtain fair experimental results, we first adopt a ground-truth-irrelevant metric, i.e., Modularity Q [16], as the validation metric for models' hyperparameter selection, and then adopt two widely-used metrics, i.e., Accuracy (ACC) [16] and Normalized Mutual Information (NMI) [16], to evaluate the performance of community detection models.

This study adopts six real-world networks with ground-truth community labels in the experiments, as summarized in Table I. Nine state-of-the-art community detection methods are adopted for a comprehensive comparison to assess the performance of the proposed HNG-based community detection method, i.e., SSAGCN [17], ADNMF [18], NAGC [11], NMFGAAE [13], DGCN-NMF [19], GraphSAGE [20], vGraph [21], GCN [22], M-NMF [23]. All models' hyperparameters are set to their optimal values, and Python 3.7 is used for implementation.

B. Hyperparameter Sensitivity Test

As discussed in Section III, several hyperparameters are involved in HNG, i.e., λ, γ, n_W and n_L. They are vital for its performance. Note that the graph regularization coefficient is well studied in previous work [24, 25]. According to their suggestions, we set its value at one in our study uniformly. The dimension of node embeddings of all network representation learning-based methods is set to be 64. Next, we aim to explore the effects of λ, n_W and n_L regarding HNG's performance.

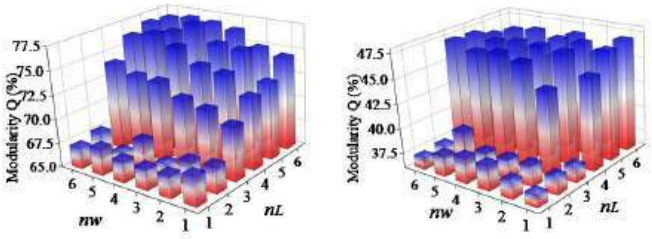
To test the effect of n_W and n_L, we conduct a grid search experiment for their optimal settings by selecting their values from the range of {1, 2, 3, 4, 5, 6}. Results on Cora and Gene networks are depicted in Fig. 2, and similar observations can be obtained on the other networks. As shown in Fig. 2, HNG's performance is sensitive to n_W and n_L. HNG performs consequently better as n_W and n_L increase in a certain range, and its accuracy degrades when n_W and n_L become too large. The reason for such a phenomenon is that too many random walks may lead to the collection of a large amount of redundant information; too long a walk length may extend the attention to the global structure of the whole network and ignore the local community properties, as well as introduce noise, both of which will lead to a deterioration of the final community detection.

TABLE I. Datasets

Networks	n	$\|E\|$	k	Description
Cora	2708	5278	7	LINQS [13]
Citeseer	3327	4552	6	LINQS [13]
Gene	1103	1672	2	Network Repository [26]
PubMed	19717	44324	3	LINQS [13]
ACM	3025	26256	3	ACM [27]
Polblogs	1490	16715	2	LINQS [13]

TABLE II. PERFORMANCE COMPARISON RESULTS (%). (BOLD/SHADED RECORDS INDICATE THE BEST/SECOND-BSET RESULTS)

Methods	Metrics	Cora	Citeseer	PubMed	Gene	ACM	Polblogs
SSAGCN	NMI	28.4±3.0	6.7±1.2	15.3±2.0	1.5±0.7	3.8±1.6	47.9±4.8
	ACC	47.2±3.0	25.6±0.8	54.8±1.2	57.8±0.9	40.9±1.4	86.0±2.6
	Q	56.2±2.5	41.4±2.1	37.5±3.5	34.1±4.7	47.9±4.2	42.4±0.1
ADNMF	NMI	28.0±0.0	12.0±0.0	6.5±0.0	4.0±0.0	4.3±0.0	42.4±0.0
	ACC	49.6±0.0	37.7±0.0	41.5±0.0	55.9±0.0	42.6±0.0	86.3±0.0
	Q	39.3±0.0	46.1±0.0	35.0±0.0	36.4±0.0	50.6±0.0	42.0±0.0
NMFGAAE	NMI	30.5±1.2	14.3±1.8	16.0±3.3	1.5±0.8	9.3±4.5	46.9±2.6
	ACC	45.5±2.4	32.4±1.2	54.4±2.6	56.1±2.5	43.3±2.2	85.9±1.8
	Q	72.9±0.6	70.7±0.7	57.3±0.8	45.3±1.9	56.2±1.0	43.0±0.1
NAGC	NMI	31.9±2.6	18.0±1.8	13.8±2.8	2.7±1.1	10.6±3.1	43.7±1.8
	ACC	46.1±2.0	36.6±1.1	51.5±2.4	57.7±4.3	45.0±2.8	85.6±1.2
	Q	65.0±4.1	54.2±1.5	49.2±2.7	35.6±9.6	56.0±3.8	42.2±0.0
DGCN-NMF	NMI	10.6±1.2	4.0±2.2	0.6±0.1	0.7±0.3	1.8±0.9	29.5±5.6
	ACC	31.5±2.1	25.6±2.7	40.3±2.1	54.2±3.3	40.4±2.5	79.7±2.8
	Q	40.9±3.2	36.8±5.0	12.1±1.6	14.6±2.9	32.8±1.8	38.5±4.3
GraphSAGE	NMI	28.2±3.4	19.2±4.0	17.9±4.5	0.6±2.5	4.4±1.1	12.5±3.6
	ACC	48.0±4.1	42.3±4.2	53.0±7.9	55.3±2.6	37.2±1.3	66.0±4.7
	Q	49.5±1.8	53.7±3.5	42.5±2.5	25.9±2.2	41.3±6.5	13.7±6.6
vGraph	NMI	9.3±0.6	5.5±0.6	0.1±0.0	0.5±0.1	5.1±4.1	0.7±0.1
	ACC	30.3±2.4	28.0±1.9	35.5±0.6	54.2±0.9	43.5±2.9	54.8±0.4
	Q	57.4±0.4	66.6±0.7	7.6±0.7	34.0±0.4	41.8±5.9	1.3±0.2
GCN	NMI	6.4±1.2	3.3±1.0	4.5±0.6	2.7±0.2	3.1±0.6	11.6±1.2
	ACC	24.9±0.8	25.9±1.4	43.0±3.4	56.6±0.3	35.8±1.3	59.7±2.2
	Q	59.4±0.9	66.6±0.8	47.8±4.4	12.1±3.6	26.6±2.2	20.0±2.6
M-NMF	NMI	17.4±2.5	3.5±1.3	5.5±3.2	1.0±0.6	3.4±2.0	44.1±2.9
	ACC	36.7±03.7	25.8±2.4	46.1±4.2	52.8±1.2	43.4±2.6	86.7±6.0
	Q	65.7±1.6	68.4±3.4	49.1±4.3	38.1±1.7	56.8±5.2	42.2±0.2
HNG (Our)	NMI	**44.1±1.6**	**26.5±0.6**	**24.3±0.6**	**5.4±1.0**	**22.3±1.1**	**52.8±0.8**
	ACC	**62.4±1.3**	**47.5±0.4**	**62.4±0.1**	**60.7±2.9**	**60.4±0.4**	**89.4±0.2**
	Q	**76.9±0.3**	**75.6±0.7**	**60.9±0.2**	**46.9±0.2**	**61.7±0.7**	**43.1±0.1**



(a) Cora network (b) Gene network

Fig. 2. The effect of n_W and n_L on the Cora and Gene networks

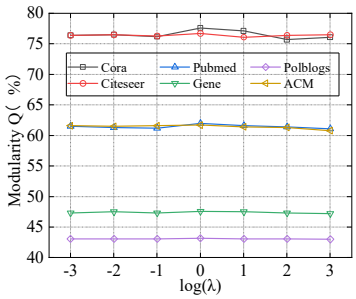


Fig.3. The performance of HNG with various λ on all networks

Specifically, we set n_W=4 and n_L=5 for the Cora, Gene, and Polblogs networks, n_W=5 and n_L=6 for the Citeseer and ACM networks, and n_W=7 and n_L=8 for the PubMed network.

Then we investigate λ's effect on the performance of the HNG model by tuning it in the set of $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ with the Modularity metric. The corresponding result tested on all networks is depicted in Fig. 3. From it, we see that when $\lambda \geq 10$, HNG's performance decreases slightly. This indicates that λ has little effect on HNG's performance. In general, HNG can achieve satisfactory accuracy when λ is set in the scale of $[10^{-1}, 10^{1}]$. Given this, we set λ=1 uniformly in our experiments.

C. Comparison Results and Analysis

The comparison results across NMI, ACC, and Modularity metrics are shown in Table II. From this, we conclude that HNG consistently outperforms its peers in community detection accuracy on all testing networks. The accuracy improvements are significant. For example, on the Cora network, the average NMI, ACC and Q values obtained by HNG are 44.1%, 62.4% and 76.9%, respectively. The improvements compared to the second-highest NMI at 31.9% obtained by NAGC, ACC at 49.6% obtained by ADNMF, and Q at 72.9% obtained by NMFGAAE arrive at 27.7%, 20.5%, and 5.2%, respectively. These results confirm HNG's competitive advantage in community detection.

D. Ablation Analysis

In this section, we conduct an ablation study to verify the effectiveness of the proposed HOP-based network enhancement scheme. In this experiment, we compare the performance between HNG and its variant that takes out the network enhancement module with the Modularity metric. It is worth mentioning that we can achieve similar situations when HNG is evaluated with NMI and ACC. The experimental results are depicted in Fig. 4. From it, we see that by incorporating the HOP-based network enhancement, HNG's performance on community detection significantly improves. Such results tell us that HOP has great significance to community detection.

To further illustrate their community detection results visually, we visualize them in a two-dimensional space with the final node representations in the way of the *t*-distributed stochastic neighbor embedding on the Polblogs network in Fig.

5. From it, we can see that the communities acquired by HNG are more legible and compact, and the sample points of different

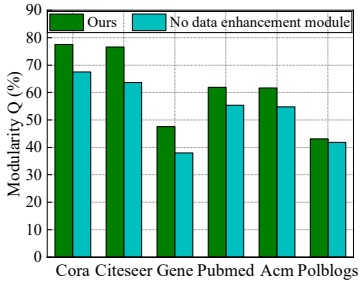
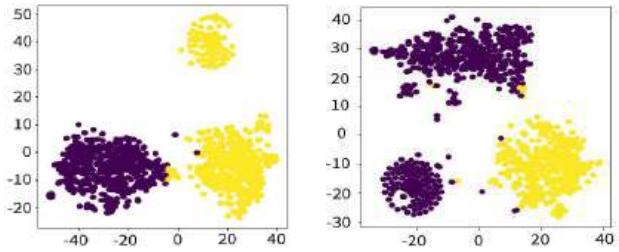


Fig. 4. Experimental results of ablation analysis



(a) HNG (b) The variant

Fig. 5. Visualization of community detection results on the Polblogs network

communities overlap less, making the boundary line between the two detected clusters significant. Such phenomena indicate that HNG performs better than the variant without integrating the HOP information.

V. Conclusion

A novel HOP-enhanced nonlinear NMF model called HNG is proposed in this paper to boost an NMF-based community detection method by improving the model's non-linear representation learning ability and enhancing the structural information of a target network via a novel HOP-incorporated network enhancement scheme to better characterize the community structure for the facility of community detection. An alternating optimization-based learning scheme is developed to efficiently solve the model. Extensive experiments on real networks show that HNG outperforms state-of-the-art methods in achieving high accuracy gain. In our future work, we will explore the adaptive selection mechanism of hyperparameters by adopting Bayesian optimization techniques.

References

[1] M. Riolo and M. Newman, "Consistency of community structure in complex networks," *Physical Review E*., vol. 101, no. 5, 2020.

[2] D Jin, Z. Yu, P. Jiao, *et al*., "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1149-1170, 2023.

[3] S. Bhowmick and B. Seah, "Clustering and summarizing protein-protein interaction networks: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 3, pp. 638-658, 2016.

[4] M. Javed, M. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *J. Netw. and Comput. Appl*., vol. 108, pp. 87-111, 2018.

[5] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Trans. Evolut. Comput*., vol. 22, no. 3, pp. 464-483, 2018.

[6] M. Okuda, S. Satoh, Y. Sato, and Y. Kidawara, "Community detection using restrained random walk similarity," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 43, no. 1, pp. 89-103, 2021.

[7] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol.34, no. 1, pp. 249-270, 2022.

[8] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using Bayesian non-negative matrix factorization," *Physical Review E*., vol. 83, no. 6, 2011.

[9] R. Ibrahim and D. Gleich, "Nonlinear diffusion for community detection and semi-supervised learning," *Proc. of the World Wide Web Conf.*, 2019, pp. 739-750.

[10] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, "A deep matrix factorization method for learning attribute representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 417-429, 2017.

[11] S. Maekawa, K. Takeuchi, and M. Onizuka, "New attributed graph clustering by bridging attribute and topology spaces," *J. Inf. Process.*, vol. 28, pp. 427-435, 2020.

[12] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection" in *Proc. of ACM Conf. Inf. Knowl. Manage.*, 2018, pp. 1393-1402.

[13] C. He, Y. Zheng, X. Fei, H. Li, Z. Hu, and Y. Tang, "Boosting nonnegative matrix factorization based community detection with graph attention auto-encoder," *IEEE Trans. on Big Data*, vol.8, no. 4, pp. 968-981, 2022.

[14] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information net-work embedding," in *Proc. of the 24th Int. Conf. World Wide Web*, 2015, pp. 1067-1077.

[15] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788-791, 1999.

[16] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Comput. Survey*, vol. 50, no. 4, pp. 1-34, 2017.

[17] C. He, J. Cheng, G. Chen, *et al*., "Multiple topics community detection in attributed networks," in *Proc. of the 46th Int. ACM SIGIR Conf. Res. Dev. Inf. Ret.*, 2023, pp. 2199-2203.

[18] J. Cheng, Y. Tang, C. He, *et al*., "Community detection in attributed networks via adaptive deep nonnegative matrix factorization," *Neural Computing and Applications*, vol. 36, no. 2, pp. 897-912, 2024.

[19] S. Xu, S. Liu, and L. Feng, "Deep graph convolution neural network with non-negative matrix factorization for community discovery," 2021, *arXiv:2103.05768*.

[20] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. of the 21st Conf. Neural Inf. Process. Syst*., 2017, pp. 1024-1034.

[21] F. Sun, M. Qu, J. Hoffmann, *et al*., "vGraph: A generative model for joint community detection and node representation learning," in *Proc. of the 33rd Conf. Neural Inf. Process. Syst*., 2019, pp. 512-522.

[22] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907.*

[23] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, "Community preserving network embedding," in *Proc. of AAAI Conf. Artif. Intell*., vol. 31, no. 1, pp. 203-209, 2017.

[24] X. Luo, Z. Liu, L. Jin, *et al*., "Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1203-1215, 2022.

[25] Z. Liu, X. Luo, and M. Zhou, "Symmetry and graph bi-regularized non-negative matrix factorization for precise community detection," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1406-1420, 2024.

[26] S Su, J. Guan, B. Chen, *et al*., "Nonnegative matrix factorization based on node centrality for community detection," *ACM Trans. Knowl. Discov. D*. vol. 17, no. 6, pp. 1-21, 2023.

[27] Y. Liu, W. Tu, S. Zhou, *et al*., "Deep graph clustering via dual correlation reduction," in *Proc. of AAAI Conf. Artif. Intell*., vol. 36, no. 7, pp. 7603-7611, 2022.

Genetic NEAT-Based Method for Multi-class Classification

1st Abdallah Alfaham
IDLab - Faculty of Applied Engineering
University of Antwerp - imec
Antwerp, Belgium
abdallah.alfaham@uantwerpen.be

2nd Stijn Van Raemdonck
Faculty of Applied Engineering
University of Antwerp
Antwerp, Belgium
stijn.vanraemdonck@student.uantwerpen.be

3rd Siegfried Mercelis
IDLab - Faculty of Applied Engineering
University of Antwerp - imec
Antwerp, Belgium
siegfried.mercelis@uantwerpen.be

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899662

***Abstract*—Advances in the field of NeuroEvolution (NE) highlighted the potential of applying genetic and evolutionary mechanisms to Machine Learning (ML) problems while simultaneously alleviating the need for manual neural architecture design. The NeuroEvolution of Augmenting Topologies (NEAT) is a prominent NE method shows competitive performance in the field of reinforcement learning while its performance is not as efficient if we apply it to supervised multi-class problems. This may be attributed to the multiple objectives of the problem which hinders the learning process in NEAT. In this study, we introduce a novel method C-NEAT which is developed to address this issue and enhance the performance of NEAT without changing its core implementation. C-NEAT seeks to learn and classify different classes by creating and using a container holds the best genomes i.e. networks where each one of them focuses on learning a specific class label of the problem. Each organism in NEAT which is a unit contains the evolved genome and its fitness value is assigned automatically to a specific class based on its index in the population. During the evolution process, the container will keep updating itself and store the best evolved genomes for these classes. Through this, C-NEAT will focus on recognizing each class label and it will preserve the learning progress, thus ensuring higher learning efficiency.**

***Index Terms*—Supervised Learning, NEAT, Genetic Algorithms, Automated Architecture**

I. Introduction

Machine learning (ML) has gained attention in recent years mainly due to its wide use in different domains as it focuses on building models to solve complex problems without any explicit instructions [1]. The field of ML can be clustered into three main groups: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, models are trained using labelled data. In essence, for every set of inputs, the desired set of outputs is known in advance, while in reinforcement learning (RL), the models or agents associate the environmental conditions with the received rewards to learn the desired behavior.

Research has demonstrated that the prediction capabilities of Artificial Neural Networks (ANNs) are greatly affected by their topologies and architectures [2]–[5]. Consequently, using complex ML models can be computationally expensive to achieve optimal performance, whereas the same problem can be solved with a less complex model.

These architectures are usually designed based on human experience, and it can be automated using Network Architecture Search (NAS) [6]. Nevertheless, conventional NAS methods require substantial amounts of computing resources. As a result, concerns have been raised regarding the environmental impact of these methods [7]. Graph Neural Networks (GNNs) might provide lower computational costs as they are often far more sparsely connected compared to Fully Connected Neural Networks (FCNNs). Contrary to FCNNs, each node in a GNN can connect to any number of other nodes. This further complicates the design process of GNN architectures, making them increasingly difficult to be designed by humans. Complex algorithms are often used to determine the optimal topology of these GNNs. These algorithms include evolutionary algorithms, gradient-based algorithms, random search, etc [6].

In this study, we will focus on NEAT which is an evolutionary algorithm that evolves both topology and network parameters and it belongs to Topological and Weight Evolving Artificial Neural Networks (TWEANNs) methods. NEAT method serves as a promising candidate to solve the challenges of conventional NAS methods due to its effectiveness at finding sparsely connected network solutions.

Using a container for NEAT will be beneficial to maintain the comprehensive trained behavior and handle the imbalanced datasets as we explain that in the result section VI.

We will test NEAT & C-NEAT and other traditional fully connected neural networks FCNNs on multi-class tasks to predict the correct class label corresponding to the feature values.

II. NEAT

NEAT generates a population of organisms, each organism has a genome with a specific evolved topology that represents a candidate neural network to solve the task. For convergence and to maintain the best performance, at every generation or epoch, the best genomes will be elected to the next generation.

The procedure for updating the architecture and network parameters in NEAT is based on genetic operations. These operations are briefly, Crossover, where two parent genomes produce offspring by combining their genes or connections in a meaningful way. Mutation, which is considered to be a random process that can alter the fitness value of the affected organism by adding a new element to the network and it can be a new

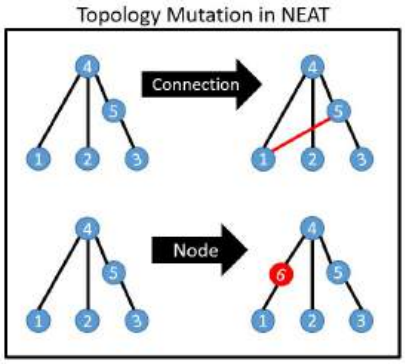


Fig. 1. The two types of structural mutation in NEAT.

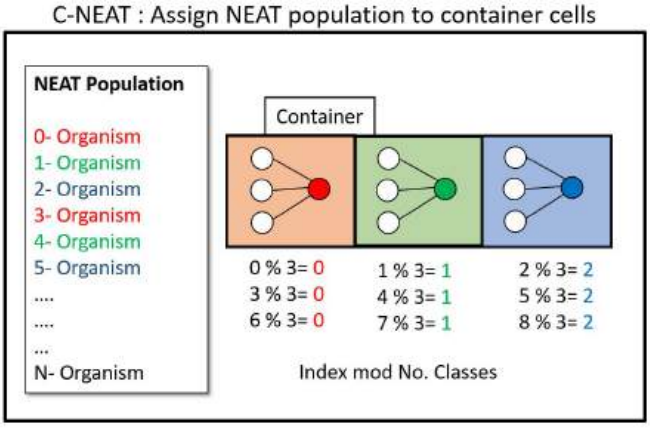


Fig. 2. The process of automatically assigning organisms and placing them in the container.

node or a new connection as shown in figure 1. NEAT also introduces several novel ideas including Historical Markers and Speciation. These innovations were demonstrated to be essential in explaining the performance difference compared to previous NE methods. Where Historical Markers works as a solution to the Competing Conventions Problem [8], and Speciation, allows groups of closely related organisms to evolve as a whole and increases the viability of offspring by only allowing mating to happen between organisms belonging to the same species. [4] [9].

NEAT tends to decrease the average number of evaluations because it explores the search space by starting with a minimal structure. This strategy helps to find less complex solutions.

The minimal structure refers to a simple dense network with full connection between inputs & outputs without having hidden layers or hidden neurons.

III. Related Work

In the literature, there are several attempts to apply NEAT on supervised multiclass tasks like the study of Chen [2] where he introduces L-NEAT and applies backpropagation on the best elected genomes. Nevertheless, having multiple separated steps for optimization may increase the complexity of the solution and increase the potential for human intervention. On the other hand, there are other approaches try to integrate NEAT with other approaches and benefit from the population of organisms to create a kind of a major vote for classification like the Ensemble based on NEAT in the study of Pimenta [10], Other studies place some controls on the updating procedures such as NEAT with Difference-Based Mutation (NEAT-DBM), however the results are still comparable to the standard NEAT [11]. In our approach, we try to enhance NEAT's performance using the same implementation of standard NEAT without combining it with other approaches and we won't add any human intervention to the process of optimization.

IV. Methodology

As previous studies have pointed out, there are multiple problems in terms of efficiency when applying NEAT directly to classification tasks [2]. In this study, we focus on three of these shortcomings.

1) The complexity and the size of the search space increase with the number of class labels. NEAT can be inefficient at finding solutions to such complex problems.
2) NEAT might be biased towards evolving networks in favor of specific classes in imbalanced datasets.
3) Search in NEAT strives to optimize the overall fitness value. This means that mutations could negatively impact the classification performance of certain class labels while still improving the overall fitness value, thereby losing efficiency by forgetting previous progress.

Our method C-NEAT is a NEAT-based method with a container developed to increase the performance on multiclass classification tasks. It strives to solve the problems associated with NEAT by simplifying the problem and assigning the organisms of the population equally among the classes to find the optimal sub-networks that predict one class label each.

The best genome with the highest ability to recognize its assigned class will be stored in the container. Controlling the designation and the storing processes are done automatically by using the indexes of the organisms in the population, where The size of the container refers to the number of class labels, and the organisms are placed in the container based on the modulo of their index to the container size as shown in the equation below 1 and in figure 2

$$C[Org_{indx} \mod C_{size}] = Org \tag{1}$$

where, C refers to the container and Org to the organism.

This approach should, in theory, achieve better classification results compared to NEAT as the number of class labels in the dataset increases.

During the evaluation in C-NEAT, each organism is evaluated based on its ability to accurately predict the class label corresponding to the given set of inputs. The loss function of the first organism will focus on the first class label, the loss of the second organism will depend on its ability to predict the second class label, etc. After reaching the last class label, we revert to the first class label as shown in figure 2.

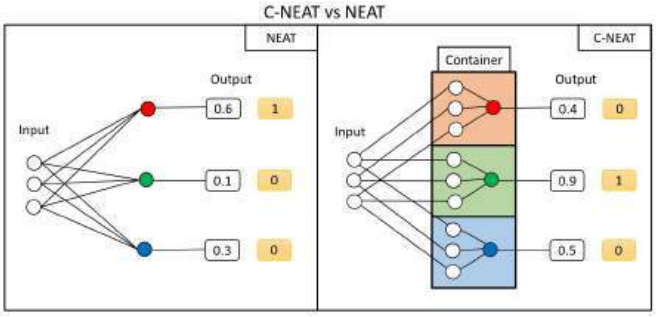


Fig. 3. Prediction and classification between NEAT & C-NEAT

For each of the possible class labels, the assigned genome with the lowest loss value is stored in a container and it is updated every time a better genome is found. However, the evolutionary process of NEAT requires a fitness value for each of the genomes in the population in order to perform natural selection. We used the negative of the loss value as a fitness function. When C-NEAT is terminated, the container will contain the best-performing ANNs for each class label. To make a single prediction, all of these networks have to collaborate. The class label corresponding to the network with the highest output value given a set of input features will be the class label predicted by the system.

For final classification, The class labels are represented by using one-hot encoding vectors. For NEAT, the output is a single genome i.e. a neural network with the highest overall performance, therefore, the output layer of this genome will refer to that one-hot encoding. whereas in C-NEAT, the output will be the container itself and every cell in the container focuses on one class, thus, from this container and after distributing the input features among its cells we can make one-hot encoding vector to determine the predicted class as shown in figure 3.

V. Experimental Setup

The two popular datasets Iris & Yeast were used to compare the performances. Iris dataset [12] contains four features for classifying three different iris flower species. This dataset is balanced with size 150 instances in which there are 50 instances for each class. Yeast dataset [13] contains 1484 instances of the localisation sites of proteins. Contrary to Iris dataset, Yeast dataset is imbalanced and it has an unequal number of instances for each class label. Furthermore, this dataset is more complicated as it contains eight features for classification with ten different possible class labels for localization site. The input features of both datasets were normalised within a range [0..1].

The experiments were conducted using a C++ implementation of NEAT where the topologies are implemented based on pointer mechanism to deal with memory usage. We used 100 generations for evolving the ANNs and the rest of NEAT Parameters are as described in the original paper of NEAT [4].

We compared NEAT and C-NEAT with three different FCNNs with fixed architecture that use backpropagation for learning. The FCNN architecture can have up to two hidden layers with 8 or 16 neurons per layer. For building FCNNs we used Scikit-learn [14] which is an open source ML library in Python. We conducted 10 different runs for each experiment and the results are displayed using box plots.

VI. Result and Discussion

We build two scenarios to test the performance. We train each method on two different percentage of dataset i.e. 10% & 66%. The reason behind this is two check the generalisability and the performance sustainability between these two percentages.

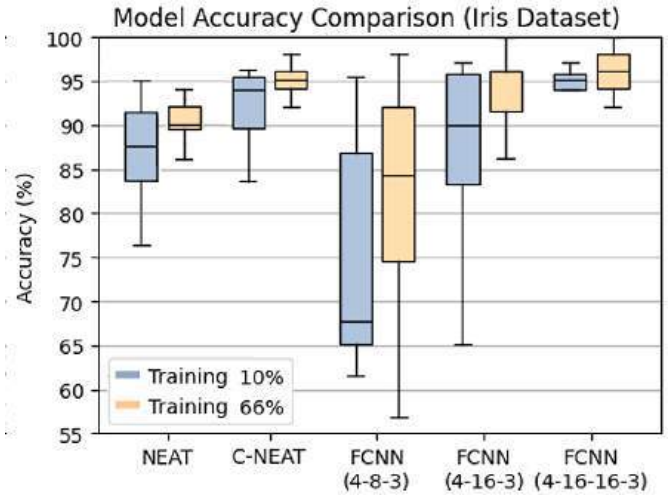


Fig. 4. Classification accuracy of NEAT, C-NEAT, and FCNNs using different hidden layers on the Iris dataset.

As demonstrated in figure 4, C-NEAT achieves a higher classification accuracy on Iris dataset than NEAT. In addition, C-NEAT compares favourably to most of FCNN architectures apart from the largest one with two hidden layers of 16 neurons each. However, C-NEAT and NEAT achieved comparable classification results, which can be explained by the low number of different class labels.

After testing on Yeast dataset, which is imbalanced and more complex than Iris dataset, the outperformance of C-NEAT compared to NEAT becomes more distinct as shown in figure 5. This can be attributed to the advantage of using a container in our C-NEAT method as the container helps to sustain high performance on imbalanced datasets by assigning a specific genome i.e. a neural network to each class label unlike the standard NEAT. Besides, the greater number of class labels in the Yeast dataset makes the problem more challenging to be solved by a single genome that mainly uses genetic operations with a certain degree of uncertainty for optimization.

As illustrated in figure 5, NEAT-based methods seek to generalize and sustain its performance between the two experimental setup scenarios of training dataset 10% & 66%. The high performance of C-NEAT on these scenarios shows the efficiency of our method on limited datasets for training.

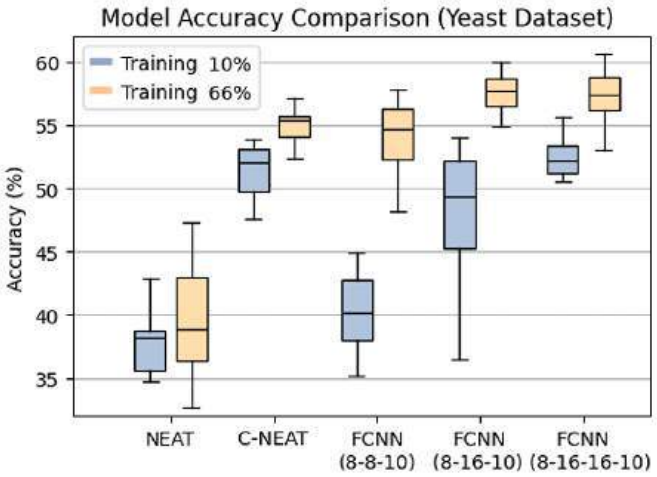


Fig. 5. Classification accuracy of NEAT, C-NEAT, and FCNNs using different hidden layers on the Yeast dataset.

TABLE I
COMPARISON OF THE NO. CONNECTIONS BETWEEN THE DIFFERENT METHODS.

Method	# Connections (Iris)	# Connections (Yeast)
NEAT	27	103
C-NEAT	52	140
FCNN (8)	67	162
FCNN (16)	131	314
FCNN (16-16)	403	586

There is also an interesting behavior of FCNNs in figure 4 where more complex models lead to better performance. Studies have attributed this to the presence of a 'double-descent' effect. [15], which tells us that there is a peak in the test error of ML problems when the number of data points equals the number of parameters of the ANN. Furthermore, Zhang et al. [16] show that highly over-parameterised neural networks, that is, networks with more parameters than training instances, generalise well on unseen test data unlike the fewer parameterised neural networks. However, our results indicate that C-NEAT do not suffer from the double-descent effect to the same degree as the FCNN-based models because the architecture in our method is build automatically and evolved based on the complexity of the problem, and on the other hand, the learning progress of each class label is preserved by using the container, and that explains the excellent test performance using a training set percentage of 10% in contrast to the two smallest FCNN models.

Another important factor when comparing the classification performance of different models is the network size. For the NEAT-based methods, the No. connections was averaged over 10 runs, while the No. connections remains constant for all FCNN strategies. As denoted in Table I, NEAT achieved the lowest No. connections on both datasets. C-NEAT achieved a close No. connections to the FCNN with eight hidden neurons on both the Iris and Yeast datasets. These results indicate that C-NEAT requires more connections than NEAT. This can be explained by the fact that both of these methods evolve neural networks, contrary to NEAT which evolves only one network as output, C-NEAT has a container represents the final output and it holds the best networks per class 3.

Although NEAT, C-NEAT start with the same minimal structure, each sub-network in the container of C-NEAT can accumulate new connections over time through mutations. In NEAT, only one network can receive extra connections over time, leading to a lower average No. connections compared to C-NEAT.

It is important to note that during the training stage, the total number of connections is not the sole determinant of computational complexity. For instance, in FCNNs, We have forward data for prediction and backward data for learning, and training a neural network using backpropagation differs from evolving a population using evolutionary principles. Moreover, applying mini-batch learning [17] to NEAT-based methods requires some computational power as each genome in the ensemble must be tested on the same parts of dataset. But just for simplicity we focus on the No. connections of the produced solution.

VII. CONCLUSION

In this study, we investigate the feasibility of using an evolutionary method, i.e. NEAT on supervised multi-class tasks. The purpose of this research is to introduce an automated system which minimizes human intervention by utilizing the unique characteristics of NEAT such as automatically building models' architectures with complexity comparable to the difficulty of the problem and taking advantage of its high computational efficiency. Our method does not change the core implementation of standard NEAT as it still uses genetic operations for optimization, but it deals with the evolutionary learning progress in more efficient way by using a container. The container plays an important role for enhancing the classification performance by simplifying the whole search space of the problem into multiple search spaces and optimizing sub-networks using only one population where each network is focusing on learning a single specific class. Our results demonstrate that C-NEAT outperforms both NEAT and some FCNNs with fixed architectures on multi-class classification tasks. It achieves superior classification results on imbalanced datasets and displays enhanced generalisability with less complexity and a low No. connections.

C-NEAT can easily be implemented using an already existing implementation of NEAT, and can be useful on applications in classification problems where data and computational resources are scarce. Moreover, C-NEAT serves as an example demonstrating the successful application of NE algorithms in the field of supervised learning.

REFERENCES

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 5 2020.

[2] L. Chen and D. Alahakoon, "Neuroevolution of augmenting topologies with learning for data classification." IEEE, 12 2006, pp. 367–371.
[3] A. Wijaya, D. Ikawahyuni, R. Gea, and F. Maedjaja, "Role comparison between deep belief neural network and neuroevolution of augmenting topologies to detect diabetes," *JOIV : International Journal on Informatics Visualization*, vol. 5, 5 2021.
[4] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, pp. 99–127, 6 2002.
[5] A. Gaier and D. Ha, "Weight agnostic neural networks," 6 2019.
[6] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," 8 2018.
[7] E. Gibney, "How to shrink ai's ballooning carbon footprint," *Nature*, vol. 607, pp. 648–648, 7 2022.
[8] P. R. Neary, "Competing conventions," *Games and Economic Behavior*, vol. 76, no. 1, pp. 301–328, 2012.
[9] M. Y. Ibrahim, R. Sridhar, T. Geetha, and S. Deepika, "Advances in neuroevolution through augmenting topologies – a case study." IEEE, 12 2019, pp. 111–116.
[10] G. A. Pimenta, F. B. J. R. Dallaqua, A. Fazenda, and F. A. Faria, "Neuroevolution-based classifiers for deforestation detection in tropical forests," 8 2022.
[11] V. Stanovov, S. Akhmedova, and E. Semenkin, "Neuroevolution of augmented topologies with difference-based mutation," *IOP Conference Series: Materials Science and Engineering*, vol. 1047, p. 012075, 2 2021.
[12] R. A. Fisher, "Iris," UCI Machine Learning Repository, 1988, DOI: https://doi.org/10.24432/C56C76.
[13] K. Nakai, "Yeast," UCI Machine Learning Repository, 1996, DOI: https://doi.org/10.24432/C5KG68.
[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
[15] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," 12 2019.
[16] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, pp. 107–115, 3 2021.
[17] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2002, pp. 9–50.

ConDVC: Bridging Visual and Semantic Spaces with Key Semantics for Video Understanding

1st Jianxiong Wang
School of Artificial Intelligence
South China Normal University
Foshan, China
wangjianxiong@m.scnu.edu.cn

2nd Jing Xiao*
School of Computer Science
South China Normal University
Guangzhou, China
xiaojing@scnu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899553

***Abstract*—Dense video captioning task aims to understand unsegmented video content for accurate event localization and captioning. Recent studies have focused on leveraging inter-task relationships to address this task. However, the complexity of cross-modal learning between video content and captions, particularly without prior knowledge guidance, presents significant challenges in jointly handling these two tasks. This paper introduces a Concept-Guided Dense Video Captioning framework (ConDVC) that uses concepts, i.e., key elements such as objects and actions, as a bridge linking visual and semantic spaces. By employing video-to-text retrieval to gather textual features and integrating these with multimodal features for concept detection, we utilize these concepts as semantic guides during the event matching process. This approach not only provides additional prior information for optimizing both subtasks, event localization and caption generation, but also leverages the prior knowledge capabilities of pretrained models like CLIP to enhance overall model performance. Extensive experiments on the YouCook2 and ActivityNet Captions datasets demonstrate the superiority of ConDVC against state-of-the-art methods without extra data for pretraining.**

***Index Terms*—Dense video captioning, Concept detection, Video-to-text retrieval**

I. Introduction

Significant advances [1]–[5] have been made in video captioning tasks as research in multimodal understanding deepens. Unlike traditional video captioning, which merely requires generating accurate descriptions for specific video clips, Dense Video Captioning (DVC) is more attuned to real-world scenarios and confronts more formidable challenges. It necessitates identifying multiple events within unedited, lengthy videos and crafting coherent natural language descriptions. In past research [6], [7], modeling the two subtasks of event localization and caption generation within DVC was a primary focus. Frameworks previously pioneered the development of end-to-end, parallel event detection methods. These methods [8], [9] utilized interactions between tasks to achieve precise event localization and caption generation, garnering broad interest. Concurrently, notable successes [10] in cross-modal studies between vision and language have enhanced pre-existing knowledge for visual understanding. Nevertheless, the inherent temporal and semantic complexities of full-length videos, which often involve processing sequences with overlapping or closely spaced events, pose ongoing challenges. Effectively harnessing and deploying prior knowledge to bridge videos with natural language and refining the synergy between localization and descriptive tasks remains an arduous endeavor.

There are some video captioning methods [4], [5], [11] that incorporate prior information, such as the introduction of relevant concepts related to video content. Concepts, as carriers of key information like actions and objects, align more easily in both the visual-semantic joint space and the purely semantic space. Accordingly, in this paper we aim to guide task interaction on a semantic level within Dense Video Captioning (DVC) tasks using explicit prior information, i.e. concepts. When employing concepts as the core of semantic guidance, two aspects should be considered: Firstly, the performance of concept detection is limited by the inherent complexity of video data. Detecting corresponding concepts solely through visual information will limit the detection accuracy and generate ambiguity. Secondly, we need to explore how to further utilize concepts as prior information. In previous DVC methods [8], [9], the introduction of prior information only serves as a supplement to visual features. The event features enhanced by information implicitly affect task interaction in the framework, but there is no more effective use of prior information in the process of event matching, localization, and description.

Based on these insights, this paper proposes a Concept-Guided Dense Video Captioning framework. To alleviate potential ambiguities in concept detection with a single modality, we supplement text information by video-to-text retrieval. We employ the CLIP model to effectively bridge visual and semantic spaces. CLIP, by learning contrastive relationships between images and text, tightly links visual content with semantic information. In this study, CLIP is used to retrieve textual clues relevant to the video content. These textual clues not only provide the model with prior knowledge but also reduce ambiguities in concept detection by combining with visual information. Based on the end-to-end parallel decoding framework, visual and text features are fused through the encoder-decoder structure. The event features derived from

*Corresponding author.

the decoder enable us to extract concept information from the video through multimodal concept detection. In our framework, concept detection and event localization are parallel processes. During the matching process, conceptual information serves as semantic supervision, mutually promoted with the event localization. For the related concepts that we have detected, we extract common semantic information through the attention module. This information, combined with the temporal difference information derived from the context features, forms a set of cross-modal complementary information.

Our contributions are as follows: (1) We propose a novel multimodal concept detection and application framework for the DVC task. This framework uses concepts to guide the cross-modal alignment of video events, promoting interaction between event localization and caption generation tasks. It also utilizes concepts as prior information to influence captioning. (2) We design an attention module capable of fusing common concept information with temporal discrepancy information from the context into cross-modal complementary information. This fusion guides word prediction and enhances subtitle generation performance. (3) Extensive experiments on the YouCook2 [6] and ActivityNet Captions [12] datasets validate the effectiveness of our concept-guided framework. ConDVC surpasses state-of-the-art (SOTA) in most metrics compared to methods without additional data.

II. Related Work

A. Dense Video Captioning

Dense video captioning involves event localization and captioning. Krishna et al. [6] introduced the "localize-then-describe" approach, but it treated the tasks independently. To address this, joint learning methods [8], [9], [13], [14] have been proposed. Zhou et al. [13] linked caption loss with event boundaries, while Yang et al. [14] incorporated speech features and pretraining datasets. Wang et al. [8] applied the DETR approach for task interaction, and Kim et al. [9] used memory-based retrieval. Our work enhances both localization and captioning by introducing concepts as prior information for better event matching.

B. Retrieval-Enhanced Captioning

Text retrieval has proven effective in video captioning.Chen et al. [1] used retrieved captions as key-value memories in an encoder-decoder framework. Kim et al. [9] integrated retrieved sentences via cross-attention for joint task learning. Our approach leverages multimodal event features to improve concept detection and extract key semantic information.

C. Concept Detection

Concept detection has been widely studied in video captioning. Xu et al. [4] and Chen et al. [11] used audio-visual content to predict concepts, and Yang et al. [5] incorporated text from video-to-text retrieval. We extend this by using concepts as semantic costs to guide event matching, addressing both event localization and captioning in dense video captioning tasks.

III. Method

We present the overall structure of our proposed ConDVC model in Fig.1. In the following sections, we present the derivation of multimodal event features via video-to-text retrieval, the mechanics of concept detection, and how these concepts guide event matching and caption generation in detail.

1) Video-to-Text Retrieval.: We sample the video at 1 frame per second (fps) and use CLIP's [10] image encoder to extract visual features. For text, we follow a similar approach, treating frame-level features as the smallest unit. To maximize the capabilities of CLIP, we augment each sentence in the training corpus with a dataset-specific prompt (e.g., "an image describing the food-making process: " for Youcook2). We derive embeddings for all modified sentences and compute cosine similarity with visual features. The top N_R most relevant sentences (excluding those from the same video) are selected, normalized, and weighted to obtain text features for each frame.

2) Feature Integration.: After retrieving visual and text features at the frame level, we resample the feature map to a fixed size T. To capture multi-scale information, we apply L temporal convolution layers and extract inter-frame relationships using a deformable transformer encoder [8], [15]. In the decoder, N event queries retrieve event features from the frame-level features, iterating through decoding layers to output multi-scale event features $\widetilde{\mathbf{q}}_j$.

A. Concept Detection

We design a concept detector based on multimodal event features $\widetilde{\mathbf{q}}_j$ to identify relevant concepts within events. First, we select N_c frequent nouns, verbs, and adjectives from the training corpus to create a key concept vocabulary $\mathbf{V}c$. If a caption contains any of these concepts, they are marked as 1 in the pseudo label $\mathbf{C} = c_1, c_2, \ldots, c_{N_C}$, indicating the concepts present in the video segment. The concept detector uses the CLIP text encoder to obtain text embeddings $\mathbf{Emb}_{V_c}$ of the vocabulary, then calculates cosine similarity between $\widetilde{\mathbf{q}}_j$ and $\mathbf{Emb}V_c$ to produce a preliminary distribution. A multi-layer perceptron (MLP) then refines this into a concept probability distribution:

$$\hat{\mathbf{P}}_{c,j} = \text{MLP}\left(\text{sim}\left(\widetilde{\mathbf{q}}_j, \mathbf{Emb}_{V_c}\right)\right) \tag{1}$$

where $\hat{\mathbf{P}}_{c,j}$ represents the concept distribution. Due to the large vocabulary and sparse positive samples, KL divergence is used as the loss function.The loss for concept detection, denoted as L_{cpt}:

$$L_{\text{cpt}} = D_{\text{KL}}(\hat{\mathbf{P}}_{c,j} \parallel \mathbf{C}_j) = \sum_{i=1}^{N_c} c_{i,j}(\log c_{i,j} - \log \hat{p}_{i,j}) \tag{2}$$

where $\mathbf{C}_j$ represents the one-hot vector corresponding to the concepts in the j-th ground-truth caption, $c_{i,j}$ is the i-th element of $\mathbf{C}_j$, and $\hat{p}_{i,j}$ is the i-th element of the predicted distribution $\hat{\mathbf{P}}_{c,j}$.

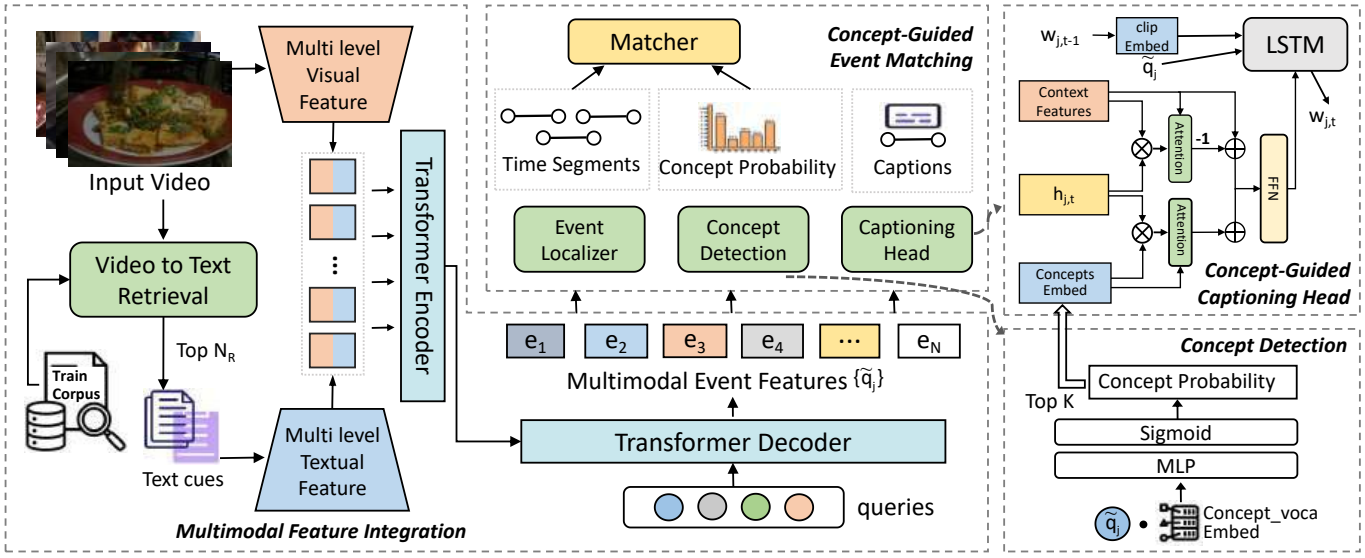


Fig. 1. Overview of the proposed ConDVC. Our method employs video-to-text retrieval to extract relevant textual information and integrates it with visual data within an encoder-decoder framework (Section 3.1). The concept probabilities derived from the concept detection (Section 3.2) are considered as semantic costs and combined with localization costs to enhance event matching (Section 3.3). The Captioning Head illustrates how the top K relevant concepts, fused with contextual temporal features through an attention mechanism, influence the caption generation (Section 3.4).

It is important to note that obtaining the concept probability distribution through the concept detector constitutes a multi-label classification task. We integrate this as a detection head, rather than as a bypass or offline component of the model. Furthermore, in the text generation head, we separate the extraction of the top K concepts from the loss calculation associated with the classification task.

B. Concept guided event matching

In ConDVC, we leverage a readily accessible, static information-concept that exhibits common attributes across modalities, providing semantic guidance to enhance matching performance. During the Hungarian matching [16] process for each predicted event, we combine semantic costs with the original localization L_{giou} and classification costs L_{cls} ,the combined cost, denoted as S, is computed as follows:

$$S = \alpha_{\text{cpt}} L_{\text{cpt}} + \alpha_{\text{cls}} L_{\text{cls}} + \alpha_{\text{giou}} L_{\text{giou}} \tag{3}$$

where α are weighting factors for the corresponding loss, and L_{cpt} represents the KL divergence loss between the predicted concepts probability and the ground-truth distribution. This approach measures the matching degree between predicted events and ground-truth events from both semantic and temporal dimensions enhances the robustness of matching at the semantic level, improves the training process.

C. Concept-Guided Captioning Head

We employ a concept-guided deformable soft attention LSTM for word prediction. In addition to sampling relevant points around the reference point as contextual features, we incorporate concept features to influence word selection at each time step. By leveraging CLIP's prior knowledge, we map event features and text at each step into a shared joint space.

Specifically, the top K event-related concepts are selected and mapped into the joint space using CLIP's text embeddings. Points around the event query $\widetilde{\mathbf{q}}j$ provide temporal context, while event concepts supply semantic information. Context features form $\mathbf{K}$ctx and $\mathbf{V}$ctx, and the hidden state $\mathbf{h}j_t$ serves as $\mathbf{Q}$. Attention between $\mathbf{Q}$ and $\mathbf{K}$ctx, weighted by $\mathbf{V}$ctx, yields shared information. By subtracting this, we obtain the discrepancy information $\mathbf{DI}_{\text{ctx}}$.

$$\mathbf{DI}_{\text{ctx}} = \mathbf{V}_{\text{ctx}} - \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}_{\text{ctx}}^T}{\sqrt{d_k}}\right)\mathbf{V}_{\text{ctx}} \tag{4}$$

Additionally, the top K concepts are treated as $\mathbf{K}_{\text{cpt}}$ and $\mathbf{V}_{\text{cpt}}$, and attention calculations between $\mathbf{Q}$ and $\mathbf{K}_{\text{cpt}}$ are used to obtain semantic common information $\mathbf{CI}_{\text{cpt}}$ relevant to the current timestep.

$$\mathbf{CI}_{\text{cpt}} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}_{\text{cpt}}^T}{\sqrt{d_k}}\right)\mathbf{V}_{\text{cpt}} \tag{5}$$

To obtain cross-modal complementary information $\mathbf{F}_{\text{cmi}}$, we inject the temporal discrepancy information $\mathbf{DI}_{\text{ctx}}$ into semantic commonalities $\mathbf{CI}_{\text{cpt}}$, which can be formulated as:

$$\mathbf{F}_{\text{cmi}} = \text{MLP}(\mathbf{CI}_{\text{cpt}} + \mathbf{DI}_{\text{ctx}}) \tag{6}$$

The LSTM cell then takes the cross-modal information $\mathbf{F}_{\text{cmi}}$, the event feature $\widetilde{\mathbf{q}}_j$, and the CLIP-encoded embedding of the previous word $\hat{\mathbf{w}}_{j,t-1}$ as inputs.

$$\hat{\mathbf{w}}_{j,t} = \text{Softmax}\left(\text{LSTM}\left([\mathbf{F}_{\text{cmi}}; \widetilde{\mathbf{q}}_j; \text{Emb}(\hat{\mathbf{w}}_{j,t-1})], \mathbf{h}_{j,t}\right)\right) \tag{7}$$

As the selection of words progresses at each timestep, the entire sentence $\mathbf{S}_l$ is gradually constructed.

TABLE I

EVENT LOCALIZATION PERFORMANCE IN YOUCOOK2 AND ACTIVITYNET CAPTIONS. SCORES DISPLAYED IN BOLD REPRESENT THE HIGHEST ACHIEVED IN METHODS WITHOUT PRETRAINING, WHILE THOSE UNDERLINED INDICATE THE SECOND HIGHEST. PT DENOTES PRETRAINING WITH EXTRA DATA. † RESULTS ARE REPRODUCED USING THE OFFICIAL IMPLEMENTATION IN OUR ENVIRONMENT.

Method	Backbone	PT	Youcook2(val)			ActivityNet(val)		
			F1 Score	Recall	Precision	F1 Score	Recall	Precision
Vid2Seq [14]	CLIP	✓	27.84	27.90	27.80	53.29	52.70	53.90
PDVC† [8]	CLIP	-	26.69	22.71	32.38	54.79	53.52	56.14
CM2 [9]	CLIP	-	<u>28.43</u>	<u>24.76</u>	<u>33.38</u>	<u>55.21</u>	<u>53.71</u>	<u>56.81</u>
ConDVC (Ours)	CLIP	-	**29.84**	**26.35**	**34.38**	**55.97**	**54.95**	**57.04**

TABLE II

EVENT CAPTIONING PERFORMANCE IN YOUCOOK2 AND ACTIVITYNET CAPTIONS. REFER TO THE NOTES IN TABLE 1

Method	Backbone	PT	Youcook2(val)				ActivityNet(val)			
			BLEU4	METEOR	CIDEr	SODA_c	BLEU4	METEOR	CIDEr	SODA_c
Vid2Seq [14]	CLIP	✓	-	9.30	47.10	7.90	-	8.50	30.10	5.80
MT [13]	TSN	-	1.15	5.00	9.30	-	0.30	3.20	6.10	-
ECHR [7]	C3D	-	-	3.82	-	-	1.82	7.20	14.70	3.20
E2ESG [17]	C3D	-	-	-	-	-	-	3.50	25.00	-
PDVC [8]	TSN	-	0.80	4.74	22.71	4.42	1.78	7.96	28.96	5.44
PDVC† [8]	CLIP	-	1.42	5.53	28.30	4.97	2.25	8.10	29.83	5.89
CM2 [9]	CLIP	-	<u>1.63</u>	<u>6.08</u>	<u>31.66</u>	<u>5.34</u>	**2.38**	**8.55**	**33.01**	<u>6.18</u>
ConDVC (Ours)	CLIP	-	**1.87**	**6.27**	**36.42**	**6.06**	<u>2.28</u>	<u>8.51</u>	<u>32.89</u>	**6.30**

TABLE III

COMPARISON OF PERFORMANCE ACROSS VARIOUS COMPONENTS ON THE YOUCOOK2 DATASET, WHICH INCLUDES VIDEO-TO-TEXT RETRIEVAL (V2T), CONCEPT-GUIDED EVENT MATCHING (CEM), AND CONCEPT-GUIDED EVENT CAPTIONING (CEC). THE FIRST ROW SHOWS THE RESULTS OF THE BASELINE, $\text{PDVC}_{\text{CLIP}}$. IN THE RESULTS OF ROWS 2-6, WE USE THE CLIP TEXT ENCODER FOR WORD EMBEDDINGS IN THE CAPTIONING HEAD.

Components			Performance					
V2T	CEM	CEC	BLEU4	METEOR	SODA_c	CIDEr	Recall	Precision
-	-	-	1.42	5.53	4.97	28.30	22.71	32.38
×	×	×	1.55	5.66	5.34	31.62	22.65	32.49
✓	×	×	1.74	5.86	5.45	33.40	23.72	32.78
×	✓	✓	1.78	6.04	5.64	34.00	24.68	33.64
✓	✓	×	1.59	6.05	5.42	34.35	23.92	34.55
✓	✓	✓	1.87	6.27	6.06	36.42	26.35	34.38

IV. EXPERIMENTS

A. Experimental Settings

1) Datasets.: We utilize two benchmark dense video captioning benchmark datasets, ActivityNet Captions and YouCook2, for training our model and evaluating its effectiveness. ActivityNet Captions comprises 20,000 untrimmed videos depicting a variety of human activities. Each video averages 120 seconds in length and includes 3.7 temporally localized sentences. We follow the standard split with 10,009 videos for training and 4,917 for validation. The YouCook2 dataset features 2,000 untrimmed videos showing cooking procedures, each averaging 320 seconds and annotated with 7.7 temporally localized sentences. We adhere to the standard split with 1,237 videos for training and 436 for validation.

2) Evaluation metrics.: We conduct an assessment of our approach for two specific tasks within dense video captioning. We scrutinize the generated captions using key metrics including CIDEr [18], BLEU4 [3], and METEOR [19]. Furthermore, to evaluate the storytelling proficiency, we apply the SODA_c [20] metric. In terms of event localization, our evaluation involved measuring average precision, average recall, and the F1 score, which provides a balanced assessment of precision and recall.

3) Implementation Details.: For the video data, we sample frames at 1 FPS and used $\text{CLIP}_{\text{Vit-L/14}}$ [10] as the backbone to extract frame-level features. In Hungarian matching, the cost ratios are $\alpha_{\text{cpt}} : \alpha_{\text{giou}} : \alpha_{\text{cls}} = 2 : 2 : 1$. we retrieve $N_R = 20$ captions from the training corpus for each frame and use the $\text{CLIP}_{\text{Vit-L/14}}$ text encoder to extract text features.We create a concept vocabulary by selecting the top N_c most frequently occurring nouns, verbs, and adjectives from each dataset, with N_c set to 800 for YouCook2 and 4000 for ActivityNet. To limit the number of relevant concepts, we set top K to 8.

B. Comparison with State-of-the-Art Methods

1) Localization Performance.: The event localization results of state-of-the-art approaches using CLIP as the backbone

are shown in Table 1. In ActivityNet Captions, we achieved the best scores across all three metrics. In YouCook2, our method achieves the best scores in precision and F1 score, and second to VidSeq [14] in the recall metric. This indicates that conceptual information as semantic costs can improve the effectiveness of bipartite matching, thereby enhancing task interaction and improving localization performance.

2) Event Captioning Performance.: In Table 2, we evaluate the caption performance of the leading models on two datasets. In the YouCook2 dataset, our method is second only to VidSeq, which utilizes additional training data. In addtion, we achieve the best scores among models without extra data across the BLEU4, METEOR, CIDEr, and SODA_c metrics. The improvement over the baseline (PDVC) is notable. Comparing to the SOTA method (CM2), our method shows a significant enhancement of 14.87%/3.10%/15.03%/13.53%. On the ActivityNet Captions dataset, our results are close to CM2 in BLEU4, METEOR, and CIDEr, and surpass it in the SODA_c metric. These outcomes validate that the introduction of concepts can enhance the model's understanding and depiction of crucial content and overall scenes.

3) Ablation Study.: To assess the effectiveness of each component in ConDVC, we conducted ablation experiments on the YouCook2 dataset, with the results summarized in Table 3. Our approach focuses on detecting concepts through multimodal concept detection and leveraging these concepts to guide event matching and caption generation. The main components include frame-level visual feature-based text retrieval, concept-guided event matching, and the integration of conceptual information during caption generation. We controlled the use of these components to validate their contribution.

Comparing the first two rows, we observe that feeding $w_{j,t-1}$ into the captioning process via the CLIP text encoder results in a noticeable performance boost. At this stage, adding text retrieval alone does not significantly enhance performance, as both methods incorporate prior knowledge that plays a similar role in subtitle generation. The impact of video-to-text retrieval becomes more pronounced when combined with concept detection, aligning with our hypothesis that text features complement visual ones, thereby improving concept detection and providing a stronger semantic foundation for event matching and caption generation.The integrating conceptual information further improves model performance.

V. Conclusion

We propose ConDVC, which incorporates concepts as prior information into the DVC task to guide event matching, optimize task interaction, and improve caption generation. Extensive experiments on the YouCook2 and ActivityNet Captions datasets validate the effectiveness of our method, surpassing the state-of-the-art (SOTA) in most metrics. By anchoring semantic information within the visual context, our method not only enhances the precision of event localization and description but also contributes to a more coherent and contextually relevant narrative flow.

References

[1] J. Chen, Y. Pan, Y. Li, T. Yao, H. Chao, and T. Mei, "Retrieval augmented convolutional encoder-decoder networks for video captioning," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 1s, pp. 1–24, 2023.

[2] S. Jing, H. Zhang, P. Zeng, L. Gao, J. Song, and H. T. Shen, "Memory-based augmentation network for video captioning," *IEEE Transactions on Multimedia*, 2023.

[3] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6504–6512.

[4] Y. Xu, J. Yang, and K. Mao, "Semantic-filtered soft-split-aware video captioning with audio-augmented feature," *Neurocomputing*, vol. 357, pp. 24–35, 2019.

[5] B. Yang, M. Cao, and Y. Zou, "Concept-aware video captioning: Describing videos with effective prior information," *IEEE Transactions on Image Processing*, 2023.

[6] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 706–715.

[7] T. Wang, H. Zheng, M. Yu, Q. Tian, and H. Hu, "Event-centric hierarchical representation for dense video captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1890–1900, 2020.

[8] T. Wang, R. Zhang, Z. Lu, F. Zheng, R. Cheng, and P. Luo, "End-to-end dense video captioning with parallel decoding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6847–6857.

[9] M. Kim, H. B. Kim, J. Moon, J. Choi, and S. T. Kim, "Do you remember? dense video captioning with cross-modal memory retrieval," *arXiv preprint arXiv:2404.07610*, 2024.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[11] S. Chen, Q. Jin, J. Chen, and A. G. Hauptmann, "Generating video descriptions with latent topic guidance," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2407–2418, 2019.

[12] L. Zhou, C. Xu, and J. Corso, "Towards automatic learning of procedures from web instructional videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[13] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8739–8748.

[14] A. Yang, A. Nagrani, P. H. Seo, A. Miech, J. Pont-Tuset, I. Laptev, J. Sivic, and C. Schmid, "Vid2seq: Large-scale pretraining of a visual language model for dense video captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 714–10 726.

[15] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2020.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[17] W. Zhu, B. Pang, A. V. Thapliyal, W. Y. Wang, and R. Soricut, "End-to-end dense video captioning as sequence generation," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 5651–5665.

[18] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[19] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[20] S. Fujita, T. Hirao, H. Kamigaito, M. Okumura, and M. Nagata, "Soda: Story oriented dense video captioning evaluation framework," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 517–531.

Generating basic probability assignment from the view of distance measures and its application in evidential decision tree

1st Yifan Sun
College of Information Engineering,
Northwest A&F University,
Yangling, China
1fan.sun@nwafu.edu.cn

2nd Mengzhuo Zhang
College of Information Engineering,
Northwest A&F University,
Yangling, China
zhangmz@nwafu.edu.cn

3rd Xiaozhuan Gao
College of Information Engineering,
Northwest A&F University,
Yangling, China
gaoxiaozhuan@nwafu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899481

***Abstract*—In Dempster-Shafer evidence theory, basic probability assignment (BPA) plays a important role in representing uncertain and unknown information. How to generate high-quality BPA is essential, which can promote further application of evidence theory. BPA can be understood as the possibility assigned to each proposition. Distance measures, as an effective tool for quantitatively analyzing inconsistencies between samples and known information, are central to this process. Hence, this paper proposes a novel method to generate BPA by comprehensively considering Gaussian distribution and distance measures. New method is applied to evidential decision tree and its effectiveness can be verified by real world data set.**

***Index Terms*—Basic probability assignment, Distance measure, Evidential decision tree**

I. Introduction

As an extension of traditional probability theory, Dempster-Shafer Theory (DST) [1], [2] provides a broader framework for handling uncertain information. In DST, basic probability assignment (BPA) maps uncertain information to the power set of the identification framework to model imprecise and unknown information. Up to now, DST has attracted more and more attention which can be applied in some fields, such as classification task, target recognition, risk assessment etc.

Generating the hight-quality BPA is essential to apply DST into practical engineering, which can have a direct impact on the experiment results. Up to now, there are some studies about how to generate BPA. Ghafir et al propose a novel method based on the Gaussian and exponential probability density functions, the categorical probability mass function, and the local reachability density [3]. Fu et al use Adaboost to generate BPA which does not consider probability distribution of data [4].Fei et al generate BPA by using K-means method and it is extended by K–nearest neighbor (K-NN) algorithm [5]. Besides, there are some other studies about how to generate BPA, however, which are based on data-driven by building probability distribution models based on the training set. It should be pointed out that those existing methods can not consider distance between sample and known information. Distance measure can effectively quantify the differences between known information and samples. Garg and Rani apply distance into pattern recognition and clustering [6]. Hassanat et al review the specific applications of Hassanat distance metric in supervised and unsupervised learning [7]. It can be seen that distance measure has the better performance when those data is addressed which are contains noise and outliers.

Functionally, the BPA is used to represent uncertainty by assigning mass to subsets, and the effectiveness of this assignment determines the quality of the BPA. Therefore, by adjusting the mass function of each subset based on the distance measures to the test sample, the BPA can more accurately express the degree of membership of the test sample to each category. This, in turn, enhances the BPA's ability to represent uncertain information and improves the quality of generated BPA.

This paper presents a novel approach for generating basic probability assignment. First, the mean and standard deviation of each class with respect to each attribute in the training set are calculated, and Gaussian models are constructed for each attribute. These models are then employed to generate the BPA for the test set by matching the test samples to the corresponding Gaussian distributions. Next, the mean and median values of each class in the training set are computed, and the differences between the test sample and these values are used to define the distances to individual or multiple classes. These distances are subsequently aligned with the power set spatial distribution within the framework of discernment. Finally, the BPA generated from the Gaussian models is combined with the distance values through the computation of their inner product.

To further validate the advantages of the proposed method, it is applied within the context of the evidential decision tree. The performance of the evidential decision tree serves as an indicator of the effectiveness of attribute selection, as well as the quality of the BPA.

The organization of the rest of this paper is shown as fallow. Section 2 is the preliminaries. Section 3 presents the novel method of generating BPA. In section 4, proposed method is applied in the evidential decision tree. Section 5 shows the experimental results by using real world data set. section 6 concludes this paper.

II. Preliminaries

A. D-S evidence theory

(1)Framework of discernment(FoD)

The framework of discernment is a set including exclusive elements. In order to promote the scientific process of decision, the empty set is not involved in framework of discernment because it does not contain any information beneficial for decision making. Then, for $k-element$ framework of discernment $[a_1, a_2, ..., a_{k-1}, a_k]$, its power set spatial distribution contains $n = 2^k - 1$ subsets, namely

$$\begin{aligned}\Omega = \{&\{a_1\}, \{a_2\}, ..., \{a_k\},\\ &\{a_1, a_2\}, ..., \{a_1, a_{k-1}\}, ...,\\ &\{a_1, a_2, ..., a_{k-1}, a_k\}\}\\ &= \{x_1, x_2, ..., x_n\},\end{aligned} \tag{1}$$

here x_n means the nth subset of power set spatial distribution.

(2)Basic probability assignment(BPA)

Basic probability assignments are presented as mass functions m whose function values vary in range [0, 1].

$$m(A) \to [0, 1], A \in \Omega \tag{2}$$

$$\sum_{A \in \Omega} m(A) = 1 \tag{3}$$

$$m(\Phi) = 0 \tag{4}$$

B. Gaussian function

Gaussian function is a probability function which describing the distribution law of random variables. It is defined by two parameters, mean $\bar{X}$ and variance σ. The expression of Gaussian function μ is

$$\mu(x) = \exp\,[-\frac{(x - \bar{X})^2}{2\sigma^2}] \tag{5}$$

III. Generation of BPA

For a m attributes data set with N samples of k classes, n samples are selected from each class as training samples, so as to establish the Gaussian model of each class on each attribute. The remaining samples are used as test samples from which BPA is generated. The procedures of BPA generation are presented as follows.

A. Build Gaussian models on each attribute

The specific process of obtaining Gaussian function $\mu(x)$ is as follows.

(1) For a selected class k and attribute s, respectively calculate the sample mean $\bar{X_{sk}}$ and standard deviation σ_{sk} of all training samples belonging to class k on the attribute s:

$$\bar{X_{sk}} = \frac{1}{m}\sum_{i=1}^{m} x_{sk}^i, \tag{6}$$

$$\sigma_{sk} = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_{sk}^i - \bar{X_{sk}})^2}, \tag{7}$$

where, x_{sk}^i represents the value of the ith sample of class k on attribute s.

(2) According to the obtained mean $\bar{X_{sk}}$ and standard deviation σ_{sk}, construct the Gaussian models of class k on the attribute s:

$$\mu_k^s(x) = \exp\,[-\frac{(x - \bar{X_{sk}})^2}{2\sigma_{sk}^2}]. \tag{8}$$

B. Match the test samples to the Gaussian models to get BPA

For example, Gaussian models of a data set with three classes A, B, C on attribute s are built. And there is a piece of test data having a value of v_1 on attribute s. Firstly, calculate the function values of the test value, namely $\mu_A^s(v_1), \mu_B^s(v_1), \mu_C^s(v_1)$. Then sort them in descending order to get the sequence of classes and assign the function value of a class to the subset including itself and classes before it. Finally, the mass functions of these subsets are represented by the function values.

C. Combine the BPA with distance measures

As a means of revealing the intrinsic patterns and structures within data, distance measures provide a quantitative foundation for assessing similarity and dissimilarity between objects. These measures can capture the geometric relationships among different objects, thereby facilitating the identification of potential clustering structures and classification boundaries.

The process of combination is achieved through the calculation of the inner product. For test data $(v_1, v_2, ..., v_m)$ which values on attributes $(A_1, A_2, ..., A_m)$, the final BPA are gained by

$$BPA_{A_m} = Gaussian_{A_m} * Distances, \tag{9}$$

where $Gaussian_{A_m}$ are the BPA directly generated from the Gaussian models and $Distances$ refers to a certain distance sequence consists of distances of subsets in FoD, namely distance measures of $\{x_1, x_2, ..., x_k\}$.

Consider a dataset with three classes A, B, C. The original distances are computed from the test data to the training data on each attribute, as follows:

$$\begin{aligned}dis &= \{d_A^s, d_B^s, d_C^s, d_{A,B}^s, d_{A,C}^s, d_{B,C}^s, d_{A,B,C}^s\}\\ &= \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\},\end{aligned} \tag{10}$$

here d_A^s denotes the distance from the test value to a specific measure of distance for the data in class A of training set, and $d_{A,B}^s$ is the distances of the data in class A and class B.

In this study, the mean and median are chosen as distance measures and are combined with the BPA generated from the Gaussian models. The mean is defined as:

$$mean(x_1, x_2, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{11}$$

and the median is given by:

$$\begin{aligned}&median(x_1, x_2, \ldots, x_n)\\ &= \begin{cases} x_{(n+1)/2}, & \text{if } n \text{ is odd}\\ \frac{x_{(n/2)} + x_{(n/2+1)}}{2}, & \text{if } n \text{ is even,}\end{cases}\end{aligned} \tag{12}$$

where $x_1, x_2, \ldots, x_n$ refers to a sequence of values.

As an example, when using the mean measure, the elements in dis are defined as:

$$\begin{aligned} d_1 &= d_A^s = v^s - mean(train_A^s), \\ d_4 &= d_{A,B}^s = v^s - mean(train_A^s \cup train_A^s), \end{aligned} \tag{13}$$

here v^s denotes the value of a given test data on attribute s, and $train_A^s$ denotes the part of training data that belongs to class A on attribute s. The remaining elements in dis can be calculated in a similar manner.

To fully leverage the distance information, we apply the negative exponent to the original distance values, thus defining $Distances$ as:

$$\begin{aligned} Distances = e^{(-dis)} = \{&e^{-(d_1)}, e^{-(d_2)}, \\ &e^{-(d_3)}, e^{-(d_4)}, e^{-(d_5)}, e^{-(d_6)}, e^{-(d_7)}\}. \end{aligned} \tag{14}$$

Finally, the BPA with distance measures are gained by combining $Distances$ with the BPA generated from Gaussian models according to Equation (9).

IV. Application in Evidential Decision Tree

Gao et al. [8] introduced a novel method for constructing an evidential decision tree using hierarchical interval estimation, which has been shown to be effective. In this paper, we build upon their work by modifying the attribute selection rule and incorporating the proposed BPA generation method to further enhance the model's performance. The procedures for constructing the modified evidential decision tree are illustrated in Fig. 1.

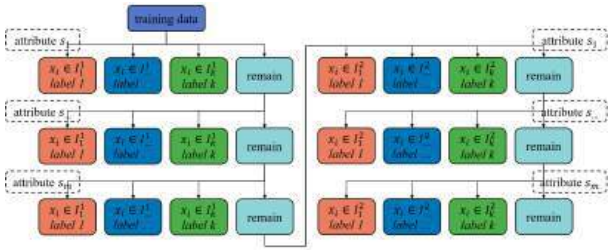


Fig. 1. Process of constructing evidential decision tree

(1)Divide the data set.

For classification problems, the whole data set is usually divided into training set and test set. The data in test set accounts for 20% to 30%.

(2)Select splitting attributes.

In Dempster–Shafer theory, many kinds of entropy methods are proposed to quantify the amount of information contained in uncertain data. In this paper, we choose Nguyen entropy(Equation (15)) and Deng entropy(Equation (16)) to measure the uncertainty.

$$E_{Nguyen} = -\sum_{A \in \Omega} m(A) log_2 m(A) \tag{15}$$

$$E_{Deng} = -\sum_{A \in \Omega} m(A) log_2 \frac{m(A)}{2^{|A|} - 1} \tag{16}$$

Specifically, the entropy of the attribute s is calculated by

$$E(s) = \frac{1}{n} \sum_{i=1}^{n} E(m_i^s), \tag{17}$$

where E refers to Nguyen entropy or Deng entropy, and m_i^s denotes the BPA combined with distance measures. Finally, the attribute $s^* = argmin E(s)$ is selected as the best splitting attribute.

(3)Determine interval estimation criterion.

In this paper, we modify the algorithm proposed by Gao et al. [8]. Both extremum values of attribute data (namely $I^1 = [x_{min}^{sk}, x_{max}^{sk}]$), mid-value μ and width ϵ (namely $I^2 = [\mu_{sk} - \epsilon, \mu_{sk} + \epsilon]$) are used to form splitting intervals. Considering that the value of ϵ can decide the speed at which data are split into different branches, we choose the standard deviation σ as the value of ϵ.

V. Experiment

The classification problem holds a pivotal position in the fields of machine learning and artificial intelligence (AI). It is not only a core driving force behind the advancement of AI technologies but also a crucial factor determining the successful deployment and application of intelligent systems.

To investigate this, we conduct experiments on the classification problem using the Iris dataset. For simplicity, the classes set, ver, vir are denoted as A, B, C, and attributes are denoted as SL, SW, PL, PW.

A. Generate basic probability assignment

Step 1: Determine the framework of discernment.

For the three classes in iris data set, the framework of discernment is $\{A, B, C\}$. Thus its power set spatial distribution is given by:

$$\begin{aligned} \Omega = \{&\{A\}, \{B\}, \{C\}, \{A, B\}, \\ &\{A, C\}, \{B, C\}, \{A, B, C\}\}. \end{aligned} \tag{18}$$

Step 2: Build Gaussian models on each attribute.

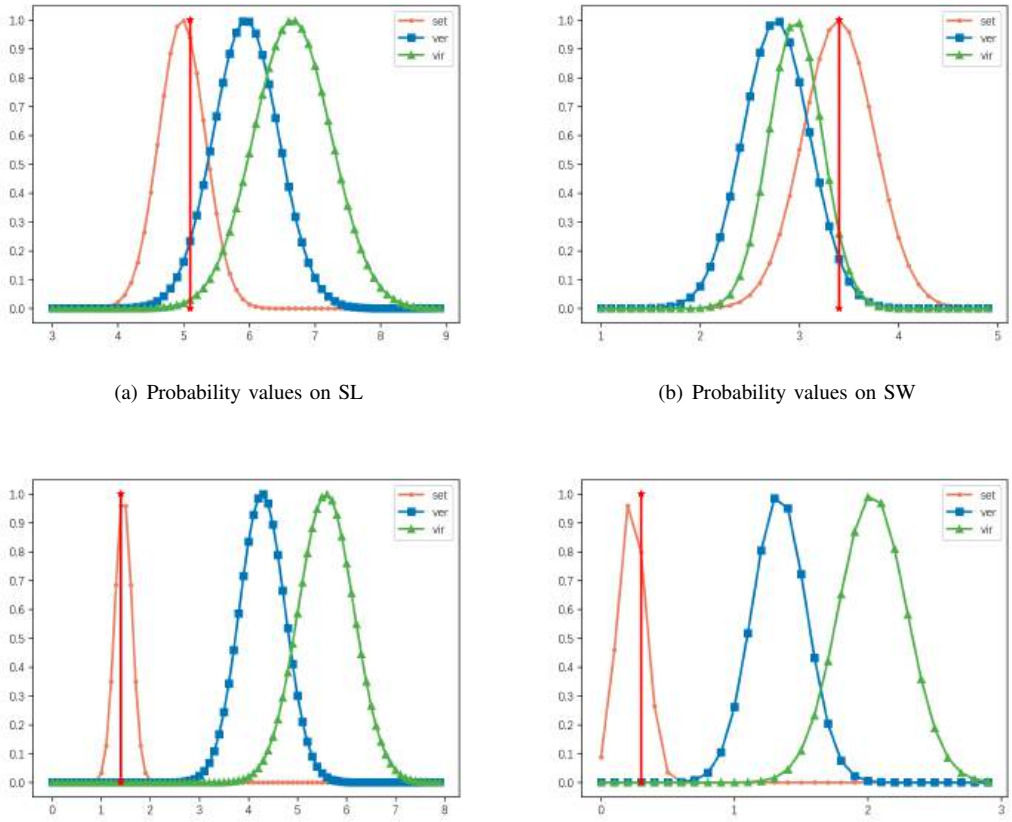
According to Equation(6) and Equation(7), means and standard deviations of each class on each attribute are calculated. With these critical parameters, Gaussian models can be established by Equation(8). For a given data set partitioning situation, the Gaussian models established on each attribute are shown in Fig 2.

Step 3: Match the test samples to the Gaussian model to get BPA.

For instance, consider the test data point (5.1, 3.8, 1.5, 0.3). The corresponding BPA derived from the Gaussian models is shown in Tab I.

Step 4: Combine the BPA with measurement of distance.

For each test data point, compute the distance values for each subset in the power set of the framework of discernment, as specified in Equation (13). Then take the negative exponent of the values and combine them with BPA in Step 3.



(a) Probability values on SL

(b) Probability values on SW

(c) Probability values on PL

(d) Probability values on PW

Fig. 2. Established Gaussian models on each attribute

B. Construct decision tree

Decision trees are constructed on different capacity of training set and their performance is determined by $Accuracy$ which means the ratio of samples correctly classified among all test samples.

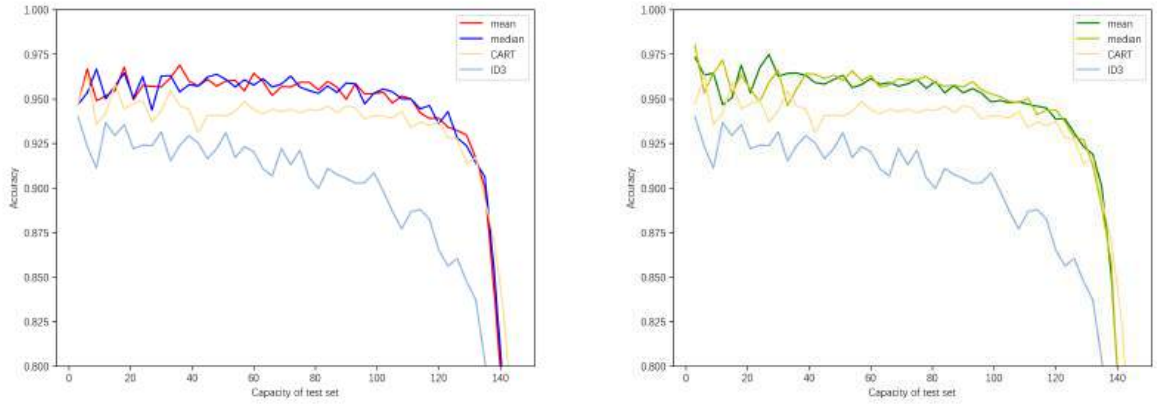
C. Results and discussion

To make comparison with existing methods and verify the performance of proposed method, we also apply it to traditional decision trees such as ID3(with "one vs all" strategy) and CART. For each method, we have done the experience 50 times and calculated the average accuracy of every time in each capacity of test set. To illustrate each methods' dependence on amount of training data and the influence of entropy, the variation of accuracy with capacity is shown in Fig 3.

The experimental results are averaged over the first half of the test set for each capacity value, with these averages serving as the final performance metrics for each method. Specifically, the Nguyen entropy method achieves accuracies of 95.784% for the mean measure and 95.788% for the median measure, while the Deng entropy method yields accuracies of 96.110% and 96.061%, respectively. In contrast, the CART method results in an accuracy of 94.425%, and the ID3 method produces an accuracy of 92.297%.

The proposed method demonstrates enhanced stability as the training set varies, with accuracy oscillating within relatively high ranges. This indicates that the method effectively improves the performance of the decision tree by incorporating uncertain information. In fact, the objective of classification tasks is to determine which category a test sample most closely resembles based on the training data, and the core principle of BPA is to represent uncertainty by assigning mass to subsets. By adjusting the mass function of each category subset according to the distance from the test sample, BPA can more accurately express the degree of membership of the test sample to each category, thereby ultimately improving the performance of the evidential decision tree.

It is also evident that the methods utilizing Nguyen entropy exhibit performance similar to those using Deng entropy. One possible reason for this is that the distributions of BPA generated by both methods are analogous. In this case, the multi-subsets in the generated BPA carry less information, resulting in minimal differences between the two entropy measures. For example, consider the test set data point (5.1, 3.8, 1.5, 0.3); the BPA generated from this point and the



(a) Results by Nguyen entropy

(b) Results by Deng entropy

Fig. 3. Results of each method

corresponding entropy values are recorded in Table I.

TABLE I
BPA GENERATED FROM REAL DATA AND THEIR ENTROPY

	A	AB	AC	ABC	Nguyen	Deng
SL	0.93	0.23	-	0.03	0.7278	1.1823
SW	0.99	-	0.17	0.25	0.9406	1.9391
PL	0.95	nearly 0	-	nearly 0	0.0581	0.0581
PW	0.79	nearly 0	-	nearly 0	0.2595	0.2595

However, let us assume that a group of BPA, where the multi-subsets contain more information, is obtained in some way. Table II presents three forms of BPA. As shown in Table II, the Nguyen entropy values for these BPA are identical, while the Deng entropy values differ. Furthermore, the greater the amount of information contained in the multi-subset, the larger the discrepancy between the Nguyen entropy and Deng entropy values. This can be attributed to the fact that Deng entropy considers the cardinality of the subsets, enabling it to utilize the information from the multi-subset more effectively.

TABLE II
HYPOTHETICAL BPA AND THEIR ENTROPY

	A	B	C	AB	ABC	Nguyen	Deng
α	0.1	0.1	-	0.8	-	0.9219	0.9219
β	0.1	0.1	-	-	0.8	0.9219	3.1678
γ	0.1	-	-	0.1	0.8	0.9219	3.3263

VI. CONCLUSION

In conclusion, this paper proposes a novel method for generating basic probability assignments (BPA). It uses Gaussian models based on the mean and standard deviation of each class to generate BPA for the test set. Distances between test samples and class centroids are calculated, then aligned with the power set distribution in the framework of discernment. These BPA are combined with distance values through their inner product. The method is validated using the evidential decision tree. When applied to the Iris classification problem, the proposed method achieves average accuracies for both entropy types that are 1.011% and 3.319% higher than those of other methods. This result demonstrates its effectiveness in attribute selection for splitting and BPA quality, while also highlighting its advantages in handling uncertain data.

ACKNOWLEDGMENTS

The work is supported by Qin Chuangyuan high-level innovation and entrepreneurship talent program of Shaanxi(Grant No.QCYRCXM-2023-108) and the "Innovation and Entrepreneurship Training Program for college students" project of Northwest A&F University (Project No. X202410712542).

REFERENCES

[1] D. A. P., Upper and lower probabilities induced by a multivalued mapping, Institute of Mathematical Statistics (1967).
[2] G. Shafer, A mathematical theory of evidence, Princeton University Press 42 (1976).
[3] Z. Wang, W. Yang, H. Zhang, Y. Zheng, Spa-based modified local reachability density ratio wsvdd for nonlinear multimode process monitoring, Complexity 2021 (1) (2021) 5517062.
[4] W. Fu, S. Yu, X. Wang, A novel method to determine basic probability assignment based on adaboost and its application in classification, Entropy 23 (7) (2021) 812.
[5] Y. Tang, Y. Zhou, X. Ren, Y. Sun, Y. Huang, D. Zhou, A new basic probability assignment generation and combination method for conflict data fusion in the evidence theory, Scientific Reports 13 (1) (2023) 8443.
[6] H. Garg, D. Rani, Novel distance measures for intuitionistic fuzzy sets based on various triangle centers of isosceles triangular fuzzy numbers and their applications, Expert Systems with Applications 191 (2022) 116228.
[7] A. Hassanat, E. Alkafaween, A. S. Tarawneh, S. Elmougy, Applications review of hassanat distance metric, in: 2022 International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA), IEEE, 2022, pp. 1–6.
[8] B. Gao, Q. Zhou, Y. Deng, Hie-edt: Hierarchical interval estimation-based evidential decision tree, Pattern Recognition 146 (2024) 110040.

A LLMs-assisted Framework for Parkinson's Disease Assessment Based on PPMI Dataset

Zhenyu Gao[1], Qin Ni[2], Wen Liu[1], and Lei Zhang[1], *Member, IEEE*
[1]College of Information Science and Technology, Donghua University, Shanghai, China
[2]The Key Laboratory of Multilingual Education with AI, Shanghai International Studies University, Shanghai, China
lei.zhang@dhu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899636

***Abstract*—The assessment of diseases such as Parkinson's disease, which are chronic neurodegenerative conditions, is an extremely complex and time-consuming process. Self-assessment scales or interviews in the outpatient often lead to the loss of key information. Large Language Models (LLMs) show abilities in capturing subtle linguistic differences and handling long texts, enabling the extraction of a significant amount of key information while ensuring data integrity and user privacy. We focus attention on people with Parkinson's disease (PwP). In response to these issues, we propose a framework in this paper. Firstly, we use large language model (LLM) with chain-of-thought (CoT) prompt to rephrase patient self-report texts from real-outpatient structured data as a supervised fine-tuning (SFT) corpus. Secondly, we fine-tune bidirectional encoder representation from transformers (BERT) with Low-Rank Adaptation (LoRA) to enable it to understand and extract the semanteme of self-report, thus predicting each MDS-UPDRS item. Finally, we designed experiments on a large amount of test data to evaluate the effectiveness of the framework. The results indicate that the accuracy on this task has been improved to 95.36%, which is a 6.7% increase compared to the best-performing model.**

***Index Terms*—Large language models, BERT, Parkinson's disease, Healthcare, Supervised Fine-Tuning**

I. Introduction

Since the advent of ChatGPT, the potential of its application across various tasks has presented a possible path towards artificial general intelligence (AGI). The deployment of large language models in the healthcare and medical is one of the current research attention. Healthcare and medical and concerns everyone's life, and research in this field, including Med-PaLM, BioMedLM, DoctorGLM, Med-Gemini, and other LLMs [1]–[4], have demonstrated their outstanding capabilities in the healthcare. These applications span a wide range of tasks, including medical question answering, medical advice generation, assistance in clinical diagnosis, and processing of massive amounts of medical data [5].

Parkinson's disease (PD) is the second most common neurodegenerative disorder, following Alzheimer's disease. The prevalence of Parkinson's disease may be growing faster than any other neurological disorder globally [6]. Over the past two decades, the prevalence of Parkinson's disease has significantly increased compared to the period from 1980 to 2003, with an incidence rate reaching 9.34 cases per 1000 people aged 60 and older [7]. Additionally, there is a long prodromal period before clinical manifestation of Parkinson's disease [8]. Therefore, integrating LLMs technologies into the automated diagnosis and treatment process for PD is an urgent research question.

As a result, we surveyed some neurologists, and they raised several problems in the real-world diagnosis and treatment of Parkinson's disease: 1) Cumbersome scale assessments: The diagnosis of PwP always involves a lot of scales, such as MDS-UPDRS, etc. These scales contain a number of questions, requiring much time and energy from both patients and doctors to complete, which is not only time-consuming but also cause fatigue. 2) Symptom fluctuation: Symptoms of PwP may fluctuate throughout the day or on different days, which can lead to unstable results in scale assessments and prevent tracking in time. 3) Follow-up and monitoring: Parkinson's disease is a chronic condition that requires long-term follow-up and monitoring. For doctors, tracking the progression of the disease and the effectiveness of treatment is an complex task.

In response to the said problems, in the paper, we took advantage of LLMs to conduct research based on real MDS-UPDRS data, and developed a framework that aligns medical scales using patient self-reported texts. The MDS-UPDRS has been the most widely used scale in various settings of clinical and research practices [9]. Our work is primarily divided into two steps, data generation and results prediction, as shown in Fig. 1. First, we need to obtain high-quality unstructured data, which in this case is text data, and ensure it is aligned with the real participant structured data we obtained from Parkinson's Progression Markers Initiative (PPMI). We utilized the few-shot learning ability of LLMs, guiding the model step by step to generate corpora that conform to the patient self-report format through CoT and few-shot prompting, and also provided some interpretability during the generation process. For model training and prediction, we used a model named ClinicalBERT [10] as the base model and supervised fine-tuned it using the corpora obtained from the data generation step and the LoRA framework. Finally, we transferred the task to other models, including BERT, GPT3.5, ChatGLM3-6B, and ChatGLM4 [11]–[13], and compared their performance with the framework mentioned in this paper to validate the feasibility of our framework.

The key contributions of this work are as follows:

- Utilizing the few-shot learning ability of LLMs and the chain-of-thought prompt engineering, we transformed structured data into self-reported form corpora. Based on

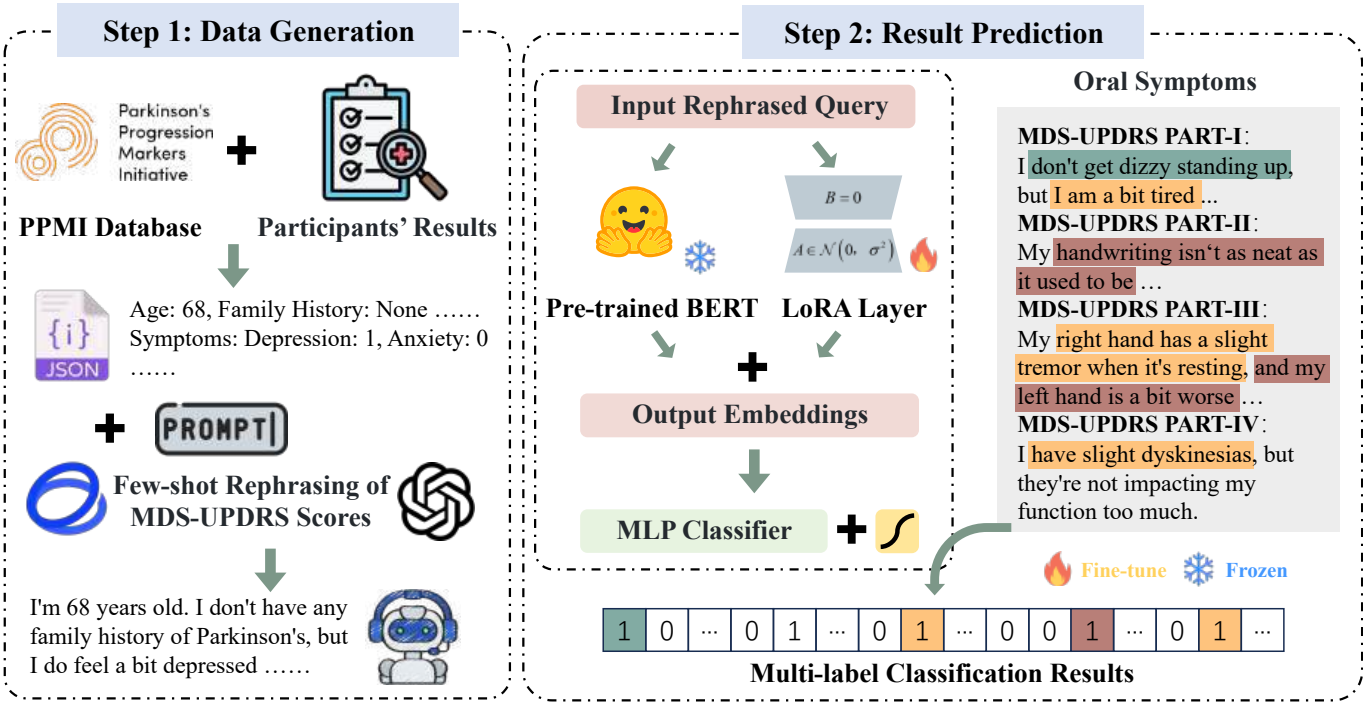


Fig. 1: The overall workflow of our framework: from outpatient data extraction to MDS-UPDRS scores prediction.

real structured table data, we formed a pipeline for generating supervised fine-tuning instructions in a human-in-the-loop manner.

- Using a pre-trained BERT as the base model, we conducted supervised fine-tuning through LoRA. We obtained a fine-tuned model capable of handling unstructured input tasks and outputting structured Parkinson's scale scores, which improved accuracy for the task compared to other models.

II. Related Work

LLMs have reshaped the application of AI in the healthcare and medical. In recent years, a variety of tools and applications related to large models have emerged. Rasmy et al. [14] proposed Med-BERT, a BERT model pre-trained on a large number of electronic health records (EHR), which introduces the successful experience of BERT in the natural language processing NLP to clinical tasks, providing a powerful pre-trained model for healthcare.

In the realm of healthcare question-answering systems, the LLM named Med-PaLM2 [15] achieved a score as high as 86.5% on the USMLE dataset, which is more than 19% higher than Med-PaLM [1] and sets a new technical standard.

Chatdoctor [16] is a medical assistant based on LLaMA [17] that has been fine-tuned using a large real-world dataset of doctor-patient conversations and integrated with a self-directed information retrieval mechanism, possessing smooth conversational abilities. Similarly, DoctorGLM [3] uses ChatGLM6B as the base model and has been trained on a Chinese medical dialogue dataset, achieving better results in Chinese scenarios.

We then focused on the application of LLM in the Parkinson's disease. Rahman et al. [18] established a user-centered teleneurology platform and assessed the possibility of using artificial intelligence technology to screen for PwP.

Among the aforementioned works, BERT is difficult for medical professionals to use directly and is more of a tool for data scientists. The generated text from healthcare LLMs can create a "trust" problem. Additionally, in the field of Parkinson's disease applications, traditional machine learning techniques like XGBoost and SVM are still used for basic identification, while LLMs mainly focus on providing chat services after obtaining the results. Our framework uses the powerful semantic classification capability of BERT as a classifier and does not provide medical advice. Moreover, it can use unstructured input instead of structured feature input, offering a more convenient user interface.

III. Materials and Methods

A. Chain-of-Thought for Data Generation

In this phase, we used real participant data from the PPMI database [19], including comprehensive MDS-UPDRS data along with demographic information such as age, gender, family history, etc. These data were compiled into JSON files. Utilizing the comprehension and the reasoning abilities of large language models (LLMs), we rephrased raw data into conversational language by simulating a outpatient scenarios. Fig. 2 provides a visual representation of the associated workflow.

Due to the extensive number of items in the MDS-UPDRS, more than 50 items, ensuring that the rephrased results

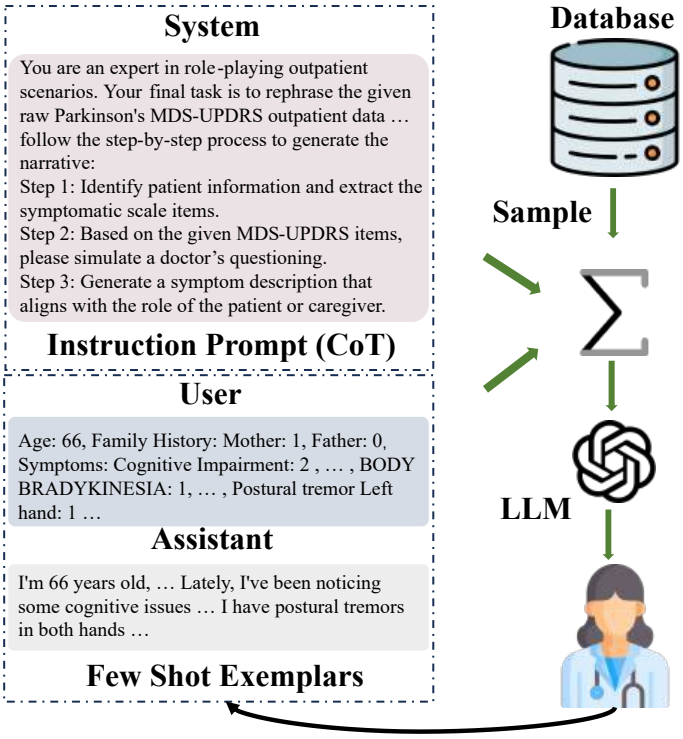


Fig. 2: The workflow of Chain-of-Thought for data generation in our framework.

accurately correspond to the true information provided by the subjects is challenging. The standard prompting method tends to hallucination, and the generation process complicates adjustments by prompt engineers. To address this, we opted for a Chain-of-Thought (CoT) approach with few-shot Exemplars as our prompting strategy. CoT has been demonstrated to bring out emergent abilities in LLMs [20]. Moreover, its step-by-step generation process enhances the controllability and interpretability of the transcription process, revealing the thought process of the LLMs and avoiding a completely non-transparent "black-box" process.

Under the instruction of system prompts, the LLM was tasked with identifying key items. Subsequently, in the model simulation of a clinic, a doctor uses a scale to conduct an interview and score a patient. The LLM generated a symptom description that aligns with the role of the patient. Through CoT, we have established a human-in-the-loop prompting engineering framework in which researchers actively participate in the generation process, recheck the generated results, provide feedback on prompt modifications, and ultimately generate more diverse and accurate results.

B. Supervised Fine-Tuning by LoRA

To better use the data generated in section 3.1 for scoring tasks, we selected a pre-trained model from the medical field named ClinicalBERT as the base model in [10]. This is a pre-trained model initialized based on BERT and trained on a large corpus of various diseases totaling 1.2B words. This pre-trained model has better generalization performance in the medical field compared to the basic BERT. Fine-tuning on this basis can adapt to our task more quickly, making it a more low-carbon, cost-effective, and faster approach.

We used the high-quality data generated in section 3.1 as the training corpus, which has good consistency with real outpatient data. Additionally, we encoded the scores of items for each sample as labels and chose the LoRA method for supervised fine-tuning. Furthermore, the final classification result of each sample corresponds to multiple labels. For example, in the sample statement "I often feel sleepy during the day, and my anxiety has been very serious recently." the output classification results should at least include MDS-UPDRS 1.8 and MDS-UPDRS 1.4. Therefore, we define this classification problem as a multi-label classification problem and set up a classification layer that fits this task. We freezed all parameters of the pre-trained model and placed the LoRA layers on the Q, V matrices, significantly reducing the number of trainable parameters using LoRA.

C. Interact Mode

In the interaction process, on one end of the interaction is the patient or caregiver, who relays the Parkinson's patients health condition and symptoms to the system in home. This process is similar to a patient visiting a clinic and consulting with a professional physician. After receiving the description, the system returns results aligned with the MDS-UPDRS scores and records the time and specifics of this assessment. Subsequently, the patient's attending physician can review the results of each home assessment, checking the efficacy of treatment or the progression of the condition based on score changes. In short, this framework allows patients to reduce the time spent on each scale by shifting from manually recorded scales to those derived from verbal narration. Particularly for Parkinson's patients, who may have lost some motor abilities or suffer from abnormal emotional fluctuations, narration is a more convenient and natural method, lightening their burden. Additionally, the MDS-UPDRS scale has many items, and it is easy for both doctors and patients to experience survey fatigue, which can affect the scoring results [21].

IV. Experiments and Results

A. Data Analysis

Our assessment data is sourced from PPMI, a real-world database specifically for Parkinson's disease and Parkinsonian syndromes. The database has collected within-participant data from over 4,000 participants across approximately 50 sites worldwide, with a gender distribution of 54.5% male and 45.5% female. For our research objectives, we selected the MDS-UPDRS from the Motor Assessments. Through LLM, we ultimately obtained a total of 60,342 labeled data for training and testing from visits of participants in the database. During the training process, the training and validation sets were divided in a ratio of 4:1.

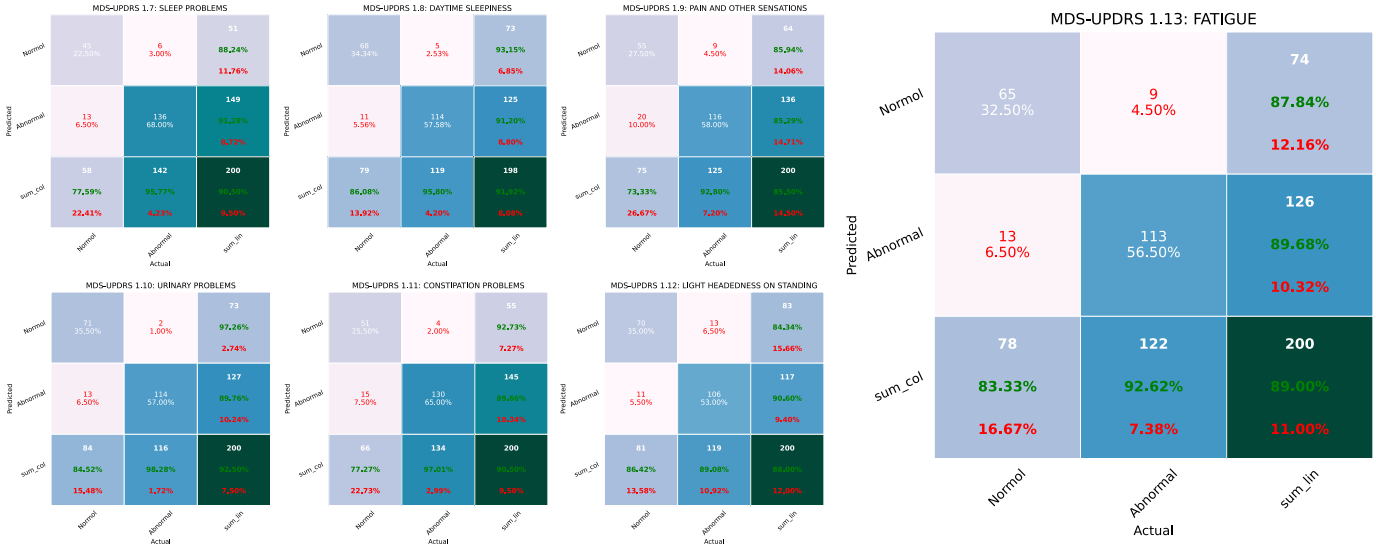


Fig. 3: The confusion matrix for each symptom in MDS-UPDRS Part I. Each subplot is derived from our sample of 200 cases, with the titles indicating each item, including sleep problems, daytime sleepiness, pain and other sensation, urinary problems, constipation problems, light headedness on standing, and fatigue.

TABLE I: Experimental parameter settings.

Parameters	
Learning rate	1×10^{-4}
Epoch	100
Batch_size	64
LoRA_rank	4
LoRA_alpha	2
Early stop patience	10
Early stop monitor	val_loss
Lr_scheduler_type	Cosine Annealing
Trainable parameters	7.4M
Non-trainable parameters	135M

B. Training Settings

Our training process was run on a server with Ubuntu 22.04, Python 3.10, and a single RTX3090.

During the training process, we did not select full parameter fine-tuning. Instead, we chose LoRA fine-tuning. Compared to full parameter fine-tuning, LoRA better preserves the prior knowledge of the pre-trained model and offers higher computational efficiency. By freezing the parameters of the pre-trained model, we kept the number of trainable parameters in the model to 7.4M. We set the learning rate to 1×10^{-4} and used Cosine Annealing as the learning rate decay strategy, with LoRA_rank set to 4 and LoRA_alpha set to 2. All the training parameters are as shown in Table I.

C. Model Evaluation Results

With the trained model obtained, we selected a variety of evaluation methods. Since our problem is a multi-label classification, where one sample corresponds to multiple labels, the common methods for multi-classification models are not applicable. Therefore, we referred to the evaluation methods for multi-label classification as described in [22], using Label-based metrics to calculate TP, TN, FP, FN. The j-th class labely_j can be calculated in (1).

$$\begin{aligned} TP_j &= |\{x_i \mid y_j \in Y_i \wedge y_j \in h(x_i), 1 \le i \le p\}|; \\ FP_j &= |\{x_i \mid y_j \notin Y_i \wedge y_j \in h(x_i), 1 \le i \le p\}|; \\ TN_j &= |\{x_i \mid y_j \notin Y_i \wedge y_j \notin h(x_i), 1 \le i \le p\}|; \\ FN_j &= |\{x_i \mid y_j \in Y_i \wedge y_j \notin h(x_i), 1 \le i \le p\}|. \end{aligned} \tag{1}$$

Where p is the total number of instances involved in the evaluation, $h(\cdot)$ is the classifier.

Based on the above four quantities, we can compute most of the binary classification metrics. Let $B(TP_j, FP_j, TN_j, FN_j)$ represent some specific binary classification metric ($B \in \{Accuracy, Precision, Recall, F1\}^4$). Finally, we used the four metrics for each label to calculate the micro-average metrics for the model using, based on:

$$B_{\text{micro}}(h) = B\left(\sum_{j=1}^{q} TP_j, \sum_{j=1}^{q} FP_j, \sum_{j=1}^{q} TN_j, \sum_{j=1}^{q} FN_j\right) \tag{2}$$

The results, as shown in Table II, include the outcomes of the extra six models we selected, which performed the same task for comparison.

In addition, we plotted the confusion matrices for some of the label classification results. Fig. 3 shows the classification outcomes for the seven symptoms in the patient questionnaire section of MDS-UPDRS Part I. It can be observed that the classification achieved a high accuracy.

V. Conclusions and Future Work

In this study, we present an innovative application, which uses LLMs to construct self-reported corpora for training data,

TABLE II: Comparing with existing methods on the MDS-UPDRS task.

Models	Accuracy	Precision	Recall	F1
BERT with Pretrain-MLP	0.2597	0.3615	0.3220	0.3406
GPT-3.5 (zero-shot)	0.7186	0.5604	0.8242	0.6671
GPT-3.5 (few-shot)	0.7492	0.6081	0.8432	0.7066
Fintuned-ChatGLM3-6B	0.8405	0.6014	0.9004	0.7211
GLM-4-Plus (zero-shot)	0.8840	0.7102	0.9275	0.8045
GLM-4-Plus (few-shot)	0.8936	0.7341	**0.9335**	0.8218
Ours	**0.9536**	**0.9085**	0.8543	**0.8805**

and employs a pre-trained medical BERT model for fine-tuning to better perform the daily classification tasks of the MDS-UPDRS. It validates the possibility of using artificial intelligence to report the results of Parkinson's disease scale in non-outpatient settings. Our system also demonstrates the potential for further continuous assessment and tracking of Parkinson's patients' conditions. In addition, compared to the current LLMs with massive parameters, our system is more lightweight. As an auxiliary diagnostic tool, it is also easier to deploy. For future work, we intend to investigate the utilization of the model's generated results for Parkinson's disease and examine more user-friendly interaction approaches for both patients and doctors.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 62371118.

REFERENCES

[1] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl *et al.*, "Large language models encode clinical knowledge," *Nature*, vol. 620, no. 7972, pp. 172–180, 2023.

[2] E. Bolton, A. Venigalla, M. Yasunaga, D. Hall, B. Xiong, T. Lee, R. Daneshjou, J. Frankle, P. Liang, M. Carbin *et al.*, "Biomedlm: A 2.7 b parameter language model trained on biomedical text," *arXiv preprint arXiv:2403.18421*, 2024.

[3] H. Xiong, S. Wang, Y. Zhu, Z. Zhao, Y. Liu, L. Huang, Q. Wang, and D. Shen, "Doctorglm: Fine-tuning your chinese doctor is not a herculean task," *arXiv preprint arXiv:2304.01097*, 2023.

[4] L. Yang, S. Xu, A. Sellergren, T. Kohlberger, Y. Zhou, I. Ktena, A. Kiraly, F. Ahmed, F. Hormozdiari, T. Jaroensri *et al.*, "Advancing multimodal medical capabilities of gemini," *arXiv preprint arXiv:2405.03162*, 2024.

[5] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.

[6] J. D. Steinmetz, K. M. Seeher, N. Schiess, E. Nichols, B. Cao, C. Servili, V. Cavallera, E. Cousin, H. Hagins, M. E. Moberg *et al.*, "Global, regional, and national burden of disorders affecting the nervous system, 1990–2021: a systematic analysis for the global burden of disease study 2021," *The Lancet Neurology*, vol. 23, no. 4, pp. 344–381, 2024.

[7] J. Zhu, Y. Cui, J. Zhang, R. Yan, D. Su, D. Zhao, A. Wang, and T. Feng, "Temporal trends in the prevalence of parkinson's disease from 1980 to 2023: a systematic review and meta-analysis," *The Lancet Healthy Longevity*, vol. 5, no. 7, pp. e464–e479, 2024.

[8] B. R. Bloem, M. S. Okun, and C. Klein, "Parkinson's disease," *The Lancet*, vol. 397, no. 10291, pp. 2284–2303, 2021.

[9] R. Rajan, L. Brennan, B. R. Bloem, N. Dahodwala, J. Gardner, J. G. Goldman, D. A. Grimes, R. Iansek, N. Kovács, J. McGinley *et al.*, "Integrated care in parkinson's disease: a systematic review and meta-analysis," *Movement Disorders*, vol. 35, no. 9, pp. 1509–1531, 2020.

[10] G. Wang, X. Liu, Z. Ying, G. Yang, Z. Chen, Z. Liu, M. Zhang, H. Yan, Y. Lu, Y. Gao *et al.*, "Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial," *Nature Medicine*, vol. 29, no. 10, pp. 2633–2642, 2023.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[12] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[13] T. GLM, A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Rojas, G. Feng, H. Zhao, H. Lai, H. Yu, H. Wang, J. Sun, J. Zhang, J. Cheng, J. Gui, J. Tang, J. Zhang, J. Li, L. Zhao, L. Wu, L. Zhong, M. Liu, M. Huang, P. Zhang, Q. Zheng, R. Lu, S. Duan, S. Zhang, S. Cao, S. Yang, W. L. Tam, W. Zhao, X. Liu, X. Xia, X. Zhang, X. Gu, X. Lv, X. Liu, X. Liu, X. Yang, X. Song, X. Zhang, Y. An, Y. Xu, Y. Niu, Y. Yang, Y. Li, Y. Bai, Y. Dong, Z. Qi, Z. Wang, Z. Yang, Z. Du, Z. Hou, and Z. Wang, "Chatglm: A family of large language models from glm-130b to glm-4 all tools," 2024.

[14] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi, "Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction," *NPJ digital medicine*, vol. 4, no. 1, p. 86, 2021.

[15] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. Pfohl, H. Cole-Lewis, D. Neal *et al.*, "Towards expert-level medical question answering with large language models," *arXiv preprint arXiv:2305.09617*, 2023.

[16] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang, "Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge," *Cureus*, vol. 15, no. 6, 2023.

[17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[18] W. Rahman, A. Abdelkader, S. Lee, P. Yang, M. S. Islam, T. Adnan, M. Hasan, E. Wagner, S. Park, E. R. Dorsey, C. Schwartz, K. Jaffe, and E. Hoque, "A user-centered framework to empower people with parkinson's disease," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, no. 4, Jan. 2024. [Online]. Available: https://doi.org/10.1145/3631430

[19] K. Marek, D. Jennings, S. Lasch, A. Siderowf, C. Tanner, T. Simuni, C. Coffey, K. Kieburtz, E. Flagg, S. Chowdhury *et al.*, "The parkinson progression marker initiative (ppmi)," *Progress in neurobiology*, vol. 95, no. 4, pp. 629–635, 2011.

[20] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[21] P. J. Lavrakas, *Encyclopedia of survey research methods*. Sage publications, 2008.

[22] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

Syntactic-Semantic Graph Fusion Generative Adversarial Network: SSGF-GAN

Qihui Li
Xiamen University Malaysia
Sepang, Malaysia
liqihui02@outlook.com

Ruixin Kang*
Xiamen University Malaysia
Sepang, Malaysia
kangruixin0406@gmail.com

Haodong Lu*
Xiamen University Malaysia
Sepang, Malaysia
luhaodong101519@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899591

***Abstract*—As AI technology continues to advance rapidly, more generative tasks have emerged, with text-to-image (T2I) generation becoming a popular area of research. Numerous studies have demonstrated the impressive capabilities that GANs bring to this field. In this paper, we focus on enhancing GANs by incorporating a novel component called the Syntactic-Semantic Graph Fusion Block, inspired by the Graph Neural Network (GNN). This block effectively captures the semantic and syntactic nuances of different sentences, leading to more detailed and refined image generation. Additionally, we introduce several other innovations to the framework, further boosting the model's performance. Experimental results on the CUB bird dataset demonstrate that our approach outperforms many current advanced models.**

***Index Terms*—Generative Adversarial Network, Text-to-Image, Computer Vision, Natural Language Processing, Graph Neural Network, CUB dataset**

I. Introduction

In the past decade, Generative Adversarial Networks (GANs) have achieved remarkable success across various domains of artificial intelligence. One area where GANs have particularly excelled is text-to-image synthesis, demonstrating their ability to generate visually compelling images from textual descriptions [1] The significant advancements in this field have attracted increasing attention from researchers and practitioners alike, spurring further innovations and applications.

While recent years have seen the emergence of diffusion models, which have shown impressive results in image generation tasks [2], GANs continue to be a fertile ground for exploration and improvement in text-to-image synthesis. The unique adversarial training mechanism of GANs, coupled with their ability to generate high-quality images efficiently, ensures their ongoing relevance in this rapidly evolving field.

The persistent interest in GAN-based approaches for text-to-image synthesis is driven by several factors: their potential for fine-grained control over generated images, their capacity for handling complex semantic relationships, and their adaptability to various downstream tasks. As such, despite the rise of alternative methods, GANs remain a crucial area of research, offering promising avenues for advancing the advanced methods in text-to-image synthesis [3], [4].

*These authors contributed equally to this work.

Fig. 1: Examples of generated images

Although previous research has demonstrated impressive results, significant challenges remain in the field of text-to-image generation. One of the most significant challenges is maintaining semantic consistency, which directly affects the similarity between a generated image and its corresponding textual caption [1]. To address this challenge, researchers have proposed a variety of approaches. Among them, the Deep Attention Multimodal Similarity Model (DAMSM) [5] is an important attempt to quantify the semantic differences between the generated image and the text by means of a specialized loss function to improve the consistency. Then based on this, the semantic space-aware generative adversarial network (SSA) [6] further enhances the process. SSA dramatically improves the expressiveness of the image details by incorporating semantic information at the pixel level, while maintaining semantic consistency. With these improvements, the overall semantic fidelity of the generated images is significantly enhanced.

In addition, many early studies employed a multi-stage optimization framework that progressively optimizes the details of the images by applying fine-grained word embedding on the initial noisy images. However, this multi-stage approach not only increases the computational overhead due to the introduction of generators and discriminators at each stage, but also leads to instability in the training process. What's more, the quality of the final generated image is largely dependent on the effect of the image generated in the earliest stage. To cope with these problems, Tao et al. proposed a simplified single generator-discriminator architecture [1], which solves the above problems to a certain extent.

While recent advancements have significantly improved the general synthesis capabilities of text-to-image (T2I) tasks, several limitations remain. Firstly, most current approaches rely on a single textual description to generate images [6], [7]. Although this may be sufficient to create a basic image, it

often lacks the level of detail needed for higher image quality and precision. Integrating richer and more detailed descriptions could further enhance the clarity and fidelity of the synthesized images. Secondly, sentence structure plays a crucial role in T2I tasks. Previous research [8] has demonstrated that word embeddings enriched with syntactic dependency information significantly improve performance across various tasks. Similarly, incorporating syntactic context has been shown to enhance the quality of word embeddings [9]. Despite these findings, little attention has been paid to leveraging sentence structure information in current T2I models.

To address the limitations outlined above, we made use of a novel one-stage text-to-image (T2I) backbone that utilizes a single generator and discriminator at each stage. This streamlined approach effectively reduces computational costs while still generating high-resolution images, as validated in prior work [1]. Additionally, inspired by Graph Neural Networks (GNNs) [10], [11], we introduce a GNN-based variation called the Semantic-Syntactic Graph Fusion GAN (SSGF-GAN). This model features a semantic-syntactic graph fusion block, specifically designed to produce more representative text embedding, ultimately leading to higher-quality image generation. Fig. 1 shows the generated images by our method.

In summary, our main contributions are as follows:

- We propose the Semantic-Syntactic Graph Fusion GAN (SSGF-GAN), featuring an innovative semantic-syntactic graph fusion block. This block integrates semantic structure information between word nodes and sentence nodes, constructing a graph that facilitates the complementarity and transfer of semantic and sentence structure data. Our model also supports image generation from multiple captions, leveraging the SSGF block and an attention mechanism to effectively highlight important features within sentences, resulting in richer and more detailed visual content.
- We incorporate grammatical structure embedding and a GNN-based semantic complementing model. This approach includes finer-grained word-level information, allowing words to indirectly influence the generated images, while also providing lower complexity compared to models that handle sentence and text information separately. This integration enhances the model's ability to perceive and utilize relationships between different parts of a sentence.

II. Related Work

The text-to-image task has always been one of the most challenging tasks in computer vision. In recent years, with the breakthrough of deep learning, the related images processing techniques have been improved to the next level [12]. In the section, Generative Adversarial Network (GAN)-based synthesis, text-to-image (T2I) GAN will be comprehensively discussed.

A. GAN-based Synthesis

As one of the main methods for various computer vision tasks [13], GAN has developed greatly in recent years, derived from many derivative models with slightly different functions, and has demonstrated powerful capabilities in many fields of computer vision. For example, image-to-image translation [14]–[18], super-resolution [19]–[22], and more importantly, which is also the subject of the paper, text-to-image synthesis. For the text-to-image task, whose purpose is to generate a suitable image based on the text caption, it can be regarded as the inverse process of image captioning [12].

B. Text-to-Image GAN

In 2016, the pure text-to-image synthesis first came to the world [23]. Ever since then, more and more work have been done in the field. Some of the more representative ones are as follows. In the year 2017, a two stage StackGAN had been published by [24], which ultimately generate image with 256 * 256 pixels on the base of 64 * 64-pixel image in the first stage. Later, AttnGAN by [5] introduced attention mechanism to the GAN, which implemented more sophisticated text image generation. Besides, MirrorGAN, which introduced in 2019, is a brand new GAN framework, which improved the generation quality by re-describing the generated images [25]. Even though these works have made great development in the field, the text-to-image tasks still facing a series of challenges: Generating scenes containing multiple complex objects with text-based descriptions is still far from ideal. Besides, there has also been limited work extending these methods to resolutions higher than 256 x 256 [12]. While diffusion and autoregressive models (such as DALL-E [26] and Stable Diffusion [27]) that rely on iterative inference have become the dominant paradigm in recent years, they suffer from the disadvantage of high inference costs. On the other hand, GAN is more efficient because its words are passed forward to generate images. As a result, many text-to-image tasks are still unfolded with GANs [28]. As for improving the image quality, DF-GAN [1] modified the architecture of the GAN, which made sure the high quality image generation. Affine transformation lies in the Recurrent Affine Transformation, which connects different fusion blocks using a recurrent neural network to achieve global assignment of text information during image generation, resulted in the improvement of the consistence between images and descriptions [29]. Meanwhile, XMC-GAN [30] enhanced cross-modal learning, CP-GAN [31] improved text comprehension and content parsing, and LAFITE [32] explored the implementation of supervise learning on GAN.

III. Method

In this paper, we aim to develop a GAN model that effectively utilizes all the captions associated with an image, ensuring that each sentence complements the others through the proposed Semantic-Syntactic Graph Fusion (SSGF). We begin by introducing the pre-trained text encoder and syntax embedding in Section 3.1. Next, in Section 3.2, we detail the construction of a generator utilizing semantic-syntactic graph

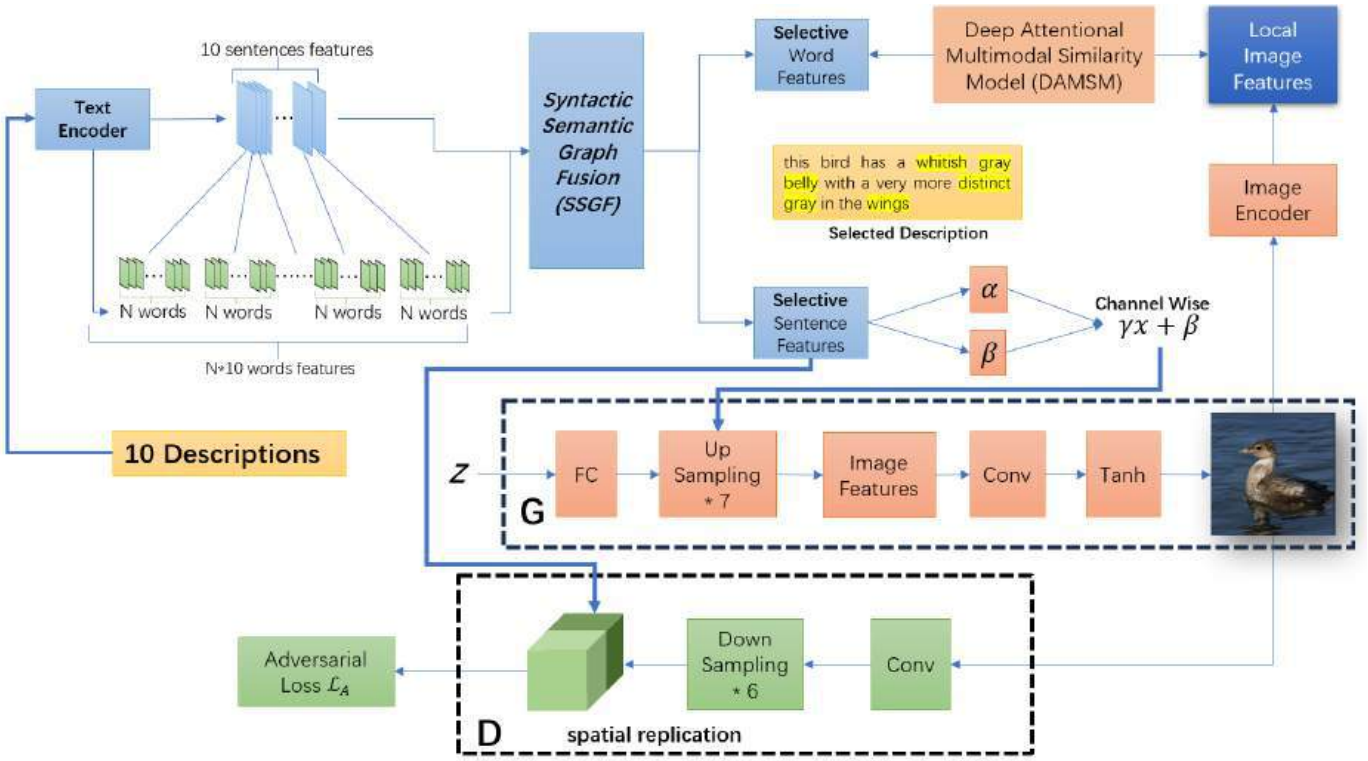


Fig. 2: The figure shows the flowchart of the text-to-image generation process using our proposed method. The model processes multiple text descriptions into word and sentence features, which are fused in the SSGF block. Selective features are extracted for image generation, guided by the Deep Attentional Multimodal Similarity Model (DAMSM), and refined through a generator (G) and discriminator (D) network with adversarial loss.

fusion. Finally, in Section 3.3, we describe the development of a discriminator incorporating spatial attention. Fig. 2 presents the task flowchart, outlining the process from SSGF to the final image generation.

A. Embedding Layer

Building on the work of [5], we adopt a pre-trained text encoder. This encoder is a bi-directional LSTM [33] trained using the Deep Attentional Multimodal Similarity Model (DAMSM) [5]. The output sentence feature has a dimension of $e^s \in \mathbb{R}^{256}$, while the word features are represented as $e^w \in \mathbb{R}^{18\times256}$, with a length of 18. For each image, we use 10 captions at a time.

We also take into account dependency relations and their types. The adjacency matrix is defined as $A = adj_{i,j}n \times n$, where $adj_{i,j} = 1$ if there is a syntactic relation between nodes, derived from dependency relations that include 45 different syntax relations. Additionally, beyond this word adjacency matrix, we introduce a non-directed edge between word nodes and their corresponding sentence node, as well as a directed edge from the sentence nodes to the selected sentence node. This implies that $adj_{w,cs}$, $adj_{cs,w}$, and $adj_{ss,os}$ will always equal 1, where *w* represents a word, *cs* denotes the corresponding sentence, and *ss* signifies the selected sentence. To account for each node's self-connection, we derive $\hat{A}$ using Eq. (1).

Besides, a dependency matrix, $D = d_{i,j}{}_{n\times n}$, is employed to record the dependency types, with each type's embedding denoted as e^{d_i}, where $e^{d_i} \in \mathbb{R}^{256}$. In total, the model incorporates 49 distinct relations, including a self-loop type.

$$\hat{A} = A + I_n \tag{1}$$

where I_n is the $n \times n$ identity matrix.

B. Semantic-Syntactic Graph Fusion (SSGF)

The structure of the Semantic-Syntactic Graph Fusion (SSGF) is hierarchical, as illustrated in Fig. 3. In the first layer of SSGF, all nodes correspond to words, where the *i*-th node is represented as $e_i^w \in \mathbb{R}^{256}$. In the second layer, the nodes correspond to sentences, represented as $e_i^s \in \mathbb{R}^{256}$. Our model's backbone consists of seven SSGF blocks to enhance the performance of the Affine Transformation. To mitigate issues of gradient vanishing and to promote stable training, Layer Normalization (LayerNorm) [34] is applied to both sentence and word nodes, as shown in Eq. (2).

$$[\hat{e_i}^s; \hat{e_j}^w] = \textit{LayerNorm}([e_i^s; e_j^w]) \tag{2}$$

Where $e_i{}^s$ is the *i*-th word inside one sentence, *LayerNorm* is the layer normalization.

Attentional Semantic Fusion Since the pretrained features outcome from text encoder have already contained semantic

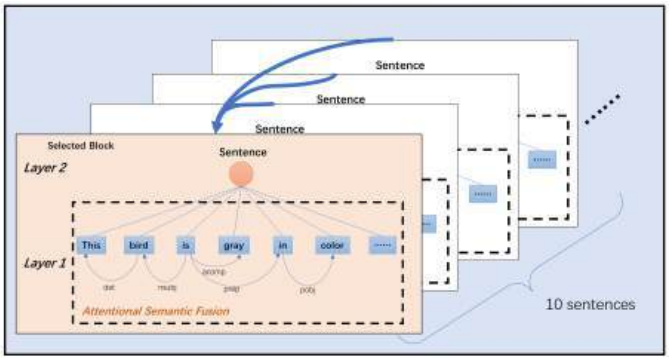


Fig. 3: Figure illustrating the work process of SSGF block. The model processes each sentence by extracting syntactic and semantic relationships between words, enhancing the representation of the selected block. This approach integrates detailed sentence-level information to improve overall semantic understanding.

meanings, we freeze all the contextual representations. To make it suitable for SSGF block, we use the fully connected layer further encode the features from text encoder attention scores $\alpha_{i,j}^{w}$ and the attention scores among words within each sentence is computed by Eqs. (3)-(5).

$$\hat{e_i}^{w_q} = \hat{W_q} e_i^{w} + \hat{b_q} \tag{3}$$

$$\hat{e_i}^{w_k} = \hat{W_k} e_i^{w} + \hat{b_k} \tag{4}$$

$$\alpha_{i,j}^{w} = \frac{\exp(\hat{e_i}^{w_q} \cdot \hat{e_j}^{w_k})}{\sum_{j=1}^{n} \exp(\hat{e_i}^{w_q} \cdot \hat{e_j}^{w_k})} \tag{5}$$

Where $\hat{W_q}$, $\hat{W_k}$ are the trainable weights, $\hat{b_k}$, $\hat{b_q}$ are the bias terms. $\alpha_{i,j}^{w}$ means the similarity between *i*-th and *j*-th words. This semantic attention matrix will be used as adjacency matrix in words-level, and applied multiple layers of GCN models using Eq. (6).

$$h_i^{w(l^w+1)} = \sigma\left(\sum_{j=1}^{n} \alpha_{i,j} \left(W_w^{(l^w+1)} h_j^{w(l^w)} + b_w^{(l^w)}\right)\right) \tag{6}$$

Where l^s means the number of GCN layer, $h_j^{w(l^w)}$ is the hidden features of *j*-th word, $W_w^{(l^s)}$ is the trainable weights, $b_w^{(l^s)}$ is the bias term, and σ is the activation function, i.e., *ReLU*.

Syntactic-Semantic Fusion Next, we proceed to the syntax-semantic fusion stage. By integrating both syntax types and semantic types into a single adjacency matrix, we can utilize the same GCN layer to fuse the features effectively. Following the approach of [11], we encode each type with a weight matrix and a bias, applying an affine transformation to the target node as described in Eq. (7).

$$h_v^{(l^t+1)} = \sigma\left(\sum_{u \in \mathcal{N}_+(v)} \left(W_{luv}^{(l^t+1)} h_u^{(l^t)} + b_{luv}^{(l^t)}\right)\right) \tag{7}$$

Where l^t means the number of GCN layer, $h_u^{(l^t)}$ is the hidden features of node *u* in layer l^t, $W_{luv}^{(l^t)}$ and $b_{luv}^{(l^t)}$ are the trainable parameter and bias term for type where edge from node u to node v, $\mathcal{N}_+(v)$ refers to the immediate neighbors of node *v* (including *v* itself).

Finally, we extract a single sentence along with its corresponding words and train it using DAMSM [5], ensuring that the parameters of both the text encoder and image encoder remain frozen. After training, we obtain features that encapsulate crucial text information from 10 different captions as well as the syntactic details for each word. Additionally, these global sentence features, denoted as $\bar{e}^s$, contain each local word feature $\bar{e_i}^w$, and conversely, each local word feature also incorporates the global sentence features.

C. Channel-Wise Affine Transformation

In [1], the authors integrate the concepts of CBN [35] and AdaIN [36], [37]. Given an input image $x \in \mathbb{R}^{N \times C \times H \times W}$, where *N* represents the batch size, *C* is the number of channels, and *H* and *W* denote the image's height and width, we first normalize the image using Eqs. (8)-(10) for CBN. After normalization, a channel-wise affine transformation is applied, as described in Eq. (11).

$$\mu_B^c = \frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} x_{nhwc} \tag{8}$$

$$\sigma_B^c = \sqrt{\frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_{nhwc} - \mu_B^c)^2 + \epsilon} \tag{9}$$

$$\hat{x}_{nhwc} = \frac{x_{nhwc} - \mu_B^c}{\sigma_B^c} \tag{10}$$

$$y_{nhwc} = \gamma_c \hat{x}_{nhwc} + \beta_c \tag{11}$$

Where ϵ is a small positive constant value, γ_c and β_c comes from $MLP_\gamma([z; \bar{e}^s])$ and $MLP_\beta([z; \bar{e}^s])$, z is the generator's noise vector. Note that since we have a hierarchical GNN model to combine all semantic and syntactic information to one sentence features, which means besides global sentence vector $\bar{e}^s$, all the other nodes will indirectly decide the γ_c and β_c.

It is worth mentioning that in [1], the authors argue that the normalization process, as defined in Eqs. (8)-(10), can counteract the effects of the Affine Transformation by increasing the distance between different samples. Based on this insight, we also opted not to use batch normalization in our model, instead directly applying the Affine Transformation, similar to the approach described in Eq. (11).

D. Discriminator

Following the approach in [1], we employ a one-way discriminator due to its simplicity and effectiveness. This discriminator concatenates the sentence features extracted from our SSGF module with the generated image features. Additionally, it utilizes the Matching-Aware Gradient Penalty [1], which aids the generator in synthesizing images that better match the text descriptions.

E. Objective Functions

In the objective function of the discriminator, we utilize hinge loss combined with the Matching-Aware Gradient Penalty (MA-GP) [1], as formulated in Eq. (12). The corresponding generator loss consists of both an adversarial loss and a DAMSM loss [5], as detailed in Eq. (13).

$$\begin{aligned} L_{\mathrm{D}} = & \mathbb{E}_{x \sim p_{\text{data}}}[\max(0, 1 - D(x, s))] \\ & + \frac{1}{2}\mathbb{E}_{x \sim p_G}[\max(0, 1 + D(\hat{x}, s))] \\ & + \frac{1}{2}\mathbb{E}_{x \sim p_{\text{data}}}[\max(0, 1 + D(x, \hat{s}))] \\ & + \lambda \mathbb{E}_{x \sim p_{\text{data}}}\left[\left(\|\nabla_x D(x, s)\|_2 + \|\nabla_s D(x, s)\|_2\right)^P\right] \end{aligned} \tag{12}$$

$$L_{\mathrm{G}} = -\mathbb{E}_{x \sim p_G}[D(\hat{x}, s)] + \beta L_{DAMSM} \tag{13}$$

Where x is the real image with corresponding text s, $\hat{x}$ is the synthesized image, $\hat{s}$ is the mis-matched text for the real image x, λ, p are the hyper-parameters for MAGP loss, while β is the hyper-parameter of the weight of DAMSM loss.

IV. Experiments

To evaluate our method, we utilized the CUB bird benchmark dataset, comparing its performance against several state-of-the-art GAN methods for text-to-image (T2I) synthesis. We employed a range of metrics to comprehensively assess and validate the effectiveness of our approach.

A. Experiments Details

Experiments Device. This experiment was conducted on a device equipped with a GeForce RTX 4090D GPU with 24GB of RAM, running Ubuntu 20.04.

Data Following previous works [1], [5], [6], [25], [29], [38]–[41], we use the CUB bird dataset [42] for our experiments. The CUB dataset consists of 11,788 images representing 200 bird species, with each image accompanied by ten descriptive captions. We utilize RGB images of size 256×256×3 as input. For the text information, we incorporate all ten captions per image, with each caption containing up to 18 words.

Parameters Setting. We used the Adam optimizer to train our GAN model, with $\beta_1 = 0.0$ and $\beta_2 = 0.9$ for both the generator and discriminator. The learning rate was set to 0.0001 for the generator and 0.0004 for the discriminator, following the setup in DF-GAN. The β parameter in the DAMSM loss [5] was set to 0.05, while the values of p and λ in the discriminator loss were set to 6 and 2, respectively. The model was trained for 600 epochs on the CUB dataset.

Evaluation Metrics. Similar to most text-to-image models, we use the Inception Score (IS) [43] and Fréchet Inception Distance (FID) [44] to evaluate the performance of our model. The IS measures the KL-divergence between the conditional distribution and the marginal distribution, while the FID calculates the Fréchet Distance of the feature distribution between generated images and real images. To evaluate both FID and IS, each model generates 30,000 images at a resolution of 256×256 from captions randomly selected from the test dataset.

Fig. 4: Qualitative comparison with DF-GAN, RAT-GAN, and Ground Truth (GT) on the CUB dataset. Descriptions of the images are provided above.

B. Results and Discussion

We conducted experiments on the CUB dataset and compared our model with existing text-to-image models, including StackGAN++, AttnGAN, DAE-GAN, DM-GAN, DTGAN, DF-GAN, MirrorGAN, SD-GAN, SSAG, and RAT-GAN. The comparison results are presented in Table 1. Higher IS values indicate better model performance, while lower FID values are preferable. The best-performing model is highlighted in red. The '✓' symbol indicates that the model uses word-level features during the fusion of images and captions, which adds complexity, while the '✗' symbol indicates that the model does not use word-level features. Compared to above advanced models, our model achieves improvements in FID and IS, reaching values of 12.32 and 5.38, respectively. The experimental results demonstrate that our model outperforms other models on the CUB dataset in terms of IS and FID, even when using only sentence-level features in text-image fusion processes.

C. Quantitative Results

In this section, we compare the generated images from our model with those from Ground Truth, DF-GAN [1], and RAT-GAN [29] as shown in Fig. 4. With the integration of the SSGF module, our model can comprehensively capture various image characteristics of birds, allowing it to focus more effectively on key aspects of the descriptive text. This results in more detailed bird images that are closely aligned with the descriptions. As seen in the Fig. 4, our model outperforms DF-GAN and RAT-GAN, particularly in the richness of feather details.

TABLE I: Comparison of IS and FID Results on the CUB Dataset

Model	IS ↑	FID ↓	Words
StackGAN++ [24]	4.04	15.30	✗
AttnGAN [5]	4.36	23.98	✓
DAE-GAN [40]	4.42	16.61	✓
DM-GAN [38]	4.75	16.09	✓
DTGAN [41]	4.82	15.87	✗
DF-GAN [1]	5.10	14.96	✗
MirrorGAN [25]	4.56	19.41	✗
SD-GAN [45]	4.57	20.56	✓
SSA-GAN [6]	5.17	15.61	✗
RAT-GAN [29]	5.36	13.91	✗
SSGF-GAN (Ours)	**5.38**	**12.32**	✗

D. Limitation

Although our SSGF-GAN model performs well in the text-to-image task, it does have certain limitations that should not be overlooked. Specifically, our model focuses solely on integrating features at the sentence level, disregarding word-level features. This limitation may affect the model's ability to generate highly detailed, fine-grained images.

V. Conclusion

In this work, we introduced the Syntactic-Semantic Graph Fusion Generative Adversarial Network (SSGF-GAN) to fully leverage textual descriptions in the text-to-image task. By incorporating a novel semantic-syntactic graph fusion block, our model significantly improves the extraction of sentence-level features, leading to the generation of high-quality, semantically consistent images. Experimental results on the CUB dataset showed that our model outperforms other advanced methods.

References

[1] M. Tao, H. Tang, F. Wu, X.-Y. Jing, B.-K. Bao, and C. Xu, "DF-GAN: A simple and effective baseline for text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16515–16525.

[2] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, "Text-to-image diffusion models in generative AI: A survey," *arXiv preprint arXiv:2303.07909*, 2023.

[3] B. Jiang, W. Zeng, C. Yang, R. Wang, and B. Zhang, "DE-GAN: Text-to-image synthesis with dual and efficient fusion model," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 23839–23852, 2024.

[4] V. Talasila, M. R. Narasingarao, and V. M. Mohan, "Modified GAN with proposed feature set for text-to-image synthesis," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 37, no. 04, pp. 2354004, 2023.

[5] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," *arXiv preprint arXiv:1711.10485*, 2017.

[6] W. Liao, K. Hu, M. Y. Yang, and B. Rosenhahn, "Text to image generation with semantic-spatial aware GAN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18187–18196.

[7] S. Haoran, Y. Wang, L. Haipeng, and Q. Biao, "Fine-grained cross-modal fusion based refinement for text-to-image synthesis," *Chinese Journal of Electronics*, vol. 32, no. 6, pp. 1329–1340, 2023.

[8] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308.

[9] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 873–882.

[10] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[11] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," *arXiv preprint arXiv:1703.04826*, 2017.

[12] S. Frolov, T. Hinz, F. Raue, J. Hees, and A. Dengel, "Adversarial text-to-image synthesis: A review," *Neural Networks*, vol. 144, pp. 187–209, 2021.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[14] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017.

[16] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

[17] S. Song, S. Lee, H. Seong, K. Min, and E. Kim, "SHUNIT: Style harmonization for unpaired image-to-image translation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, pp. 2292–2302, Jun. 2023.

[18] D. Torbunov, Y. Huang, H. Yu, J. Huang, S. Yoo, M. Lin, B. Viren, and Y. Ren, "Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 702–712.

[19] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] J. Park, S. Son, and K. M. Lee, "Content-aware local GAN for photo-realistic super-resolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 10585–10594.

[21] Y. Wang, Y. Hu, J. Yu, and J. Zhang, "GAN prior based null-space learning for consistent super-resolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, pp. 2724–2732, Jun. 2023.

[22] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[23] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text-to-image synthesis," in *International Conference on Machine Learning*, 2016, pp. 1060–1069.

[24] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint arXiv:1612.03242*, 2017.

[25] T. Qiao, J. Zhang, D. Xu, and D. Tao, "MirrorGAN: Learning text-to-image generation by redescription," *arXiv preprint arXiv:1903.05854*, 2019.

[26] J. Cho, A. Zala, and M. Bansal, "DALL-Eval: Probing the reasoning skills and social biases of text-to-image generation models," *arXiv preprint arXiv:2202.04053*, 2023.

[27] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," *arXiv preprint arXiv:2112.10752*, 2022.

[28] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park, "Scaling up GANs for text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10124–10134.

[29] S. Ye, H. Wang, M. Tan, and F. Liu, "Recurrent affine transformation for text-to-image synthesis," *IEEE Transactions on Multimedia*, vol. 26, pp. 462–473, 2023.

[30] H. Zhang, J. Yu Koh, J. Baldridge, H. Lee, and Y. Yang, "Cross-modal contrastive learning for text-to-image generation," *arXiv preprint arXiv:2101.04702*, 2022.
[31] J. Liang, W. Pei, and F. Lu, "CpGAN: Content-parsing generative adversarial networks for text-to-image synthesis," in *Computer Vision–ECCV 2020: 16th European Conference*, 2020, pp. 491–508.
[32] Y. Zhou, R. Zhang, C. Chen, C. Li, C. Tensmeyer, T. Yu, J. Gu, J. Xu, and T. Sun, "LAFITE: Towards language-free training for text-to-image generation," *arXiv preprint arXiv:2111.13792*, 2022.
[33] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
[34] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
[35] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[36] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.
[37] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.
[38] M. Zhu, P. Pan, W. Chen, and Y. Yang, "DM-GAN: Dynamic memory generative adversarial networks for text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5802–5810.
[39] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1947–1962, 2018.
[40] S. Ruan, Y. Zhang, K. Zhang, Y. Fan, F. Tang, Q. Liu, and E. Chen, "DAE-GAN: Dynamic aspect-aware GAN for text-to-image synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13960–13969.
[41] Z. Zhang and L. Schomaker, "DTGAN: Dual attention generative adversarial networks for text-to-image generation," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.
[42] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," California Institute of Technology, 2011.
[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.
[44] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[45] Z. Kangneng, X. Zhu, D. Gao, K. Lee, X. Li, and X.-C. Yin, "SD-GAN: Semantic Decomposition for Face Image Synthesis with Discrete Attribute," arXiv preprint, arXiv:2207.05300, 2022. [Online]. Available: https://arxiv.org/abs/2207.05300

An Efficient Attention-Based Deep Reinforcement Learning Model for Traffic Signal Control

Aodi Lin
Department of Automation
University of Science and Technology of China
Hefei, China
audilad@mail.ustc.edu.cn

Feng Chen*
Department of Automation
University of Science and Technology of China
Hefei, China
chenfeng@ustc.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899594

***Abstract*—Traffic Signal Control (TSC) is a significant challenge within intelligent transportation systems. As Vehicle-to-Everything (V2X) technology advances, TSC systems are increasingly able to utilize extensive vehicle driving data to enhance decision-making. This paper proposes a reinforcement learning-based model for TSC at urban intersections. We design a vehicle driving information matrix that integrates vehicle position and speed information. The state space includes vehicle driving information and the current traffic signal phase. The action space includes various traffic signal phases. To efficiently extract traffic features from the large state space, the model incorporates an attention mechanism within the neural network. The simulation results on the Simulation of Urban Mobility (SUMO) demonstrate the convergence and generalization capabilities of our model, showing significant advantages in trip duration compared to several benchmark methods. Ablation studies further validate the effectiveness of our proposed position-speed fusion matrix and attention mechanism in extracting traffic feature representations.**

***Index Terms*—attention mechanism, deep reinforcement learning, traffic signal control, Dueling Double Deep Q-Network**

I. Introduction

Urban TSC is a crucial topic within intelligent transportation systems. As urbanization accelerates and the population grows, the demand for transportation is on the rise, leading to increasingly severe traffic congestion. Strategies to alleviate traffic congestion include improving transportation infrastructure, restricting vehicle access, and optimizing traffic signal control, etc. Among these approaches, optimizing traffic signal control is regarded as the most direct and cost-effective measure for enhancing intersection efficiency and alleviating traffic congestion.

Traditional TSC methods can generally be divided into two categories: fixed-time control and adaptive control. Fixed-time control performs poorly when dealing with dynamic traffic conditions. Widely used adaptive control systems such as SCOOT [1] and SCATS [2] dynamically select pre-designed traffic signal plans based on traffic volumes detected by sensors. However, issues like difficulties in phase design still persist. Some studies have applied reinforcement learning to TSC. However, traditional reinforcement learning often has limitations when dealing with complex state spaces. Deep Reinforcement Learning (DRL) combines the strengths of deep learning and reinforcement learning, utilizing neural networks to automatically extract meaningful features from the state space, enabling precise decision-making in complex traffic environments. This integration allows traffic signal control systems to adapt to real-time traffic flow changes, enhancing the intelligence of traffic management and improving intersection throughput efficiency.

As V2X technology continues to advance, TSC systems can now use more vehicle driving data for their decision-making processes. However, current DRL-based TSC algorithms still predominantly rely on traditional neural network architectures, and their perception capabilities remain insufficient when dealing with large-scale state spaces.

The key contributions of this paper include:

1) A vehicle position-velocity fusion matrix (PV) is proposed to represent vehicle driving information, effectively reducing the dimensionality of the state matrix.
2) A DRL-based TSC algorithm (EA-D3QN) is designed, incorporating an efficient attention mechanism to effectively capture traffic state features and enhance control performance.

II. Related Works

Over two decades ago, reinforcement learning was introduced in the field of TSC [3] [4]. However, due to technological constraints at the time, only small-scale state representations, such as the number of queued vehicles, were typically employed.

With the development of deep learning, DRL has emerged as an effective function approximator for handling large state spaces. W. Genders [5] and X. Liang [6], et al. used vehicle position matrices as the state space for TSC and applied DRL for TSC.

In the era of V2X, the agent of DRL will have access to more detailed vehicle information for decision-making, resulting in a significantly larger state space. DRL models based on traditional neural networks may face challenges in effectively capturing key features within these expanded state spaces. After the introduction of attention mechanisms [7], attention mechanisms have demonstrated great potential in processing language, images [8] [9], and video [10]. Researchers have proposed incorporating attention mechanisms into DRL methods for TSC to enhance the model's ability to

perceive traffic states and improve control effectiveness. The GC_PPO [11] algorithm leverages the Global Context (GC) block to capture spatial correlations in the state, enhancing the model's state perception capabilities. Inspired by SENet, W. Ni integrated the attention mechanism with the D3QN algorithm and proposed D3QN_AM [12]. The CoLight [13] algorithm uses Graph Attention Networks (GAT) to achieve coordinated signal control across multiple intersections. By incorporating two attention mechanisms, AttendLight [14] ensures that the model's input and output are independent of the intersection structure, thereby enabling efficient control across different types of intersections.

III. Proposed Method

A. Problem Definition for TSC

We model the TSC problem as a DRL task, with a detailed design of states, actions, and rewards to meet the needs of intelligent transportation environments. This framework lays the foundation for the model to learn optimal control strategies.

1) State: In the TSC task, state definitions determine the model's capability to perceive the environment. To fully leverage the extensive data provided by connected vehicles while minimizing the size of the state matrix, we propose using a fused Position-Velocity (PV) matrix to represent vehicle driving states. As illustrated in Fig. 1, all incoming and outgoing roads at the intersection are divided into uniform grid cells, each 5 meters in length. For each cell within the PV matrix, the value signifies the normalized velocity of a vehicle, defined as the quotient of the vehicle's instantaneous velocity and the maximum permissible velocity on the road, when the cell is occupied by a vehicle; in the absence of a vehicle, the cell is assigned a value of -1. The PV matrices of all roads are concatenated to form the intersection's PV matrix.

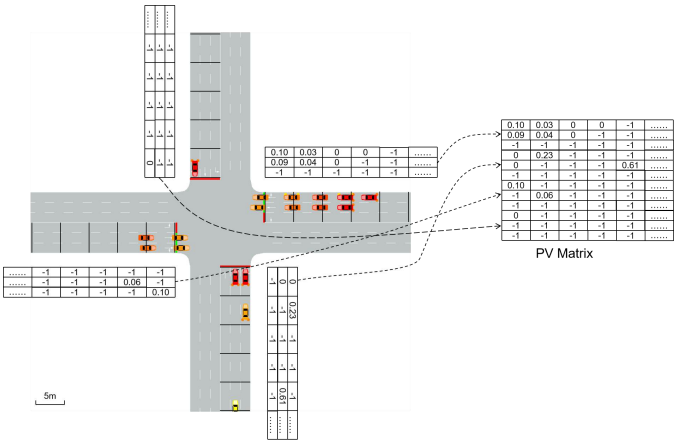


Fig. 1. Intersection Structure and PV Matrix. In the PV matrix, each row corresponds to a distinct lane, with each cell representing a 5-meter segment of the road.

The current signal phase is also an important piece of state information. In this study, we utilize a four-phase signal design, $Phase = \{NSG, NSLG, EWG, EWLG\}$, as illustrated in Fig. 2. Here, NSG denotes the north-south straight-ahead phase, $NSLG$ denotes the north-south left-turn phase, EWG denotes the east-west straight-ahead phase, and $EWLG$ denotes the east-west left-turn phase. The current signal phase is encoded as a one-hot vector, enabling the model to accurately perceive the signal status at the intersection.

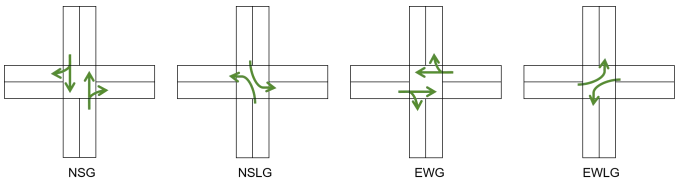


Fig. 2. Traffic signal phase.

2) Action: We define actions as selecting a signal phase from the phase set $Phase$ and executing it for 15 seconds. For safety, a 3-second yellow light is automatically inserted whenever the phase changes.

3) Reward: In this study, the reward r_t of the action is defined as the reduction in the number of waiting vehicles q.

$$r_t = \sum_l q(t,l) - q(t+1,l) \tag{1}$$

Here, t denotes the time step at which the action is executed, while $t+1$ refers to the time step occurring 10 seconds after the action is executed. The variable l represents the lane of the road. This reward function provides immediate feedback on each action's effectiveness, encouraging the model to choose strategies that minimize vehicle queue lengths, thereby reducing congestion and improving traffic signal control performance.

B. EA-D3QN

Inspired by Efficient Attention (EA), we designed an algorithm called EA-D3QN, which integrates Dueling Double Deep Q Network (D3QN) [15] with EA mechanisms to process complex traffic state information more effectively, thereby enhancing the accuracy and efficiency of control decisions.

1) Q-Network: As illustrated in Fig. 3, the Q-network is designed using a Dueling DQN architecture. The input consists of a PV matrix, which encodes vehicle information, and a one-hot vector representing the current traffic signal phase. The PV matrix is first processed through EA layer, flattened, and then concatenated with the phase vector. This combined representation is then passed through a hidden layer. Subsequently, the output is fed into the two branches of the Dueling architecture: one branch computes the state-value function $V(s)$, while the other calculates the advantage function $A(s,a)$ for each possible action in the given state. The final Q-value is obtained using the following aggregation formula:

$$Q(s,a) = V(s) + A(s,a) - \frac{1}{|A|}\sum_{a'} A(s,a') \tag{2}$$

Here, s denotes the state and a denotes the action.

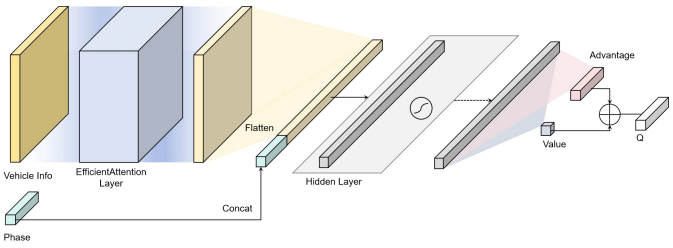


Fig. 3. Q-Network Architecture.

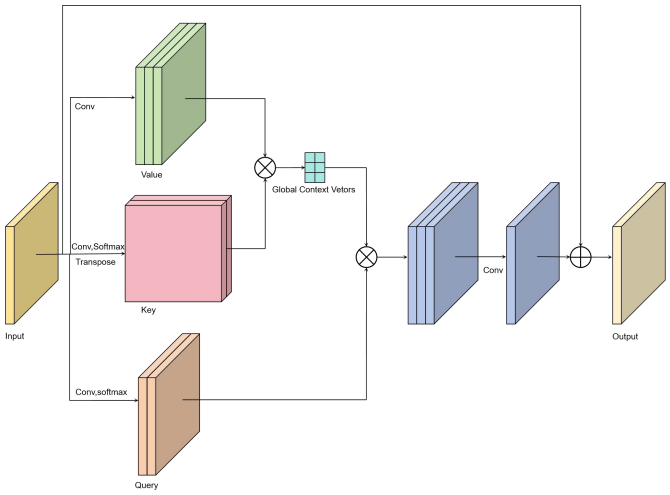


Fig. 4. Structure of the Efficient Attention Layer.

2) EA Network: The EA [16] mechanism is a type of linear attention mechanism designed to optimize computational and memory efficiency in large-scale neural networks which is mathematically equivalent to dot-product attention [7]. Unlike Dot-product attention mechanisms, which often involve quadratic complexity in relation to the input size, EA reduces this to linear complexity, making it highly suitable for applications with substantial data.

As shown in Fig. 4, the input pass through three convolutional layers to form the queries $Q_{atn} \in \mathbb{R}^{n \times d_k}$, keys $K_{atn} \in \mathbb{R}^{n \times d_k}$, and values $V_{atn} \in \mathbb{R}^{n \times d_v}$. The efficient attention mechanism is characterized by:

$$E(Q_{atn}, K_{atn}, V_{atn}) = \rho_q(Q_{atn})\Big(\rho_k(K_{atn})^\intercal V_{atn}\Big) \tag{3}$$

where ρ is a normalization function.

The output E is first transformed through a convolution to match the input dimensions, and then added to the input to obtain the final output.

3) Training Process: The model is trained using the Double DQN [17] framework. The agent selects actions using the epsilon-greedy strategy, with epsilon gradually decaying throughout the training process. In the early stages of training, the agent tends to explore a variety of actions, while as training progresses, it increasingly favors actions with higher Q-values. Once an action a is chosen based on the current state s, the agent obtains the associated reward r and the subsequent state s' from the environment, then stores the tuple (s, a, r, s') in the experience replay buffer. The experience replay technique effectively breaks the correlation between samples, thereby improving sample efficiency. Training starts once the buffer contains sufficient data, with a random batch of experiences sampled at each step for training.

In DQN, action selection and Q-value calculation rely on the target network Q, which may lead to Q-value overestimation. To address this, Double DQN modifies the approach: action selection is based on the training network, while the Q-value is calculated using the target network. To ensure training stability, the target network's parameters ω^- are synchronized with the training network's parameters ω every fixed number of steps.

The loss function for the Double DQN Q-network can be expressed as:

$$L(\omega) = \mathbb{E}_{s_t, a_t, r_t, s_{t+1}}\left[\left(y_t - Q_\omega(s_t, a_t)\right)^2\right] \tag{4}$$

where the target y_t is computed as follows:

$$y_t = r_t + \gamma Q_{\omega^-}(s_{t+1}, \arg\max_{a'} Q_\omega(s_{t+1}, a')) \tag{5}$$

In this expression, r_t is the reward obtained by the agent for taking action a_t in state s_t, and γ is the discount factor, which reflects the relative importance of future rewards. The Q-value for all actions in the next state s_{t+1} is estimated by the training network as $Q_\omega(s_{t+1}, a')$, and the target value is calculated by selecting the action with the highest Q-value $\arg\max_{a'} Q_{\omega^-}(s_{t+1}, a')$ using the target network.

The pseudocode of the training process is shown in Algorithm 1.

IV. Simulation Experiment

A. Experiment Setup

We conduct simulation experiments on SUMO, which provides Traffic Control Interface (TraCI), allowing us to obtain the current vehicle states and traffic signal phases in the simulation, as well as control the phase transitions of the traffic signals.

As shown in Fig. 2, we set the number of traffic signal phases as 4. Each phase lasts for at least 15 seconds. If a phase transition occurs, a 3-second yellow light is inserted.

B. Datasets

We experiment with a series of real-world datasets based on camera data from Hangzhou. The dataset consists of 11 traffic flow records collected across different time periods from five intersections. The turning ratios in each dataset are set to approximately 14.3% for left turns and 85.7% for through traffic, with right-turning vehicles excluded as they are typically not controlled by traffic signals. Each scenario in the dataset consists of a four-way intersection, with each approach direction having two lanes: one dedicated to through traffic and the other to left turns. The length of each lane is 300 meters.

Algorithm 1 EA-D3QN

Initialize the training network $Q_\omega(s,a)$ with random parameters ω and the target network Q_{ω^-} by $\omega^- \leftarrow \omega$
Initialize the experience replay buffer R with minimum replay buffer size $minR$ and Maximum replay buffer size $maxR$
Initialize the target network update interval c, training steps $count$, discount factor γ, training episodes E, steps per episode T
for $e = 1 \rightarrow E$ **do**
  Get the initial state of environment s_1
  for $t = 1 \rightarrow T$ **do**
    Select action a_t according to the $\epsilon-$greedy policy based the training network $Q_\omega(s.a)$
    Execute action a_t, receive reward r_t, and the environment state transitions to s_{t+1}
    Store (s_t, a_t, r_t, s_{t+1}) in R
    if the size of R $> minR$ **then**
      Sample$\{(s_t, a_t, r_t, s_{t+1})\}_{t=1,\ldots,N}$ from R
      Select $a' = \arg\max_a Q_\omega(s_{t+1}, a)$ using the online network
      Compute $y_t = r_t + \gamma Q_{\omega^-}(s_{t+1}, a')$ using the target network
      Compute the loss $L(\omega)$ by (4)
      Update the training network Q_ω
      if $count\%c == 0$ **then**
        $\omega^- \leftarrow \omega$
      end if
      $count \leftarrow count + 1$
    end if
  end for
end for

C. Compared Methods

To assess the effectiveness and efficiency of our proposed method, we compare it with the following approaches.

- FixedTime [18]: Fixed-time control utilizes a predefined cycle and phase duration schedule, commonly applied in steady traffic conditions.
- SOTL [19]: This approach adaptively adjusts traffic light timings by utilizing a manually defined threshold based on the number of vehicles waiting at the intersection.
- D3QN: A TSC Method based on D3QN, which state, action, and reward are consistent with this study.
- EA-D3QN-NoPV: An ablation experiment variant of our proposed method, which replaces the fused vehicle position PV matrix with separate position and velocity matrices, to evaluate the effectiveness of the PV matrix.

D. Evaluation Metrics

In TSC, the average trip duration is a key metric for evaluating the efficiency of the traffic system and the effectiveness of intelligent TSC strategies. It represents the mean time, in seconds, for vehicles to travel from the origin to the destination. A shorter average trip duration indicates smooth traffic flow and higher throughput, while a longer duration may suggest traffic congestion or inefficiencies in the signal control strategy.

E. Performance Evaluation

For each algorithm, we train the model using the dataset named "bc-tyc_07". Subsequently, for each dataset, we select the final five sets of model parameters for testing. The average of these test results is then taken as the algorithm's final score on that dataset. Then, for each dataset, we select the final five sets of model parameters for testing, and the mean of these test results is used as the final performance score of the algorithm on that dataset. Fig. 5 illustrates that EA-D3QN performs better than the FixedTime and SOTL control methods.

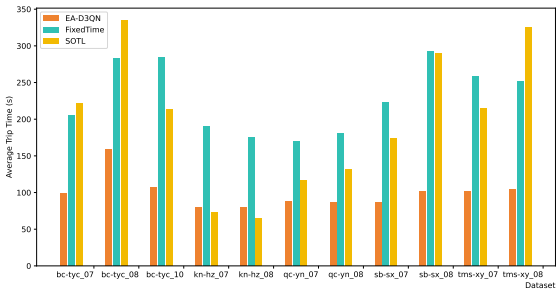


Fig. 5. Performance Evaluation. The x-axis denotes the names of the datasets, while the y-axis indicates the average trip time in seconds. The bars represent the average trip time for each dataset using the EA-D3QN, FixedTime, and SOTL algorithms.

We designed and conducted ablation experiments with D3QN and EA-D3QN as the reference groups. Fig. 6 presents the testing results during the training process. A comparison shows that EA-D3QN exhibits smoother curves and converges faster than D3QN and EA-D3QN-NoPV. Furthermore, as shown in Fig. 7, EA-D3QN typically achieves shorter average trip durations than D3QN and EA-D3QN-NoPV, which validates the effectiveness of the proposed PV matrix in state representation and the EA module in state extraction.

V. Conclusion

In this paper, the authors propose an algorithm named EA-D3QN, which simplifies traffic state representation using PV and integrates an EA mechanism to address the challenges of TSC. The experimental results demonstrate that, compared to the fixed-time, SOTL, and D3QN methods, the proposed EA-D3QN algorithm provides superior control performance for traffic signal control. Additionally, ablation studies validate the effectiveness of the proposed PV matrix and EA mechanism.

In the future, we will attempt to research the multi-intersection traffic signal control problem. Additionally, more vehicle information, such as turning intentions, should be incorporated.

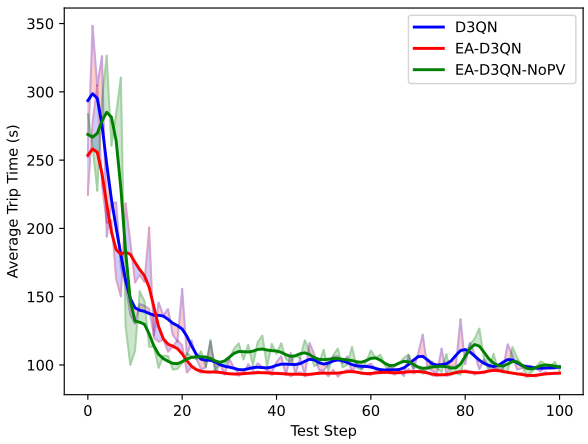


Fig. 6. Average Trip Duration for Different Methods during Training. This figure illustrates the average trip duration for the D3QN, EA-D3QN, and EA-D3QN-NoPV algorithms at various stages of training, with testing conducted every 10 epochs. The x-axis represents the test steps , and the y-axis indicates the average trip time in seconds.

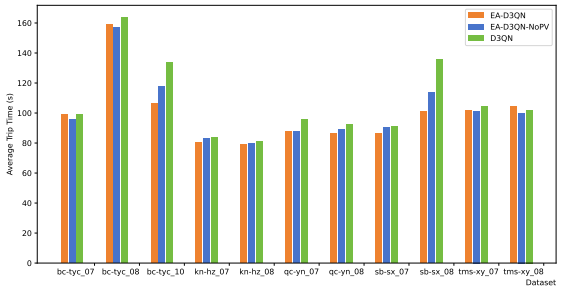


Fig. 7. Results of the Ablation Study. The x-axis represents the dataset names, and the y-axis represents the average trip time in seconds. The bars compare the average trip time for each dataset using three different algorithms: EA-D3QN, EA-D3QN-NoPV, and D3QN.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Union Fund of China (U1811463).

REFERENCES

[1] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "Scoot - a traffic responsive method of coordinating signals," *trrl laboratory report*, 1981.

[2] L. P. SCATS, "the sydney coordinated adaptive traffic system—principles, methodology, algorithms," in *Proceedings of the IEE International Conference on Road Traffic Signalling*, 1982, pp. 67–70.

[3] M. A. Wiering *et al.*, "Multi-agent reinforcement learning for traffic light control," in *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, 2000, pp. 1151–1158.

[4] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[5] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," *arXiv preprint arXiv:1611.01142*, 2016.

[6] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," *arXiv preprint arXiv:1803.11115*, 2018.

[7] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[8] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[10] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[11] N. ZHANGXijun, L. Zhe, and Z. Hong, "Traffic signal control with deep reinforcement learning and self-attention mechanism," *Journal of Transportation Systems Engineering and Information Technology*, vol. 24, no. 2, p. 96, 2024.

[12] W. Ni, P. Wang, Z. Li, and C. Li, "Traffic signal control optimization based on deep reinforcement learning with attention mechanisms," in *International Conference on Neural Information Processing*. Springer, 2023, pp. 147–158.

[13] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1913–1922.

[14] A. Oroojlooy, M. Nazari, D. Hajinezhad, and J. Silva, "Attendlight: Universal attention-based reinforcement learning model for traffic signal control," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4079–4090, 2020.

[15] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[16] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li, "Efficient attention: Attention with linear complexities," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3531–3539.

[17] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[18] A. J. Miller, "Settings for fixed-cycle traffic signals," *Journal of the Operational Research Society*, vol. 14, no. 4, pp. 373–386, 1963.

[19] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," *Advances in applied self-organizing systems*, pp. 45–55, 2013.

EAMS-YOLOv8: An Object Detection Algorithm for Drone Aerial Images

1st Yanyang Han
College of Computer Science, South-Central Minzu University
Wuhan, China
kahan2001@scuec.edu.cn

2nd Mian Zheng
China Ship Development and Design Center
Wuhan, China
zyclzmm@163.com

3rd Xingyu Wu
College of Computer Science, South-Central Minzu University
Wuhan, China
wuxingyux@163.com

4th Yimin Feng
College of Computer Science, South-Central Minzu University
Wuhan, China
15997343048@163.com

5th Liyang Zhang
College of Computer Science, South-Central Minzu University
Wuhan, China
1005903625@qq.com

6th Chendong Ding
College of Computer Science, South-Central Minzu University
Wuhan, China
dingcd7786@163.com

7th Yunlong Shi
College of Computer Science, South-Central Minzu University
Wuhan, China
1656048231@qq.com

8th Jing Liu*(corresponding author)
College of Computer Science, South-Central Minzu University
Wuhan, China
jingliu@scuec.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899483

***Abstract*—Object detection in drone aerial images encounters numerous challenges, notably the presence of small objects occupying minimal pixel areas, as well as occlusions and overlaps among objects, leading to a reduction in detection accuracy. To tackle these issues, we present EAMS-YOLOv8, an enhanced model derived from the YOLOv8n. The innovations of EAMS-YOLOv8 are fourfold: First, the Efficient-C2F block is incorporated into the backbone network, replacing the C2F block. This enhancement bolsters the capability of feature extraction and facilitates the extraction of multi-scale features. Second, the SPPF block is replaced by the Adaptive-SPPF block, which employs diverse adaptive pooling to capture global and local information and decreases information loss during pooling. Furthermore, a layer of up-sampling and concatenation operations is added to change the connection of the neck network to enrich the feature information of the small object layer. Finally, the SEAM attention mechanism is introduced before the detection head to alleviate the occlusion problem among objects and enhance the recognition and localization capability of EAMS-YOLOv8. Experiments show that EAMS-YOLOv8 improves mAP by approximately 7% on the VisDrone2019 dataset compared to YOLOv8n, effectively increasing detection accuracy in complex and dense scenes, and providing a practical solution object detection area.**

Keywords—Object detection, Drone aerial images, YOLOv8n, VisDrone2019

I. INTRODUCTION

In recent years, various detection models have emerged with the escalating demand for real-time capabilities and high-precision tasks. Broadly, object detection methodologies can be classified into two categories: single-stage and two-stage models. Single-stage models streamline the process by immediately identifying objects after feature extraction, with representative models including the You Only Look Once (YOLO)[1] and the Single Shot MultiBox Detector (SSD)[2]. In contrast, two-stage models initially generate potential object regions, leveraging either traditional algorithms like Selective Search (SS)[3] or deep learning techniques, such as Region Proposal Networks (RPN)[4]. Then, the final object detection is performed on these generated regions, with representative models including Cascade R-CNN[5].

With the development of deep learning, object detection has been widely applied in various scenarios, such as aerial scenarios[6], land scenes[7], liquid environments[8], and industrial applications[9]. Nevertheless, detecting small objects faces numerous challenges in these scenes. In aerial scenarios, these challenges intensify due to factors such as object occlusion and overlap, limited image resolution, intricate and dynamic backgrounds, and susceptibility to environmental fluctuations. These issues conspire to complicate the task of achieving precise object detection in drone aerial images[10]. To solve these problems, this paper proposes an EAMS-YOLOv8 model, an enhancement of the YOLOv8n framework. Firstly, the Efficient-C2F block and Adaptive-SPPF block within the backbone network are improved to enhance its feature extraction capability. Secondly, the connection method within the neck network is modified to further boost its feature representation. In addition, we introduce the SEAM[11] attention mechanism, which combines depthwise separable convolution and pointwise convolution to improve the capability in occlusion problems. The main contributions of this paper are as follows.

- We propose the Efficient-C2F block, which integrates global and local features, thereby enhancing feature extraction and reducing information loss in the backbone network.
- We propose the Adaptive-SPPF block, which alleviates information loss for small objects by combining multiple pooling operations.
- We added a layer of up-sampling and a concatenation operation in the neck network, and introduced the SEAM attention mechanism to accurately focus on crucial regions, improving the accuracy of object localization and recognition.

II. METHODOLOGY

This paper proposes several initiatives to improve the model. Firstly, the C2F block in the backbone network is replaced with the Efficient-C2F block, within each C2F sub-block convolution replaced by dilated convolution[12], enhancing the capability of feature extraction. Secondly, the SPPF block is refined into the Adaptive-SPPF block, enabling more nuanced processing of

feature information. In addition, a layer of concatenation and up-sampling operations are added to the neck network to enrich the semantic and spatial information of the small object layer. Finally, before the detection head, the SEAM attention is introduced to enable the model to focus on crucial regions with greater precision. Fig .1 shows our proposed EAMS-YOLOv8 network model, with detailed descriptions of the specific improvement blocks provided in subsequent sections.

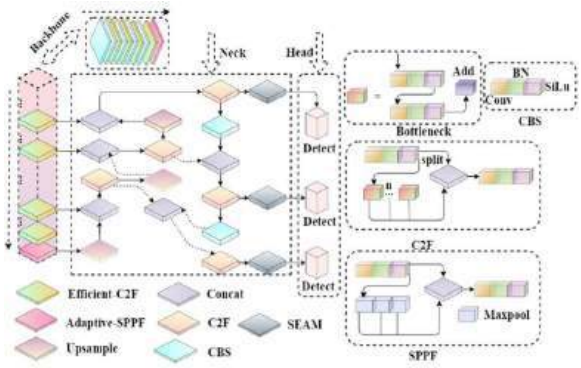


Fig 1. The overall network architecture of EAMS-YOLOv8. '…' in the backbone network means that omitting the CBS block.

A. *Efficient-C2F block*

We propose the Efficient-C2F block, with detailed specifications outlined in Fig .2. The processing sequence within this block is defined as follows: First, a 1×1 convolutional layer adjusts the number of channels in the input feature map. Next, the adjusted feature map is split into three parts along the channel dimension. Each part is independently processed by a C2F sub-block, where the convolution within the Bottleneck structure employs dilated convolution. Subsequently, the three processed sub-feature maps are concatenated into a new feature map which passes 1×1 convolution to adjust the number of channels. Finally, the adjusted feature map is connected to the initial feature map through a residual connection.

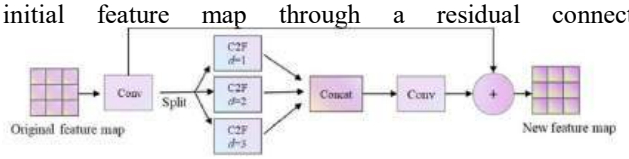


Fig 2. The architecture of Efficient-C2F. 'Split', 'Concat', and '+' stand for the split, channel-wise concatenation and add operations, $d = 2$ and $d = 3$ means that the convolution dilation factor in Bottleneck is set to 2 and 3, respectively, and $d = 1$ means that normal convolution is used.

The Efficient-C2F block enables each C2F sub-block to capture features at different receptive fields. The features extracted by different C2F sub-blocks can complement each other so that the concatenated feature map obtains more comprehensive feature information. Specifically, the Bottleneck structure within the three sub-blocks employs dilated convolutions tailored with distinct different dilation factors (d=1, d=2, and d=3). The dilated convolution with different dilation factors can extract features in different receptive fields to obtain more context information. This method enhances the fusion capability of global and local information. Finally, the original information is preserved and the loss of detailed information is decreased by residual connection with the original feature map. The backbone network can extract information more efficiently through the above operations, thus transferring more abundant information to the neck network.

B. *Adaptive-SPPF netw*

In 2015, He et al.[13] proposed Spatial Pyramid Pooling (SPP), which integrates features across different scales. Spatial Pyramid Pooling–Fast (SPPF) replaces the parallel pooling in SPP with a serial pooling operation. The features of small objects are usually concentrated in a small region. The multi-layer max-pooling operation of SPPF tends to focus on the local maxima in the feature map, potentially losing the subtle features of small objects. The Adaptive-SPPF block combines adaptive pooling into the SPPF block, as shown in Fig .3. The flow of the Adaptive-SPPF block is as follows: Firstly, the feature map, after processing by the CBS block, is divided into two paths. One path passes through adaptive average pooling while the other path passes through adaptive max-pooling. In addition, after a max-pooling operation, the feature map transmits features to the corresponding adaptive max-pooling layer. Finally, the feature maps resulting from different pooling operations are merged with the original feature maps, and the number of channels is adjusted using a CBS block. This approach empowers the block to not only focus on local features but also capture global features. Furthermore, the combination of max-pooling and adaptive max-pooling further strengthens the representation of crucial features at each layer and ensures that crucial information is retained at different max-pooling layers, thus decreasing the loss of feature information.

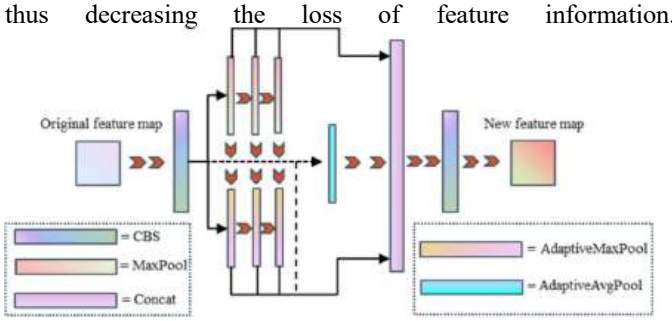


Fig 3. The architecture of the Adaptive-SPPF block.

C. *Modified neck network*

As shown in Fig .4, we improve the neck network in the original structure by adding up-sampling and concatenation operations within the small object layer. This improvement combines shallow features in the backbone network and deep features in the neck network to obtain a richer global context and local information for the small object layer. This boosts the recognition and localization capability of the model. In addition, we precede the detection head with the SEAM attention mechanism, to obtain crucial regions and alleviate the object occlusion. The SEAM attention mechanism first employs depthwise separable convolution, while typically ignoring interactions between channels. Subsequently, pointwise convolution is utilized to fuse the information across individual channels. Finally, two fully connected layers are employed to further refine these channel interactions. The attention mechanism enables the model to compensate for the information loss by leveraging the intricate correlation it has learned between the occluded and non-occluded areas within an image, thus enhancing the detection accuracy.

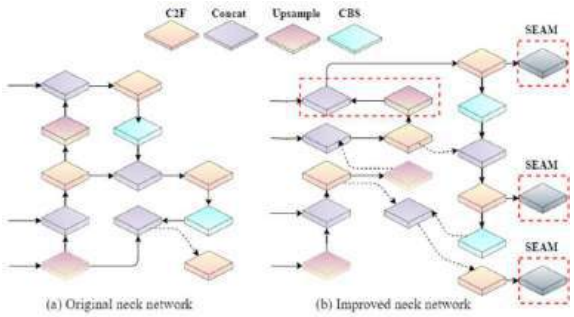


Fig 4 . Comparison of neck network before and after the improvement. The red boxes represent the blocks added to the neck network.

III. EXPERIMENTS AND ANALYSIS

A. Dataset and experimental environment

The VisDrone2019[14] is a large-scale dataset of drone aerial imagery containing 8,629 images. This dataset is split into 6,471 images for training, 548 for validation, and 1,610 for testing. The dataset offers complexities of real-world aerial imagery with densities across 14 cities. It includes annotations for 10 object categories: pedestrians, people, bicycles, cars, vans, buses, trucks, motors, tricycles, and awning tricycles.

The training of the EAMS-YOLOv8 was conducted on an NVIDIA RTX 4060 GPU, using Windows 10 as the operating system and Pytorch (2.3.0) as the deep learning framework. GPU acceleration was achieved using NVIDIA CUDA (12.1). To verify the effectiveness of the EAMS-YOLOv8, we trained and compared YOLOv8n and EAMS-YOLOv8 under the same experimental parameters, including a weight decay factor of 0.0005, and a final learning rate of 0.001. The specific experimental parameters are shown in Table I, and the default parameters of YOLOv8n were used for all the parameters not listed.

TABLE I. EXPERIMENTAL PARAMETERS

Hyperparameter	Value
Epochs	150
Image size	640×640
Batch_size	8
Learning_rate	0.01
Momentum	0.937
Optimizer	SGD

B. Evaluation metrics

In object detection, the main evaluation metrics for model performance include Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP). P measures the ability of the model to correctly predict positive samples, reflecting the accuracy of the model predictions. R indicates the ability of the model to identify all positive samples, representing the proportion of correctly identified positive samples out of the actual positives. AP is an evaluation metric for a single class, integrating precision and recall to evaluate the overall performance of the model for a specific class. mAP represents the average performance across multiple classes, providing a comprehensive evaluation of the model performance in multi-class object detection tasks. The calculation methods for P and R are shown in Equations (1) and (2), respectively.

$$\mathrm{P} = \frac{TP}{(TP + FP)} \tag{1}$$

$$\mathrm{R} = \frac{TP}{(TP + FN)} \tag{2}$$

Where *TP* denotes the number of positive samples that the model correctly predicts as positive classes, *FP* denotes the number of negative samples that the model incorrectly predicts as positive classes, and *FN* denotes the number of positive samples that the model incorrectly predicts as negative classes.

AP is obtained by calculating the area under the Precision-Recall curve as shown in Equation (3).

$$\mathrm{AP} = \int_0^1 \mathrm{P}(\mathrm{R})\mathrm{dR} \tag{3}$$

The mAP is averaged over all categories of AP and is calculated as shown in Equation (4).

$$\mathrm{mAP} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AP}_i \tag{4}$$

where *N* denotes the number of categories.

C. Experiment results and comparison

Table II shows the results of the comparison between the EAMS-YOLOv8 and YOLOv8n models on the VisDrone2019 validation dataset. It can be seen from Table II that EAMS-YOLOv8 shows an improvement in performance in all of the categories. For example, the AP for pedestrian detection increased by 13.2%, from 0.358 to 0.490, indicating a substantial enhancement in its ability to accurately detect pedestrians. Similarly, AP for motorcycle detection increased by 11.6%. Across all categories, the EAMS-YOLOv8 consistently outperforms YOLOv8n, showcasing its superior detection capabilities. Moreover, the comprehensive indicators $mAP_{0.5}$ and $mAP_{0.5:0.95}$ also reflect the overall improvement. $mAP_{0.5}$ increased by 7%, from 0.351 to 0.421; $mAP_{0.5:0.95}$ increased by 4.3%. from 0.201 to 0.244. These results indicate that the EAMS-YOLOv8 has increased accuracy in handling various detection tasks, making it a more suitable choice for multi-class object detection.

To compare the regional focus of EAMS-YOLOv8 versus the YOLOv8n, we utilize the Gradient-weighted Class Activation Mapping (Grad-CAM)[15] visualization technique for an interpretability analysis of two models. Grad-CAM indicates the behavior of the model by generating heatmaps that the model attends to during decision-making, thereby fostering a clear understanding of its operational logic. The visualization results are displayed in Fig .5. The comparison of image groups a, b, and c shows that the color of the heatmap generated by EAMS-YOLOv8 is darker than that of YOLOv8n. Dark areas indicate that the EAMS-YOLOv8 model more confidently identifies the presence of objects within those darker areas, thus enabling more accurate localization and recognition within these areas.

TABLE II. RESULTS ON THE VISDRONE2019 VALIDATION DATASET

Category / Model	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning Tricycle	Bus	Motor	$mAP_{0.5}$	$mAP_{0.5:0.95}$
EAMS-YOLOv8	0.490	0.408	0.150	0.826	0.461	0.366	0.299	0.147	0.554	0.506	0.421	0.244
YOLOv8n	0.358	0.304	0.0896	0.762	0.399	0.326	0.254	0.124	0.501	0.39	0.351	0.201
Improvement	+13.2%	+10.4%	+6.0%	+6.4%	+6.2%	+4.0%	+4.5%	+2.3%	+5.3%	+11.6%	+7.0%	+4.3%

TABLE III. RESULTS OF SEVEN MODELS ON THE VISDRONE2019 VALIDATION DATASET

Category / Model	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning Tricycle	Bus	Motor	$mAP_{0.5}$
Faster R-CNN[16]	0.214	0.156	0.067	0.517	0.295	0.190	0.131	0.077	0.314	0.207	0.217
CenterNet[17]	0.333	0.152	0.121	0.552	0.405	0.341	0.292	0.216	0.422	0.175	0.311
YOLOv4n[18]	0.248	0.126	0.086	0.643	0.224	0.227	0.114	0.076	0.443	0.217	0.307
YOLOv5n[19]	0.444	0.367	0.185	0.742	0.377	0.374	0.253	0.127	0.486	0.433	0.379
Model A[20]	0.438	0.345	0.116	0.809	0.442	0.308	0.252	0.151	0.511	0.454	0.383
Model B[21]	0.390	0.274	0.233	0.696	0.480	0.422	0.325	0.204	0.589	0.365	0.412
EAMS-YOLOv8	0.490	0.408	0.150	0.826	0.461	0.366	0.299	0.147	0.554	0.506	0.421

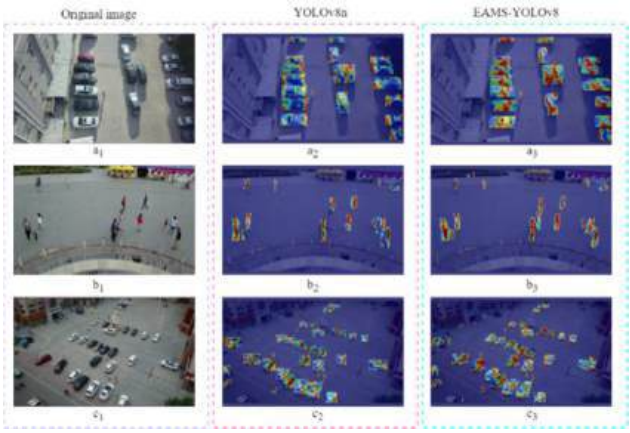


Fig 5. Comparison of heatmaps generated by Grad-CAM method in the YOLOv8n model and EAMS-YOLOv8 model. a_1, b_1, and c_1 represent the original images. a_2, b_2, and c_2 represent the results of the heatmap generated by YOLOv8n. a_3, b_3, and c_3 represent the results of the heatmap generated by EAMS-YOLOv8.

To further validate the effectiveness and superiority of EAMS-YOLOv8, we conducted an experimental comparison involving seven distinct detection models on the validation dataset of the VisDrone2019, and the results are presented in Table III. On the $mAP_{0.5}$ metric, the EAMS-YOLOv8 shows the advantage of detection performance. Specifically, the $mAP_{0.5}$ of EAMS-YOLOv8 is 0.421 surpasses that of Faster R-CNN (0.217) by 20.4%, CenterNet (0.311) by 11%, and YOLOv5n (0.379) by 4.2%. In addition, EAMS-YOLOv8 shows certain competitiveness in terms of detection accuracy when compared to other models. From the experimental results, it can be seen that EAMS-YOLOv8 performs better in the detection of drone aerial images. It can effectively handle situations with widely dispersed objects, small object sizes, and mutual occlusions.

D. Ablation experiments

We refined the YOLOv8n backbone and neck networks to enhance detection performance. To verify the effectiveness of the improved blocks, ablation experiments were performed on each block, and the results are shown in Table IV, where '√' indicates that the block was selected for the experiment and '×' indicates that the block was not selected for the experiment. Firstly, the Adaptive SPPF block led to a modest improvement, increasing $mAP_{0.5}$ from 0.351 to 0.353 (0.2%) and $mAP_{0.5:0.95}$ from 0.201 to 0.203 (0.2%). Secondly, the Efficient-C2F block significantly bolstered performance, elevating $mAP_{0.5}$ up to 0.381 (2.7%) and $mAP_{0.5:0.95}$ to 0.215 (1.2%). Furthermore, the Modified-Neck further enhanced performance, with $mAP_{0.5}$ increasing to 0.408 (2.7%) and $mAP_{0.5:0.95}$ reaching 0.237 (2.2%). Finally, the SEAM attention mechanism contributed an increase in $mAP_{0.5}$ to 0.421 (1.3%) and $mAP_{0.5:0.95}$ to 0.244 (0.7%). These results indicate the richness of feature information in EAMS-YOLOv8, with the optimized neck network efficiently leveraging the features extracted by the backbone network. Notably, the SEAM attention mechanism effectively alleviates the occlusion issues among objects.

TABLE IV. RESULTS OF THE ABLATION EXPERIMENT

Base	Adaptive-SPPF	Efficient-C2F	Modified-Neck	SEAM	$mAP_{0.5}$	$mAP_{0.5:0.95}$
√	×	×	×	×	0.351	0.201
√	√	×	×	×	0.353	0.203
√	√	√	×	×	0.381	0.215
√	√	√	√	×	0.408	0.237
√	√	√	√	√	0.421	0.244

E. Visualization

To visually demonstrate the detection capabilities of the YOLOv8 and EAMS-YOLOv8, we conducted a comparative analysis using three groups of images featuring diverse environments. As shown in Fig .6 group (a) and group (b), the objects are widely distributed and occupy a tiny area within the image, which causes the YOLOv8n to fail in effectively extracting regional features, resulting in false negatives,

especially for distant objects. This result indicates the limitations of YOLOv8n in complex scenes with small objects. In Fig .6 group (c), the objects exhibit occlusion, a scenario that further complicates detection. The YOLOv8n cannot accurately discern occluded objects resulting in a significant rise in false negatives, demonstrating its vulnerability to such challenges. Contrastingly, EAMS-YOLOv8, with its improved blocks, displays a superior detection performance across all three image groups. It effectively handles the challenges posed by small, distant, and occluded objects, leading to fewer false negatives and a more reliable detection output. This comprehensive visual analysis validates the effectiveness of the proposed improvements in YOLOv8n.

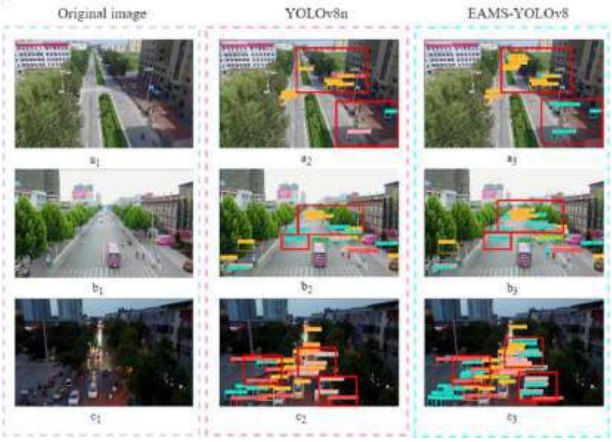


Fig 6. Validation results of the two models on the VisDrone2019 validation dataset. a_1, b_1, and c_1 represent the original images. a_2, b_2, and c_2 represent the results of YOLOv8n validation. a_3, b_3, and c_3 represent the results of the EAMS-YOLOv8 validation. The red boxes mark the results of the visualization of the two models, and it is clear that YOLOv8n produces more false negatives.

IV. Conclusions

In this paper, we propose EAMS-YOLOv8, an enhanced YOLOv8n model to improve the detection performance of drone aerial images. The model introduces four enhancements that will enhance multi-scale feature extraction, information fusion, and background suppression, thereby improving detection in small objects and occluded scenes.

The experiments on the VisDrone2019 dataset validate the effectiveness of the proposed improvement block in the EAMS-YOLOv8 model. Specifically, the EAMS-YOLOv8 model achieves an approximately 7% increase in $mAP_{0.5}$ compared to the YOLOv8n model. However, the EAMS-YOLOv8 model also has some shortcomings, mainly in a large number of parameters and less than ideal real-time inference speed. Future work will focus on reducing the weight of the model by exploring combinations with other lightweight architectures and applying techniques such as pruning and knowledge distillation to optimize the model structure.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 11804399), and the Fund for Academic Innovation Teams of South-Central Minzu University (Grant Number: XTZ24002).

References

[1] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement.," *arXiv: Computer Vision and Pattern Recognition*, Apr. 2018.

[2] N. Agrawal, V. Prabhakaran, T. Wobber, JohnD. Davis, MarkS. Manasse, and R. Panigrahy, "Design tradeoffs for SSD performance," Jun. 2008.

[3] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," *International Journal of Computer Vision*, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1137–1149, Jun. 2017, doi: 10.1109/tpami.2016.2577031.

[5] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018. doi: 10.1109/cvpr.2018.00644.

[6] Y. Liu, D. Shangguan, L. Chen, C. Su, and J. Liu, "Prediction of femtosecond laser etching parameters based on a backpropagation neural network with grey wolf optimization algorithm," *Micromachines*, vol. 15, no. 8, p. 964, 2024, doi: 10.3390/mi1508096.

[7] J. Liu, X. Wu, Y. Feng, M. Zheng, and Z. Li, "Influence of viscous force on the dynamic process of micro-sphere in optical tweezers," *Chinese Physics B*, vol. 32, no. 10, p. 108704, 2023.

[8] J. Liu, L. Long, H. Guo, and Z. Li, "Observation of Moon-like synchronous revolution and rotation of Janus microparticles trapped in an annular optical trap," *ACS Photonics*, vol. 11, no. 10, pp. 4027-4035, 2024. doi: 10.1021/acsphotonics.4c00702.

[9] J. Liu, M. Zheng, Z. Xiong, and Z. Li, "3D dynamic motion of a dielectric micro-sphere within optical tweezers," *Opto-Electron. Adv.*, vol. 4, no. 1, p. 200015, 2021.

[10] D. Shangguan, Y. Liu, L. Chen, C. Su, and J. Liu, "Modeling and experiment of femtosecond laser processing of micro-holes arrays in quartz," *Journal of Applied Physics*, vol. 135, no. 24, p. 243102, Jun. 2024, doi: 10.1063/5.0208329.

[11] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X. Wang, "YOLO-FaceV2: A Scale and Occlusion Aware Face Detector".

[12] F. Yu, V. Koltun, and T. Funkhouser, "Dilated Residual Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. doi: 10.1109/cvpr.2017.75.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1904–1916, Sep. 2015, doi: 10.1109/tpami.2015.2389824.

[14] D. Du *et al.*, "VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct. 2019. doi: 10.1109/iccvw.2019.00030.

[15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017. doi: 10.1109/iccv.2017.74.

[16] W. Yu, T. Yang, and C. Chen, "Towards Resolving the Challenge of Long-tail Distribution in UAV Images for Object Detection," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2021. doi: 10.1109/wacv48630.2021.00330.

[17] D. Du *et al.*, "VisDrone-DET2020: The Vision Meets Drone Object Detection in Image Challenge Results," in *Computer Vision – ECCV 2020 Workshops,Lecture Notes in Computer Science*, 2020, pp. 692–712. doi: 10.1007/978-3-030-66823-5_42.

[18] S. Ali, A. Siddique, H. F. Ates, and B. K. Gunturk, "Improved YOLOv4 for Aerial Object Detection," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, Jun. 2021. doi: 10.1109/siu53274.2021.9478027.

[19] G. Villalba, H. Wu, Y. Zhu, and L. Wang, "A Dense Small Object Detection Algorithm Based on a Global Normalization Attention Mechanism".

[20] H. Nie, H. Pang, M. Ma, and R. Zheng, "A Lightweight Remote Sensing Small Target Image Detection Algorithm Based on Improved YOLOv8".

[21] Z. Li, B. Fan, Y. Xu, and R. Sun, "Improved YOLOv5 for aerial images based on attention mechanism," *IEEE Access*, pp. 96235–96241, Jan. 2023, doi: 10.1109/access.2023.3277931.

Self-consistent Semantic Feature Extraction of Image Object Based on Contrastive Learning

Shanyuan Liu
School of Information Science and Engineering, Lanzhou University
Lanzhou, China
lshanyuan2024@lzu.edu.cn

Jinhui Lv
Gansu Highway and Bridge Construction Group Co., Ltd. & Gansu ZhiTong Technology Engineering Detection Consulting Co., Ltd.
Lanzhou, China
lujinhui2024@outlook.com

Ligang Zhao
Gansu Highway and Bridge Construction Group Co., Ltd. & Gansu ZhiTong Technology Engineering Detection Consulting Co., Ltd.
Lanzhou, China
zhaoligang2021@outlook.com

Zhaobin Chang
School of Information Science and Engineering, Lanzhou University
Lanzhou, China
changzhb21@lzu.edu.cn

Yonggang Lu*
School of Information Science and Engineering, Lanzhou University
Lanzhou, China
ylu@lzu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899730

***Abstract*—Contrastive learning has become an influential paradigm in the field of Computer Vision, and great progress has been made in feature extraction by exploiting the comparison of positive and negative samples. However, how to make the most of the features of different parts of the same object to infer more accurate similarity scores is still underexplored. One way to alleviate the problem is to use the framework of contrastive learning to study how to improve the feature similarity of different parts of the same object in the latent space and find out more features with high similarity called "self-consistent semantic features". In this paper, we propose a neural network model SSFE for self-consistent semantic feature extraction, and a training method by retraining the hard dataset is also included. Specifically, the representation vectors are learned to describe local features better, and then a neural network module is introduced to infer similarity by mapping the stacked feature vectors to the sample space. Furthermore, another module is developed to help learn the representation vectors by classifying them into different classes. Our experiments show that our model with the training method can extract better self-consistent semantic features in PASCAL VOC 2007 and PASCAL VOC 2012.**

***Index Terms*—Self-consistent semantic feature, Contrastive learning, Siamese Networks, Multi-task learning**

I. Introduction

Learning special representations is a hard challenge in computer vision [1], [2] as it allows for comparing and evaluating similarity such as pattern recognition,image classification and image retrieval, between different objects. The primary goal of image similarity calculation is to assign a numerical value which reflects how similar two images are. These values typically range from 0 to 1, though the range can vary depending on the specific method used [3]. Although great progress has been made in recent years, image similarity calculation remains a challenging problem due to high spatial dimension, low computational efficiency and subjective human perception.

Researchers have proposed many similarity calculation methods, which mainly focus on pixel analysis and neural network [4], [5]. On the one hand, traditional algorithms such as histogram comparison algorithm whose principle is to count the number of pixels in each image, depict the number of pixels on a histogram, and obtain the similarity of two images through human observation or specific algorithms. In addition, perceptual hash algorithm compresses the size of the image to ignore the details of the image to varying degrees and represents the image content initially by comparing the mean value of the pixel size with the size of each pixel. On the other hand, neural networks generally emerge in the field of computer vision by extracting the feature vectors of image objects and comparing them based on certain criteria. For example, cosine similarity measures the cosine of the angle between two vectors [6]. However, it is still hard for these methods to perform similarity calculation optimally and designing similar methods needs careful mathematical thinking. At the same time, similarity calculation on patches needs to be well studied because the current popular Sora is to learn small visual patches [7].

Recently, contrastive learning has been popularized because it can lead un-/self-supervised representation learning. By focusing on the underlying similarities and differences in data,this enhances the model's ability to generalize across different tasks but also causes collapsing. Many excellent algorithms are proposed to avoid collapsing solutions such as SimCLR [8],SwAV [9], BYOL [10] and SimSiam [11].

In this paper, we propose Self-consistent Semantic Feature Extraction Model(SSFE), a new algorithm for supervised learning of similarity calculation. Instead of characterizing

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

these representation vectors with scalar-based cosine similarity, we propose to learn the similarity representations based on neural networks. SSFE is based on Siamese Network [12] to extract feature, refers that both of a pair of images are through the same network. Then a neural network module named *Calculator* is introduced to infer similarity by mapping the stacked feature vectors to the sample space. Furthermore, the module named *Classifer* is developed to help learn the representation vectors by classifying them into different classes. We evaluate the representation by performing the similarity calculation task between patches in different positions in the same picture on PASCAL VOC 2007 and PASCAL VOC 2012 [13]. Our contributions are: (i) We introduce SSFE, a supervised representation learning method (Section 3) which achieves better results under the linear evaluation protocol on PASCAL VOC 2007 and PASCAL VOC 2012. (ii) We show the new training method of focusing on hard data help improve the training(Section 4).

The remaining sections of this paper follow a structured outline. Section 2 provides a concise introduction to the methods involved in our experiments. Subsequently, Section 3 presents the concept of self-consistent semantic features, model architecture, and hyperparameters, serving as the foundation for our experiments and discussions in Section 4, culminating in our conclusions in Section 5.

II. Related Work

In this section, the most recent work related to self-consistent semantic feature extraction is presented.

A. Image similarity measurement

Image similarity measurement refers to the process of quantifying the degree to which two images resemble each other. The methods for measuring similarity can be broadly categorized into two main types: pixel-based and feature-based.

1) Pixel-Based Methods: Pixel-based methods for image similarity measurement involve directly comparing the pixel values of two images. These methods are straightforward but can be sensitive. Zhao et al. have developed HSV(Hue, Saturation, Value) with emphasis on the visual perception of the variation in hue, saturation and intensity values of an image pixel [14]. Moreover, an image perceptual hash algorithm has been proposed by Wen et al. to generate a compact representation of an image. The main idea of the algorithm is to integrate color histogram and Discrete Cosine Transform coefficients of image patches as perceptual feature, then to compress perceptual features as inter-feature with principal component analysis, and to threshold to create a robust hash [15]. In order to consider more content, Wang et al. have proposed SSIM that designed to model the human visual system's ability to perceive changes about structural information [16].

2) Feature-Based Methods: These methods compare images based on extracted features rather than raw pixel values. Features can include edges, textures, colors, or more complex representations learned by deep neural networks. On the one hand, some methods calculate the similarity between images based on the extraction of feature points. Lowe has proposed Scale-Invariant Feature Transform(SIFT), which is robust to rotation, scaling and translation [17]. In order to solve the disadvantages of high computational complexity and long time consuming of SIFT, Bay et al. have developed SURF which simplifies the process of finding and describing extreme points [18]. On the other hand, more methods utilize pre-trained convolutional neural networks (CNNs) to extract high-level features and measure similarity in the feature space. Kuanr et al. have use a CNN model for the feature extraction of the images and discovers the most similar patients with Maxwell–Boltzmann similarity, threshold similarity and cosine similarity [19].

B. Contrastive learning

The purpose of contrastive learning is to learn high-level features that are sufficient to distinguish objects [20]. And the key idea of contrastive learning is to attract the positive sample pairs and repulse the negative sample pairs [21]. This methodology has been recently popularized for un-/self-supervised representation learning. Siamese Networks are general models for comparing entities in contrastive learning. Because of their excellent comparative capabilities, Siamese Networks can be used for signature authentication and face recognition [22].

C. Multi-task learning and Cross-attention mechanism

Multi-task learning is defined as a machine learning method that takes m($m \geq 1$) tasks that are related but not identical and learn them together. It improves the learning effect by jointly updating the underlying shared representation and makes related tasks enhance each other, preventing model overfitting and improving inference effect. Chen et al. have applied multi-task learning to train the multimodal model to perform well on visual language tasks including image captioning, visual question answering and visual grounding [23].

Different from the self-attention mechanism, the inputs of cross-attention mechanism are different sequences which are entered as the query vector or the key vector and the value vector [24], [25]. In this way, sequence information from different sources can fully interact.

III. Proposed Method

In this section, we will introduce the concept of self-consistent semantic features, the model architecture of SSFE (Figure 2), and hyperparameters.

A. The Concept of Self-consistent Semantic Features

In order to describe the research content of this paper, we put forward the concept of self-consistent semantic features. Different from other concepts, self-consistent semantic features focus on one object and refer to the features of different parts of the same object with high similarity in hidden space. As shown in Figure 1, the head, hands and feet in the brown box are self-consistent semantic features that belong to the rider.

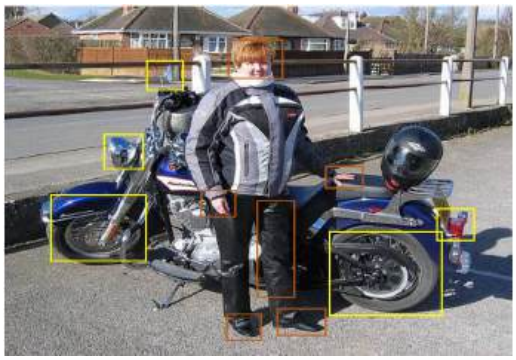


Fig. 1. The concept of self-consistent semantic features.

On the one hand, the self-consistent semantic features enrich the visual theory content, and expect each part of the object to have a high similarity so that the whole can be seen from the part. On the other hand, they help to solve visual tasks, such as recognizing the whole object by only using part of the object and segmenting the object with occlusion more effectively.

B. Framework

Inspired by recent constrastive learning frameworks, this network learns representations by distinguishing between similar and dissimilar features. As illustrated in Figure 2, this networks comprises the following six major components.

- A convolutional neural network *Encoder* extracts the represent vector of the input data. We opt for performance and decide to adopt the Resnet-50 whose output dimension is 2048-d.
- A small neural network *Projector* filters some low-level features and transforms the output from one form to another. The Projector has batch normalization(BN) [26] and ReLU [27] applied to all full-connected(fc) layers but except the output fc. Input dimension and output dimension are both 2048-d while hidden layers' dimension is 2048-d.
- Another small neural network *Predictor* further maps the representation vector. This Prediction has two fc layers. The dimension of input and output is 2048, and hidden layer is set to 512-d which can reduce parameters. It is worth mentioning that the input needs to be batch normalized first.
- A cross-attention module *Attender* help represent vectors interact with each other. The Attender has three linear layers called k, q and v.Different from self-attention, one represent vector dots with k and v while the other dots with q.
- *Calculator* has five fc layers, which would calculate the similarity of two represent vectors. Input is a stack of two represent vectors, while output is the level of similarity of these two vectors.
- *Classifer* helps improve the ability of this model to calculte similarity by classifering patches. The input is the 2048-d represent vector of Encoder, and the output is 7-d vector which represent seven classes.Classifer has four fc layers. The dimensions of hidden layers are 512 and 128.

Assume that $z1$, $z2$ are the output from the *Projector*, and $p1$, $p2$ are the output from the *Predictor*. Before the $stop-grad$ method is added to our framework, the output M of *Calculator* is displayed as

$$M = \frac{1}{2}Calculator(p1, z2) + \frac{1}{2}Calculator(p2, z1) \quad (1)$$

We apply it by modifying as

$$n1 = stop - grad(z1) \quad (2)$$

$$n2 = stop - grad(z2) \quad (3)$$

$$M = \frac{1}{2}Calculator(p1, n2) + \frac{1}{2}Calculator(p2, n1) \quad (4)$$

C. Setting

- BatchSize is 256 by default, which is friendly to typical GPU implementations. Larger or smaller is fine but our method does not require a large-batch optimizer such as SimCLR [8].
- Stochastic gradient descent(SGD) optimizer is used to update gradient during training. As increase in BatchSize, learning rate(lr) would increase linearly. The weight decay is 0.002 and the SGD momentum is 0.9.
- To speed this model to converge better, the learning rate has a cosine decay schedule. The initial lr is 0.05, and decays to 0.025 at the middle of epoches.

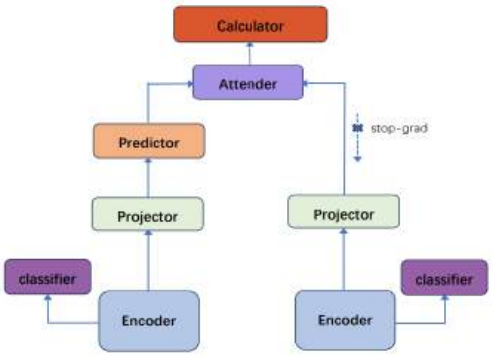


Fig. 2. Self-consistent semantic feature extraction model network architecture.

IV. Experimentation and Analysis

In this section, we will introduce the datasets, the experimental procedures, the involved methods and the proposed model. The experimental results show that the model can achieve better results.

A. Datasets

1) PASCAL VOC 2007[6]: PASCAL VOC 2007 has 20 classes including person, animal, vehicle and indoor object. For the segmentation task, a total of 311, 321 and 253 images are used for training, verification and testing. PASCAL VOC 2007[6] is built based on animal(sheep,cow,horse,dog,cat,bird) images of PASCAL VOC 2007.

2) PASCAL VOC 2012[6]*:* PASCAL VOC 2012 has the same classes with PASCAL VOC 2007. For the segmentation task, a total of 1464, 1449 and 1456 images are used for training, verification and testing. PASCAL VOC 2012[6] is built based on animal(sheep,cow,horse,dog,cat,bird) images of PASCAL VOC 2012.

B. Setup

First, we randomly crop the image into multiple 32*32 patches where 80% or more of the content must belong to the same object in order to ensure a large amount and high quality of training data.

Second, for a pair of patches belong to or do not belong to the same object, the corresponding label is 1 or 0. In addition, we use the one-hot encoding to encode the corresponding label for each class for classification tasks.

Third, Each pair of small patches with positional information is input into the model, and the calculated similarity value is used together with the label to calculate the loss function. And the model with the best generalization ability is selected on the validation dataset.

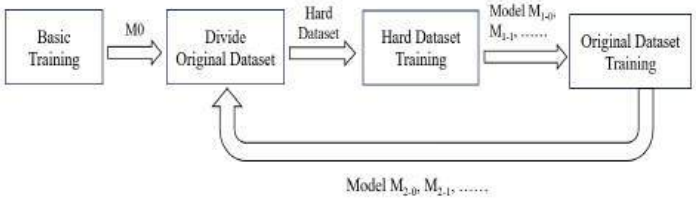


Fig. 3. Running process of self-consistent semantic feature extraction model.

Finally, according to the absolute difference of the label and the similarity value calculated by the model with the best performance in the verification dataset, the hardest 10% of original training dataset is classified as hard training dataset. We obtain hard validation dataset in the same way. Then perform the following steps repeatedly(Figure 3):

a. The performance indicators hard_metric of the hard validation set and all_metric of the whole validation set were initialized to verify the learning effect of the model.

b. On the hard training set, the model is trained for 5 epoches to make it more exposed to difficult data. If there is a better model than hard_metric on the hard validation set, then hard_metric is updated and the best model is saved.

c. On the whole training set, the model is trained for 2 epoches so that it does not forget the original knowledge. If the best model over all_metric appears on the whole validation set, all_metric is updated and the best model is saved.

C. Evaluation Criteria

Since the capability of consistent semantic feature extraction is not easily comparable, accuracy is used as the evaluation metric in this experiment. The accuracy is calculated based on the similarity values and their corresponding labels. If the similarity value is greater than or equal to 0.5 and the label is 1, or if the similarity value is less than 0.5 and the label is 0, it is considered correct; otherwise, it is considered incorrect. As shown in Figure 4, the self-consistent semantic feature extraction model calculated 0.76 for the image patches of the head and body parts of the object "sheep" while the value of the self-consistent semantic feature extraction model for the head of the "sheep" target and the body part of the "green grass" target is 0.35.

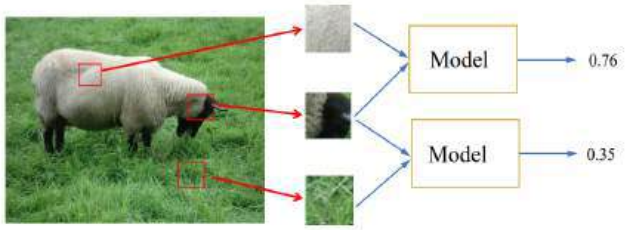


Fig. 4. Running process of self-consistent semantic feature extraction model.

D. Methodology Comparisons

In this section, by comparing the results from some current methods(we discuss them next), we hope to determine that exploring a new structure for self-consistent semantic feature extraction is necessary. The experimental results on PASCAL VOC 2007[6] and PASCAL VOC 2012[6] are shown in the following.

1) Methodology Introduction:

a) Cosine Similarity: The Cosine Similarity is a common way to calculate the similarity of two vectors,whose prior knowledge is that the cosine of the angle between two vectors can measure the difference between two individuals.

$$cosineSimilarity(u,v) = \frac{\sum_{i=1}^{n}(u_i \times v_i)}{\sqrt{\sum_{i=1}^{n} u_i^2} \times \sqrt{\sum_{i=1}^{n} v_i^2}} \quad (5)$$

b) Euclidean Distance: The Euclidean Distance is the most common distance metric, which measures the absolute distance between two vectors in a multidimensional space.

$$Euclidean\ Distance(u,v) = \frac{1}{1+\sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}} \quad (6)$$

c) Manhattan Distance: The Manhattan Distance was proposed by the famous German Jewish mathematician Hermann Minkowski in the 19th century to indicate the absolute wheelbase sum of two vectors in the standard coordinate system.

$$Manhattan\ Distance(u,v) = \frac{1}{1+\sum_{i=1}^{n}(u_i - v_i)} \quad (7)$$

d) Pearson Correlation Coefficient: The Pearson Correlation Coefficient was developed by Carl Pearson from a similar but slightly different idea proposed by Francis Galton in the 1880s. It is widely used to measure the degree of correlation between two vectors.

$$Correlation\ Coefficient(u,v) = \frac{\mathrm{Cov}(u,v)}{\sqrt{D(u)}\sqrt{D(v)}} \quad (8)$$

2) Result Comparisons: For each individual method, we follow the principle that hyper-parameter and augmentation are the same. The module *Calculator* is replaced with a module composed of other methods and the module *Classifer* isn't invoked by default. Table 1 shows the accuracy of these methods on the PASCAL VOC 2007[6] and PASCAL VOC 2012[6] test set.

It can be found from Table 1 that the highest accuracy of 75.3% on PASCAL VOC 2007[6] and 81.4% on PASCAL VOC 2012[6] are produced by Cosine Similarity and Pearson Correlation Coefficient respectively. However, our results demonstrated that these methods can't help the model learn special image representations well.

E. The Design and Refinement of SSFE

In this section, we will introduce the main model architecture and working principle of SSFE in detail. And for the task of self-consistent semantic feature extraction, we use a multi-task learning mechanism and a better training procedure. The effects of all methods or models will be listed to provide data support.

1) The Design of SSFE: The main body of the self-consistent semantic feature extraction model is composed of Encoder, Projector, Predictor and Attender. A pair of patches is input into Encoder to encode feature vectors, filter low-level features through the Projector to transform the feature vectors, and use the stop-grad strategy to strengthen the model learning process, and then further transform through the Predictor to reduce the randomness brought by data enhancement. Finally, cross-attention is used to deal with the relationship between the output vectors of the Projector and the Predictor.

An important innovation of the SSFE model is *Calculator*. It is trained to calculate the similarity value of a pair of representation vectors extracted by the Siamese Network. First,a pair of representation vectors is stacked into a higher dimensional vector. Then *Calculator* applies a nonlinear transformation to this vector.Finally, we would get the result which passes through the Sigmoid transformation.

TABLE I
ACCURACY COMPARISON OF DIFFERENT METHODS OR MODELS.

Method	PASCAL VOC 2007[6]	PASCAL VOC 2012[6]
Cosine Similarity	75.3%	80.5%
Euclidean Distance	73.9%	79.8%
Manhattan Distance	70.9%	77.6%
Correlation Coefficient	71.7%	81.4%
SSFE	79.6%	85.3%

We find that SSFE can achieve 79.6% on PASCAL VOC 2007[6] and 85.3% on PASCAL VOC 2012[6](see Table 1). From the results, it is clear that SSFE is much better than these methods shown in Table 1, which is at least 4.3% higher on PASCAL VOC 2007[6] and 3.9% higher on PASCAL VOC 2012[6].

2) Introduction to Multi-task Learning: In this section, we use Multi-task learning to refine SSFE. *Classifer* is a secondary task module and only shares the parameters of the *Encoder* with the main module. *Classifer* can classify patches as bird, cat, cow, dog, horse, sheep and background to facilitate the main task. During training, Either of a pair of patches is input to *Encoder* and then to *Classifer*.

The reason why the multi-task learning mechanism works is that, due to the addition of a pair of patch classification losses to the loss function, this loss updates the parameters of *Encoder* in the process of backpropagation, thus facilitating the *Encoder* to extract more representative feature vectors.

TABLE II
ACCURACY COMPARISON OF DIFFERENT METHODS OR MODELS WITH CLASSIFER.

Method	PASCAL VOC 2007[6]	PASCAL VOC 2012[6]
Cosine Similarity	76.5%	81.8%
Euclidean Distance	74.7%	80.3%
Manhattan Distance	71.3%	77.9%
Correlation Coefficient	75.2%	82.5%
SSFE	80.4%	86.6%

Table 2 recapitulates that *Classifer* is able to improve the accuracy of the main task.SSFE with *Classifer* obtains the accuracy of 80.4% on the PASCAL VOC 2007[6] and 86.6% on the PASCAL VOC 2012[6] which is comparable to the state-of-the-art methods. Compared with SSFE without multi-task learning in the previous section, it is 0.8% and 1.3% higher respectively.

3) The Impact of Different Training Processes on Results: In this section, we will introduce the innovative work on the training process in this study. Through the observation of the previous training process, it is found that the model doesn't learn enough from the training dataset, even after 200 epoches of training, it can only correctly identify 93% of the training data. Therefore, innovation must be in the training process to make the model fully learn the training dataset, so as to enhance its ability to extract feature vectors and calculate similarity. Therefore, in the training process, we retrain the best model of each method obtained in the above steps and make it learn for five epoches of difficult data; In order to avoid the overfitting of the model to the difficult data, the model is trained with full data for two epoches(Figure 4).

Table 3 shows that innovation in the training process can help the model improve its generalization ability, and enhance its ability to extract feature vectors and calculate similarity. Our method achieved 80.7% accuracy on PASCAL VOC 2007[6] and 87.1% accuracy on PASCAL VOC 2012[6], outperforming existing methods, which is 0.3% and 0.5% higher than before the training process innovation.

TABLE III
ACCURACY COMPARISON OF DIFFERENT METHODS WITH CLASSIFER AND RETRAINING ON PASCAL VOC 2007[6] AND PASCAL VOC 2012[6].

Method	PASCAL VOC 2007[6]	PASCAL VOC 2012[6]
Cosine Similarity	76.8%	82.0%
Euclidean Distance	74.7%	80.4%
Manhattan Distance	71.9%	77.9%
Correlation Coefficient	75.5%	82.9%
SSFE	80.7%	87.1%

4) Experimental Summary of SSFE: Above all, it is proved that the SSFE can better improve the self-consistent semantic feature extraction ability, the multi-task learning method can effectively help train the model, the training innovation method can help the model learn difficult data to improve the model effect, and the two-dimensional position encoding can better assist the task.

V. Conclusions and future scope

To review this paper, we propose a new model named as SSFE to improve the similarity calculation using Siamese networks and a new training method to train more carefully. This model is different from calculating by measuring the distance or angle in high dimensional space, but based on the trainable neural network. And this method promote the model to learn more samples. As the experimental results has shown that the proposed SSFE is superior to other methods in performance and the training method is useful. In the future, we will test whether the proposed model and training method is practical on more sophisticated networks and larger datasets.

VI. Acknowledgments

This work is supported by Gansu Haizhi Characteristic Demonstration Project (No. GSHZTS2022-2).

References

[1] Wiskott, L., & Sejnowski, T.J. (2002). Slow Feature Analysis: Unsupervised Learning of Invariances. Neural Computation, 14, 715-770. https://doi.org/10.1162/089976602317318938

[2] Hinton, G.E., Osindero, S., & Teh, Y.W. (2006). A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, 18, 1527-1554. https://doi.org/10.1162/neco.2006.18.7.1527

[3] Bohush, R.P., Ablameyko, S., Ablameyko, S., Adamovskiy, E.R., & Savca, D. (2020). Image Similarity Estimation Based on Ratio and Distance Calculation between Features. Pattern Recognition and Image Analysis, 30, 147 - 159. https://doi.org/10.1134/S1054661820020030

[4] Hu, M., Yang, Y., Shen, F., Xie, N., Hong, R., & Shen, H.T. (2019). Collective Reconstructive Embeddings for Cross-Modal Hashing. IEEE Transactions on Image Processing, 28, 2770-2784. https://doi.org/10.1109/TIP.2018.2890144

[5] Peng, J., Wang, Z., & Wang, S. (2023). Similarity calculation method for images based on the scene graph. Signal, Image and Video Processing, 17, 2395-2403. https://doi.org/10.1007/s11760-022-02456-0

[6] Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., & Liu, W. (2018). CosFace: Large Margin Cosine Loss for Deep Face Recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5265-5274. https://doi.org/10.48550/arXiv.1801.09414

[7] Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., He, L., & Sun, L. (2024). Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models. ArXiv, abs/2402.17177. https://doi.org/10.48550/arXiv.2402.17177

[8] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G.E. (2020). A Simple Framework for Contrastive Learning of Visual Representations. ArXiv, abs/2002.05709.

[9] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. ArXiv, abs/2006.09882.

[10] Grill, J., Strub, F., Altch'e, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.Á., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., & Valko, M. (2020). Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. ArXiv, abs/2006.07733.

[11] Chen, X., & He, K. (2020). Exploring Simple Siamese Representation Learning. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 15745-15753. https://doi.org/10.1109/CVPR46437.2021.01549

[12] Bromley, J., Bentz, J.W., Bottou, L., Guyon, I.M., LeCun, Y., Moore, C., Säckinger, E., & Shah, R. (1993). Signature Verification Using A "Siamese" Time Delay Neural Network. Int. J. Pattern Recognit. Artif. Intell., 7, 669-688. https://doi.org/10.1142/S0218001493000339

[13] Everingham, M., Eslami, S.M., Gool, L.V., Williams, C.K., Winn, J.M., & Zisserman, A. (2014). The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision, 111, 98 - 136. https://doi.org/10.1007/s11263-014-0733-5

[14] Zhao, N., Zhou, Z., & Liao, L. (2019). Partial-duplicate image retrieval based on HSV colour space for coverless information hiding. Int. J. Comput. Sci. Eng., 19, 15-24. https://doi.org/10.1504/IJCSE.2019.10018359

[15] Wen, Z., Zhu, W., Jie, O., Liu, P., Du, Y., Meng, Z., & Gao, J. (2010). A Robust and Discriminative Image Perceptual Hash Algorithm. 2010 Fourth International Conference on Genetic and Evolutionary Computing, 709-712. https://doi.org/10.1109/ICGEC.2010.180

[16] Wang, S., Rehman, A., Wang, Z., Ma, S., & Gao, W. (2012). SSIM-Motivated Rate-Distortion Optimization for Video Coding. IEEE Transactions on Circuits and Systems for Video Technology, 22, 516-529. https://doi.org/10.1109/TCSVT.2011.2168269

[17] Lowe, D.G. (1999). Object recognition from local scale-invariant features. Proceedings of the Seventh IEEE International Conference on Computer Vision, 2, 1150-1157 vol.2. https://doi.org/10.1109/ICCV.1999.790410

[18] Bay, H., Tuytelaars, T., & Gool, L.V. (2006). SURF: Speeded Up Robust Features. European Conference on Computer Vision. https://doi.org/10.1007/11744023_32

[19] Kuanr, M., Mohapatra, P., Mittal, S., Maindarkar, M.A., Fauda, M.M., Saba, L., Saxena, S., & Suri, J.S. (2022). Recommender System for the Efficient Treatment of COVID-19 Using a Convolutional Neural Network Model and Image Similarity. Diagnostics, 12. https://doi.org/10.3390/diagnostics12112700

[20] Wu, Z., Xiong, Y., Yu, S.X., & Lin, D. (2018). Unsupervised Feature Learning via Non-parametric Instance Discrimination. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3733-3742. https://doi.org/10.1109/CVPR.2018.00393

[21] Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality Reduction by Learning an Invariant Mapping. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2, 1735-1742. https://doi.org/10.1109/CVPR.2006.100

[22] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 1701-1708. https://doi.org/10.1109/CVPR.2014.220

[23] Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., & Elhoseiny, M. (2023). MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning. ArXiv, abs/2310.09478. https://doi.org/10.48550/arXiv.2310.09478

[24] Chen, C., Fan, Q., & Panda, R. (2021). CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 347-356. https://doi.org/10.1109/ICCV48922.2021.00041

[25] Lee, K., Chen, X., Hua, G., Hu, H., & He, X. (2018). Stacked Cross Attention for Image-Text Matching. ArXiv, abs/1803.08024. https://doi.org/10.1007/978-3-030-01225-0_13

[26] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv, abs/1502.03167.

[27] Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. International Conference on Artificial Intelligence and Statistics.

[28] Kunc, V., & Kl'ema, J. (2024). Three Decades of Activations: A Comprehensive Survey of 400 Activation Functions for Neural Networks. ArXiv, abs/2402.09092. https://doi.org/10.48550/arXiv.2402.09092

Python source code vulnerability detection based on CodeBERT language model

Kunpeng Zhao*
School of Cyberspace Security
Zhengzhou University
Zhengzhou, China
Email: zkunp@gs.zzu.edu.cn

Shuya Duan†
School of Cyberspace Security
Zhengzhou University
Zhengzhou, China
Email: duanshuyaa@gs.zzu.edu.cn

Ge Qiu†
School of Cyberspace Security
University of Information Engineering
Zhengzhou, China
Email: 3111896391@qq.com

Jinyuan Zhai†
School of Cyberspace Security
University of Information Engineering
Zhengzhou, China
Email: 1243251137@qq.com

Mingze Li†
School of Cyberspace Security
University of Information Engineering
Zhengzhou, China
Email: 311897521@qq.com

Long Liu†*
School of Cyberspace Security
University of Information Engineering
Zhengzhou, China
Email: d12_liu@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899694

***Abstract*—Programming language source code vulnerability mining is crucial to improving the security of software systems, but current research is mostly focused on the C language field, with little attention paid to scripting languages. Python code is currently widely used in software systems, and vulnerability detection in Python code is also very important. As an interpreted language, Python has a concise and clear grammatical structure and is closer to natural language, so Python source code vulnerability mining has greater advantages. Previous detection systems based on LSTM models have achieved certain results in detecting Python source code vulnerabilities, but due to the limitations of the model, the vulnerability dataset has low accuracy, lack of full consideration of the deep semantic features of the source code, and the vulnerability multi-classification process is relatively cumbersome and inefficient, and there is room for improvement in accuracy and recall. Therefore, this paper proposes a source code snippet multi-classification vulnerability detection system based on CodeBERT, which aims to automatically detect the security of software code snippets. First, by analyzing the differences between the new and old versions, a multi-label vulnerability dataset is constructed using diff files, and then the RoBERTa is used to obtain high-quality expressions of code keywords. Finally, the constructed multi-classification CodeBERT model is introduced for fine-tuning detection. By improving the accuracy of vulnerability datasets and introducing a high-performance model, this method has achieved significant improvements in source code vulnerability mining and vulnerability classification compared to traditional LSTM model-based detection methods. The average accuracy of the proposed model reaches 98%.**

***Keywords*-Diff; CodeBERT; Python source code; vulnerability mining**

I. INTRODUCTION

The rapid development of the Internet has profoundly changed people's production and lifestyle, bringing unprecedented convenience to individuals and society. However, with the significant increase in the number of software vulnerabilities and potential risks, the inefficiency of manual vulnerability detection methods can no longer meet the current urgent needs for software security. As an effective strategy to deal with network attacks, source code vulnerability detection has gradually occupied an important position in the field of network security research and has become the focus of academic circles.

There are many tools in the field of vulnerability mining, such as Scan Dal[1], HybriDroid[2] and PREfix[3]. These tools are mainly used to detect vulnerabilities in C language or Java. There is relatively little research on vulnerability mining in Python. There are mainly the following methods. Michelsen proposed a Python vulnerability detection tool based on static taint analysis - PyT[4]; Peng et al[5]. proposed a Python source code vulnerability detection method based on code similarity; Wartschinski et al[6]. proposed a deep learning based on LSTM Model method for detecting vulnerabilities in Python source code; Guo et al[7]. innovatively proposed a method for detecting and classifying vulnerabilities in Python source code based on BiLSTM and OVO SVMs models.

They do not fully consider the contextual semantic information of vulnerability datasets, and the models do not extract deep semantic features for the source code. In addition, in terms of vulnerability classification, a vulnerability type classifier is constructed by extracting relevant text features and combining it with the OVO SVMs algorithm, and finally gives code snippets of vulnerability types, which is cumbersome and inefficient. Although the above results have achieved certain results, there is still much room for improvement.

Based on the above situation, this paper proposes a new Python source code vulnerability mining and classification method based on the CodeBERT model, aiming to achieve highly accurate identification and precise classification of Python vulnerabilities.

The specific contributions of this study are outlined as

follows:

1. Constructed a Python source code vulnerability data set from the open source community by comparing diff file differences and the keyword ”Security”.

2. A multi-label and multi-classification CodeBERT[8] model was constructed to mine and classify Python source code vulnerabilities, with an accuracy rate of 97%-99% and a recall rate of 94%-98%.

II. METHODOLOGY

This section presents a Python source code vulnerability detection method designed for the GitHub open-source community, comprising three main modules. The comprehensive framework of the system is depicted in Fig 1.

Initially, the Transformer model undergoes pre-training to develop a comprehensive understanding of the syntax and semantics inherent in Python source code. Following this, the multi-label source code vulnerability data, segmented into blocks, is inputted into the pre-trained Transformer model for fine-tuning post data preprocessing[9].

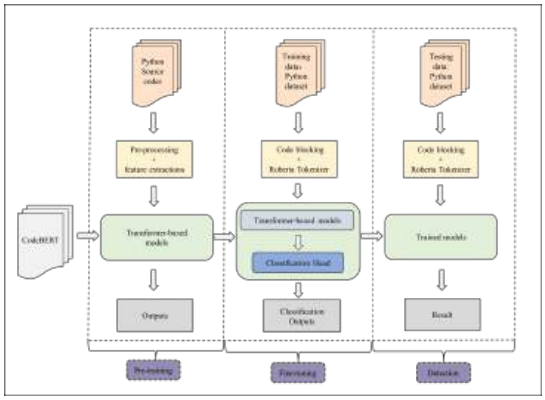


Figure 1: Frame structure.

A. Problem definition

In the context of the current version, a_i represents the code line of fragment ”i,” which is deemed non-vulnerable to attack. Conversely, p_i denotes the code line of fragment ”i” from the previous version, identified as vulnerable to attack. The equation $diff(a_i) = p_i$ signifies the process of comparing the code line from the diff file with that of the current version, thereby identifying the vulnerable lines of code from the previous version. This method is instrumental in compiling a vulnerability dataset.The vulnerability data set marked through diff files is shown in Fig 1.

Our objective is to identify the presence and types of vulnerabilities within the source code. In the initial step of the process, known as vulnerability dataset labeling, we approach the problem as a multi-classification task. Here, ”c” denotes the label category, where ”c” takes on values from 1 to 10, representing different vulnerability types. A label of ”c=0” signifies that the code fragment is deemed safe. Moving to the second step, fine-tuning of the vulnerability model takes place. This entails refining the pre-trained model using the labeled vulnerability dataset to develop the multi-classification detection model, denoted as $f()$. In the final step of vulnerability detection, the label of a given code fragment s_i is determined through the fine-tuned model $f()$. If $f(s_i)$ returns a value from 1 to 10, it indicates that s_i is vulnerable code, with the label number identifying the specific vulnerability type. Conversely, if $f(s_i) = 0$, it signifies that s_i is deemed safe code.

B. Data collection

The dataset utilized in this study was curated and structured based on the real-world vulnerability dataset collected and annotated by Wartschinski et al[6]. Furthermore, it was expanded from the original 7 types to encompass 10 types of vulnerabilities. Data collection involved crawling a substantial volume of source codes and diff files pertaining to Python files from the GitHub open-source community. Subsequently, irrelevant content was filtered out, and the differences between old and new versions were analyzed using the diff files to construct a vulnerability dataset $diff(a_i) = p_i$. The intricate process of determining vulnerable code fragments is elucidated in Fig 2.

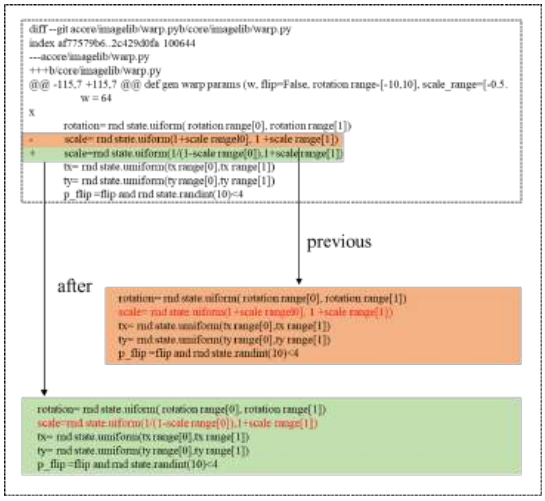


Figure 2: Source code diff file comparison fragment.

C. Data processing

The CodeBERT model is specifically designed to directly learn and analyze source code. In order to effectively leverage the capabilities of this model, it is imperative that the dataset contains contextual semantic information from the training data. Consequently, when labeling the dataset, careful consideration is given to the context surrounding each vulnerable code snippet. Furthermore, after identifying the vulnerable lines of code within the diff file, directly

marking the code snippets above and below as "vulnerable label numbers" may result in inconsistent block sizes for the remaining portions of the file. To ensure proper data processing and prevent imbalanced datasets[10], it is imperative to treat both vulnerable and non-vulnerable segments equally and proportionally during the labeling phase. Therefore, the proposed approach involves dividing the data into chunks of uniform length. If these chunks overlap with vulnerable code segments, they are assigned a vulnerable type label; otherwise, they are labeled as non-vulnerable. This strategy ensures consistency and balance in the dataset, as illustrated in Fig 3.

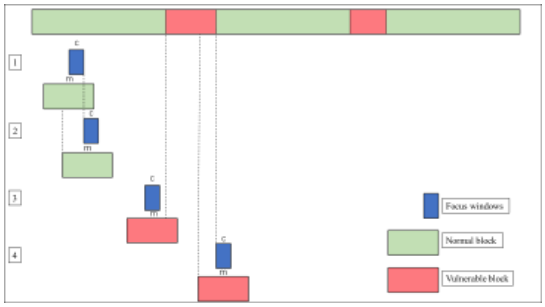


Figure 3: Source code chunking.

Utilize a focus window to systematically traverse the entire source code in increments of length "c." The focus window, denoted by a small blue box, commences and concludes at the characters that terminate individual tokens of Python code (e.g., colons, parentheses, or spaces) to prevent token fragmentation. Within this focus window, ascertain the surrounding context with an approximate length of "m" and regard it as a code block[11]. If the code block encompasses code statements bearing security risks, it is flagged as containing vulnerabilities; otherwise, it is designated as normal code.

D. Representation of source code

Liu et al[11]. argue for the necessity of Abstract Syntax Tree (AST) representation in mining patterns from code. Conversely, Russel et al[12]. and Hovsepyan et al[13]. have demonstrated that AST representation is not always essential, as code can also be effectively modeled using textual representations. Moreover, there are many similarities between language and source code, such as the inherent structure of language and source code, both of which are carriers for expressing meaning and conveying information.

Deep neural networks are adept at modeling diverse forms of sequence data, encompassing natural language, sensor data, and even music [citation needed], and have demonstrated remarkable performance in these domains. Consequently, they have found successful application in directly modeling source code as text.

E. Vulnerability detection model based on Transformer

1) Transformer: The Transformer is a deep learning model architecture grounded in the attention mechanism, originally introduced by Vaswani et al. A key characteristic of this model is its departure from the conventional recurrent neural network (RNN) structure, opting instead for a self-attention mechanism to capture dependencies across different positions within the input sequence[25]. Illustrated in Fig 4, the Transformer architecture primarily comprises two components: the encoder and the decoder. The encoder is tasked with transforming the input sequence into a contextually informed encoded representation, while the decoder utilizes this encoded representation to gen- erate the output sequence. Central to both components are key elements such as position encoding, multi-head attention, and fully connected layers.[14]. These operations are succinctly expressed in matrix form as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

here d_k is the key vector's dimension, and Q, and K and V are matrices packed with all (multi-head) queries, (multi-head) keys and (multi-head) values, respectively.

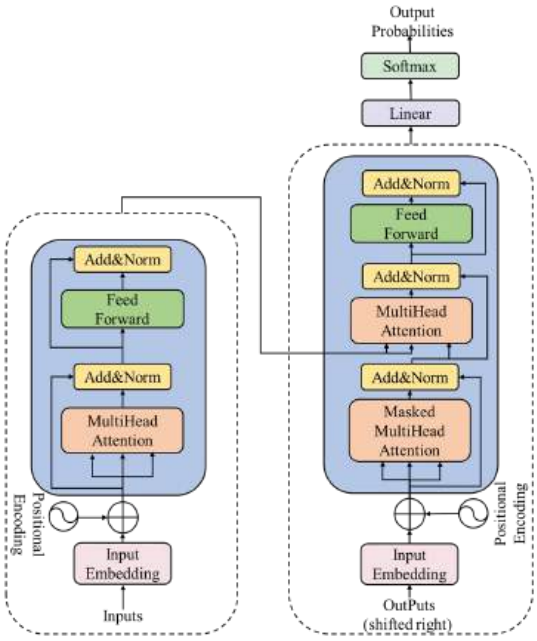


Figure 4: Transformer architecture.

2) CodeBERT: CodeBERT is a dual-mode pre-trained transformer model designed for programming and natural language tasks[8]. It undergoes training on bimodal data, consisting of both code and documents, sourced from languages such as Python, Java, JavaScript, Ruby, Go, and PHP.

The system employs BERT and CodeBERT models, which are constructed upon the Transformer architecture.

These models extend and refine the Transformer to acquire a more profound understanding of the semantic representation within the source code, thereby capturing essential information and semantics embedded in the code. This structural enhancement enables the models to exhibit heightened accuracy and superior generalization capabilities in detecting vulnerabilities within Python source code.

III. EMPIRICAL EVALUATION

A. Datasets

Currently, there exists a scarcity of open datasets specifically tailored for Python source code vulnerability detection, and those available are often lacking in comprehensiveness. Hence, this study utilizes a dataset founded upon the real-world vulnerability dataset assembled and annotated by Wartschinski et al[6]. The dataset is meticulously organized by scrutinizing the disparities between the old and new versions of diff files and employing a method that retrieves security-related keywords. Through this process, the dataset is expanded to encompass ten distinct categories.At the same time, in order to train the vulnerability detection model and the code vulnerability type classification model, we filtered the data and assigned labels. Among the 10 categories of code features, we have a total of 434,040 training data labels, including 309,682 vulnerability codes and 124,358 security codes.To further refine the dataset, a partitioning scheme is implemented, allocating the data into proportions of 0.7:0.15:0.15.

B. Evaluation metrics

Evaluation metrics play a pivotal role in machine learning assessment. Each metric underscores a different aspect, with the following four key concepts commonly forming the basis for evaluation in prediction and classification tasks: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). ”Positive” and ”negative” pertain to predictions, where a positive prediction denotes a ”fragile” outcome and a negative prediction signifies a ”not fragile” outcome.

Precision, defined as the ratio of true positives to all predicted positives, serves as a metric for assessing the accuracy of a model.

$$\text{precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall, also referred to as sensitivity, quantifies the proportion of correctly identified positive instances relative to the total actual positive instances.

$$\text{recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score, a balanced metric derived from the harmonic mean of precision and recall, provides a comprehensive assessment of a model's performance, incorporating both precision and recall into a single score.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Accuracy, a fundamental metric in machine learning evaluation, represents the proportion of correct predictions relative to all predictions made by the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

C. Result & analysis

1) Experimental parameters: Hyperparameters exert significant influence on the predictive performance of deep learning models. This study investigated the variations in accuracy, precision, recall, and F1 Score evaluation metrics of the model across specific batch sizes and various training durations.

The number of training epochs significantly influences the predictive performance of the model. If the number of epochs is set too low, the model may fail to adequately capture the underlying features of the training data, leading to underfitting and poor predictive accuracy. Conversely, setting the number of epochs too high may result in the model memorizing the training data rather than learning generalizable patterns, resulting in overfitting and reduced generalization ability. Hence, this experiment examined training epochs ranging from 1 to 10 and analyzed their impact on accuracy, precision, recall, and F1 Score. The fluctuations in these metrics across the ten epochs are depicted in Fig 9 below.

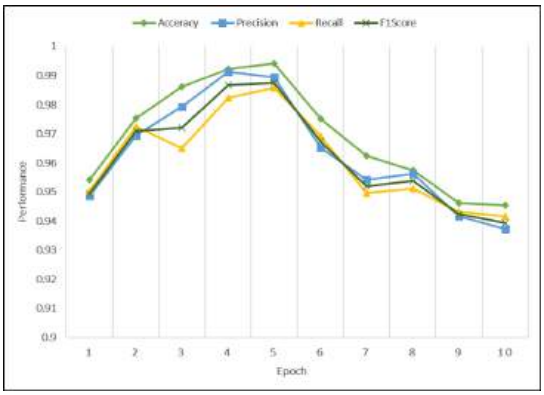


Figure 5: Changes in model indicators at different times of training.

The analysis of model performance across different training epochs, as illustrated in Fig 9, reveals that the model achieves optimal performance when trained for 5 epochs. During this period, the accuracy, precision, recall, F1 Score, and other metrics consistently stabilize at higher levels. As a result, it is recommended to set the number of training epochs for the model to 5.

Table I: Model hyperparameter settings

model	hidden size	hidden layers	attention heads	batch_size	Epoch
LSTM	100	1	0	128	100
BERT	768	12	12	16	10
CodeBERT	768	12	12	16	10

The analysis of model performance across different training epochs, as depicted in Fig 6, indicates that the model consistently achieves optimal performance when trained for 5 epochs. During this period, metrics such as accuracy, precision, recall, and F1 Score stabilize at higher levels, suggesting robust performance. Therefore, it is recommended to set the number of training epochs for the model to 5.

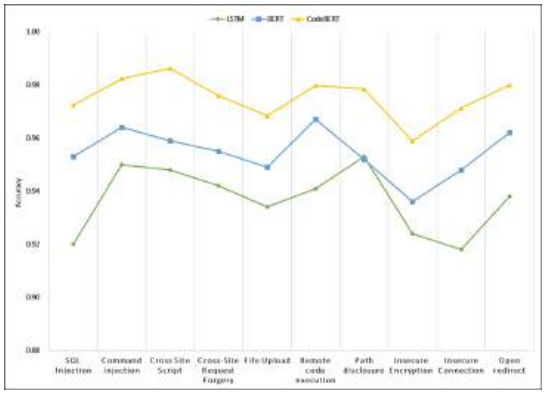


Figure 6: Accuracy indicator comparison.

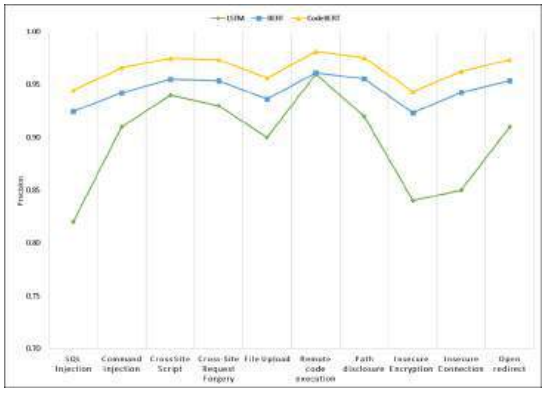


Figure 7: Precision indicator comparison.

Furthermore, experimental results demonstrate that varying the batch size has negligible impact on the accuracy, precision, and F1 Score of the model. Consequently, it is deemed appropriate to set the batch size to 16. Additionally, given the utilization of the BERT and CodeBERT models, parameters such as hidden layer size, number of hidden layers, dropout ratio, and activation function selection remain

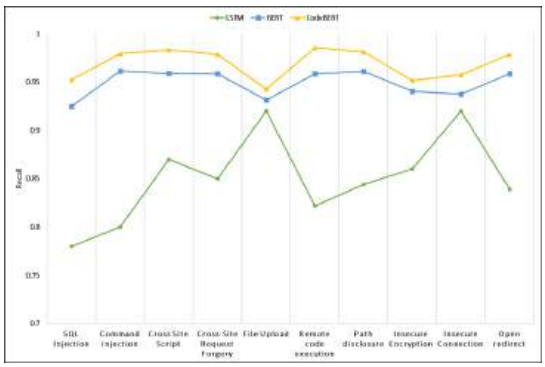


Figure 8: Recall indicator comparison.

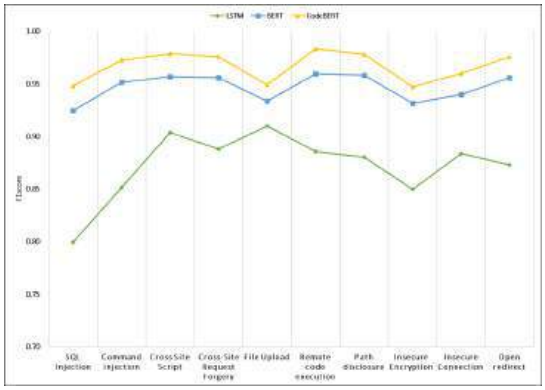


Figure 9: F1Score indicator comparison.

fixed. Thus, the various hyperparameter settings of the model are summarized in Table 1.

2) Model performance: To validate the effectiveness of the CodeBERT model, a comparative analysis was conducted against the BERT and LSTM deep learning models. This comparison was performed across 10 distinct vulnerability datasets, ensuring consistency in dataset selection, data preprocessing methodologies, and optimal hyperparameter configurations. The performance metrics of the three models were evaluated under varying dataset conditions, as depicted in Figures 7 through 10.

In the comparison of multiple metrics across different datasets, it is evident that both the CodeBERT and BERT models exhibit notable improvements over the LSTM model utilized by Wartschinski et al[6]. in source code vulnerability detection. Notably, there are enhancements observed in accuracy, F1Score, and recall metrics, with accuracy ranging 95.89% - 98.63%, and F1Score 94.74% - 98.33%. These findings underscore the efficacy of BERT and CodeBERT, built on the Transformer model, in source code vulnerability detection. Leveraging the attention mechanism, these models not only capture sequence features effectively but also prioritize key vulnerability characteristics, thus leading to enhanced detection performance compared to LSTM-based approaches.

Table II: CodeBERT multi-classification model evaluation

Type	Accuracy	Precision	Recall	F1Score
SQL Injection	0.97	0.94	0.95	0.95
Command injection	0.99	0.97	0.98	0.97
Cross Site Script	0.98	0.97	0.98	0.98
Cross-Site Request Forgery	0.97	0.97	0.98	0.98
File Upload	0.98	0.96	0.94	0.95
Remote code execution	0.98	0.98	0.99	0.98
Path disclosure	0.98	0.98	0.98	0.98
Insecure Encryption	0.96	0.94	0.95	0.95
Insecure Connection	0.97	0.96	0.96	0.96
Open redirect	0.98	0.97	0.98	0.98

In the vulnerability type classification experiment, since the code and data set of Guo et al[7]. have not been made public, comparative experiments cannot be conducted. As shown in Table 2, the performance of the CodeBERT model in multi-classification experiments is demonstrated. As can be seen from the table, among the 10 vulnerability types, the accuracy rate of CodeBERT multi-classification is about 96%-98%, the precision rate is 94%-98%, the recall rate is 94%-99%, and the F1-score is 95%-98%, all indicators show excellent results.

IV. CONCLUSION

This article builds a deep learning-based CodeBERT vulnerability mining model for Python source code. In the past, due to the lack of public Python source code data sets, this paper combined diff files and Security keywords to build a multi-label vulnerability data set based on contextual semantics. At the same time, in order to deeply learn semantic information and extract deep semantic structures, a multi-classification CodeBERT detection model was constructed. Compared with LSTM and BiLSTM, this model can extract and learn semantic information at a deeper level.

When performing multi-classification, we directly added corresponding classification headers in CodeBERT based on the multi-label vulnerability data set to determine the type of vulnerability. Since the code of Guo et al[7]. has not been made public, we cannot conduct comparative experiments, but according to the indicators provided in the paper, our method is more effective. Currently, the system can only detect whether there are vulnerabilities in the code, but cannot accurately locate the vulnerabilities. We will continue to work hard to achieve this goal.

REFERENCES

[1] J. Kim, Y. Yoon, K. Yi, J. Shin, and S. Center, "Scandal: Static analyzer for detecting privacy leaks in android applications," *MoST*, vol. 12, no. 110, p. 1, 2012.

[2] S. Lee, J. Dolby, and S. Ryu, "Hybridroid: static analysis framework for android hybrid applications," in *Proceedings of the 31st IEEE/ACM international conference on automated software engineering*, pp. 250–261, 2016.

[3] W. R. Bush, J. D. Pincus, and D. J. Sielaff, "A static analyzer for finding dynamic programming errors," *Software: Practice and Experience*, vol. 30, no. 7, pp. 775–802, 2000.

[4] S. Micheelsen and B. Thalmann, "A static analysis tool for detecting security vulnerabilities in python web applications," *Aalborg University, Aalborg University, 31st May*, 2016.

[5] S. Peng, P. Liu, and J. Han, "A python security analysis framework in integrity verification and vulnerability detection," *Wuhan University Journal of Natural Sciences*, vol. 24, no. 2, pp. 141–148, 2019.

[6] L. Wartschinski, Y. Noller, T. Vogel, T. Kehrer, and L. Grunske, "Vudenc: vulnerability detection with deep learning on a natural codebase for python," *Information and Software Technology*, vol. 144, p. 106809, 2022.

[7] W. Guo, C. Huang, W. Niu, and Y. Fang, "Intelligent mining vulnerabilities in python code snippets," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 2, pp. 3615–3628, 2021.

[8] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, *et al.*, "Codebert: A pre-trained model for programming and natural languages," *arXiv preprint arXiv:2002.08155*, 2020.

[9] C. Thapa, S. I. Jang, M. E. Ahmed, S. Camtepe, J. Pieprzyk, and S. Nepal, "Transformer-based language models for software vulnerability detection," in *Proceedings of the 38th Annual Computer Security Applications Conference*, pp. 481–496, 2022.

[10] M. Tan, L. Tan, S. Dara, and C. Mayeux, "Online defect prediction for imbalanced data," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2, pp. 99–108, IEEE, 2015.

[11] K. Liu, D. Kim, T. F. Bissyandé, S. Yoo, and Y. Le Traon, "Mining fix patterns for findbugs violations," *IEEE Transactions on Software Engineering*, vol. 47, no. 1, pp. 165–188, 2018.

[12] R. Russell, L. Kim, L. Hamilton, T. Lazovich, J. Harer, O. Ozdemir, P. Ellingwood, and M. McConley, "Automated vulnerability detection in source code using deep representation learning," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pp. 757–762, IEEE, 2018.

[13] A. Hovsepyan, R. Scandariato, W. Joosen, and J. Walden, "Software vulnerability prediction using text analysis techniques," in *Proceedings of the 4th international workshop on Security measurements and metrics*, pp. 7–10, 2012.

[14] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International conference on machine learning*, pp. 4055–4064, PMLR, 2018.

Two-stage Prompt-based Entity Representation in Contrastive Learning for Knowledge Graph Completion

1st Zhaoxuan Zhang
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
dut_zzx@mail.dlut.edu.cn

3rd Jian Zhao
School of Automotive Engineering
Dalian University of Technology
Ningbo Institute of Dalian University of Technology
Dalian, China
jzhao@dlut.edu.cn

2nd Nianmin Yao
School of Computer Science and Technology
Dalian University of Technology
Ningbo Institute of Dalian University of Technology
Dalian, China
lucos@dlut.edu.cn

4th Mingyu Li
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
limingyu230423@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899717

***Abstract*—The task of Knowledge Graph Completion (KGC) focuses on inferring missing entities automatically. Existing methodologies mainly fall into two categories: structure-based and description-based approaches. Recently, contrastive learning has shown great advantage in description-based KGC. However, this approach necessitates large batch sizes for negative sampling, yet increasing batch size risks introducing false negatives. Furthermore, many contrastive-based methods underutilize the rich semantic content in textual data. To address these issues, leveraging recent research in sentence representation learning and Chain of thought (CoT) in prompt engineering, we proposes a novel description-based KGC method with two-stage prompt-based template. Simultaneously, we proposed a contrastive learning model architecture with infoNCE$-\gamma$ loss. Extensive experiments on three KGC benchmarks validate the superior performance of our approach in KGC.**

***Index Terms*—Knowledge Graph Completion, Prompt Engineering, Contrastive Learning**

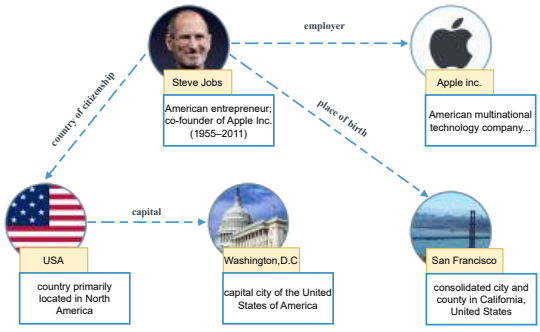


Fig. 1. An example of knowledge graph in Wikidata. Each entity has its name and textual descriptions.

I. Introduction

Knowledge graphs (KGs) are collections of large-scale facts in triplets (h, r, t), which exhibit complex graph structures and encompass abundant semantic information, playing an important role in many Natural Language Processing(NLP) applications, such as question answering [1], recommendation systems [2], and web search [3], etc. Nevertheless, human-crafted knowledge graph such as Wikidata [4], WordNet [5], and YAGO [6] suffer from inherent incompleteness, posing a significant barrier in KG applications and research.

Existing KGC methods broadly classified into two streams: structure-based and description-based. **Structure-based** KGC methods includes TransE [7], RotatE [8], and TuckER [9] etc. Although these methods are effective in representing KG structural information, they have difficulty in inferring unseen entities. Conversely, as shown in Figure 1, **description-based** methods utilize the fine-tuning of pre-trained language models(PLMs) to represent the entities and relations via textual descriptions which encompass a wealth of information. However, even with the introduction of additional information and PLMs, description-based methods still lag behind in a long time. A notable stride in this direction is SimKGC [10], which employs a contrastive learning framework inspired by visual representation learning and dense passage retrieval [11]. SimKGC is the first description-based method that outperforms structure-based models. They propose a contrastive learning framework which includes three distinct strategies about negatives sampling: in-batch negatives, pre-batch negatives, and self-negatives. However, it still has certain challenges. We identify two main issues of SimKGC: (1)

979-8-3315-2931-4/24/31.00 ©2024 IEEE

The quantity of negatives is contingent upon batch size. (2) Hard negatives sampling based on self-negative can potentially introduce false negatives. For example, instances such as (*turkiye*,*country*,*turkiye*) and (*R*,*programming language*,*R*) found in Wikidata5M. (h,r,h) could actually be a positive. To tackle these issues, galvanized by the recent application of chain-of-thought (CoT) [12], we propose the TPER-KGC, which stands for **T**wo-stage **P**rompt-based **E**ntity **R**epresentation in **C**ontrastive Learning for **K**nowledge **G**raph **C**ompletion, which improve large knowledge graph link prediction through improving entity representation embeddings. Contrary to previous work that simple concatenates entities and relations [13] [10], our approach utilizes a prompt-based template to obtain the entity embeddings. Simultaneously, we propose a contrastive learning model architecture with infoNCE$-\gamma$ loss in the KGC task, enabling the model to more easily distinguish between positive and negative examples. The advantages of our approach are that the proposed model integrate the progressive thinking implied by the Chain-of-Thought technique, devising a pioneering two-stage template for entity embeddings computation, and we reduce the number of negative samples, lowering the consumption of computational resources and computational time. We evaluate our proposed model by conducting experiments on three popular benchmarks. According to the evaluation results, our method consistently outperforms existing baseline models on WN18RR(MRR $66.6 \rightarrow 68.8$), FB15k-237(MRR $33.6 \rightarrow 36.8$), Wikidata5M(MRR $36.8 \rightarrow 41.1$). Finally, to demonstrate the effectiveness of individual components, we carry out ablation studies and a series analyses.

II. Related Work

A. Knowledge Graph Completion

Knowledge Graph Completion is a pivotal field that focuses on modeling multi-relational data to facilitate the construction of large-scale knowledge graphs. **Structure-based methods** aim to map each entity and its relation to low-dimensional vector spaces as embeddings. Many well-known methods like TransE [7] and TransH [14] conceptualize a triple (h, r, t) as a relation-specific translation from the entity h, relation r to tail entity t. RotateE [8] defines relation as a rotation from h to t in a complex space. Their embeddings satisfy the equation $\mathbf{h} \odot \mathbf{r} \approx \mathbf{t}$, where $\odot$ denotes vector multiplication. [15] proposed constraining the relation matrix to a diagonal matrix, thereby substantially reducing the parameter count in the bilinear model. **Description-based methods** integrate the Pretrained Language Models(PLMs) into KGC methods by encoding or generating facts from textual information. KG-BERT [13], StAR [16]both employed pre-trained language models to compute entity embeddings. SimKGC [10] introduced three types of contrastive learning negatives, and employs PLM to generate textual representations, aiming to maximize the cosine similarity among positives while minimizing it among negatives. Besides, several approaches [17] have extended beyond textual descriptions to also include structural information through soft prompt learning as an auxiliary measure. In addition, recent studies have attempted to fine-tune large language models (LLMs) through prompt learning. For example, the DIFT [18] method inputs all information related to neighboring entities into the model, thereby constraining the model's output within a specific range of entities. These methods tend to perform poorly on inductive KGC due to the limitations imposed on their outputs

B. Prompt Engineering

Prompt-based learning is designed to fully leverage the prior knowledge stored within PLMs which originate from GPT models [19] and swiftly used by tasks of text representation and semantic textual similarity. Although the fact that BERT [20] and RoBERTa [21] possess parameter scales much smaller than models like GPT, many studies have demonstrated that prompt-based learning exhibits extensive applicability across a variety of language models. PromptBERT [22] employs designed templates and utilizes [MASK] position as the output vector. Recently, the Chain of thought(CoT) has been proposed as a novel prompting methodology, leading to conclusive answer step by step. The essence of CoT revolves around the decomposition of complex problems into progressively solvable components. Results on CoT-BERT [23] demonstrate CoT is effective in fine-tuning tasks for PLMs. However, as a more sophisticated prompt structure, CoT remains largely unexplored in this filed. To the best of our knowledge, we are the first to integrate prompt learning and contrastive learning with knowledge graph completion.

III. Methodology

A. Notion

We delineate our approach for Knowledge Graph Completion (KGC) on a directed graph $\mathcal{G}$, representing a collection of triples (h,r,t), where h, r and t correspond to head entity, relation and tail entity respectively. Each entity $e \in (h \cup t)$ has its name and textual description, denoted as $x^e = \left(x_1^n, x_2^n, ..., x_k^n, x_{k+1}^d ..., x_l^d\right)$. The total length of each entity name and its description is truncated to a uniform length l. Given a query (h,r,?), the link prediction task aims at inferring the tail t in missing fact (h,r,t) by ranking all entities given h and r.

To tackle this task, we design the architecture of the proposed TPER-KGC, as shown in Figure2. In the following, we first introduce our two-stage prompt-based entity representation method inIII-B. Then, we describe our proposed contrastive learning model with a bi-encoder architecture, and introduce a novel denoising template strategy in III-C. Finally, we introduce our efficient training and inference strategies for KGC task inIII-D.

B. Two-stage Entity Representation

As shown in TableI, for the tail entity query (h,r,?), the query template consists of two non-adjacent [MASK] tokens and template sentence. the output of the query is:

$$S_q = \left\{x_1^h, ..., x_{l_h}^h, :, x_1^d, ..., x_{l_{hd}}^d, x_1^r, ..., x_{l_r}^r, m^1, x_1^t, ..., x_{l_t}^t, m^2\right\} \quad (1)$$

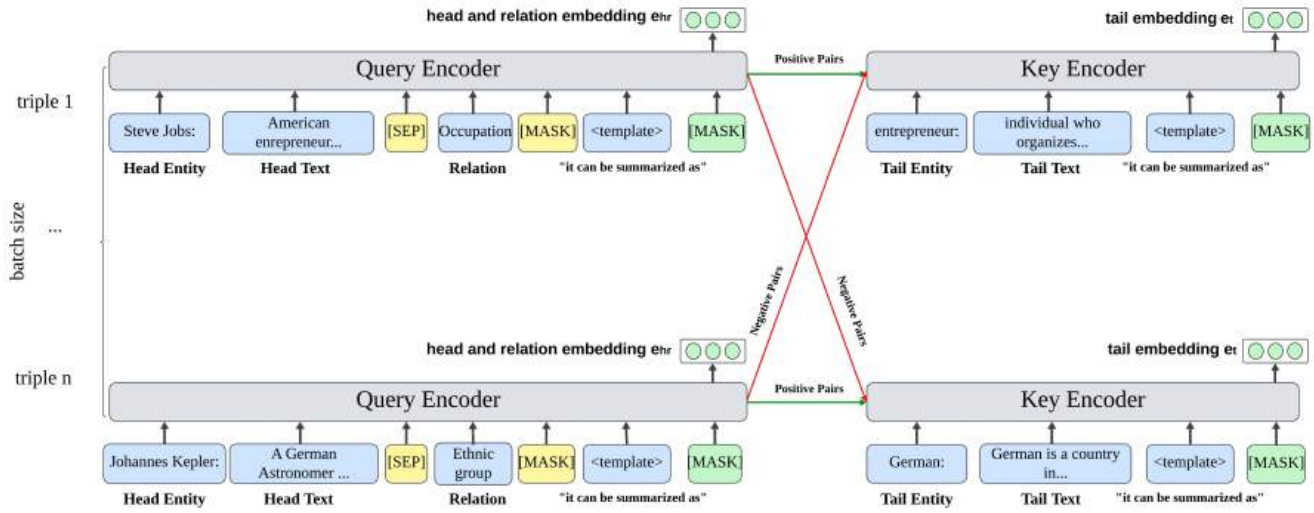


Fig. 2. Architecture of our bi-encoder contrastive learning model TPER-KGC with a batch of triples. For each triple, the Query Encoder generates embeddings by concatenating the head entity with its description, relation, two [MASK] tokens, and designed template to form the query representation. Simultaneously, the key encoder constructs the tail entity embeddings by appending the entity's description and one [MASK] template. The embeddings are then passed a contrastive framework, facilitating the learning of a semantically rich embedding space.

TABLE I
AN EXAMPLE OF OUR DESIGNED TEMPLATES. IN THE CONTEXT OF KNOWLEDGE GRAPH COMPLETION TASKS, THE QUERY IS DEFINED AS $(h,r,?)$, AND THE KEY IS DEFINED AS t.

Query Representation
Johannes Kepler: A German astronomer... [SEP] Ethic group [MASK], it can be summarized as [MASK].
Key Representation
German: German is a country in..., it can be summarized as [MASK].

where l_h, l_{hd}, l_r and l_m represent the lengths of the entity name, entity textual description, relation and manual template respectively. m^1 and m^2 denote two [MASK] tokens. The total length of the template input is:

$$\mathcal{L}_q = l_h + l_d + l_r \tag{2}$$

where $\mathcal{L}_q$ is a hyperparameter. Since each entity's name and description have different token lengths, to ensure the uniform total length of each template output, we truncate the textual descriptions l_d of the entities. By adding two [MASK] tokens to the input text, we transform the representation of entities into two stages: inference and summarization. After the forward propagation, we extract the latter [MASK] token as the ultimate embedding $\mathbf{e}_{hr}$:

$$\mathbf{e}_{hr} = BERT_{hr}(S_q) \tag{3}$$

Similarly, as key t does not necessitate inference, the template for the key requires only a single mask token for the purpose of summarization. The output of the key is:

$$S_k = \left\{x_1^t, ..., x_{l_t}^t, :, x_1^d, ..., x_{l_{td}}^d, x_1^t, ..., x_{l_m}^t, m\right\} \tag{4}$$

where l_t, l_{td} represent the length of key entity name and textual description respectively. The key template and the query template follow the same truncation principle. The total input length of the key template is:

$$\mathcal{L}_k = l_h + l_d \tag{5}$$

Consistent with the previous description, the key embedding also utilizes the mask token from the summary part as its final embedding $\mathbf{e}_{hr}$:

$$\mathbf{e}_t = BERT_t(S_k) \tag{6}$$

It is important to note that BERT$_{hr}$ and BERT$_t$ for Query and Key do not share parameters.

C. Bi-encoder Architecture

Our model adopts a bi-encoder architecture (Figure2) with negative sampling strategy which is widely adopted in visual representation learning [24]. We establish two completely separate encoders, the Query Encoder BERT$_{hr}$ and the Key Encoder BERTt, both configured with identical hyperparameters. The Query is used for encoding the head entity and relation, while the Key Encoder is for tail entity encoding. Their weights are independently updated during the training process. We applies the contrastive matching within the mini-batch, which allows the efficient utilization of negative pairs within a batch while avoiding the additional cost of negatives sampling and encoding. The candidate keys of the queries are defined as all of head entities and tail entites in a batch. Our bi-encoder model allows the efficent reuse of entity embeddings.

To enhance the performance of the model, we further process the output embeddings. Noise may be introduced when building entity representation by using prompt template because the output vector at the [MASK] position may be influenced by the presence of the template. To counter the potential noise introduced by the prompt template, we implement a denoising strategy using an empty template padded with [PAD] tokens:"$[\text{PAD}]_1[\text{PAD}]_2...[\text{PAD}]_l$, it can be summarized as [MASK]." where $l \in \{\mathcal{L}_q, \mathcal{L}_k\}$. For the attention mask of the empty template, the position of [PAD] is set to 0, and the rest positions are set to 1. In order to alignment with original template, we use the [MASK] token at the same position as in the input template as the embedding for the denoising template. The ablation study of the denoising template can be found in Section V-B Ultimately, the embeddings of the query and key are:

$$e_q = e_{hr} - \hat{e}_{hr} \tag{7}$$

$$e_k = e_t - \hat{e}_t \tag{8}$$

where $\hat{e}_{hr}$ and $\hat{e}_t$ are respectively the outputs of the denoising templates for query and key.

D. Training and Inference

Currently, most sentence representation tasks and KGC tasks based on contrastive learning utilize the InfoNCE loss [10], [24], [25] for training guidance. we also discovered that incorporating an additive margin in the loss function better differentiates between positive and negative examples. Therefore, in this paper, we utilize InfoNCE-γ as the loss function to guide the training:

$$\mathcal{L} = -\log \frac{e^{(\phi(h,r,t)-\gamma)/\tau}}{e^{(\phi(h,r,t)-\gamma)/\tau} + \sum_{i=1}^{|\mathcal{N}|} e^{\phi(h,r,t'_i)/\tau}} \tag{9}$$

In this equation, $\phi(h, r, t)$ is the score function for a candidate triple, we define $\phi(h, r, t) = \cos(\mathbf{e}_q, \mathbf{e}_k) \in [-1, 1]$ as:

$$\cos(\mathbf{e}_q, \mathbf{e}_k) = \frac{\mathbf{e}_q \cdot \mathbf{e}_k}{\|\mathbf{e}_q\| \, \|\mathbf{e}_k\|} \tag{10}$$

The parameter $\gamma > 0$ encourages the correct triple (h,r,t) to achieve higher score. The temperature τ can adjust the importance of negatives. In this paper, we set τ as a trainable parameter. In general, this formulation guides the model to maximize the score between the query h, r and its positive sample t, while simultaneously minimizing the score between other unrelated entity samples t' within the same batch. For the prediction (h,r,t), we compute the cosine similarity which is widely adopted in contrastive learning between $\mathbf{e}_q$ and all entities, and pick up the largest as answer.

$$\underset{t_i}{\text{argmax}} \ \cos(\mathbf{e}_q, \mathbf{e}_{k_i}), \ k_i \in \mathcal{E} \tag{11}$$

The inference phase of KGC tasks requires computing a large number of entity embeddings, for our bi-encoder structure, we can pre-compute and store the embeddings of all entities in advance, thereby saving a significant amount of inference time.

IV. Experiments

A. Experimental Setup

Datasets We evaluate our model on three datasets which widely used in KGC task: **WN18RR** [26], **FB15K237** [27] and **Wikidata5M** [28]. Detailed statistics of these datasets are presented in TableII. WN18RR consists about 41k synsets and 11 relations from WordNet [5] and it is constructed by removing the inverse relations because of the test set leakage [26], [27]. FB15k237 is a subset of Freebase [29], it consists about $15k$ entities and 237 relations which has removed the inverse relation either. Wikidata5M is much larger in scale which consists of $\sim 4.6M$ entities, 822 relations, and ~ 20 million triples. It provides two settings: transductive and inductive. Following most of KGC methods, we use the transductive version of Wikidata5M. All entites in the test set also appear in the training set.

For textual description, we maintain consistency with other baselines by utilizing the description for WN18RR and FB15k-237 provided by KG-BERT [13] and the text from Wikidata.

Baselines We compare our model with leading KGC methods, encompassing both structure-based (TransE, RotatE, DistMult) and description-based (KG-BERT, StAR, KEPLER, SimKGC, CSProm-KG, DIFT) approaches. Furthermore, to demonstrate the superiority of our method, we have selected the most popular large language models. ChatGPT$_{oneshot}$ is a baseline proposed by AutoKG [30], and We also incorporate the ideas from SimKGC in conjunction with LoRA to fine-tune LLaMA-7B, and we report the results in this paper.

For a fair comparison, we reimplement a version of SimKGC with the batch size of 256, aligning it with the batch size used in our model. This approach facilitates a more accurate and fair assessment of model performances under similar operational conditions. **Evaluation metrics** Following previous work, our evaluation encompasses four standard automatic metrics: mean reciprocal rank (MRR), and Hits@k(k $\in$\{1,3,10\}). Metrics for link prediction are based on the rank of the correct entity in a list of all entities, ranked by their plausibility. MRR is the average reciprocal rank of all test triples, providing a comprehensive measure of the model's overall ranking accuracy. Hits@k measures the proportion of correct entities ranked in the top K. MRR and H@k are reported under the *filtered setting* [7], the *filtered setting* is a common practice that removes other correct entities (which also constitute triples existing in the KG) from the list. Generally, a good model is expected to achieve higher MRR and Hits@k.

Implementation detail The encoders are initialized with open source project *bert-base-uncased* [1] provided by hugging face. Larger models may lead to better performance, but they also impact the fairness of experimental comparisons. Therefore,

[1]https://huggingface.co/bert-base-uncased

TABLE II
STATISTICS OF THE DATASETS USED IN THIS PAPER.

dataset	#entity	#relation	#train	#valid	#test
WN18RR	$40,943$	11	$86,835$	3034	3134
FB15k-237	$14,541$	237	$272,115$	$17,535$	$20,466$
Wikidata5M-Trans	$4,594,485$	822	$20,614,279$	$5,163$	$5,163$

we do not consider using larger model in this paper. For the three datasets, we standardized the majority of hyperparameters except learning rate and training epochs. We set the learning rate with ranges $\{10^{-5}, 3\times10^{-5}, 5\times10^{-5}\}$. Follow SimKGC [10], entity descriptions (excluding the template section) are truncated to a maximum of 50 tokens for fair comparison. Temperature τ is initialized to 0.05, and the additive margin for InfoNCE loss is 0.02. We use AdamW optimizer with linear learning rate decay. Models are trained with batch size 256 on a single Nvidia Tesla V100 GPU. For the WN18RR, FB15k-237, and Wikidata5M datasets, we train for 50, 10, and 1 epochs, respectively.

B. Main Results

In Table III and Table IV, our proposed model TPER-KGC outperforms state-of-the-art methods by a large margin on WN18RR and Wikidata5M datasets. TPER-KGC has a significant performance advantage in WN18RR and Wikidata5M, while marginlly trailing in FB15k-237. To the best of our knowledge, TPER-KGC is the best description-based KGC method on this dataset.

We analyze the reason that description-based methods fail to widen the gap with structure-based methods on FB15k-237 in two aspects: sparsity and description. First, according to the TableII, the graph for the FB15k-237 contains fewer entities and more average degree ($\sim$ 37) per entity, which indicates that entites in FB15k-237 have access to rich structural information while entities in WN18RR are more likely to be long-tail. Second, FB15k-237 are not predictable based on the available information [34]. This limitation may harm the training of description-based models. In the large-scale KG Wikidata5M, TPER-KGC significantly outperforms previous methods, highlighting the scalability and effectiveness of our approach.

In terms of inference time and computational resource consumption, the most expensive part is the forward pass with BERT. Our model need to do $2\times|\mathcal{T}|+|\mathcal{E}|$ times BERT forward pass, where $|\mathcal{T}|$ and $|\mathcal{E}|$ denote the size of test size and the number of entities, respectively. It is important to note that for large-scale datasets, such as Wikidata5M, $|\mathcal{T}| << |\mathcal{E}|$ ($5,163$ test samples and $\sim$ 4.6M entites). Thus, by employing the pre-computation strategy mentioned in SectionIII-D, we can reduce the time complexity to O($\mathcal{T}$). Our method completes training in $\sim$ 14 hours and inference in $\sim$ 40 minutes on a single GPU, while cross-encoder models such as KG-BERT [13] would require an estimated time of 3000 hours and large-scale negative sampling method SimKGC [10] has the same training and inference time as TPER-KGC, but due to its requirement for large-scale negative sampling, it requires 4 GPUs for training and 2 GPUs for inference.

On the WN18RR and FB15K-237 datasets, TPER-KGC significantly outperformed LLM-based models, regardless of the integration with any of the embedding-based models. However, on the WikiData5M dataset, LLM-based models achieved better performance in terms of MRR, attributed to LLaMA's 7 billion parameters compared to the 110 million parameters of the BERT-based model we used. Therefore, such a comparison is not entirely fair. Additionally, we observed that although ChatGPT demonstrates superior performance in knowledge question-answering tasks, its effectiveness in knowledge graph link prediction under zero-shot conditions is not satisfactory.

V. ABLATION STUDY AND ANALYSIS

A. Two-stage Template Evaluation

To validate the effectiveness of our two-stage template strategy, we conducted experiments focusing on the individual components of our designed template. For the query manual template, "*entity name: entity description* [SEP] *relation* [MASK], it can be summarized as [MASK]", we define its prefix as "*entity name: entity description* [SEP] *relation* [MASK]" and its suffix as "*entity name: entity description* [SEP] *relation* [SEP] it can be summarized as [MASK]". Based on the experimental results shown in TableV, it can be observed that employing a two-stage manual template is superior to using either of the individual sub-stages alone. This further confirms that the immense potential of CoT as a guiding signal for prompt-based learning. Additionally, we considered whether adding sub-stages would yield better results. Regrettably, due to computational resource limitations, this experiment could not be completed. Under the constraints of the existing computational resources, increasing sub-stages would shorten the textual descriptions of entities, impairing the ability of PLM to effectively discern differences between entities. Furthermore, adding sub-stages would significantly increase the time required for entity embedding encoding and denoising embedding encoding, thereby affecting inference efficiency. We plan to investigate this further in our future work.

B. Template Denoising Strategy Evaluation

We also embark on an ablation study on the template denoising method. As shown in TableVI, the main operation of [PAD] denoising is to inject [PAD] placeholders of identical length as the input sentence, accompanied by the corresponding 1 to 0 to the attention mask.

TABLE III
MAIN RESULTS FOR WN18RR AND FB15K-237 DATASETS. MOST RESULT NUMBERS ARE FROM SIMKGC [10]. ADDITIONALLY, WE REIMPLEMENT THE SIMKGC WITH A BATCH SIZE OF 256 WHICH DESIGNATED AS SIMKGC (SMALL).

Method	WN18RR				FB15k-237			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
structure-based methods								
TransE [7]	24.3	4.3	44.1	53.2	27.9	19.8	37.6	44.1
DistMult [15]	44.4	41.2	47.0	50.4	28.1	19.9	30.1	44.6
RotatE [8]	47.6	42.8	49.2	57.1	33.8	24.1	37.5	53.3
TuckER [9]	47.0	44.3	48.2	52.6	35.8	26.6	39.4	54.4
NBFNet [31]	55.1	49.7	57.3	66.6	**41.5**	**32.1**	**45.4**	**59.9**
description-based methods								
KG-BERT [13]	21.6	4.1	30.2	52.4	-	-	-	42.0
MTL-KGC [32]	33.1	20.3	38.3	59.7	26.7	17.2	29.8	45.8
StAR [16]	40.1	24.3	49.1	70.9	29.6	20.5	32.2	48.2
SimKGC [10]	66.6	58.7	71.7	80.0	33.6	24.9	36.2	**51.1**
SimKGC (tiny)	65.9	57.4	71.5	80.0	32.8	23.9	35.5	51.1
CSProm-KG [17]	57.5	52.2	59.6	67.8	35.8	26.9	39.3	53.8
LLM-based methods								
ChatGPT$_{oneshot}$	-	21.2	-	-	-	26.7	-	-
LLaMA + SimKGC	39.1	6.5	69.5	79.8	23.6	7.4	33.5	50.3
DIFT [18]	68.6	61.6	73.0	80.6	40.2	33.8	41.8	52.8
TPER-KGC	**68.8**	59.2	**75.6**	**84.6**	36.8	28.3	39.5	**53.6**

TABLE IV
MAIN RESULTS FOR THE WIKIDATA5M TRANSDUCTIVE DATASET.

Method	Wikidata5M			
	MRR	H@1	H@3	H@10
structure-based methods				
TransE [7]	25.3	17.0	31.1	39.2
RotatE [8]	29.0	23.4	32.2	39.0
description-based methods				
DKRL [33]	16.0	12.0	18.1	22.9
KEPLER [28]	21.0	17.3	22.4	27.7
SimKGC(small)	35.6	31.2	37.3	43.3
SimKGC [10]	35.8	31.3	37.6	44.1
CSProm-KG [17]	38.0	34.3	39.9	44.6
LLM-based methods				
ChatGPT$_{oneshot}$	-	29.0	-	-
LLaMA + SimKGC	42	33.8	44.8	50.6
TPER-KGC	41.1	**34.7**	44.1	**52.9**

TABLE V
ABLATION STUDY BASED ON **WN18RR** FOR OUR TWO-STAGE ENTITY REPRESENTATION METHOD. ALL EXPERIMENTAL CONFIGURATIONS FOR THREE GROUPS ARE IDENTICAL. "PREFIX+SUFFIX" INDICATES THE USE OF OUR COMPLETE TEMPLATES AS DESCRIBED IN TABLEI.

	MRR	H@1	H@3	H@10
TPER-KGC (only prefix)	62.8	54.5	67.8	77.6
TPER-KGC (only suffix)	64.8	56.9	70.6	81.5
TPER-KGC (prefix+suffix)	**68.8**	**59.2**	**75.6**	**84.6**

TABLE VI
ABLATION STUDY FOR OUR TEMPLATE DENOISING STRATEGY BASED ON **WN18RR**.

	MRR	H@1	H@3	H@10
TPER-KGC (w/o denoise)	67.9	58.8	75.4	81.0
TPER-KGC (w/ denoise)	**68.8**	**59.2**	**75.6**	**84.6**

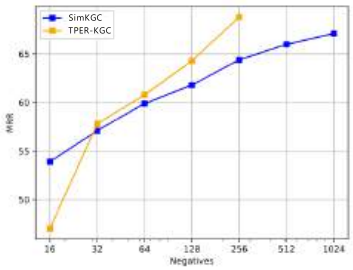


Fig. 3. MRR score of TPER-KGC and SimKGC with different number of negatives based on WN18RR.

The empirical findings demonstrate that the denoising template not only achieves good results in sentence representation tasks, but also exhibits good effectiveness on other contrastive-based tasks. We posit that one possible reason is the introduction of fixed structures and grammatical information through manually designed templates. Removing these elements could lead to better entities and relation representation outputs.

C. How many Negative Samples do we need?

As shown in Figure3, MRR metrics on WN18RR vary for TPER-KGC and SimKGC under different numbers of negatives. Remarkably, TPER-KGC outperforms SimKGC with a batch size of 1024, even when operating at a lower batch size of 256. Note that we do not count the pre-batch negatives of SimKGC for a fair comparison. However, in the fileds of link prediction and recommendation, increasing the number of negatives may not only introduce noise [35],such as false negatives, but also excessive GPU memory usage. It can be observed from the graph that the negative sampling in the SimKGC method increased from 256 to 1024, yet the improvement of MRR is quite limited. Therefore, we set the baseline batch size for this paper to 256, striking a balance between model performance and computational efficiency.

D. Entity Visualization

To further validate the efficacy of our proposed model, we selected the six largest categories [2] from all entities in the Wikidata5M dataset for two-dimensional visualization. From

[2]categories are determined by the relation "instance of".

each category, fifty entities were randomly chosen. Each entity embeddding is computed with Key Encoder in SectionIII-C. As illustrated in Figure4, the embeddings of entities from different categories are well-separated, demonstrating the high quality of our proposed model.

It is not difficult to observe that there is some overlap between the entities of 'community' and 'village' in the graph. This phenomenon is reasonable, as these two categories of entities are conceptually not entirely distinct.

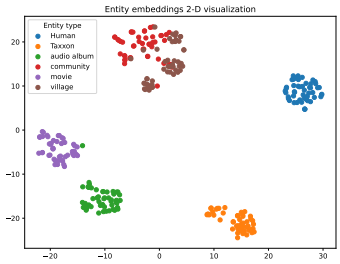


Fig. 4. 2-D visualization of entity embeddings from the Wikidata5M-trans dataset with t-SNE [36]

E. Sparsity Analyses

To analyze the impact of triplet sparsity on KGC tasks, methods such as TransE [7] and SimKGC [10] categorize entities according to four relation types: one-to-one(1-1), one-to-many(1-n), many-to-one(n-1) and many-to-many(n-n). As shown in TableVII, although TPER-KGC significantly outperforms the baseline across multiple relation categories, the model still struggles to identify the correct candidate answers in scenarios with multiple candidate entities, referred to as "1-n" and "n-n". In queries involving relations like "has part" and "child", there are typically multiple correct key entities. However, for a specific triplet (h,r,t), the evaluation metrics considers there to be only a single key entity t, leading to poor evaluation performance.

In addition, we further divided triplets into six categories based on the degree of entities. In the Wikidata5M-trans dataset, all entities in the test set had previously appeared in the training set. Figure5 shows that TPER-KGC perform well on key entities with very large in-degree (nodes with more than 100 neighbors). Additionally, we observed that reverse inference for head entities is significantly more challenging compared to tail entity inference. The main reasons for this phenomenon are two-fold. Firstly, the majority of triplets constituting the knowledge graph are of the 'n-1' structure, and their scale is much larger than that of '1-n'. For instance, in the case of head entity queries (*?*, *country of citizenship*, *united states*), there are 87 correct candidate entities in the test set. This abundance of candidate entities reduces the probability of the correct entity being selected. Secondly, the template designed in SectionIII-B is more suited for sequential inference and does not include a specific design for head entity inference. In our future work, we plan to integrate context information and design an additional two-stage template to address the issue of poor head entity inference performance.

TABLE VII
MRR FOR DIFFERENT KINDS OF RELATIONS ON WIKIDATA5M DATASET

Method	1-1	1-n	n-1	n-n
SimKGC	30.4	**8.3**	71.1	10.6
TPER-KGC	**38.8**	6.4	**80.5**	**11.7**

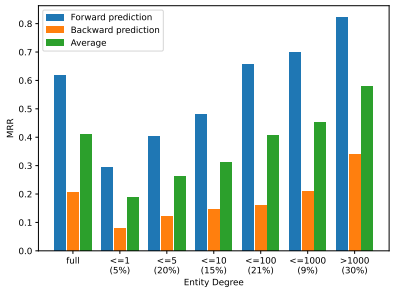


Fig. 5. MRR for forward tail entity prediction and backward head entity prediction grouped by entity degree on Wikidata5M. Group weight is given in brackets.

VI. Conclusion and Future Work

In this paper, we introduce TPER-KGC, a pioneering description-based approach for Knowledge Graph Completion (KGC), marking the first instance of integrating Chains of Thought (CoT) with contrastive learning in the realm of KGC. By designing a bi-encoder structure and denoising strategy, TPER-KGC has significantly saved computational resources and time compared to previous description-based methods. Experiments on the WN18RR, FB15K-237, and Wikidata5M datasets show that TPER-KGC substantially outperforms state-of-the-art methods.

Up to now, there has been no description-based methods that successfully integrates the rich graph structural information of knowledge graphs. In our future work, we will explore the enhanced utilization of the graph structure of knowledge graphs in KGC tasks.

References

[1] H. Sun, T. Bedrax-Weiss, and W. Cohen, "PullNet: Open domain question answering with iterative retrieval on knowledge bases and text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 2380–2390. [Online]. Available: https://aclanthology.org/D19-1242

[2] J. Huang, W. X. Zhao, H. Dou, J. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, K. Collins-Thompson, Q. Mei, B. D. Davison, Y. Liu, and E. Yilmaz, Eds. ACM, 2018, pp. 505–514. [Online]. Available: https://doi.org/10.1145/3209978.3210017

[3] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[4] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[5] G. A. Miller, "WordNet: A lexical database for English," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992. [Online]. Available: https://aclanthology.org/H92-1116

[6] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, Eds. ACM, 2007, pp. 697–706. [Online]. Available: https://doi.org/10.1145/1242572.1242667

[7] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data." in *Conference and Workshop on Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., March 2013, pp. 2787–2795.

[8] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[9] I. Balazevic, C. Allen, and T. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5185–5194. [Online]. Available: https://aclanthology.org/D19-1522

[10] L. Wang, W. Zhao, Z. Wei, and J. Liu, "Simkgc: Simple contrastive knowledge graph completion with pre-trained language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 4281–4294.

[11] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 6769–6781. [Online]. Available: https://aclanthology.org/2020.emnlp-main.550

[12] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[13] L. Yao, C. Mao, and Y. Luo, "Kg-bert: Bert for knowledge graph completion," *ArXiv preprint*, vol. abs/1909.03193, 2019. [Online]. Available: https://arxiv.org/abs/1909.03193

[14] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, C. E. Brodley and P. Stone, Eds. AAAI Press, 2014, pp. 1112–1119. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531

[15] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6575

[16] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proceedings of the Web Conference 2021*, 2021, pp. 1737–1748.

[17] C. Chen, Y. Wang, A. Sun, B. Li, and K.-Y. Lam, "Dipping plms sauce: Bridging structure and text for effective knowledge graph completion via conditional soft prompting," *arXiv preprint arXiv:2307.01709*, 2023.

[18] Y. Liu, X. Tian, Z. Sun, and W. Hu, "Finetuning generative large language models with discrimination instructions for knowledge graph completion," *arXiv preprint arXiv:2407.16127*, 2024.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[22] T. Jiang, J. Jiao, S. Huang, Z. Zhang, D. Wang, F. Zhuang, F. Wei, H. Huang, D. Deng, and Q. Zhang, "Promptbert: Improving bert sentence embeddings with prompts," *arXiv preprint arXiv:2201.04337*, 2022.

[23] B. Zhang, K. Chang, and C. Li, "Cot-bert: Enhancing unsupervised sentence representation through chain-of-thought," *arXiv preprint arXiv:2309.11143*, 2023.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[25] Y. Yang, G. H. Ábrego, S. Yuan, M. Guo, Q. Shen, D. Cer, Y. Sung, B. Strope, and R. Kurzweil, "Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, August 2019, pp. 5370–5378.

[26] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[27] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 1499–1509. [Online]. Available: https://aclanthology.org/D15-1174

[28] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[29] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[30] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, and N. Zhang, "Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities," *World Wide Web*, vol. 27, no. 5, p. 58, 2024.

[31] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang, "Neural bellman-ford networks: A general graph neural network framework for link prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 476–29 490, 2021.

[32] B. Kim, T. Hong, Y. Ko, and J. Seo, "Multi-task learning for knowledge graph completion with pre-trained language models," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 1737–1743. [Online]. Available: https://aclanthology.org/2020.coling-main.153

[33] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, D. Schuurmans and M. P. Wellman, Eds. AAAI Press, 2016, pp. 2659–2665. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12216

[34] Y. Cao, X. Ji, X. Lv, J. Li, Y. Wen, and H. Zhang, "Are missing links predictable? an inferential benchmark for knowledge graph completion," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021, pp. 6855–6865. [Online]. Available: https://aclanthology.org/2021.acl-long.534

[35] C. Wu, F. Wu, and Y. Huang, "Rethinking infonce: How many negative samples do you need?" *arXiv preprint arXiv:2105.13003*, 2021.

[36] L. V. D. Maaten and G. E. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

Sequential Semantic Descriptor from 3D Point Clouds for Place Recognition

1st Chujia Lin
South China University of Technology
Guangzhou, China
202221017805@scut.edu.cn

2nd Yiqi Liu
South China University of Technology
Guangzhou, China
liuyiqi@scut.edu.cn

3rd An Chen*
South China University of Technology
Guangzhou, China
chenan@scut.edu.cn

4th Hongxia Gao
South China University of Technology
Guangzhou, China
gaohongxia@scut.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899587

***Abstract*—The closed-loop detection of 3D point clouds remains a significant challenge due to the complexities involved in generating effective descriptors that are robust to occlusion and viewpoint changes. Unlike most existing methods focusing on extracting local, global, and statistical features from the raw point clouds of a single frame, our approach emphasizes the utilization of semantic-level scene graphs derived from successive multiple frames, with superiority in robustness to environmental changes. Drawing inspiration from human cognitive processes that facilitate scene cognition by identifying semantic objects and continuous observation, this study introduces an innovative closed-loop detection method based on sequential semantic graphs. First, we propose a novel semantic graph representation method for point clouds scenarios by preserving the semantic and topological information of the original point clouds. Then, we represent a scene with consecutive multiple frames of semantic graphs . Thus, the closed-loop detection is modeled as a multi-graph similarity matching problem. A fast and efficient multi-graph similarity network is then designed to calculate the similarity. An exhaustive evaluation of the KITTI dataset shows that our method substantially outperforms state-of-the-art methods, exhibiting robust performance in the face of occlusion and viewpoint variation, with superior performance on challenging reverse closed-loop problems.**

***Index Terms*—Place recognition, 3D point clouds, Semantic graph, Sequence match**

I. Introduction

Closed-loop detection is an important issue in SLAM, pertaining to the capacity of a robot or a mobilevehicle to recognize whether previously visited locations.

Lidar-based methods have recently received much attention recently due to their enhanced stability in the face of seasonal and lighting variations. A majority of lidar-based algorithms [1]–[4], operate directly on the raw point clouds data and generate local or global descriptors via neural networks or manual design. Some segmentation-based methods [5]–[7] identify locations by matching segments belonging to partial or complete objects, which can better represent the dynamic situation. These approaches are more relevant to the way in which humans perceive their surroundings.

Inspired by this, we propose a novel method to convert the original point cloud data into a novel graph representation that aggregates semantic information. This graph representation effectively retains essential information while accounting for topological relationships, thereby enhancing the efficiency and comprehensibility of point cloud data representation.

Furthermore, it has been observed that human cognition of a scene is improved through sustained observation of objects. Inspired by this, we propose a novel method to obtain a sequential scene graph by integrating multiple consecutive frames of semantic graphs. This sequence representation is derived from the multi-dimensional scene information resent in the scene, thereby enhancing the identification of significant elements within it. Consequently, the information encapsulated in the scene becomes more comprehensive and dependable.

Furthermore, we propose a learning-based multi-graph similarity calculation strategy to solve the retrieval task, rather than simply computing the Euclidean distances of the eigenvectors. To our knowledge, we are pioneers in the application of sequence semantic maps and multi-graph matching networks for the closed-loop detection of 3D point clouds. Our contributions can be summarized as follows:

(1)In pursuit of human-like perception, we propose a novel semantic graph representation of 3D point clouds scenes, which effectively captures semantic information while modeling the topological relationships among semantic objects.

(2)We use multiple consecutive frames of semantic graphs to address the closed-loop detection challenge, reformulating the closed-loop detection problem as a multi-graph similarity matching problem.

(3)We propose a novel network is designed to estimate multi-graph matching similarity, which is subsequently utilized for closed-loop detection.

(4)Experiments on the KITTI range dataset [8] show that our approach achieves state-of-the-art performance, particularly in terms of reverse closed-loop detection and robustness to occlusions and viewpoint changes.

II. RELATED WORK

Closed detection methods based on 3D point clouds can be divided into the following types: local descriptor based methods, global descriptor based methods and segmentation based methods.

Segmentation-based methods: SegMatch [5] and SegMap [6] propose a high-level perception that divides point clouds into a set of different and distinct elements at the object level. They use a 3D CNN to encode the fragment features and identify the candidate correspondences by using the k-nearest neighbors (kNN) in the feature space. This approach is a successful attempt at human-like perception. However, it still requires a dense local mapping, and does not consider the relationships between the objects. To address the above problem, we create a new graph representation at the semantic level, making it more concise and efficient. Then graph similarity network instead of Euclidean distance to measure scene similarity to obtain better estimation.

Sequence-based method: In the closed-loop detection, there are two main ways to use sequence information: sequence matching and sequence descriptor extraction [9]. Sequence matching involves two key steps. The SeqSLAM [10] is a seminal example of sequence matching. Fast-SeqSLAM [11] utilizes an approximate nearest-neighbor algorithm that reduces the time complexity without reducing the accuracy [15]. proposed a local matching method based on an improved dynamic time warping algorithm that relaxed the constant speed assumption while reducing the time complexity. Refs. 23 and 24 use a cost matrix-based dynamic programming approach to alleviate the problem of missing frames. Ref. 12 First proposed the idea of merging multiple individual descriptors to generate a sequence descriptor. Subsequently, SeqNet [13] proposed to use one-dimensional convolution to learn frame-level features as sequence descriptors. SeqVLAD [14] presents a detailed technical classification method using sequence descriptors. It analyzes various mechanisms for fuse information from each frame and further studies the feasibility of using transformer as backbone.

III. METHODOLOGY

In this section, we propose a closed-loop detection method based on sequence semantic descriptors, which includes semantic graphs representation, sequence semantic descriptors generation, and similarity calculation of sequence semantic descriptors, as shown in Fig. 1. Our key insight is to perceive the scene from a human viewpoint, describe it at the semantic level and emphasize on coding relationships among semantic objects, and enhance cognition of the scene through continuous observation. To achieve this, we used semantic segmentation of the original point clouds to obtain instances and further collect semantic and topological information to obtain the nodes that comprise the semantic graphs and further generate semantic graphs. Then, a scenario is expressed with sequence semantic graphs to transform the closed-loop detection task into a problem of matching sequence semantic graphs. Additionally, we employed learning-based similarity calculations of sequence semantic descriptors were used to derive similarity scores between pairs of scenes.

A. Sementic Graph Method

As shown in Fig. 2, Semantic segmentation of point clouds: Some semantic segmentation methods of point clouds have been proposed recently. RangeNet++ [17] was trained on the SemanticKITTI [18] dataset that annotates the semantic category of each 3D point on the KITTI [8] range dataset, including 19 classes. In our experimental section, we used SemanticKITTI annotations as semantic information. Inspired by SGPR [25], we merge the dynamic classes into the corresponding static classes while excluding categories such as 'people' and 'others,' as they are either irrelevant or represent a minor proportion of the data. A total of 12 categories were established after merging precess. Then, we set different clustering radii according to the semantic categories and obtained the semantic instances by Euclidean clustering. Specifically, for a single frame of the point clouds $P = \left\{p_1, \ldots, p_M | p_i \in \mathbb{R}^3\right\}$, Each point p_i has a semantic label, and we cluster the points with the same semantic labels into a set of clusters $I_i = \left\{p_1, \ldots, p | p_i \in \mathbb{R}^3\right\} \subset I$, and is used to represent different objects, which also have corresponding semantic labels.

Semantic map construction: A 64-loop lidar typically captures more than 100,000 points per frame, which is huge and redundant. To reduce the data, most existing methods randomly downsample the points or project them onto a two-dimensional plane [16], [19]. In contrast, we propose the construction of topological semantic graph representations to retain critical information by maintaining both semantic information and semantic instances, thereby offering a more efficient and meaningful representation.

For each instance I_i, we keep its semantic category l_i and the centroid $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ of the point set, forming the node features $f_i \in \mathbb{R}^{3+N}$.. The semantic features are encoded using a one-hot encoding scheme, with the length corresponding to the number of semantic categories N. In addition, we preserve the relationships between nodes that are within a specified spatial distance, which we interpret as adjacency, and the adjacency information is retained in the global data. Thus, these nodes together form a semantic graph that can represent a point cloud scene.

Sequence semantic graph construction: In the analysis of a scene, we not only derive the point clouds from a single frame, but also obtain the point clouds from consecutive frames to represent the scene. Then, the semantic graph construction of the consecutive multi-frame point clouds is performed, and the semantic graph of the consecutive frames, namely the sequence semantic map, serves as a representation of the scene.

Closed-loop detection was established by evaluating the similarity between two scenarios, which we reformulated as a problem of measuring similarity between two sequence semantic graphs.

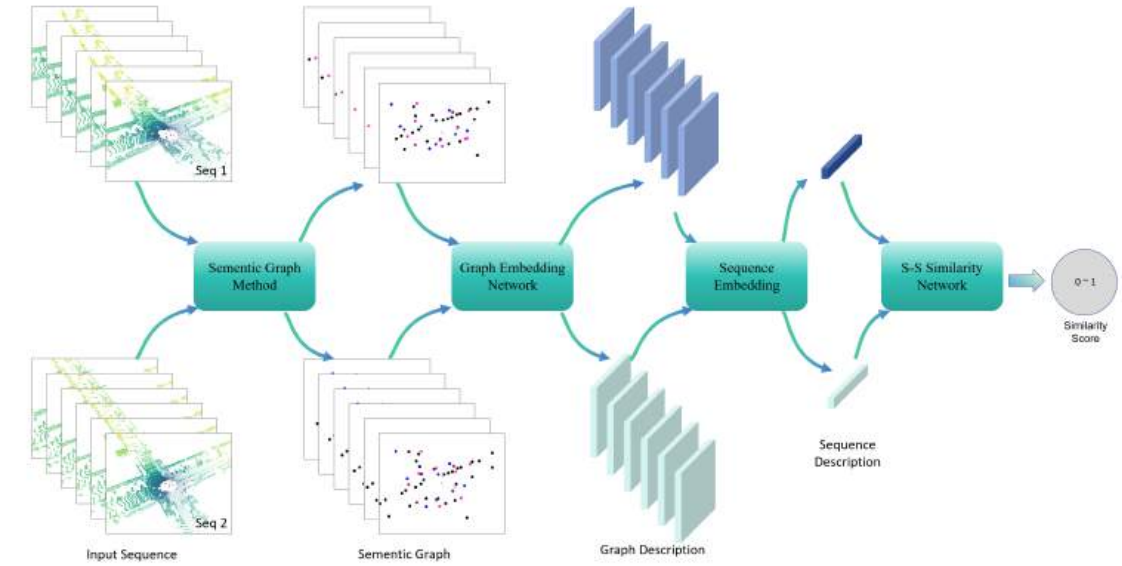


Fig. 1. The foundational structure of the proposed method. The algorithm receives a pair of consecutive multi-frame point clouds as input, and it produces an output value ranging from 0 to 1, indicating the degree of similarity between the two scenes.

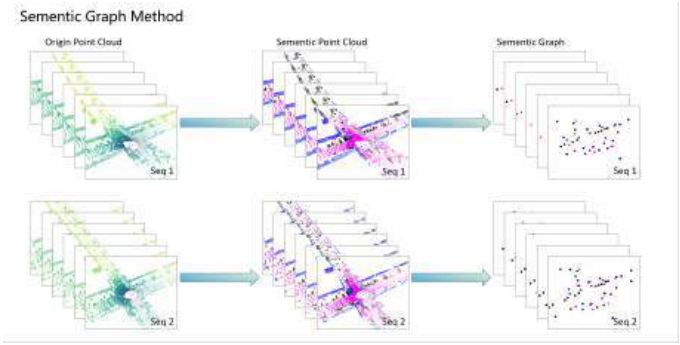


Fig. 2. Sementic Graph Method. Point clouds that are annotated with semantic labels are transformed into a topological graph, which preserves the centroids of instances as well as their interrelationships.

B. Sequential Semantic Graph Similarity Network

In the context of closed-loop detection, the implementation of graph matching algorithms necessitates that these algorithms exhibit representation invariance. This is essential as the computed similarity score must be unaffected by the sequential arrangement of the nodes. Furthermore, the algorithms should be rotation-invariant because reverse closed-loop detection frequently occurs in practical applications. We propose graph similarity networks inspired by SimGNN [20] to perform closed-loop detection of graph matching.

1) Node Embedding Network: Node embedding: The Dynamic Graph Convolutional Neural Network(DGCNN) [21] is effective in the learning of features from point clouds. We use the EdgeConv introduced in DGCNN to capture the local geometric information, while maintaining the permutation invariance.

In the EdgeConv layer, we identify the set of kNN for each node V_i in the feature space, and aggregate the features within each set. Each node's feature f_i is encoded with centroid information and is informed by semantic labels l_i. Each edge signifies the relationship between a node f_i and its k-nearest neighbors $f_j^m, m = 1, 2, ..., k$ in the feature space, with the edge function being defined as follows:

$$h_\Theta(f_i, f_j) = \bar{h}_\Theta\left(f_i, f_i - f_j^m\right), \tag{1}$$

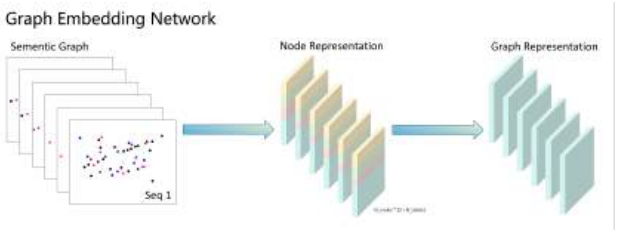


Fig. 3. Graph Embedding Network. This approach incorporates node embedding and graph embedding, with the former emphasizing local contextual information and the latter prioritizing global contextual understanding.

2) Graph Embedding Network: As shown in Fig. 3, Graph embeddings are typically generated using either weighted or unweighted averages of node embeddings. Inspired by SimGNN [20], our objective is to estimate a learnable weight matrix for each node through via an attention module. This approach enables the neural network to discern which nodes should get higher attention and are more representative of the overall graph. By integrating the embeddings of all nodes, we aim to derive a comprehensive feature of the overall graph

context.

$$c = tanh\left(\left(\frac{1}{N}\sum_{i=1}^{N} u_i\right) W\right), \tag{2}$$

The similarity of each node to the global feature is determined through the inner product method, which serves as the foundation for assessing the correlation of each node. Each node is then multiplied by its respective correlations and finally obtain the graph embedding, namely Graph Embedding.

$$\begin{aligned} \mathrm{e} &= \sum_{i=1}^{N} sigmoid\left(a_i\right) u_i \\ &= \sum_{i=1}^{N} sigmoid\left(u_i tanh\left(\left(\frac{1}{N}\sum_{m=1}^{N} u_m\right) W\right)^{T}\right) u_i, \end{aligned} \tag{3}$$

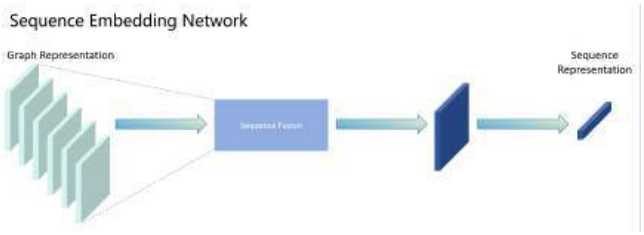


Fig. 4. Sequence Embedding Network. Sequential graph embeddings are integrated and fully connected to form a sequence embedding.

3) Sequence Embedding Network: As shown in Fig. 4, We designed modules that fuse multi-frame embeddings into a single sequence embedding. The output of this module is a sequence embedding of predetermined dimensions, which contains information about the point clouds scene for each individual frame and also the contextual information regarding the multi-frame point cloud scenes.

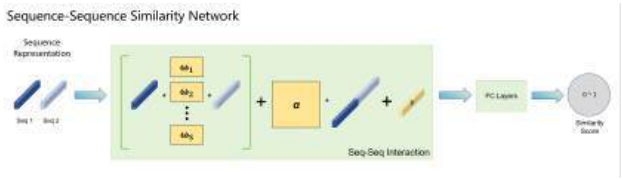


Fig. 5. Sequence-Sequence Similarity Network. A pair of sequence embeddings is transformed into a scalar score that ranges from 0 to 1 through the application of a learnable network.

4) Sequence-Sequence Similarity Network: As shown in Fig. 5, The network designed to evaluate the similarity between the two sequence embeddings is divided into two components. The first component computes an eigenvector to encapsulates the relationship between the two sequence embeddings.

$$g\left(e_1, e_2\right) = ReLU\left(e_1^T \omega^{[1:S]} e_2 + \alpha \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b\right), \tag{4}$$

In the subsequent component, we apply a series of fully connected layers to systematically decrease the dimension of the similarity vector and finally obtain a scalar value within the range of [0,1]. The problem is reduced to a binary classification task, the ground truth is 0 or 1, where 0 signifies that two sequences are entirely dissimilar, and 1 indicates that the sequences are identical. We train the model with a binary cross-entropy loss function.

IV. EXPERIMENT

A. Datasets and experimental details

We evaluated the proposed method on the KITTI odometry dataset [8], which contains 11 sequences (from 00 to 10) obtained from a 64-ring LIDAR(Velodyne HDL-64E). In our experiment, a positive pair was defined as two point cloud scenes exhibiting a Euclidean distance of less than 3 meters, while a negative pair was characterized by a distance exceeding 20 meters. For the evaluation process, positive pairs with timestamps exceeding 30 seconds were classified as true closed loops. Adjacent scenarios that represent easy positive pairs, were excluded from evaluation to provide a more accurate reflection of the algorithm's performance. The sequences identified as containing closed loops included 00, 02, 05, 06, 07, and 08, with sequence 08 exhibiting a reverse closed loop, in contrast to the others, which maintained the same orientation.

The SemanticKITTI dataset [18] provides semantic annotations for the point cloud data derived from the KITTI odometry dataset with 28 categories, and we mapped them into 12 categories. On the KITTI odometry dataset, the number of nodes exhibited variability, ranging from 10 to 70. In the node embedding section, we set k = 10 for the kNN algorithm and fill in the zero-embedded false nodes to obtain a fixed number of nodes. We used one sequence as the test set and the other sequences as the training set. All experiments were trained with PyTorch [22] and Adam optimizer [26] with a learning rate of 0.001. There are a large number of negative pairs, so we keep all positive pairs and randomly sample some negative pairs of the proportion, making the proportional relationship of the positive and negative pairs reasonable.

B. Performance of the closed-loop detection

To evaluate our sequence semantic graph representation and multi-graph similarity network, we conducted a comparative analysis utilizing the SematicKITTI labels. This evaluation was performed against several existing methodologies, including M2DP [1], LiDAR-iris [27], OverlapNet [28], Scan Context3 [16], LOCUS [19], DISCO [29], PointNetVLAD [2], OverlapNet [28] and SG-PR [25].

Quantitative results: We employed the precision-recall(P-R) curves to compute the maximum value of the F1 score, which serves as a metric for assessing the various P-R curves. The F1 score is defined as follows:

$$F_1 = 2 \times \frac{P \times R}{P + R}, \tag{5}$$

As illustrated in Table. I, our average maximum F1 score outperformed other existing methods, demonstrating a competitive overall performance. Notably, in the context of the

TABLE I
THE F1 MAX SCORE ON THE KITTI DATASET.

Methods	00	02	05	06	07	08	Mean
M2DP	0.836	0.781	0.772	0.896	0.861	0.169	0.719
LiDAR-iris	0.863	0.860	0.846	0.971	0.782	0.696	0.836
PointNetVLAD	0.882	0.791	0.734	0.953	0.747	0.129	0.709
OverlapNet	0.948	0.898	0.960	0.978	0.942	0.607	0.889
Scan Context3	0.937	0.858	0.955	0.987	0.922	0.811	0.914
SGPR	0.969	0.891	0.905	0.971	0.927	0.900	0.934
LOCUS	0.957	0.745	**0.968**	0.948	0.921	0.903	0.907
DISCO	0.964	0.892	0.964	0.990	0.897	0.900	0.907
ours-seq3	0.978	**0.924**	0.892	**0.993**	0.932	0.924	0.940
ours-seq5	**0.981**	0.911	0.907	0.991	0.930	**0.941**	0.944
ours-seq7	**0.981**	0.907	0.907	**0.993**	**0.967**	0.929	**0.947**

challenging sequence characterized by the reverse closed loop 08, there is a severe decline in performance for M2DP, LiDAR-iris, Scan Context3, and PointNetVLAD. These global descriptor-based methods exhibit limitations in accommodating viewpoint variations. In contrast, our approach is invariant to rotation, enabling it to yield consistent results. Furthermore, our method effectively aggregates information from consecutive multiple frames, thereby enhancing the cognitive understanding of the scene. This capability allows us to achieve superior closed-loop detection performance across both the overall dataset and the reverse closed-loop dataset.

V. CONCLUSION

In this study, we propose a 3D point clouds closed-loop detection method based on sequence semantic graph. This approach utilizes semantic information and continuous attention mechanisms more comprehensively for the purpose of closed-loop detection. In contrast to existing methods focusing on extracting local, global and statistical features, our method emphasizes the significance of semantic-level information, which offers distinct advantages in the context of environmental changes. Furthermore, the scene graphs generated from successive multiple frames align more closely with human perceptual processes. A thorough evaluation of our approach confirms its efficacy, particularly in the realm of reverse closed-loop detection, where our model demonstrates commendable performance.

REFERENCES

[1] He, Li, Xiaolong Wang, and Hong Zhang. "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection." 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016.

[2] Uy, Mikaela Angelina, and Gim Hee Lee. "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[3] Liu, Zhe, et al. "Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[4] Liu, Zhe, et al. "Seqlpd: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles." 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019.

[5] Dubé, Renaud, et al. "Segmatch: Segment based place recognition in 3d point clouds." 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017.

[6] Dubé, Renaud, et al. "SegMap: 3D segment mapping using data-driven descriptors." arXiv preprint arXiv:1804.09557 (2018).

[7] Dube, Renaud, et al. "SegMap: Segment-based mapping and localization using data-driven descriptors." The International Journal of Robotics Research 39.2-3 (2020): 339-355.

[8] Geiger, Andreas, et al. "Vision meets robotics: The kitti dataset." The International Journal of Robotics Research 32.11 (2013): 1231-1237.

[9] Lowry, Stephanie, et al. "Visual place recognition: A survey." ieee transactions on robotics 32.1 (2015): 1-19.

[10] Milford, Michael J., and Gordon F. Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights." 2012 IEEE international conference on robotics and automation. IEEE, 2012.

[11] Siam, Sayem Mohammad, and Hong Zhang. "Fast-SeqSLAM: A fast appearance based place recognition algorithm." 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.

[12] Facil, Jose M., et al. "Condition-invariant multi-view place recognition." arXiv preprint arXiv:1902.09516 (2019).

[13] Garg, Sourav, and Michael Milford. "Seqnet: Learning descriptors for sequence-based hierarchical place recognition." IEEE Robotics and Automation Letters 6.3 (2021): 4305-4312.

[14] Mereu, Riccardo, et al. "Learning sequential descriptors for sequence-based visual place recognition." IEEE Robotics and Automation Letters 7.4 (2022): 10383-10390.

[15] Lu, Feng, et al. "Visual sequence place recognition with improved dynamic time warping." 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2019.

[16] Kim, Giseop, and Ayoung Kim. "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

[17] Milioto, Andres, et al. "Rangenet++: Fast and accurate lidar semantic segmentation." 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2019.

[18] Behley, Jens, et al. "Semantickitti: A dataset for semantic scene understanding of lidar sequences." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[19] Yin, Huan, et al. "Locnet: Global localization in 3d point clouds for mobile vehicles." 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018.

[20] Bai, Yunsheng, et al. "Simgnn: A neural network approach to fast graph similarity computation." Proceedings of the twelfth ACM international conference on web search and data mining. 2019.

[21] Wang, Yue, et al. "Dynamic graph cnn for learning on point clouds." ACM Transactions on Graphics (tog) 38.5 (2019): 1-12.

[22] Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." Advances in neural information processing systems 32 (2019).

[23] Naseer, Tayyab, et al. "Robust visual robot localization across seasons using network flows." Proceedings of the AAAI conference on artificial intelligence. Vol. 28. No. 1. 2014.

[24] Vysotska, Olga, and Cyrill Stachniss. "Effective visual place recognition using multi-sequence maps." IEEE Robotics and Automation Letters 4.2 (2019): 1730-1736.

[25] Kong, Xin, et al. "Semantic graph based place recognition for 3d point clouds." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

[26] Diederik, P. Kingma. "Adam: A method for stochastic optimization." (No Title) (2014).

[27] Wang, Ying, et al. "Lidar iris for loop-closure detection." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

[28] Chen, Xieyuanli, et al. "OverlapNet: Loop closing for LiDAR-based SLAM." arXiv preprint arXiv:2105.11344 (2021).

[29] Xu, Xuecheng, et al. "Disco: Differentiable scan context with orientation." IEEE Robotics and Automation Letters 6.2 (2021): 2791-2798.

Hierarchical crossover-based NSGA-III for dynamic flexible job shop scheduling problem

Zeyu Feng
College of Computer Science
Beijing Information Science
and Technology University
Beijing, China
fengzeyu@bistu.edu.cn

Xu Liang
College of Computer Science
Beijing Information Science
and Technology University
Beijing, China
liangxu@bistu.edu.cn

Zhiyuan Zou*
College of Computer Science
Beijing Information Science
and Technology University
Beijing, China
zyzou@bistu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899739

***Abstract*—The flexible job shop scheduling problem (FJSP) is a common production scheduling issue in manufacturing, aiming to optimize multiple objectives during the production process while satisfying production constraints. However, in actual production environments, machine breakdowns are frequent dynamic factors that can cause job interruptions and production delays, thereby affecting the effectiveness of scheduling optimizations. To address this issue, this paper proposes a hierarchical crossover-based NSGA-III (HCNSGA-III). Firstly, we design a hierarchical crossover strategy that stratifies the Pareto front, applying different crossover strategies for different levels. Secondly, we design a dynamic reference point update method that dynamically adjusts the reference points based on the current population's solution set, enhancing adaptability to the problem. Finally, comparative analysis with other algorithms demonstrates that the proposed method exhibits good performance in solving dynamic scheduling problems caused by machine breakdowns.**

***Index Terms*—Flexible job shop scheduling, Machine breakdown, multiple objectives, NSGA-III**

I. Introduction

In the context of Industry 4.0, the manufacturing industry continues to develop rapidly, and flexible job shops have become an important direction for the transformation and upgrading of manufacturing. The FJSP is a typical combinatorial optimization problem that involves multiple jobs and machines, with decision variables including job allocation, process path selection, and scheduling sequence. In actual production, machine breakdowns are inevitable factors that can lead to downtime, disruption of job sequences, and delayed deliveries [1]. For this type of dynamic flexible job shop scheduling problem (DFJSP), traditional flexible job shop scheduling methods often assume that machine breakdowns do not occur, thus ignoring the impact of machine breakdowns [2]. However, in practical applications, how to effectively deal with machine breakdowns and optimize production scheduling has become a critical issue for improving production efficiency and reducing costs.

Previous studies have mostly focused on production efficiency in shops, setting makespan as the scheduling objective [3]. Additionally, to improve on-time delivery rates and optimize production costs, we set total tardiness and total earliness and tardiness penalties as scheduling objectives, establishing a dynamic scheduling model with three objectives. When solving such multi-objective problems, metaheuristic algorithms are typically employed, such as genetic algorithm, particle swarm optimization, etc [4] [5] [6]. Zhiqiang Tian et al. [7] developed a multi-objective optimization model for an energy-efficient α-shop and proposed a dual-population differential artificial bee colony algorithm to solve it. Kaikai Zhu et al. [8] proposed an enhanced memetic algorithm, incorporating a new five-layer encoding and initialization technique.

However, when dealing with multidimensional problems involving these three objectives, the increase in search space makes it more challenging to maintain solution diversity, easily leading to uneven distribution of solutions in the objective space. To address this, NSGA-III is an excellent method [9] [10], and we propose a HCNSGA-III for solving the problem. This method guides the search direction of solutions by defining a set of pre-determined reference points, which helps to maintain solution diversity and distribution in high-dimensional objective spaces. These reference points are generally uniformly distributed over the unit hypercube to ensure that the algorithm can effectively explore the entire objective space, and they are updated dynamically to guide the algorithm towards optimal solutions. Moreover, in the crossover operation of genetic algorithms, if individuals from different Pareto ranks adopt the same crossover strategy, it may result in the loss of superior information from higher-rank individuals or prevent lower-rank individuals from effectively passing on evolutionary information. For this reason, we introduce a hierarchical crossover strategy, where different crossover strategies are applied to individuals at different Pareto ranks, thereby making better use of the excellent information from higher-rank individuals.

II. PROBLEM DESCRIPTION

A. Problem Statement

FJSP introduces flexibility in machine selection on the basis of the traditional job shop, where different jobs consist of multiple operations, and each operation can be processed on any one of several available machines. This flexibility enhances the adaptability and resource utilization of the scheduling plan.

Corresponding author: Zhiyuan Zou, zyzou@bistu.edu.cn.

On this basis, machine breakdowns is introduced as dynamic events, leading to certain machines being unavailable for a period of time, which further increases the complexity of the problem. The relevant constraints for DFJSP are as follows:

(1) **Operation assignment constraint**: Each operation is assigned to only one machine.

(2) **Operation sequence constraint**: For consecutive operations within the same job, the start time of the next operation must be greater than or equal to the completion time of the current operation.

(3) **Processing time constraint**: The completion time of an operation is equal to its start time plus the processing time.

(4) **Machine breakdown constraint**: Operations cannot be scheduled on a machine during its breakdown period.

(5) **Non-overlapping constraint**: The processing times of different operations on the same machine do not overlap.

The symbols used in this article are shown in TABLE I.

TABLE I: Symbols used in this article.

symbols	description
J	Set of jobs, $j \in J$ represents a job
O_j	Set of operations for job j, $o \in O_j$ represents an operation
M	Set of machines, $m \in M$ represents a machine
$S_{o,m}$	Start time of operation o on machine m
$C_{o,m}$	Completion time of operation o on machine m
$P_{o,m}$	Processing time of operation o on machine m
B_m	Breakdown start time of machine m
R_m	Recovery time of machine m
$\delta_{o,m}$	Binary variable, $\delta_{o,m} = 1$ if operation o is assigned to machine m, otherwise 0
T_j	Tardiness of job j
E_j	Earliness of job j

B. Mathematical Model

In this paper, the three objectives of flexible job shop dynamic scheduling are: f_1 to minimize the makespan, f_2 to minimize total tardiness, and f_3 to minimize the total earliness and tardiness penalty.

$$f_1 = \min C_{\max} = \min \max_{j \in J} \max_{o \in O_j} C_{o,m} \tag{1}$$

$$f_2 = \min \sum_{j \in J} T_j = \sum_{j \in J} \max(0, C_{j,\text{last}} - d_j) \tag{2}$$

$$f_3 = \min \sum_{j \in J} (w_j^{\text{early}} \cdot E_j + w_j^{\text{late}} \cdot T_j) \tag{3}$$

where Equation (1) indicates the minimization of the makespan, where $C_{\max}$ represents the maximum makespan. Equation (2) indicates minimization of the total tardiness, where d_j is the due date of job j, and $C_{j,\text{last}}$ is the completion time of the last operation of job j. Equation (3) indicates minimization of the total earliness and tardiness penalty, where w_j^{early} and w_j^{late} represent the earliness and tardiness penalty weights for job j, respectively.

The constraints are as follows:

$$\sum_{m \in M} \delta_{o,m} = 1, \quad \forall o \in O_j, j \in J \tag{4}$$

$$S_{o+1,m'} \geq C_{o,m}, \quad \forall o, o+1 \in O_j, j \in J, m, m' \in M \tag{5}$$

$$C_{o,m} = S_{o,m} + P_{o,m} \cdot \delta_{o,m}, \quad \forall o \in O_j, j \in J, m \in M \tag{6}$$

$$S_{o,m} \geq R_m \quad \text{or} \quad C_{o,m} \leq B_m, \quad \forall o \in O_j, j \in J, m \in M \tag{7}$$

$$\begin{gathered} S_{o',m} \geq C_{o,m} \cdot \delta_{o,m} + C_{o',m} \cdot (1 - \delta_{o,m}), \\ \forall o, o' \in O_j, j \in J, m \in M, o \neq o' \end{gathered} \tag{8}$$

where Equation (4) indicates operation assignment constraint. Equation (5) indicates operation sequence constraint. Equation (6) indicates processing time constraint. Equation (7) indicates machine breakdown constraint. Equation (8) indicates Non-overlapping constraint.

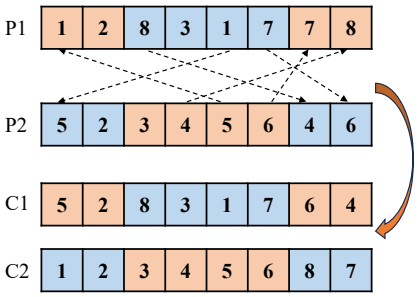
III. ALGORITHM DESIGN

To solve the DFJSP, we propose the hierarchical crossover-based NSGA-III algorithm. Below is a detailed description.

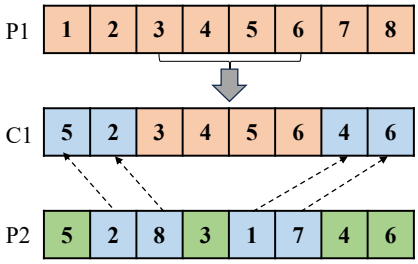
A. Hierarchical Crossover Strategy

In multi-objective optimization problems, individuals in the population exhibit significant differences in solution quality and diversity after non-dominated sorting. Using a single crossover strategy may fail to fully exploit the characteristics of high-quality individuals, leading to an imbalance in convergence speed and solution diversity. Therefore, we design a hierarchical crossover strategy, setting solutions on the first Pareto front as high-level and those on subsequent fronts as low-level, and adopting different crossover methods accordingly.

We divide the encoding into two parts: operation sequencing and machine selection. Operation sequencing is a permutation problem, where each individual represents a permutation, and during crossover, it is necessary to ensure that the exchanged solutions remain valid permutations. Machine selection, on the other hand, often involves efficient resource allocation, and during crossover, it is essential not only to consider the assignment of tasks but also to ensure that there are no conflicts in machine usage. For this purpose, in the operation sequencing part, we use partially mapped crossover for high-level individuals and order crossover for low-level individuals, as shown in Fig. 1. Partially mapped crossover randomly selects two crossover points to define the crossover region, establishes a mapping relationship for each chromosome, and then applies this mapping relationship to duplicate genes outside the crossover region. Order crossover randomly selects starting and ending positions, exchanges the genes between the parents within this region, and then fills in the missing genes

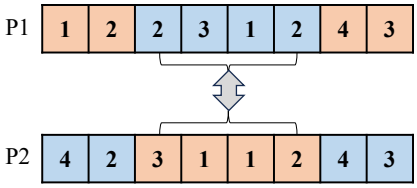


(a) Partially mapped crossover

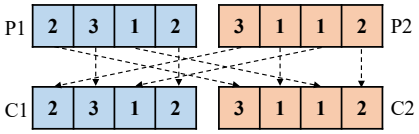


(b) Order crossover

Fig. 1: Operation sequence crossover.



(a) Two-point crossover



(b) Uniform crossover

Fig. 2: Machine selection crossover.

in the order they appear in the other parent. In the machine selection part, we employ two-point crossover for high-level individuals and uniform crossover for low-level individuals, as illustrated in Fig. 2. Two-point crossover sets two crossover points randomly within the individual chromosome and then exchanges the genetic material between these two points. Uniform crossover randomly selects genes from one of the parents at each gene position of the offspring, thereby generating the next generation individual.

B. Dynamic Reference Point Update

A reference point is a mechanism used to guide optimization algorithms in exploring the solution space and maintaining solution diversity. Reference point updating is intended to adapt the reference points to the evolution of the population, ensuring uniform distribution in the solution space during the optimization process and effectively guiding the algorithm towards optimal solutions. To avoid keeping the reference points fixed throughout the optimization process and to enhance adaptability to the problem, we have designed a dynamic reference point update method.

We dynamically update the reference points based on the current solution set of the population. The positions of the reference points are updated according to the minimum and maximum values in the current population, adjusting their positions so that they can cover the distribution of the current population in the objective space.

Assume the current population is $\Pi = \{\pi_1, \pi_2, \ldots, \pi_{\mathrm{P}}\}$, where each solution's objective value is $\pi_p = (\pi_{p1}, \pi_{p2}, \pi_{p3})$. For each objective dimension d, Equation (9) calculates the minimum value l'_d and maximum value u'_d of the current population.

$$l'_d = \min_{\pi_p \in \Pi} \pi_{pd}, \quad u'_d = \max_{\pi_p \in \Pi} \pi_{pd} \quad d = 1, 2, 3 \tag{9}$$

Based on the minimum and maximum values, update the position of the reference points as shown in Equation (10).

$$r'_{pd} = l'_d + \left(\frac{p}{N_r}\right)(u'_d - l'_d), p = 1, 2, \ldots, N_r \text{ and } d = 1, 2, 3 \tag{10}$$

where r'_{pd} is the position of the updated pth reference point in the dth objective dimension. In other words, it represents the specific position of the reference point in the objective space. N_r is the number of reference points, indicating how many reference points are divided in the objective space. The core of this formula is to generate reference points through linear interpolation, ensuring a uniform distribution of reference points in the objective space and enabling them to dynamically adapt to changes in the optimal values of the population's solutions. By doing so, the reference points can guide the algorithm to maintain solution diversity in the objective space and assist the optimization algorithm in gradually converging towards the Pareto front.

IV. Experimental Analysis

A. Parameter setting and performance metrics

The data used in the experiments is from the classic and widely used dataset Brandimarte_Data [11], which has been augmented with machine breakdowns, breakdown times, and repair times after pre-scheduling to simulate machine breakdowns. The parameter settings are as follows: population size

= 100, number of generations = 100, crossover probability = 0.7, mutation probability = 0.01, and other parameters are used according to the original text.

The experiments use two metrics, inverted generational distance (IGD) and hypervolume (HV), to evaluate algorithm performance. IGD measures the average distance from each point in the reference set to the nearest solution, with a lower value indicating better convergence and distribution. HV quantifies the covered area in the objective space by non-dominated solutions relative to a reference point, with a higher value indicating better performance.

As the true Pareto front is unknown, an approximate Pareto front is constructed in the experiments by applying non-dominated sorting to the combined non-dominated solution sets from different algorithms [12].

B. Algorithm Comparison and Analysis

To verify the performance of the algorithm, this paper selects the following three algorithms for comparison: NSGA-II [13], HGA-VNS [14], and MOEA/D [15]. Each algorithm is tested on 10 instances from the Brandimarte_Data dataset, running 20 times each, and the mean IGD and HV were calculated, resulting in TABLE II and TABLE III, with the best performing metrics highlighted in bold.

TABLE II: Average values of IGD on Brandimarte_Data Dataset.

Instance	IGD			
	HCNSGA-III	NSGA-II	MOEA/D	HGA-VNS
Mk01	**0.0810**	0.1831	0.1240	0.1162
Mk02	**0.0976**	0.1901	0.1260	0.1083
Mk03	0.0538	0.0609	0.0894	**0.0466**
Mk04	**0.1252**	0.1227	0.1364	0.1400
Mk05	**0.1427**	0.2136	0.1525	0.1752
Mk06	**0.0907**	0.1877	0.1836	0.1556
Mk07	**0.0285**	0.1010	0.0568	0.0412
Mk08	**0.0349**	0.0844	0.0483	0.0712
Mk09	**0.0386**	0.0851	0.0362	0.0529
Mk10	**0.0919**	0.1694	0.1394	0.1172
Mean	**0.0785**	0.1398	0.1092	0.1024

TABLE III: Average values of HV on Brandimarte_Data Dataset.

Instance	HV			
	HCNSGA-III	NSGA-II	MOEA/D	HGA-VNS
Mk01	**0.5554**	0.4762	0.4121	0.5019
Mk02	0.7037	0.7140	0.6710	**0.7743**
Mk03	**1.0990**	0.7149	1.0724	0.8645
Mk04	**1.1795**	1.0589	1.0836	1.1630
Mk05	**0.9241**	0.4745	0.7843	0.8526
Mk06	**1.0695**	0.8765	0.9079	1.0121
Mk07	**1.0756**	0.5901	0.9308	0.9007
Mk08	**0.8947**	0.5350	0.5143	0.8156
Mk09	**1.1588**	0.4949	0.7604	1.0116
Mk10	**1.1305**	0.8193	0.9926	0.9244
Mean	**0.9791**	0.6754	0.8139	0.8811

From TABLE II and TABLE III, HCNSGA-III achieved better results in the majority of test instances, with mean IGD and HV values of 0.0785 and 0.9791, respectively, showing a relatively significant advantage over other algorithms. Among these, the IGD value was slightly inferior to HGA-VNS only in the Mk03 instance, and the HV value was slightly lower than HGA-VNS only in the Mk02 instance. The above experiments indicate that compared to other algorithms, the HCNSGA-III algorithm performs better when solving DFJSP.

From Fig. 3, each figure shows a view combining two out of three objective values, with smaller values indicating better results. HCNSGA-III consistently produces smaller non-dominated fronts across various test scales, signifying higher solution quality. Notably, on Mk10, HCNSGA-III demonstrates a superior, even distribution of the non-dominated front, suggesting greater diversity in solutions. This distribution helps avoid bias and facilitates more efficient acquisition of high-quality solutions under dynamic conditions.

Experiments on Mk10 using four algorithms show the convergence curves of three objectives over population iterations (Fig. 4). HCNSGA-III reaches convergence in about 40 iterations, faster than the others. It outperforms the other algorithms in makespan and total tardiness but performs slightly worse than MOEA/D in total earliness and tardiness penalty. Overall, HCNSGA-III excels in convergence speed and results, making it effective for dynamic scheduling problems.

V. Conclusion

To address the DFJSP caused by machine breakdowns, we propose the HCNSGA-III algorithm. It uses a hierarchical crossover strategy to maintain solution diversity by applying different crossover methods at various Pareto front levels. A dynamic reference point update mechanism is introduced, adjusting reference points based on the current population's min and max values, ensuring better coverage in the objective space. Experimental results show that HCNSGA-III outperforms other methods in IGD and HV metrics, with superior solution distribution and convergence, demonstrating its effectiveness in solving the DFJSP.

In future research, we will investigate more dynamic factors to address more complex production environments. Additionally, we will explore new algorithms to achieve better performance in solving scheduling problems.

Acknowledgment

This research is partially supported by the R&D Program of Beijing Municipal Education Commission (KM202411232003), Young Backbone Teacher Support Plan of Beijing Information Science & Technology University (YBT 202425) and Research Foundation of Beijing Information & Science Technology University (2023XJJ19).

References

[1] W. Ren, Y. Yan, Y. Hu, Y. Guan, Joint optimisation for dynamic flexible job-shop scheduling problem with transportation time and resource constraints, International Journal of Production Research 60 (2021) 1–22.

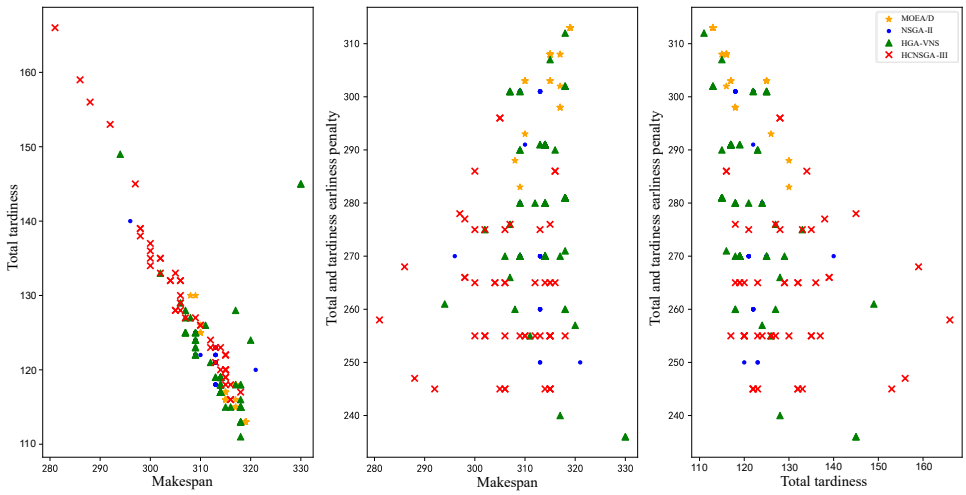


Fig. 3: Pareto front under Mk10.

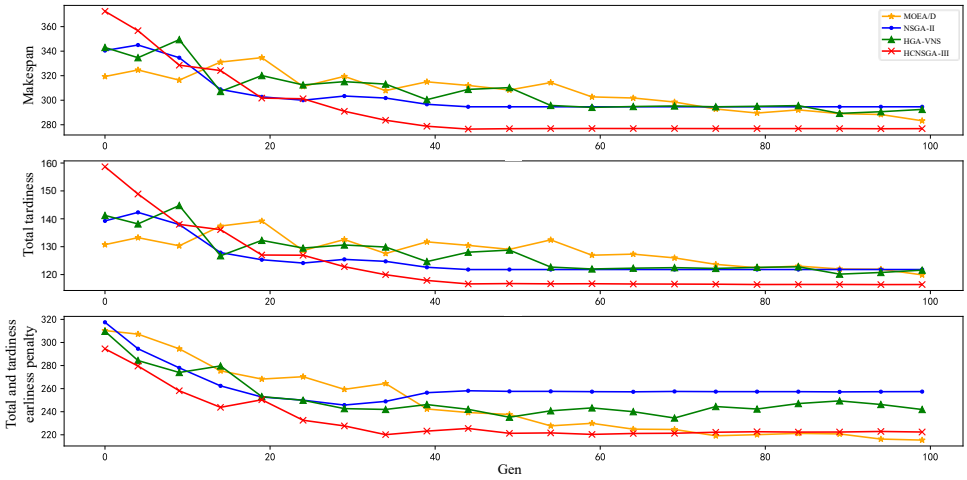


Fig. 4: Convergence curve under Mk10.

[2] Y. Zeiträg, J. R. Figueira, N. Horta, R. Neves, Surrogate-assisted automatic evolving of dispatching rules for multi-objective dynamic job shop scheduling using genetic programming, Expert Systems with Applications 209 (2022) 118194.

[3] K. Lei, P. Guo, Y. Wang, J. Zhang, X. Meng, L. Qian, Large-scale dynamic scheduling for flexible job-shop with random arrivals of new jobs by hierarchical reinforcement learning, IEEE Transactions on Industrial Informatics PP (2023) 1–12.

[4] D. Aribi, O. Belkahla Driss, H. EL HAOUZI, Multi-objective optimization of the dynamic and flexible job shop scheduling problem under workers fatigue constraints, 2023, pp. 301–308.

[5] L. Zhu, F. Zhang, X. Zhu, K. Chen, M. Zhang, Sample-aware surrogate-assisted genetic programming for scheduling heuristics learning in dynamic flexible job shop scheduling, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 384–392.

[6] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, Multitask multiobjective genetic programming for automated scheduling heuristic learning in dynamic flexible job-shop scheduling, IEEE Transactions on Cybernetics 53 (7) (2023) 4473–4486.

[7] Z. Tian, X. Jiang, W. Liu, Z. Li, Dynamic energy-efficient scheduling of multi-variety and small batch flexible job-shop: A case study for the aerospace industry, Computers & Industrial Engineering 178 (2023) 109111.

[8] K. Zhu, G. Gong, N. Peng, L. Zhang, D. Huang, Q. Luo, X. Li, Dynamic distributed flexible job-shop scheduling problem considering operation inspection, Expert Systems with Applications 224 (2023) 119840.

[9] M. Zhang, J.-S. Wang, Y. Liu, H.-M. Song, J.-N. Hou, Y.-C. Wang, M. Wang, Multi-objective optimization algorithm based on clustering guided binary equilibrium optimizer and nsga-iii to solve high-dimensional feature selection problem, Information Sciences 648 (2023) 119638.

[10] R. Gao, J. Tao, J. Zhang, L. Ma, M. Xu, Nsga-iii-sd based fuzzy energy management system optimization for lithium battery/supercapacitor hev, Applied Soft Computing 142 (2023) 110280.

[11] P. Brandimarte, Routing and scheduling in a flexible job shop by tabu search, Annals of Operations Research 41 (3) (1993) 157–183.

[12] K. Li, Q. Deng, L. Zhang, Q. Fan, G. Gong, S. Ding, An effective mcts-based algorithm for minimizing makespan in dynamic flexible job shop scheduling problem, Computers & Industrial Engineering 155 (2021) 107211.

[13] X. Li, Z. Zhang, W. Sun, Y. Liu, J. Tang, Parallel dynamic nsga-ii with multi-population search for rescheduling of seru production considering schedule changes under different dynamic events, Expert Systems with Applications 238 (2024) 121993.

[14] K. Sun, D. Zheng, H. Song, Z. Cheng, X. Lang, W. Yuan, J. Wang, Hybrid genetic algorithm with variable neighborhood search for flexible job shop scheduling problem in a machining system, Expert Systems with Applications 215 (2023) 119359.

[15] Q. Wang, Q. Gu, L. Chen, Y. Guo, N. Xiong, A moea/d with global and local cooperative optimization for complicated bi-objective optimization problems, Applied Soft Computing 137 (2023) 110162.

Doc-patch: An Unsupervised Approach for Documents Forgery Detection

1st Aboudramane DIARRA
Université Nazi BONI
Bobo Dioulasso, Burkina Faso
aboudramane.diarra@outlook.com

2nd Tegawendé F. BISSYANDE
Université Joseph KI-ZERBO
Centre d'Excellence Interdisciplinaire
en Intelligence Artificielle pour
le Développement (CITADEL)
Ouagadougou, Burkina Faso
tegawende.bissyande@citadel.bf

3rd Pasteur PODA
Université Nazi BONI
Bobo Dioulasso, Burkina Faso
pasteur.poda@u-naziboni.bf

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899704

***Abstract*—The exponential growth of digital documents has unfortunately led to a parallel rise in document forgery. Traditional authentication methods often struggle to detect "unseen-before" sophisticated falsification techniques. To address this challenge, we introduce Doc-patch, an unsupervised approach that effectively identifies anomalies in documents without requiring labeled training data. Doc-patch leverages advanced machine learning techniques to analyze document patches, focusing on small segments of text or images. By employing feature extraction, self-attention layers, and subtle hint capturing layers, the model can pinpoint local anomalies indicative of tampering. The FAISS K-Nearest Neighbors algorithm is then used to identify and localize these inconsistencies. Our experimental results demonstrate that Doc-patch achieves a high average precision score of 96.51% on a diverse dataset of documents. This superior performance underscores the effectiveness of our approach in detecting document forgery, surpassing traditional methods and providing a robust solution for authenticating digital documents in professional and administrative settings.**

***Index Terms*—document, forgery detection, patch, unsupervised learning, KNN**

I. Introduction

The increasing digitization of official and legal documents has made them more vulnerable to forgery. Malicious actors exploit advanced tools to manipulate documents, often leaving subtle traces that are difficult to detect using traditional methods [1], [10]. While supervised learning techniques have shown promise in forgery detection, they require large, labeled datasets, which are often unavailable or costly to obtain. This limitation has created a pressing need for unsupervised approaches that can effectively detect forgeries without prior knowledge of specific tampered cases. In response to this challenge, we propose Doc-patch, an unsupervised approach for document forgery detection.

Thus, we use PachCore anomaly detection as the basis for our research. PatchCore is an unsupervised anomaly detection method that leverages patch-level embeddings to identify anomalies in data. It uses feature embedding extraction, memory bank construction, nearest neighbor search, and anomaly scoring and localization to detect anomalies in data. For us, forgery detection is similar to the anomaly detection operation. More specifically, the similarity between fake and anomalous data is the mismatch with real data. This similarity leads us to take the anomaly detection method and enhance it to detect document forgery. Thus, by leveraging PatchCore Algorithm [3] and incorporating methods such as a Forgery Embedding Injector and an Adaptive Forgery Detector in the Memory Bank construction process, our method is designed to detect subtle and significant document forgeries [6]. Doc-patch analyzes the intrinsic features of the document's visual and textual content, identifying deviations that indicate potential tampering. Unlike existing systems that rely heavily on annotated training data, Doc-patch operates independently of labeled forgeries, making it more adaptable to real-world applications.

Document forgery detection has been extensively studied in recent years due to the increasing sophistication of digital manipulation techniques. Various methods have been proposed, ranging from traditional approaches, supervised or unsupervised learning approaches to more advanced deep learning techniques, aimed at detecting subtle alterations in both visual and textual elements of documents [2], [4]. Traditional approaches has relied on physical examination and forensic analysis. Experts assess factors such as paper quality, watermarks, and handwriting to authenticate documents. Techniques like infrared imaging and ink analysis further enhance this process, allowing for the identification of alterations [16]. However, these methods face limitations, especially with digital documents, where physical signs of forgery may not be visible. To address this, digital forensics has emerged, incorporating tools like Optical Character Recognition (OCR) and machine learning algorithms to detect anomalies in document structure [17].

Thus, A significant body of research has focused on supervised techniques, which rely on large, labeled datasets for training. Here many authors proposed a deep learning model based on convolutional neural networks (CNNs) to detect forgeries in scanned documents by learning pixel-level features of tampered regions [7], [11], [12], [13], [15]. Also, He et al.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

developed a supervised approach using a combination of text-based and visual cues to identify forged documents in government-issued certificates [9]. However, the reliance on labeled data poses a limitation in practical settings, where acquiring labeled forgery datasets is often unfeasible.

To overcome the limitations of supervised methods, unsupervised approaches have gained attention, focusing on detecting anomalies without the need for labeled examples. Ruff et al. introduced Deep One-Class Classification for anomaly detection, an approach that trains on normal data to identify outliers, which has been applied to fraud detection tasks [8]. Likewise, K. Roth et al. proposed PatchCore, an unsupervised method designed to detect anomalies in images by modeling normal data patterns and identifying deviations [3].

Our proposed approach, Doc-patch, builds on this prior work, particularly PatchCore and anomaly detection while introducing the novel concepts of the Forgery Embedding Injector and Adaptive Forgery Detector to enhance the detection of subtle and complex document manipulations. By utilizing these methods in an unsupervised framework, Doc-patch aims to address the gaps in current forgery detection systems that depend heavily on labeled training data. This paper discusses the architecture of Doc-patch, its training process, and the experimental results, demonstrating the approach's effectiveness in detecting forgeries across various documents.

II. Proposed methodology

A. Dataset

A substantial part of the dataset consists of over 1089 authentic documents, legally issued documents obtained from the National Agency for the Promotion of ICT, ensuring the representation of real-world variability in legitimate documents. The dataset is composed of a few forged documents including Blur, copy-paste, insertion, noise, and splicing.

1) Document format: In this work, we targeted PDF documents that have undergone modifications through falsification or image processing tools. In general, PDF documents or images that have undergone digital texture modifications.

2) Documents data representation: Representing data as patches is a common approach in computer vision tasks, including forgery detection in documents. For the Forgery detection, each document is represented as a collection of image patches, enabling fine-grained analysis and classification of localized features [3]. This representation enhances the ability to detect subtle alterations or forgeries within documents by focusing on smaller regions of interest. Patch representation helps in understanding the intent behind code modifications, which is essential for tasks like generating descriptions of patches, predicting their accuracy, and identifying the intentions behind the changes.

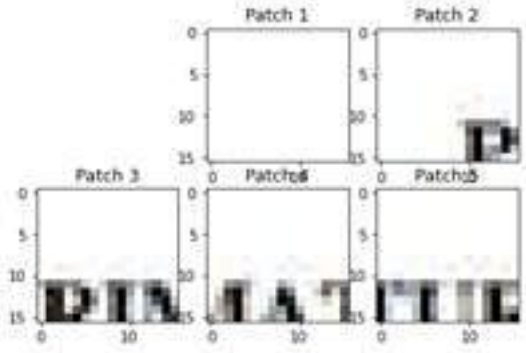


Fig. 1. Document representation as patch

Documents are divided into fixed-size rectangular patches, typically ranging from small (e.g., 9x9 pixels) as represented in figure 1, depending on our desired level of granularity and computational resources.

B. Use case motivation

Several kinds of official documents are particularly challenging to authenticate due to various factors, including their complexity, the potential for forgery, and the requirements for verification. Here, we think that some types of documents are easy to authenticate because of their identifiers such as passports or people identity cards. Thus, we choose a relevant use case of a type of document where the authentication is difficult on social networks or a professional environment. This difficulty is due to the lack of digital or physical means of authenticity verification, such as databases or document tracking registers.

*1) **Types of official's documents**:*

Educational documents: Educational credentials, such as diplomas and transcripts, can be difficult to authenticate because they often contain security features that are not easily verifiable by third parties. Additionally, the prevalence of counterfeit academic documents complicates the verification process. Credential evaluators must examine various safety features, such as watermarks and holograms, to determine authenticity, and even then, alterations can occur that may not be immediately apparent.

Government or administrative documents: Government documents can also present authentication challenges. They may lack standard bibliographic information, such as authorship or publication dates, making them difficult to cite and verify. This inconsistency can lead to complications in establishing their authenticity.

Social documents: Documents issued by religious institutions, such as baptismal or marriage certificates, can

be hard to authenticate due to their informal nature and the variability in how different faiths issue such documents. Many authorities do not recognize these documents without additional verification, which can complicate their acceptance for legal purposes.

Computer-generated records: Records produced by computer systems, especially if they lack original signatures or seals, face scrutiny regarding their authenticity [18]. The ease of alteration in digital formats raises concerns about their integrity, making it challenging to prove that they accurately reflect the original information.

*2) **Intuition on the choice of model**:*

Document falsification is a deliberate manipulation that deviates from the normal or authentic structure and content of the original document. Such deviations, whether in the form of tampered text, altered signatures, or manipulated images, create anomalies in the document's intrinsic visual and textual patterns. These anomalies, while often subtle, can be effectively detected by a model that is capable of learning and modeling the patterns of legitimate documents and identifying deviations from these patterns.

We hypothesize **that anomalies caused by document falsification can be detected using an unsupervised approach like Doc-Patch**, which is designed to identify out-of-distribution instances without relying on labeled forgery data. Given that falsified documents exhibit abnormal characteristics that deviate from the patterns of genuine documents, an anomaly detection model should be able to detect these forgeries by focusing on local inconsistencies in the document's structure, texture, or content.

By enhancing PatchCore with our proposed methods, including the Forgery Embedding Injector and the Adaptive Forgery Detector, we aim to improve the model's sensitivity to forgeries, particularly when such alterations are designed to be subtle or complex. Thus, we expect this approach to outperform traditional methods that rely on explicit labels, as it can detect forgeries even in previously unseen types of document manipulations.

C. Approach architecture

In computer vision, neural networks are very effective for classifying authentic and false images of all kinds. We have therefore proposed this approach for the classification of official documents. We have introduced modifications to the feature extraction method. We use a pre-trained WIDE ResNet50 algorithm as its native backbone. This pre-trained CNN efficiently extracts features from the image patches, creating a vector representation for each patch. These feature vectors capture the essential features of the normal image patches.

Then, we add to the WIDE ResNet50 backbone two algorithms for reinforcing and extracting forgery-related features. We therefore added an Adaptive Forgery Detector and a Forgery embedding Injector algorithm at the end of the backbone [6].

The Adaptive Forgery Detector (AFD) algorithm bridges the gap between the different domains (pre-trained knowledge and forgery-related knowledge) by learning adaptive knowledge in the self-attention layer. In other words, AFD takes a broader perspective. It aims to learn global and generalizable features associated with falsified images. These features can be subtle inconsistencies in lighting, texture, or overall image characteristics that are not restricted to specific areas. It has two functions:

- Adaptation of the self-attention layer: The AFD modifies the self-attention layer, a central element of the architecture responsible for learning the relationships between the different parts of an image;
- Learning global knowledge of forgery: By adapting the self-attention layer, the AFD prioritizes features generally indicative of forgeries in various manipulation techniques.

1) AFD algorithm description: The AFD algorithm uses an Adaptive Self Attention (ASA) method to allow the model to weigh the importance of different elements in an input sequence dynamically. The ASA method has a Multi-Head Self-Attention layer defined in torch.nn library. It enhances information processing by considering the context of each input, enabling effective learning of complex patterns and relationships in data, particularly in natural language processing tasks.

We define ASA with a **Multi-Head Self-Attention** that gives an input sequence $X \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the dimensionality of the input vectors, we define: **Query, Key, and Value matrices**

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$ are learned weight matrices for queries, keys, and values respectively, and d_k is the dimensionality of each head.

We have an attention scores: The attention scores for each head i are computed as:

$$\text{Attention}_i(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i$$

The final of the multi-head attention is given by concatenating the outputs of all heads and applying a linear transformation :

$$\text{MultiHead}(X) = \text{Concat}(\text{Attention}_1, \ldots, \text{Attention}_h)W_O$$

where $W_O \in \mathbb{R}^{hd_v \times d}$ is a weight matrix that projects the concatenated outputs back to dimension d.

We add Domain-Specific Attention that uses a linear transformation followed by a sigmoid activation function :

$$w_d = \sigma(DW_d)$$

where $D \in \mathbb{R}^{m \times d}$ represents domain-specific information and $W_d \in \mathbb{R}^{d \times d}$ is a learned weight matrix for domain attention.

The Final Output of the ASA method combines both attention mechanisms:

$$\text{output} = (\text{MultiHead}(X)) * w_d$$

Here, w_d is unsqueezed to match dimensions for element-wise multiplication with the multi-head attention output.

The Adaptive Forgery Detector (AFD) takes an input image $I \in \mathbb{R}^{H \times W \times 3}$ (height H, width W, and 3 color channels) and transform it into patch embeddings using a convolutional layer :

$$X_{\text{patch}} = \text{Conv2d}(I)$$

where the output shape is:

$$X_{\text{patch}} \in \mathbb{R}^{B \times d \times H' \times W'}$$

with B being the batch size, d being the embedding dimension, and $H' = \frac{H}{3}$, $W' = \frac{W}{3}$ due to the kernel size and stride.

Next, we flatten and permute the tensor:

$$X_{\text{flatten}} = X_{\text{patch}}.\text{flatten}(2).\text{permute}(2, 0, 1)$$

This results in:

$$X_{\text{flatten}} \in \mathbb{R}^{N \times B \times d}$$

where $N = H' \times W'$ is the number of patches.

The flattened input is passed through six transformer blocks. Each block applies adaptive self-attention as defined previously. For each block i:

$$X_i = \text{AdaptiveSelfAttention}(X_{i-1}, D)$$

where D represents domain-specific information.

After passing through all transformer blocks, we obtain:

$$X_{\text{transformer}} = X_6$$

The output from the final transformer block is averaged across patches to obtain a global representation:

$$X_{\text{global}} = \frac{1}{N} \sum_{j=1}^{N} X_{\text{transformer}}[j]$$

Finally, this representation is passed through a fully connected layer to produce class logits:

$$Y = X_{\text{fc}} = W_f X_{\text{global}} + b_f$$

where $W_f \in \mathbb{R}^{d \times C}$ is a weight matrix for classification and $b_f \in \mathbb{R}^C$ is the bias term, with C being the number of classes.

The final output can be summarized as:

$$Y = f(X_{\text{global}})$$

where $f(\cdot)$ represents the linear transformation applied by the fully connected layer.

2) FEI algorithm description:

The Forgery Embedding Injector (FEI) is designed to capture subtle hints of local falsification in images. It extracts discriminating information from local regions of the image. By combining the global features learned by ResNet with the local clues captured by FEI, this approach becomes more effective at detecting deepfakes.

The Forgery Embedding Injector class builds upon the Subtil Hint Seeker (SHS) method by adding an initial convolutional layer and chaining multiple instances of Subtil Hint Seeker. The SHS class is a neural network module designed for processing images. It consists of several layers that transform the input through the following steps :

- Three convolutional layers extract features from the input image, each followed by Batch Normalization and ReLU activation to enhance learning and stability;
- The output from the convolutional layers is flattened into a vector;
- Two fully connected (linear) layers further process the flattened output, reducing dimensionality;
- A dropout layer is applied for regularization, followed by a Sigmoid activation function to produce the final output, which represents probabilities for different classes.

Overall, the architecture is designed to effectively capture and classify subtle hints in images, making it suitable for tasks like forgery detection. The FEI class processes an input image $I \in \mathbb{R}^{H \times W \times 3}$ through several layers. The input image is transformed by the initial convolutional layer :

$$X_1 = \text{Conv2d}(I, W_1) + b_1$$

where: - $W_1 \in \mathbb{R}^{64 \times 3 \times 3}$ is the weight matrix, - $b_1 \in \mathbb{R}^{64}$ is the bias term.

The output after applying the convolution is :

$$X_1' = X_1 + b_1$$

with padding applied.

The output X_1' is passed through three instances of the SubtilHintSeeker class:

- First Layer:

$$X_2 = \text{SubtilHintSeeker}(X_1', 64, 128)$$

- Second Layer:

$$X_3 = \text{SubtilHintSeeker}(X_2, 128, 256)$$

- Third Layer:

$$Y = \text{SubtilHintSeeker}(X_3, 256, C)$$

where C is the number of output channels (e.g., forgery detection classes).

The overall forward pass can be represented as :

$$Y = f(I) = \text{SHS}(\text{SHS}(\text{SHS}(\text{Conv2d}(I, W_1) + b_1)))$$

where $f(I)$ represents the final output after processing through all layers.

3) Model description: Here we present the global model architecture as follows in image 2:

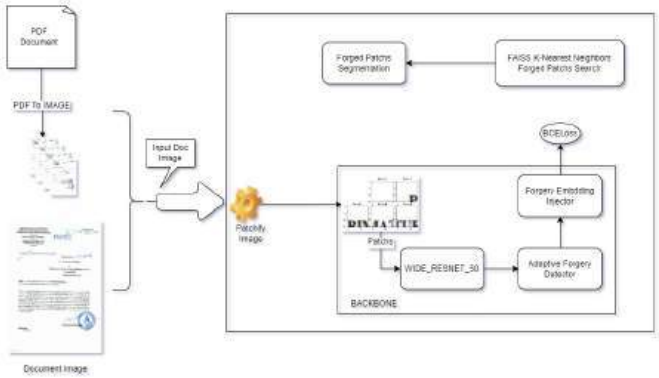


Fig. 2. Model architecture

FAISS K-Nearest Neighbors algorithms: Our model is a case of binary classification based on the number of classes we want to predict as depicted in the architecture. Hence, we have only two classes, fake and real documents. The Nearest Neighbour Search operation used here is FAISS K-Nearest Neighbors algorithms [14].
Specifically, KNN is used to compare the feature embeddings of a test document with those in the Memory Bank constructed from genuine documents. Here's how KNN contributes to the forgery detection task :

- **Similarity Comparison:** KNN measures the similarity between the test document's feature embedding and the embeddings of known authentic documents. By finding the "nearest neighbors" within the Memory Bank, KNN identifies documents that are most similar to the test sample.
- **Anomaly Scoring:** The distance between the test document's embedding and its nearest neighbors is used to calculate an anomaly score. If this score exceeds a certain threshold, the document is flagged as a potential forgery. This helps detect subtle variations in feature patterns that might indicate tampering.
- **Adaptability to Unsupervised Context:** KNN operates without the need for labeled forgeries, aligning well with the unsupervised approach of Doc-patch. It relies solely on the Memory Bank of authentic documents to determine if a document's features are unusual, making it adaptable to a range of document types without extensive labeled data.

Patches segmentation task: In this work, the forged patches segmentation task focuses on identifying specific regions, or "patches," within a document that have been tampered with. This task is essential for providing a more detailed analysis of forgeries by pinpointing the exact areas affected rather than simply labeling an entire document as genuine or forged. There is a **Forged patches segmentation** operation that helps to locate specific regions in a document that show signs of tampering, allowing the model to distinguish between authentic and altered areas within the same document. This localized detection is crucial for understanding and verifying exactly which parts have been manipulated.

III. Experimental Results and Discussion

Experimental results provide information about the performance of machine learning algorithms on datasets. These results are measured by performance metrics such as precision, recall and accuracy, etc. In this article, we used authentic official documents from the administration of the Agence National pour la Promotion des TIC (ANPTIC) in Burkina Faso. These documents are memos, information notes, press releases, etc.

A. Experimental results

The precision measures the proportion of correct positive predictions. A high precision indicates that the model is mostly identifying true positives and avoiding false positives (incorrectly classifying negative examples as positive). The recall measures the proportion of actual positive cases that were correctly identified by the model. A high recall indicates that the model is catching most of the actual positive examples and not missing them.

In the first experiment, the proposed method is analyzed as a classifier algorithm for two classes: original and fake documents for the testing phase. Based on our experimental results, we conclude that the proposed CNN-based solution has significantly higher accuracy for zero-phase training. The figure 3 shows the results of the first experiment and the comparison with some existing solutions, which are considered to be one of the best solutions for document forgery detection

problems. A first iteration of the algorithm on our test dataset of documents we obtain :

Test metric	DataLoader 0
Average Precision score	0.9651184394238159
Img AUC ROC Curve score	0.9369488536155203

Fig. 3. Test metrics

The AUROC metric was used to measure the algorithm's ability to distinguish between authentic and forged documents. An AUROC value close to 1 indicates excellent discrimination capability, while a value of 0.5 indicates no better performance than random guessing.

For the case of authentic documents the overall 92% of the documents are recognized as authentic with acceptable predictions in figure 4. For the forged document, on the other hand, the system is highly effective, identifying the document as such with a high degree of confidence and a score of 62% in figure 6. This indicates that the model is effective in capturing the structural inconsistencies introduced by falsification.

Fig. 4. Genuine document

Precision and recall metrics were calculated to evaluate the algorithm's accuracy and completeness in detecting forged documents. Precision indicates the proportion of true positive detections among all positive detections made by the algorithm, while recall measures the proportion of true positive detections among all actual forged documents.

The average precision is 96.51% and the image ROC Curve is 93.69% as depicted in figure 3 and 5, demonstrating that the majority of the detections made by the algorithm were accurate, with few false positives. This high precision value indicates that when the algorithm detects a document as forged, it is very likely to be correct.

In this example, the predicted heatmap shows the forged area on the document.

B. Discussion

1) Comparison with the GANomaly algorithm: For comparison, we trained the GANomaly algorithm on the same dataset, which yielded an AUC of 85% in figure 7. Although GANomaly is widely used for anomaly detection, its lower AUC compared to Doc-patch demonstrates that

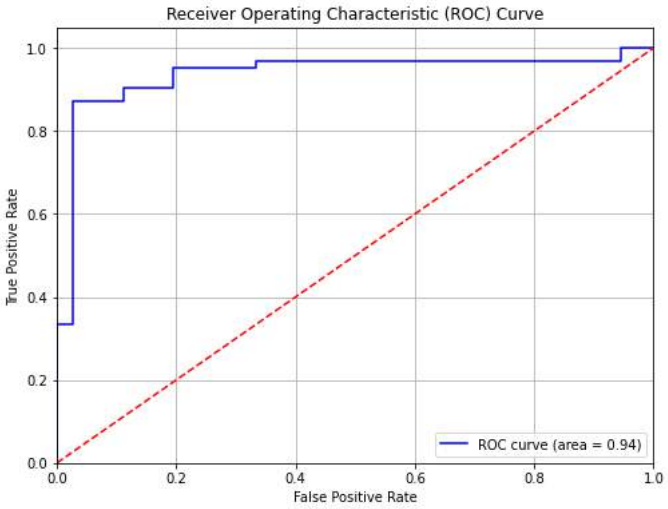


Fig. 5. Model precision

Fig. 6. Forged document detected

our method offers a significant improvement in detecting forged documents. This gap in performance highlights the effectiveness of our approach in capturing subtle anomalies and forgery patterns that GANomaly might miss [5].

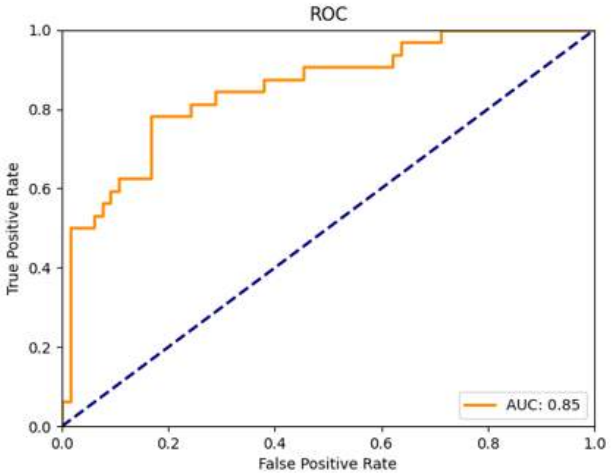


Fig. 7. GANomaly AUC curve

2) Cross-validation result: The cross-validation results in table I demonstrate the robustness and effectiveness of

the Doc-patch unsupervised forgery detection approach across multiple evaluation metrics, including Area Under the Curve (AUC), Receiver Operating Characteristic (ROC), and Average Precision. The system was evaluated across five different folds, yielding consistently high performance, which highlights its ability to generalize well to unseen data.

- **AUC Results:** The AUC scores are consistently high, ranging from 0.921 to 0.966, with Fold 3 achieving the highest AUC of 0.9669, indicating excellent separation between forged and genuine documents. The lowest AUC, 0.92107 in Fold 4, still reflects strong performance, emphasizing the reliability of the method across different data partitions.

- **ROC Results:** The ROC values show minimal variance, ranging from 0.92 to 0.97, indicating that the model maintains strong classification ability across different folds. The model effectively balances true positive and false positive rates, demonstrating its robustness even under challenging data conditions.

- **Average Precision:** Average precision scores also remain high, with a peak of 0.97963 in Fold 3 and a low of 0.953146 in Fold 4. This indicates that Doc-patch excels in identifying forged documents while keeping false positives low, a crucial requirement in practical applications.

TABLE I
PERFORMANCE METRICS ACROSS FOLDS

Folds	AUC score	ROC score	Average Precision score
1	0.946	0.95	0.968
2	0.92239	0.92	0.955
3	0.9669	0.97	0.97963
4	0.92107	0.92	0.953146
5	0.9563	0.96	0.972873

IV. CONCLUSION

In this study, we introduced and evaluated the Doc-Patch algorithm for detecting document forgeries. Our experimental results demonstrated that the algorithm is highly effective in distinguishing between authentic and forged documents, achieving a high AUROC value of 0.94. This indicates a strong discriminatory capability, allowing the algorithm to reliably identify forgeries with a high degree of accuracy. The average precision metric further validated the algorithm's performance, with a value between 95% and 97%, signifying that the majority of the forgery detections made by the algorithm were correct. This high precision rate underscores the algorithm's potential as a robust tool for practical applications in document authentication and forensic analysis. Despite the algorithm's strong performance, there is room for improvement, particularly in enhancing recall to ensure a more comprehensive detection of forgeries. Future work could focus on optimizing the algorithm to balance both precision and recall, possibly by incorporating additional features or refining the anomaly detection thresholds.

REFERENCES

[1] F. Cruz, N. Sidère, M. Coustaty, V. P. D'Andecy and J. -M. Ogier, "Local Binary Patterns for Document Forgery Detection," 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 1223-1228, doi: 10.1109/ICDAR.2017.202. keywords: Forgery;Layout;Companies;Shape;Forensics;Tools;Watermarking;Document analysis;Forgery Detection;Local Binary Patterns.

[2] Huang, N., He, J., & Zhu, N. (2018, August). A novel method for detecting image forgery based on convolutional neural network. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (pp. 1702-1705). IEEE.

[3] Roth, K., Pemula, L., Zepeda, J., Sch¨ olkopf, B., Brox, T., & Gehler,P. (2022). Towards total recall in industrial anomaly detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 14318-14328).

[4] Abd Warif, N. B., Wahab, A. W. A., Idris, M. Y. I., Ramli, R., Salleh, R., Shamshirband, S., & Choo, K. K. R. (2016). Copy-move forgery detection: survey, challenges and future directions. Journal of Network and Computer Applications, 75, 259-278.

[5] Akcay, S., Atapour-Abarghouei, A., & Breckon, T. P. (2019). Ganomaly: Semi-supervised anomaly detection via adversarial training. In Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14 (pp. 622-637). Springer International Publishing.

[6] Luo, A., Cai, R., Kong, C., Kang, X., Huang, J., Kot, A. C. (2023). Forgery-aware adaptive vision transformer for face forgery detection. arXiv preprint arXiv:2309.11092.

[7] Jaiswal, G., Sharma, A., & Yadav, S. K. (2022). Deep feature extraction for document forgery detection with convolutional autoencoders. Computers and Electrical Engineering, 99, 107770.

[8] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E. & Kloft, M.. (2018). Deep One-Class Classification. ¡i¿Proceedings of the 35th International Conference on Machine Learning¡/i¿, in ¡i¿Proceedings of Machine Learning Research¡/i¿ 80:4393-4402 Available from https://proceedings.mlr.press/v80/ruff18a.html.

[9] He, X., et al. "Forgery Detection in Certificates Using Deep Learning Techniques." Journal of Computer Security, 2020.

[10] Sudiatmika, I. B. K., Rahman, F., Trisno, T., & Suyoto, S. (2019). Image forgery detection using error level analysis and deep learning. TELKOMNIKA (Telecommunication Computing Electronics and Control), 17(2), 653-659.

[11] Sarode, S., Khandare, U., Jadhav, S.T., Jannu, A., Kamble, V., & Patil, D. (2020). Document Manipulation Detection and Authenticity Verification Using Machine Learning and Blockchain.

[12] Walia, S., Kumar, K., Kumar, M., & Gao, X. Z. (2021). Fusion of handcrafted and deep features for forgery detection in digital images. IEEE Access, 9, 99742-99755.

[13] Ahmed, B., Gulliver, T. A., & alZahir, S. (2020). Image splicing detection using mask-RCNN. Signal, Image and Video Processing, 14(5), 1035-1042.

[14] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P. E., ... & Jégou, H. (2024). The faiss library. arXiv preprint arXiv:2401.08281.

[15] Zhou, P., Han, X., Morariu, V. I., & Davis, L. S. (2018). Learning rich features for image manipulation detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1053-1061).

[16] Darem, Abdulbasit & Al-Hashmi, Asma & Javed, Mohammed & Abubaker, A. (2020). Digital Forgery Detection of Official Document Images in Compressed Domain. 10.22937/IJCSNS.2020.20.12.12.

[17] Boonkrong, S. (2024). Design of an academic document forgery detection system. International Journal of Information Technology, 1-13.

[18] Poddar, J., Parikh, V., & Bharti, S. K. (2020). Offline signature recognition and forgery detection using deep learning. Procedia Computer Science, 170, 610-617.

VAE based Disentanglement Learning by Dimension-wise Constraints to Latent Variables

1st Xiangtian Zheng
Artificial Intelligence Institute
University of Jinan
Jinan, China
zhenghengchi@gmail.com

2nd Yuehui Chen
Artificial Intelligence Institute
University of Jinan
Jinan, China
yhchen@ujn.edu.cn

3rd Yi Cao
Artificial Intelligence Institute
University of Jinan
Jinan, China
ise_caoy@ujn.edu.cn

4th Yaou Zhao
Artificial Intelligence Institute
University of Jinan
Jinan, China
ise_zhaoyo@ujn.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899478

***Abstract*—Disentanglement representation learning usually refers to the separation of factors that vary within the data, which are often physically meaningful and independent features that are important in real-world scenarios. β-VAE, as the basic model for unsupervised disentanglement, employs a mechanism to re-weight the KL divergence. However, it is often faced with the problem of how to adjust β to trade off reconstruction fidelity against disentanglement. We find that disentanglement appears to be embodied in individual dimensions of latent features that contain more information than others. Therefore, We propose Dimensionally Constrained VAE (DCVAE), which is a simple alternative to β-VAE for improving reconstruction quality and learning to achieve higher disentanglement scores. The effectiveness of this model is verified on the mainstream dSprites and CelebA datasets.**

***Index Terms*—disentanglement, representation learning, deep learning, variational autoencoder**

I. INTRODUCTION

Numerous deep generative models have achieved significant success in a variety of application domains attributed to the ability of their neural networks to extract meaningful features from high-dimensional inputs, but these features are often semantically meaningless, which can negatively impact interpretability, fairness, and downstream tasks performance [1]. Through the learned representations, it is difficult to obtain any direct insight into the structure of the data. As one candidate solution, disentanglement learning has emerged, aiming to find latent vectors with their components separated functionally [2]. By tuning a single component, it is expected that only one corresponding aspect would change in the generated samples. For example, for a face image, when the component that is responsible for hair color is tuned, the hair color changes correspondingly, leaving other aspects unchanged, such as facial expression or hair length [3]–[5]. Such special latent variables are regarded as being disentangled, which probably match a set of attributes in the real data generation process [6]. The definition of disentanglement learning is still somewhat controversial [7], while its potential is attractive, especially for precisely controlled content generation.

β-VAE is the initial variational autoencoder (VAE) based model for disentanglement learning. It amplifies the coefficient β of the KL divergence term in the loss function to force the latent distribution to move closer to the standard Gaussian distribution, which indirectly achieves disentanglement at the cost of reconstruction quality [8]. From the perspective of information bottleneck theory [9], the VAE model is viewed as a simple communication system. The raw sample $\boldsymbol{x}$ is encoded into latent $\boldsymbol{z}$ and sent through the bottleneck by the encoder, and the decoder tries to recover $\boldsymbol{x}$ from the received $\boldsymbol{z}$. The maximal information that can be carried by $\boldsymbol{z}$ is the channel capacity, which is the KL divergence term in the evidence lower bound(ELBO). Thus, β controls how much information can pass through the bottleneck. A large β drives the encoder to discard useless information to get a concise representation, which probably matches the real data generation factors. However, the encoder may also drop useful information due to the large β. Therefore, a trade-off is required [10]. An alternative to resolve this issue is to extract the correlation term from the KL divergence term and penalize it solely. As in β-TCVAE [11], the total correlation is separated out by KL divergence decomposition and estimated by Monte Carlo Markov Chain (MCMC) method [12]. Although superior performance is realized, the MCMC estimation is rather complicated.

In this paper, we follow the information bottleneck theory [13], [14] and consider the KL term as the sum of the KL losses for each dimension (component). We pick up the ones with larger KL divergence losses and exert an additional penalty. In this way, the redundant information can be eliminated gradually so that both disentanglement and fidelity can be achieved.

Our contribution can be summarized as follows:

- We propose a dimension-wise decomposition of the KL divergence term in VAE and show the channel capacity can be controlled individually, based on which, a novel model, namely, DCVAE is constructed. This model balances disentanglement and reconstruction by imposing an additional penalty on the high KL divergence components, enabling disentanglement while retaining sufficient information for high-quality reconstructions.
- We mathematically demonstrate the feasibility of this approach and show that our model outperforms other competitive methods through extensive experiments on benchmark datasets, such as dSprites and CelebA.

II. RELATED WORK

A. Model Architectures

Disentanglement representation learning aims to learn an important factor in a given dataset that corresponds to a true factor in the generated dataset. While such a goal brings interpretability to the learned features of a neural network, it is notoriously difficult to implement. Higgins et al. (2017) introduced the basic architecture β-VAE to the disentangling model, which utilizes the KL divergence as a loss, forcing the data's posterior distribution to match the hypothesized prior distribution. This in turn facilitates the disentangling of latent features, as shown in Rolinek et al [15]. The subsequent β-TCVAE architecture proposed by Chen et al [11].

This work was supported in part by the University of Jinan Disciplinary Cross-Convergence Construction Project 2023 (XKJC-202308) and in part by Shandong Provincial Natural Science Foundation, China (ZR2021MF036).

further decomposes the KL divergence term of ELBO into index code mutual information, total correlation, and dimensional KL terms. They demonstrate that the TC term (total correlation term) is an important reason for encouraging disentanglement in the KL loss term. Similarly, FactorVAE also works on constraining the TC term. Unlike TCVAE, they employ density ratio estimation techniques and adversarial networks to estimate the total correlation term rather than using an easy-to-handle but biased Monte Carlo approach. There also exist a number of network architectures, such as ControlVAE [16], that are based on the theory of information bottlenecks, which consider both information compression and information preservation to obtain compact information representations.

B. Information Bottleneck Theory

The information bottleneck theory, a new information-theoretic paradigm that considers both data compression and information preservation to obtain information representations, was originally proposed by Naftali Tishby, Fernando C. Pereira, et al [17]. The β-VAE model can be regarded as a typical information compression model in IB theory. It maps inputs to probability distributions of "bottleneck" variables (whose distributions can be viewed as channels through which information flows from the encoder to the decoder), which usually correspond to some specific semantics. VAE controls the transmission of information through information bottlenecks in the information compression process. The bottleneck controls the transmission of information, allowing the bottleneck layer to maximize the retention of critical information used for the task. For example, CascadeVAE [18] sequentially releases a latent variable at each stage to increase the IB, and DynamicVAE [19] designs a nonlinear PI controller to manipulate β to control the steadily increasing IB. However, each of these approaches suffers from diffusing some information from the unwrapped representation to some irrelevant variables. To address this, We propose a new strategy to constrain this diffusion of information so that most models can sample this strategy and achieve better disentangling results.

III. PROPOSED APPROACH

A. Problem Formulation

Given a set of samples that are generated by an unobservable process $g(\cdot)$ with a set of factors $\boldsymbol{c}$, disentanglement learning aims to learn the latent $\boldsymbol{z}$, whose components z_i align to the factors c_j. By a VAE, there are an Encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and a Decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ which are both parameterized probability distributions by two neural networks. Usually, $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, constructed from a Gaussian distribution with the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ supplied by the encoder network. The learning is driven by maximizing the evidence lower bound (ELBO) below.

$$\mathcal{L}(\phi,\theta) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathrm{D}_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p(\boldsymbol{z})\right) \quad (1)$$

The first term is the reconstruction loss and the second term can be viewed as a regularization. As pointed out in [8], The regularisation term affects the representations learned by the encoder, while the reconstruction term improves the outputs from the decoder. These terms usually contradict in practice, with strong regularisation leading to worse reconstructions but often better disentanglement.

$$\mathcal{L}(\phi,\theta) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log_{p_\theta}(\boldsymbol{x} \mid \boldsymbol{z})\right] - \beta\mathrm{D}_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \| p(\boldsymbol{z})\right) \quad (2)$$

As shown in Eq.2. β leads to the removal of the redundant components of $\boldsymbol{z}$, leaving the rest of the informative ones disentangled. From the perspective of information bottleneck theory, a large β limits the bottleneck's capacity. The components with little contribution to the reconstruction are eliminated, resulting in more disentangled. We can illustrate this mechanism more clearly by the derivation in [20]. To write the ELBO in a way that includes the aggregated posterior $q(\boldsymbol{z})$, it is convenient to treat the index n as a random variable. We identify each training example with a unique integer index and define a uniform random variable on $\{1, 2, \ldots, N\}$ with which we relate to data points. While the manipulation is entirely algebraic, this treatment makes the steps simpler and the result more interpretable. In particular, define the joint densities

$$q(n, \boldsymbol{z}) \triangleq q(n)q(\boldsymbol{z} \mid n), \quad q(\boldsymbol{z} \mid n) \triangleq q\left(\boldsymbol{z} \mid \boldsymbol{x_n}\right), \quad q(n) \triangleq \frac{1}{N} \quad (3)$$

$$p(n, \boldsymbol{z}) \triangleq p(n)p(\boldsymbol{z} \mid n), \quad p(\boldsymbol{z} \mid n) \triangleq p(\boldsymbol{z}), \quad p(n) \triangleq \frac{1}{N} \quad (4)$$

The aggregated posterior, $q(\boldsymbol{z})$, can be expressed as $q(\boldsymbol{z}) = \sum_{n=1}^{N} q(\boldsymbol{z}|n)p(n)$. By introducing the empirical distribution to the KL divergence term, its mean can be decomposed as below.

$$\frac{1}{N}\sum_{n=1}^{N}\mathrm{D}_{\mathrm{KL}}\left(q\left(\boldsymbol{z_n} \mid \boldsymbol{x_n}\right)\|p\left(\boldsymbol{z_n}\right)\right) = \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z})\|p(\boldsymbol{z})) + I_{q(n,\boldsymbol{z})}[n, \boldsymbol{z}] \quad (5)$$

The first item describes the degree of difference between the prior $p(\boldsymbol{z})$ and the aggregated posterior $q(\boldsymbol{z})$, which is non-negative. The second term is called the index-code mutual information, containing the correlation between $\boldsymbol{x}$ and $\boldsymbol{z}$. Since the average KL divergence (the left-hand side) is the upper bound of $I_{q(n,\boldsymbol{z})}(n; \boldsymbol{z})$, penalizing this term is to minimize the $I_{q(n,\boldsymbol{z})}(n; \boldsymbol{z})$. Concurrently, the interaction between the reconstruction loss and the index-code mutual information leads to $\boldsymbol{z}$ encoding pivotal information necessary for accurate reconstruction, exhibiting characteristics of disentanglement. Observations indicate that the efficacy of disentanglement is highly contingent upon the magnitude of β. As delineated in [10], an excessively large value of β detrimentally impacts the reconstruction quality, whereas an insufficiently small β fails to facilitate the disentanglement process. Such a trade-off is the central problem in this paper.

B. Dimensionally Constrained Information Bottlenecks

In VAE, the posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is a multi-dimensional Gaussion distribution $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the correlation matrix $\boldsymbol{\Sigma}$ is diagonal. It has been proved that diagonal matrices are enough for learning complex data distributions since the correlations among the components of $\boldsymbol{z}$ can be stored in the encoder and decoder. Hence, the components of $\boldsymbol{z}$ are conditionally independent. According to the properties of KL divergence, if $p(\boldsymbol{u}) = \prod p(u_i)$ and $p(\boldsymbol{v}) = \prod p(v_i)$, it can be decomposed per element as (Eq. 6)

$$\mathrm{D}_{\mathrm{KL}}(p(\boldsymbol{u})||p(\boldsymbol{v})) = \sum \mathrm{D}_{\mathrm{KL}}(p(u_i)||p(v_i)) \quad (6)$$

By this feature, the KL divergence in (2) can be decomposed as follows.

$$\beta\mathrm{D}_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\|p(\boldsymbol{z})\right) = \sum_{i=1}^{M}\beta\mathrm{D}_{\mathrm{KL}}\left(q_\phi\left(z_i \mid \boldsymbol{x}\right)\|p\left(z_i\right)\right) \quad (7)$$

where M is the dimension of $\boldsymbol{z}$. The bottleneck $\boldsymbol{z}$ can be treated as a set of independent tunnels z_i, which can be controlled individually.

When training a β-VAE(as shown in Fig. 2), the components of $\boldsymbol{z}$ do not equally contribute to the KL loss. By rewriting Eq. (7) into Eq. (8) and (9), it can be seen that if the KL loss of z_i is large (or drops slightly), it carries more information. If the KL loss still does not drop for a relatively large β, this component is probably essential to the reconstruction. Conversely, if a component makes little contribution to the reconstruction, it tends to carry little information and lower the KL loss.

$$\frac{1}{N}\sum_{n=1}^{N}\mathrm{D}_{\mathrm{KL}}\left(q_\phi(\boldsymbol{z_n} \mid \boldsymbol{x_n})\|p(\boldsymbol{z_n})\right) = \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{z})\|p(\boldsymbol{z})) + I_{q(n,\boldsymbol{z})}[n, \boldsymbol{z}] \quad (8)$$

$$\frac{1}{N}\sum_{n=1}^{N}\mathrm{D}_{\mathrm{KL}}\left(q_\phi(z_{n,i} \mid \boldsymbol{x_n})\|p(z_{n,i})\right) = \mathrm{D}_{\mathrm{KL}}(q(z_i)\|p(z_i)) + I_{q(n,z_i)}[n, z_i] \quad (9)$$

According to this idea, our proposed DCVAE structure is similar to β-VAE and incorporates dimensional constraints as illustrated in Fig. 1. This structure divides $\boldsymbol{z}$ into two distinct spaces with separate

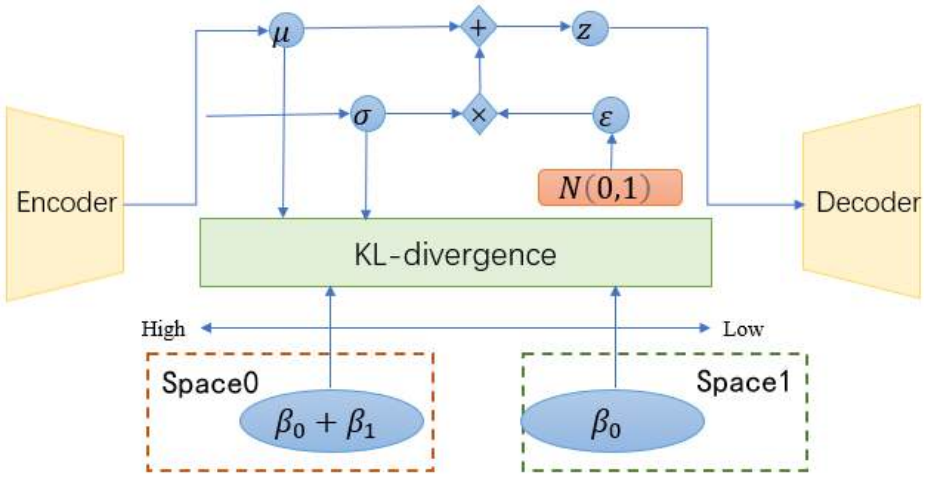


Fig. 1. An illustration of the Dimensional Constraint VAE (DCVAE) shows that it divides the KL divergence into two spaces, imposing a tighter constraint on Space 0, which has a higher KL value. This approach ensures that not all spaces are subject to the same constraints, thereby allowing better control over the information flow and improving disentanglement.

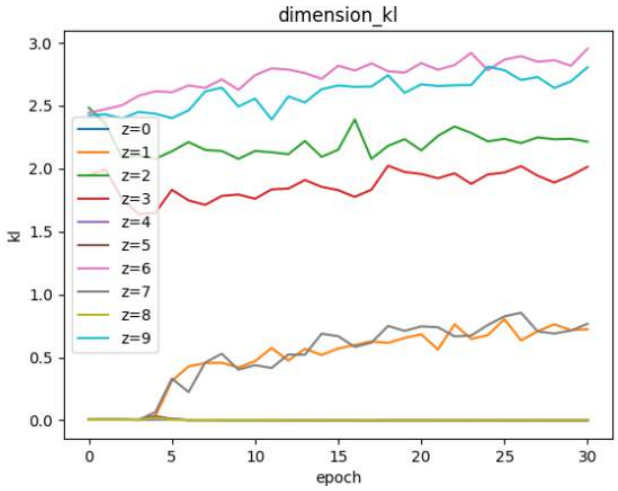


Fig. 2. The x-axis indicates the training epochs, the y-axis indicates the KL values corresponding to each epoch, and the different colors of z indicate the different dimensions of the hidden variable z.

constraints based on the amount of information each component contains. If a component significantly deviates from $\mathrm{N}(0,1)$, it indicates a higher mutual information with the input sample $\boldsymbol{x}$. Following the information bottleneck theory, we aim to reduce the mutual information in the reconstruction task to facilitate de-entanglement, advocating the removal of such information. Therefore, components with greater deviation should incur a larger β penalty. Figure 2 shows a noticeable gap in the KL divergence values among the dimensions of z_i during training. By this gap, we categorized the dimensions of z_i into two groups and imposed a higher penalty on the group with larger KL losses(The number of dimensions in this group should exceed the number of attributes in the dataset. Therefore, during the experiments, we chose half of the predefined dimensions of $\boldsymbol{z}$ as the cutoff point). During the training process, all components initially receive a default coefficient β_0 for calculating the KL divergence loss. An additional penalty β_1 is assigned to the dimensions with substantial KL losses. By applying increased pressure to these dimensions, redundant information is eliminated from the latent $\boldsymbol{z}$, resulting in a disentangled representation with high fidelity.

IV. RESULTS

A. Datasets and Experiments Description

The dataset we used was dSprites [21] and CelebA [22], designed for the disentangling test, containing 737,280 binary 64 × 64 images of 2D shapes with five ground truth factors: shape (4), scale (11), orientation (40), position X (5), and position Y (5). Our model and the base model were evaluated simultaneously on this dataset, as shown in Table 1. For the performance evaluation of disentanglement, we used two evaluation metrics. MIG(the higher, the better) refers to the mutual information gap between the two variables with the highest and the second highest mutual information. DCI_{Dis}(the higher, the better): abbreviation for DCI Disentanglement [23] measures how much each latent unit captures a ground-truth factor using a predictive model. Recon(the lower, the better): reconstruction error metric, binary cross entropy used for dSprites (binary images). In order to assess whether DCVAE provides a performance gain due to the imbalance in the penalty coefficients, we compared it to the β-VAE using $\beta = 4$ and $\beta = 6$, respectively. In addition, we evaluated it against the β-TCVAE model, which had previously performed well. As shown in Table 1, the MIG and DCI scores obtained using dimensional constraints are higher and more stable than those obtained by β-VAE and β-TCVAE, leading to better reconstruction quality. Subsequently, we applied the method to β-TCVAE to verify its generalization ability, as shown in Figure 4.

TABLE I
COMPARISON OF DIFFERENT MODELS ON THE DSPRITES DATASET

Dataset	Model	MIG↑	DCI Dis↑	Recon↓
dSprites	DCVAE($\beta_0 = 4, \beta_1 = 2$)	0.31 ± 0.05	0.41 ± 0.04	40.33 ± 2.11
dSprites	β-VAE ($\beta = 4$)	0.12 ± 0.02	0.27 ± 0.02	32.12 ± 1.5
dSprites	β-VAE ($\beta = 6$)	0.21 ± 0.03	0.29 ± 0.02	45.35 ± 2.6
dSprites	β-TCVAE ($\beta = 6$)	0.28 ± 0.04	0.39 ± 0.02	63.35 ± 3.6

In the implementation, we utilized fully connected layers as the encoder and decoder of the model. The activation function employed is ReLU, the optimizer is Adam, and the learning rate is 1e-4. β-VAE operates with a batch size of 128, whereas the TCVAE family utilizes

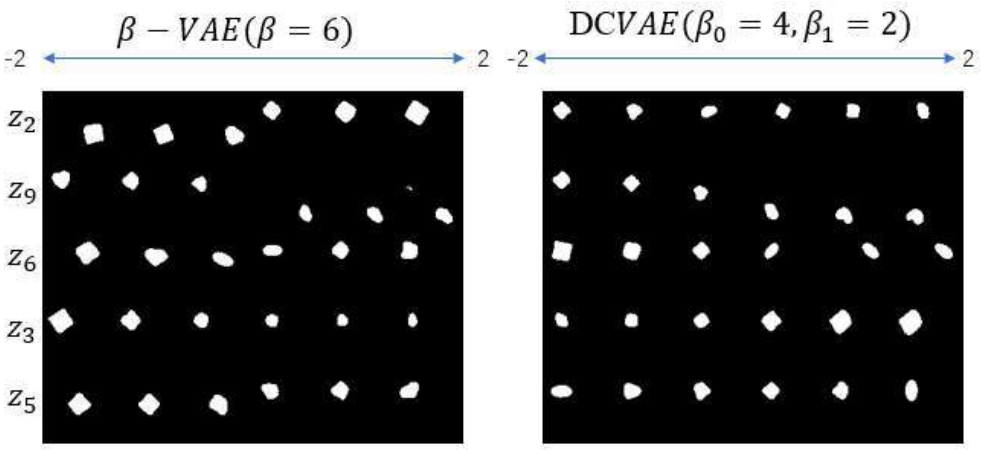


Fig. 3. The y-axis represents the five dimensions of the $\boldsymbol{z}$ with the highest KL values, in descending order, and the x-axis indicates traversing this dimension from [-2,2].

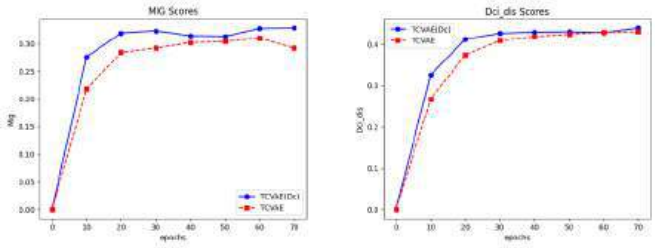


Fig. 4. The x-axis indicates the training epochs and y-axis indicate the MIG score and DCI_{dis} score.

a batch size 512. By default, all experiments are conducted over 70 training epochs. Regarding the hyperparameters, the β value for TCVAE was set at 6. From Table 1 and Fig. 4, it is concluded that the constraints on the dimensions are more effective and converge faster than the overall constraints. The overall performance is improved in both cases.

B. Visual Traversal of Images

We qualitatively examine the performance of the model in representing and disentangling the learned features. This examination uses the dSprites and CelebA datasets and is detailed through visualization experiments in this subsection. Specifically, each row of the visualized results shows reconstructed images in the latent space. These images vary systematically in only one dimension, from -2 to 2, while keeping the other dimensions constant. We selected the top five dimensions with the highest KL divergence to visualize their latent traversals, as shown in Figure 3. By comparing the left and right images, we find that the right image captures more attributes, is less affected by noise, and is more robust. For the CelebA datasets, we refrain from showing results obtained with β-VAE, as the generated images are not realistic. When DCVAE traverses the complex CelebA dataset (as shown in Figure 5), the latent feature layer can still correspond to more attributes, but some attributes remain entangled. Currently, learning disentangled representations in a completely unsupervised manner is still a challenging problem. Some key issues need urgent resolution, and we hope that the Dimension-wise Constraints can provide new insights into the nature of disentanglement.

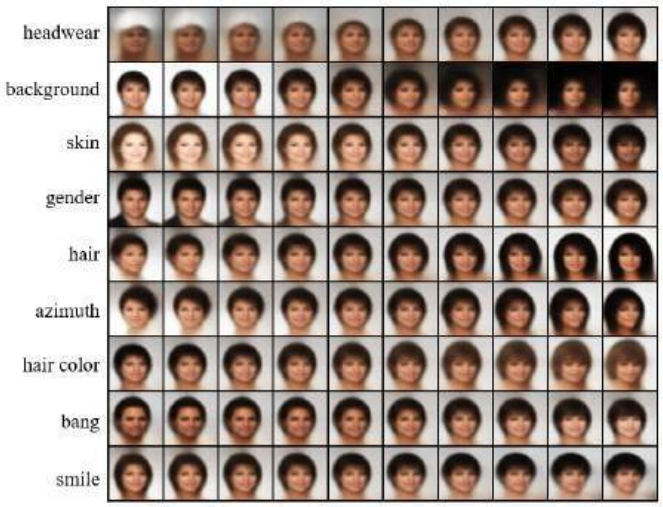


Fig. 5. DCVAE traversal of the CelebA dataset.

V. Conclusion and Future Work

We introduce DCVAE, a new method with higher disentangling and better reconstruction quality than β-VAE. By analyzing the contribution of individual dimensions in the KL divergence to the objective, we find that constraining individual dimension terms helps to obtain better disentangling scores and learn more properties than β-VAE.

However, learning disentangled representations in a completely unsupervised way is still a challenge; some key issues need to be better accounted for, and there is still a lack of better metrics to measure disentangling. On some datasets, human visualization is needed for comparison.

While our proposed method effectively decouples features and enhances image reconstruction fidelity, the inherent limitations of the VAE model's capacity remain a challenge. Therefore, our future work aims to expand the model's capacity, further reduce dimension entanglement, and identify additional decoupling factors to improve reconstruction fidelity.

REFERENCES

[1] I. Eddahmani, C.-H. Pham, T. Napoléon, I. Badoc, J.-R. Fouefack, and M. El-Bouz, "Unsupervised learning of disentangled representation via auto-encoding: A survey," *Sensors*, vol. 23, no. 4, p. 2362, 2023.

[2] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[3] S. Park, S. Hwang, D. Kim, and H. Byun, "Learning disentangled representation for fair facial attribute classification via fairness-aware information alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2403–2411.

[4] K.-Y. Zhang, T. Yao, J. Zhang, Y. Tai, S. Ding, J. Li, F. Huang, H. Song, and L. Ma, "Face anti-spoofing via disentangled representation learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*. Springer, 2020, pp. 641–657.

[5] G. Wang, H. Han, S. Shan, and X. Chen, "Cross-domain face presentation attack detection via multi-domain disentangled representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6678–6687.

[6] B. Esmaeili *et al.*, "Learning useful representations with variational autoencoders," 2024.

[7] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint arXiv:1812.02230*, 2018.

[8] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework." *ICLR (Poster)*, vol. 3, 2017.

[9] J. Song and S. Ermon, "Understanding the limitations of variational mutual information estimators," *arXiv preprint arXiv:1910.06222*, 2019.

[10] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv preprint arXiv:1804.03599*, 2018.

[11] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," *Advances in neural information processing systems*, vol. 31, 2018.

[12] H. Kim and A. Mnih, "Disentangling by factorising," in *International conference on machine learning*. PMLR, 2018, pp. 2649–2658.

[13] S. Hu, Z. Lou, X. Yan, and Y. Ye, "A survey on information bottleneck," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[14] J. Wu, S. Mo, X. Yang, M. Awais, S. Atito, X. Zhang, and L. Wang, "Variantional autoencoder with decremental information bottleneck for disentanglement," *arXiv preprint arXiv:2303.12959*, 2023.

[15] M. Rolinek, D. Zietlow, and G. Martius, "Variational autoencoders pursue pca directions (by accident)," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 406–12 415.

[16] H. Shao, Z. Xiao, S. Yao, D. Sun, A. Zhang, S. Liu, T. Wang, J. Li, and T. Abdelzaher, "Controlvae: Tuning, analytical properties, and performance analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 9285–9297, 2021.

[17] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[18] Y. Jeong and H. O. Song, "Learning discrete and continuous factors of data via alternating disentanglement," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3091–3099.

[19] H. Shao, Y. Yang, H. Lin, L. Lin, Y. Chen, Q. Yang, and H. Zhao, "Rethinking controllable variational autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 250–19 259.

[20] M. D. Hoffman and M. J. Johnson, "Elbo surgery: yet another way to carve up the variational evidence lower bound," in *Workshop in Advances in Approximate Bayesian Inference, NIPS*, vol. 1, no. 2, 2016.

[21] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, "dsprites: disentanglement testing sprites dataset (2017)," *URL https://github.com/deepmind/dsprites-dataset*, p. 27, 2020.

[22] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, no. 2018, p. 11, 2018.

[23] C. Eastwood and C. K. Williams, "A framework for the quantitative evaluation of disentangled representations," in *International conference on learning representations*, 2018.

Multi-subpopulation artificial bee colony algorithm based on individual classification

1st Sumin Li*
Minzu University of China
Beijing, China
smli@muc.edu.cn

2nd Menghan Li
Minzu University of China
Beijing, China
22301994@muc.edu.cn

3rd Xiuqin Pan
Minzu University of China
Beijing, China
amycun@muc.edu.cn

4th Jia Wang
Minzu University of China
Beijing, China
21011595@muc.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899668

***Abstract*—The artificial bee colony algorithm faces difficulties in insufficient search performance when tackling intricate optimization tasks. To improve this problem and enhance the algorithm's ability to effectively balance the exploration and exploitation, we propose a multi-subpopulation artificial bee colony algorithm that utilizes individual classification to enhance performance(DZABC).In DZABC, the whole bee population was segmented into three distinct sub-populations based on varying levels, and each sub-population adopted strategies with different characteristics as candidate strategies according to its unique population characteristics. This diversity of strategies enables algorithms to leverage the strengths of different strategies to optimize the overall search performance. The introduction of multi-subpopulation mechanism not only aids in preserving population diversity, but also ensures that the algorithm achieves good convergence performance in the search process. In addition, we design a new judgment mechanism in the scout bee phase to further enhance the algorithm's convergence efficiency. To validate the efficacy of the DZABC algorithm, we apply it to 16 test functions, and demonstrate its robust competitiveness through comparative analysis with other algorithms. Experimental outcomes indicate that the DZABC algorithm outperforms other comparative algorithms to a certain extent, which shows its potential and advantage in solving optimization tasks.**

***Keywords*—Artificial bee colony algorithm, Individual classification, Multi-subpopulation mechanism, Scout bee judgment mechanism**

I. Introduction

In recent years, inspired by the group behavior of organisms, swarm intelligent optimization algorithms [1] came into being, and they show unique advantages in the combination of optimization problems. Within this context, artificial bee colony algorithm (ABC) [2] , as a typical swarm intelligence algorithm, was put forward by Turkish scholar Karaboga in 2005. It features the benefit of few parameters, along with strong global exploration skills and straightforward implementation. Based on the above advantages, ABC algorithm has been widely used to solve diverse practical optimization tasks, including scheduling issues [3], parameter optimization tasks [4] and image processing applications [5]. However, ABC is prone to poor search performance when confronted with certain complex optimization tasks. To optimize the search performance of ABC, researchers have carried out a lot of research.

Researchers have advanced the ABC algorithm to address its shortcomings. Literature [6] incorporates valuable knowledge from the optimal individual into the algorithm's search process and proposes an modified ABC (GABC). A novel ABC with adaptive adjustment mechanism was proposed in literature [7]. ABC uses this adjustment mechanism to adaptively regulate the number of sub-populations throughout the iteration, and each sub-population conducting independent searches. Literature [8] proposed an artificial bee colony algorithm grounded in behavioral development. Employed bees, onlooker bees, and scout bees are considered as three stages in the behavioral progression of bees, achieving search process division of labor through individual specialization and role plasticity. Literature [9] proposes a multi-role oriented variable dimension disturbance ABC, which meticulously regulates and refines the search conduct of bee colonies,leveraging the synergistic interplay among multiple preference search equations, guided by multi-role nectar sources and the phases of employed bees, onlooker bees and scout bees. At the same time, literature [10] mentioned that a multi-population search approach effectively sustains population diversity. An modified multi-population ABC (MPGABC) was proposed in reference [11]. In MPGABC, population diversity is maintained by independently searching multiple sub-populations using improved search strategies.

Through the analysis of the existing modified ABC, it can be seen that the multi-population mechanism effectively maintains the population diversity [11]. However, the majority of existing modified ABCs within the multi-population framework employ a singular search strategy, rarely integrating multiple search strategies. Therefore, a multi-subpopulation artificial bee colony algorithm that utilizes individual classification (DZABC) is proposed in this paper. Literature [12] points out that an effective algorithm should properly balance exploration and exploitation throughout its evolutionary process. Therefore, in DZABC, the multi-population mechanism was devised, with the population segmented into elite sub-population, medium sub-population and ordinary sub-population according to individual rank. To attain the equilibrium between exploration and exploitation, the three

This work was supported by the National Natural Science Foundation of China under Grant 62176273 and by National first-class undergraduate major in software engineering.

sub-populations adopt different search strategies with different properties, and diverse search behaviors are exhibited by individuals within the population. At the same time, in the scout phase of bees, considering that the individuals whose parameter value reaches limit will be re-initialized, there may be some cases that the individuals themselves are relatively excellent and are abandoned because they have not been updated for many times, a judgment mechanism is set up to enhance the algorithm's search efficiency.To validate its performance, this paper selected 16 common standard test functions to conduct comparative experiments between DZABC and other variants of ABC. The results demonstrate that, compared to other algorithms, the algorithm presented in this paper exhibits strong competitiveness.

The remainder of the text is structured as follows: Section 2 provides an overview of the basic ABC and a description of DZABC. Section 3, the performance of DZABC is analyzed through relevant experiments. The fourth part summarizes the work of this paper.

II. METHOD

A. The basic ABC

ABC algorithm is a novel intelligent optimization algorithm. It mainly emulates the behavior of employed bees, onlooker bees, and scout bees in their quest for food sources. Depending on the type of bee being modeled, the ABC executes the respective phases to identify an optimal food source, that is, a good individual. The ABC phase is as follows:

Initialization phase: According to (1), SN random individuals are generated in the solution space to form the initial population.

$$x_{ij} = l_j + rand(0,1) * (u_j - l_j) \tag{1}$$

in (1), $i = 1, 2, 3, ..., SN, j = 1, 2, 3, ..., D$. x_{ij} denotes the j-th dimension of the i-th individual. The variables u_j and l_j repectively denote the upper and lower bounds,for the j-th dimension.

Empolyed bee phase: An employed bee probes the neighborhood of individual x_i and formulates a new candidate individual v_i according to (2).Then, through greedy selection mechanism, the dominant individuals are retained.

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij} - x_{kj}) \tag{2}$$

in (2), $k \neq i$, x_{kj} denotes the j-th dimension of a randomly chosen k-th individual from the population. ϕ_{ij} represents a random real number within the range [-1,1].

Onlooker bee phase: Upon selecting an individual x_i via the roulette method, an onlooker bee searches for a new candidate individual v_i in its vicinity following (2),subsequently retaining the dominant individual.

Scout bee phase: Upon an individual failing to update for limit successive times, it is discarded, a fresh individual is generated according to (1) to replace the discarded individual.

B. Motivation

Like other intelligent optimization algorithms, ABC still requires enhancements to its search performance. Given ABC's susceptibility to inadequate development performance, researchers have proposed numerous search strategies to improve algorithm optimization performance [13]. An analysis reveals that the majority of these enhanced search strategies utilize valuable knowledge from the best or elite individuals to optimize algorithm's search efficiency. Nonetheless, during the evolutionary process, an excessive dependence on optimal or elite individuals can readily result in a swift decline in population diversity, potentially compromising the algorithm's performance. Hence, while developing the best or elite individuals to enhance the algorithm's performance, it is also necessary to effectively sustain the diversity of the population. Accordingly, existing research has concluded that a multi-population mechanism effectively sustains population diversity [11]. Nevertheless, the majority of multi-population mechanisms seldom consider the use of multiple search strategies. This restricts the enhancement of algorithm performance to some extent. Hence, according to the analysis of previous research, a multi-subpopulation mechanism based on individual rank is devised to sustain good population diversity while using elite individuals to improve exploitation ability. At the same time, the merits of diverse search strategies are used to improve the search performance in sub-populations. Moreover, in the scout bee phase, considering that the individuals whose parameter value reaches limit will be re-initialized, there may be some cases that the individuals themselves are relatively excellent and are abandoned because they have not been updated for many times, a judgment mechanism is set up to boost the algorithm's search performance.

C. Multi-population mechanism of employed bee

Studies in literature [14] show that algorithm performance can be optimized by using valuable knowledge of dominant individuals to explore potential regions. Nevertheless, an excessive reliance on excellent individuals can lead to a rapid decline in population diversity, which can easily lead to premature convergence and affect search performance. Therefore, this study adopts the multi-subpopulation mechanism to effectively maintain diversity. At the same time, several search strategies are utilized to enhance the algorithm's performance. Considering these factors, a multi-sub-population mechanism guided by individual classification is proposed to enhance the algorithm's performance.

In a multi-population mechanism, individual grading is achieved by ranking fitness values from highest to lowest. Subsequently, the entire population is divided into three subpopulations, with the proportion of individual numbers of elite sub-population, medium sub-population and ordinary subpopulation was 0.3, 0.3 and 0.4, respectively. First, within the elite sub-population, individuals exhibit superior fitness values, which suggests that the region where the elite population is situated is more likely to harbor the optimal nectar source. Hence, partial exploitation search strategy was used to improve

the convergence performance within the elite sub-population. Second, in the medium sub-population, individuals have relatively balanced fitness values, suggesting that the region contains multiple potential solutions. To sustain the diversity and exploration ability of the population, a search strategy that balances exploration and exploitation should be adopted for the medium sub-populations. Finally, within the ordinary sub-population, individuals display relatively inferior fitness values, suggesting that the region is less likely to discover the optimal individual. Therefore, the search strategy of partial exploration should be adopted to augment the diversity of the ordinary sub-population and expand the search range. In this way, the probability of the algorithm discovering new solution regions can be increased.

By implementing targeted search strategies in different sub-populations, individuals within the population can manifest distinct search behaviors and take advantage of them, effectively balancing exploration and exploitation, and improving algorithm performance.

In the elite sub-population, it is imperative to fully utilize the useful knowledge of dominant individuals. Therefore, it is necessary to adopt a search strategy that includes the best or elite individuals to enhance the exploitation capacity. A search strategy combining optimal and elite individuals is proposed, as shown in (3):

$$\nu_i = x_{e1} + \varphi_1^*(x_{e1} - x_{k1}) + \varphi_2^*(x_{best} - x_i) \tag{3}$$

where $k1 \neq e1 \neq i$, x_{k1} and x_{e1} respectively represent elite individuals randomly selected from the elite sub-population. x_{best} is the global optimal individual. φ_1 represents a random real number within the range [-1,1]. φ_2 represents a random real number within the range [0,1.5].

In the medium sub-population, there exists a requirement to maintain a equilibrium between exploration and exploitation. Therefore, a search strategy that includes elite individuals and random individuals is adopted to achieve a balance between the two abilities. A search strategy combining elite individuals and random individuals is proposed, as shown in (4) :

$$\nu_i = x_{e2} + \varphi_1 * (x_{e2} - x_{k2}) \tag{4}$$

where $k2 \neq e2 \neq i$, x_{k2} represents randomly chosen individuals from the entire population. x_{e2} represents an elite individual randomly selected from the elite sub-population.

In the ordinary sub-population, individuals exhibit relatively poor fitness values, suggesting that the region may contain fewer high-quality solutions. Search the entire search space for hopeful individuals while maintaining good population diversity. Therefore, it is necessary to adopt a search strategy that focuses on exploration. Through analysis [9], this strategy gives individuals stronger breadth-first exploration ability, which can guide bee colonies to cover a wider area, as shown in (5) :

$$\nu_i = x_a + \varphi_1 * (x_b - x_c) \tag{5}$$

where $a \neq b \neq c$, x_a, x_b and x_c respectively represent three individuals randomly chosen from the entire population.

D. Improved search strategy for onlooker bee

During the onlooker bee phase, the traditional following bee strategy often lacks the deep exploitation of the high-quality solution region, which leads to the limitation of the convergence speed and the quality of the solution. To overcome the shortcoming, we designed a strategy that cleverly incorporates individual information from the elite sub-population and significantly improves the algorithm's exploitation capability. By introducing the knowledge of elite individuals, we enable the onlooker bee to locate and dig potential high-quality solution areas more accurately in the search process. Its form is shown in (6) :

$$\nu_{i,j} = x_{i,j} + \varphi_1 * (x_{best,j} - x_{k1,j}) + \varphi_2 * (x_{e1,j} - x_{i,j}) \tag{6}$$

where $x_{i,j}$ represents the j-th dimension of the i-th individual, $x_{best,j}$ signifies the j-th dimension of the global optimal solution. $k1 \neq e1 \neq i$, $x_{k1,j}$, $x_{e1,j}$ respectively represent the jth dimension of the elite individuals randomly selected from the elite sub-population.

E. Strategy optimization of scout bee

During the scout bee phase, literature [15] showed that under the action of scout bee, the optimal food source within the population would often be given up. Consequently, the next best solution assumes the role of the new best solution within the population, resulting in fluctuations in the convergence curve. At the same time, considering that in the reconnaissance bee stage, the individual is relatively excellent after initialization, so that it has been abandoned for many times without updating. Inspired by this, we designed a judgment mechanism to distinguish elite and non-elite nectar sources, and adopted different search strategies according to different results.

If the current individual continuous limit has not been improved and belongs to elite sub-population, the exploration ability can be maintained by randomly selecting two different nectar sources for spatial adjustment. Its form is shown in (7):

$$x_{new} = x_i + \varphi_1 * (x_b - x_a) \tag{7}$$

where $i \neq a \neq b$.

If the current individual limit is not improved and does not belong to the elite sub-population, the scout bee need to generate a new individual according to (1) to replace the discarded individual and conduct a larger range of random search to avoid local optimization.

III. Numerical experiments and analysis

To validate the efficacy of the DZABC algorithm, DZABC is compared with four variants of ABC [2] , GABC [6] , BDLDABC [8] and DPGABC [16]. In this paper, 16 standard test functions widely adopted in the field of function optimization are used. The experiments are conducted on the computer with 11th Gen Intel(R) Core(TM) and CPU: i5-1135G7 @ 2.40GHz 2.42GHz, and the program is implemented with Python 3.9.12. Among them, $f_1 - f_4$ are uni-modal test functions, $f_5 - f_{16}$ are multi-modal test functions. Table I shows the parameter values of the algorithm, among which the

common parameters, SN=50, MaxIter=1500. Table II shows the definition, value range and optimal value of the $f_1 - f_{16}$ test functions.All algorithms were run independently for 10 times and compared from three aspects: mean and standard deviation, non-parametric test and convergence speed.

TABLE I
ALGORITHM PARAMETER VALUES

Algorithm	Parameters
ABC	$Limit = 0.6 * D * SN = 900$
GABC	$Limit = 0.6 * D * SN = 900$
BDLDABC	$\alpha = 20, \beta = 0.4, \gamma = 0.01, \mu = 0.5$
DPGABC	$Limit = 200, ei = 0.2, \alpha = 0.1$
DZABC	$Limit = 0.6 * D * SN = 900$

TABLE II
BENCHMARK FUNCTIONS IN EXPERIMENTS

Function	Name	Dim	Range	Min
f_1	Schwefel P1.2	30	[-100,100]	0
f_2	Step	30	[-100,100]	0
f_3	Exponential	30	[-1,1]	-1
f_4	Quartic	30	[-1.28,1.28]	0
f_5	Xin-She Yang 6	30	[-10,10]	-1
f_6	Rastrigin	30	[-5.12,5.12]	0
f_7	Penalized2	30	[-50,50]	0
f_8	Alpine	30	[-10,10]	0
f_9	Zakharov	30	[-5,10]	0
f_{10}	Levy	30	[-10,10]	0
f_{11}	Holzman	30	[-10,10]	0
f_{12}	Cosine Mixture	30	[-1,1]	$-0.1 * D = -3$
f_{13}	Kowalik	4	[-5,5]	0.00030748610
f_{14}	Colville	4	[-10,10]	0
f_{15}	Matyas	2	[-10,10]	0
f_{16}	Goldstein-Price	2	[-2,2]	3

Table III shows the results of DZABC and other 4 variants of ABC on 16 test functions (D=30). Among the comparison results, the optimal result is bolded, and the experimental result is reserved for 2 decimal places.As you can see from the table, DZABC has best results on f_2,f_3,f_6,f_{12},f_{14},f_{16}.Compared to the outcomes of ABC, GABC, BDLDABC and DPGABC yields superior results on f_1,f_4,f_5,f_8,f_9,f_{11},f_{13},f_{14},f_{15}, comparable results on f_2,f_3,f_6,f_7,f_{10},f_{12},f_{16}. For uni-modal functions, DZABC has best results on f_2,f_3. For the rest of the functions, DZABC also demonstrates favorable convergence outcomes, outperforming other comparative algorithms. For multi-modal functions, DZABC has best results on f_6,f_{12},f_{14},f_{16}. For the rest of the functions, DZABC is superior to most comparison algorithms.From the comparison of experimental results, DZABC's advantage on multi-modal function is more prominent than that on uni-modal function, which also shows that DZABC can achieve superior results in addressing complex optimization issues. In summary, the experimental outcomes indicate that DZABC effectively improves the search performance of the algorithm and obtains superior results compared to most ABC variants.

To ensure the statistical significance of comparison results, two non-parametric test methods, Friedman and Wilcoxon, were adopted [17]. The results of the two tests are presented at the end of Table III . The Friedman test gives the overall performance of the algorithm by means of average ranking. The lower the ranking value, the superior the performance of the algorithm. The test results in Table III show that DZABC has the smallest ranking value across all algorithms, suggesting the best performance. Wilcoxon tests whether DZABC is significantly different from other algorithms, and the significance level is established at 0.05.The test outcomes are given in the form of "R+" , "R-" and "P-value", where "R+" represents the rank sum of DZABC on the dominant function, and "R-" represents the rank sum on the inferior function. The test results presented in Table III demonstrate that the "R+" obtained by DZABC in all comparison cases is greater than "R-", and the "P-value" are less than 0.05, indicating that the algorithm proposed in this paper is significantly better than other comparison algorithms.

TABLE III
COMPARISON BETWEEN DZABC AND OTHER ABCS ON THE BENCHMARK FUNCTIONS (D = 30)

Function	*Metric*	*ABC*	*GABC*	*BDLDABC*	*DPGABC*	*DZABC*
f_1	Mean	7.98e+03	8.19e+03	2.07e+03	3.09e+03	**1.05e-01**
	Std	1.91e+03	1.83e+03	7.68e+02	1.02e+03	6.96e-02
f_2	Mean	**0.00e+00**	**0.00e+00**	6.01e+01	**0.00e+00**	**0.00e+00**
	Std	0.00e+00	0.00e+00	9.58e+01	0.00e+00	0.00e+00
f_3	Mean	**-1.00e+00**	**-1.00e+00**	**-1.00e+00**	**-1.00e+00**	**-1.00e+00**
	Std	3.51e-17	4.97e-17	7.93e-15	1.05e-16	9.93e-17
f_4	Mean	7.06e-02	3.20e-02	2.53e-02	1.18e-02	**9.98e-03**
	Std	1.83e-02	5.21e-03	1.50e-02	2.68e-03	3.62e-03
f_5	Mean	6.67e-20	1.80e-20	4.61e-42	9.71e-23	**1.40e-43**
	Std	4.81e-20	2.77e-20	4.76e-42	2.91e-22	1.99e-59
f_6	Mean	**0.00e+00**	**0.00e+00**	2.65e-14	**0.00e+00**	**0.00e+00**
	Std	0.00e+00	0.00e+00	4.74e-14	0.00e+00	0.00e+00
f_7	Mean	3.45e-20	**1.35e-32**	7.84e-01	**1.35e-32**	**1.35e-32**
	Std	2.39e-20	2.74e-48	1.13e+00	2.74e-48	2.74e-48
f_8	Mean	3.18e-06	1.44e-06	8.34e-14	3.39e-09	**3.35e-15**
	Std	1.75e-06	2.02e-06	1.96e-13	6.77e-09	1.37e-15
f_9	Mean	2.26e+02	2.43e+02	2.13e+02	1.76e+01	**1.17e-03**
	Std	3.00e+01	1.70e+01	2.72e+01	6.33e+00	7.33e-04
f_{10}	Mean	6.30e-22	**1.50e-32**	4.37e+00	**1.50e-32**	**1.50e-32**
	Std	7.96e-22	0.00e+00	3.31e+00	0.00e+00	0.00e+00
f_{11}	Mean	8.19e-43	5.22e-70	8.97e-51	4.37e-78	**8.92e-81**
	Std	7.73e-43	6.85e-70	2.69e-50	5.47e-78	1.28e-80
f_{12}	Mean	**-3.00e+00**	**-3.00e+00**	-2.66e+00	**-3.00e+00**	**-3.00e+00**
	Std	3.97e-16	3.97e-16	2.48e-01	0.00e+00	0.00e+00
f_{13}	Mean	4.95e-04	5.37e-04	9.39e-04	3.75e-04	**3.08e-04**
	Std	4.89e-05	2.62e-04	2.16e-04	3.75e-04	4.20e-20
f_{14}	Mean	9.21e-02	8.65e-02	2.88e+00	2.52e-02	**0.00e+00**
	Std	9.09e-02	7.32e-02	3.16e+00	1.48e-02	0.00e+00
f_{15}	Mean	1.16e-18	1.46e-17	7.94e-07	1.90e-15	**1.35e-288**
	Std	1.05e-18	1.36e-17	1.76e-06	2.75e-15	0.00e+00
f_{16}	Mean	**3.00e+00**	**3.00e+00**	**3.00e+00**	**3.00e+00**	**3.00e+00**
	Std	1.58e-15	8.88e-16	5.81e-06	3.97e-16	9.52e-16
Friedman	Ranking	3.72	3.22	4.13	2.41	1.53
Wilcoxon	R+	78	55	136	58	-
	R-	0	0	0	8	-
	P-value	2.01e-03	4.64e-03	3.05e-05	2.33e-02	-

Convergence speed is a crucial indicator to assess the performance of the algorithm. Hence, in order to more directly reflect the advantages and disadvantages of the convergence speed of the algorithm proposed in this paper, the convergence curve of some test functions is depicted in Fig.1. In Fig.1,

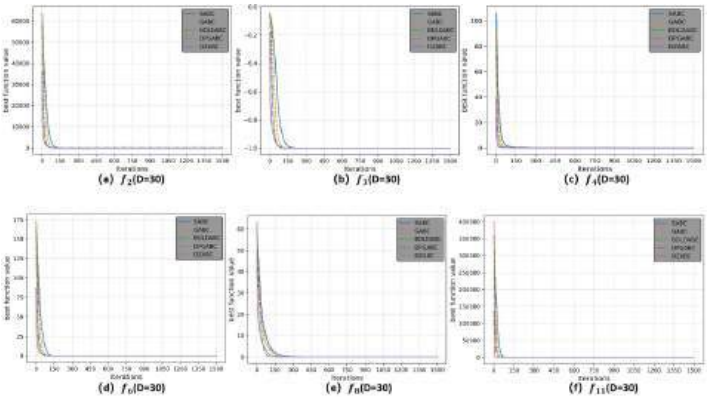


Fig. 1. Partial test function convergence curve.

(a) - (c) is uni-modal and (d) - (e) is multi-modal. From the steepness of the curve in the convergence diagram, it is evident that DZABC exhibits a faster convergence speed and better overall convergence performance for most comparison ABCs.

In summary, compared to other ABC algorithms, the DZABC algorithm in this paper converges rapidly and can approach the global optimal value early in the iteration process. With the increase in the number of iterations, the algorithm's optimization accuracy improves progressively, enabling it to reach the global optimal value in many functions. Compared with other algorithms compared in this study, the proposed algorithm demonstrates superior global and local search capabilities, and reaches the balance to a certain extent, showing strong competitiveness.

IV. CONCLUSION

To enhance the search performance of artificial bee colony algorithm in complex optimization problems, a multi-subpopulation artificial bee colony algorithm based on individual classification, called DZABC, was proposed. According to the individual rank, DZABC divided the whole population into elite sub-population, medium sub-population and common sub-population. At the same time, the three sub-populations adopt search strategies with different properties, and the individuals in the sub-population can display diverse search behaviors to achieve the balance between exploration and exploitation. At the same time, in the scout bee phase, considering that the individuals whose parameter value reaches limit will be re-initialized, there may be some cases that the individuals themselves are relatively excellent and are abandoned because they have not been updated for many times, a judgment mechanism is set up to enhance the algorithm's search efficiency.

Compared with the experimental outcomes of other improved ABC algorithms, DZABC's performance on multi-modal functions notably surpasses that of other improved ABC algorithms, which shows that DZABC has good competitiveness in dealing with complex optimization problems. Although DZABC also shows some advantages over other comparison algorithms in uni-modal functions, DZABC's advantages in unimodal functions are not as prominent as those in multi-modal functions because DZABC requires allocating certain computational resources for exploration and to sustain population diversity. Hence, how to allocate computing resources more effectively within the multi-population mechanism to further improve the performance of the algorithm on diverse types of functions is a problem worthy of further study.

REFERENCES

[1] Duan H, Luo Q. New progresses in swarm intelligence–based computation[J]. International Journal of Bio-Inspired Computation, 2015, 7(1): 26-35.

[2] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[3] Li H, Li X, Gao L. A discrete artificial bee colony algorithm for the distributed heterogeneous no-wait flowshop scheduling problem[J]. Applied soft computing, 2021, 100: 106946.

[4] Chen X, Xu B, Mei C, Ding Y, Li K. Teaching–learning–based artificial bee colony for solar photovoltaic parameter estimation[J]. Applied energy, 2018, 212: 1578-1588.

[5] Öztürk Ş, Ahmad R, Akhtar N. Variants of Artificial Bee Colony algorithm and its applications in medical image processing[J]. Applied soft computing, 2020, 97: 106799.

[6] Zhu G, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. Applied mathematics and computation, 2010, 217(7): 3166-3173.

[7] Nseef S K, Abdullah S, Turky A, Kendall G. An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems[J]. Knowledge-based systems, 2016, 104: 14-23.

[8] Wang Y, Jiao J, Liu J, Xiao R. A labor division artificial bee colony algorithm based on behavioral development[J]. Information Sciences, 2022, 606: 152-172.

[9] Kang Y, Yu H, Kang L, Qiao G, Guo D, Zeng J. A multi-role steered artificial bee colony algorithm with variable dimensionality perturbation for multimodal optimization problems[J]. Memetic Computing, 2024: 1-20.

[10] Ma H, Shen S, Yu M, Yang Z, Fei M, Zhou H. Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey[J]. Swarm and evolutionary computation, 2019, 44: 365-387.

[11] Ben Djaballah C, Nouibat W. A new multi-population artificial bee algorithm based on global and local optima for numerical optimization[J]. Cluster Computing, 2022, 25(3): 2037-2059.

[12] Cheng S, Shi Y, Qin Q, Zhang Q,Bai R. Population diversity maintenance in brain storm optimization algorithm[J]. Journal of Artificial Intelligence and Soft Computing Research, 2014, 4(2): 83-97.

[13] Zhou X, Wu Y, Zhong M, Wang M. Artificial bee colony algorithm based on multiple neighborhood topologies[J]. Applied Soft Computing, 2021, 111: 107697.

[14] Zhou X, Lu J, Huang J, Zhong M, Wang M. Enhancing artificial bee colony algorithm with multi-elite guidance[J]. Information Sciences, 2021, 543: 242-258.

[15] Singh A, Deep K. Exploration–exploitation balance in Artificial Bee Colony algorithm: a critical analysis[J]. Soft Computing, 2019, 23: 9525-9536.

[16] Guo Z, Li H, Li K. Dual subpopulation artificial bee colony algorithm based on individual gradation[J]. Egyptian Informatics Journal, 2024, 25: 100452.

[17] Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm and Evolutionary Computation, 2011, 1(1): 3-18.

Classification of liver tumors based on YOLOv8s -cls

Huili Mo
International Research Centre for Nano Handing and Manufacturing of China, Changchun University of Science and Technology
Changchun, China
1097487947@qq.com

Yingmin Qu*
International Research Centre for Nano Handing and Manufacturing of China, Changchun University of Science and Technology
Changchun, China
quy@cust.edu.cn

Xiaodong Li
International Research Centre for Nano Handing and Manufacturing of China, Changchun University of Science and Technology
Changchun, China
2903855525@qq.com

Zuobin Wang
International Research Centre for Nano Handing and Manufacturing of China, Changchun University of Science and Technology
Zhongshan Institute of Changchun University of Science and Technology
JR3CN&IRAC, University of Bedfordshire
Changchun, China
Zhongshan, China
Luton, UK
wangz@cust.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899581

***Abstract*—As liver cancer incidence and mortality rates continue to increase, the early diagnosis and treatment of liver tumors are essential for effective management and accurate classification of the disease. However, distinguishing between early and late-stage liver tumors remains challenging. To tackle the issues of low accuracy and slow classification speed in liver tumor identification, this study introduces the YOLOv8s-cls model, which is both lightweight and fast, for classifying benign and malignant tumors. Evaluation results indicate that this model demonstrates impressive classification capabilities, achieving an average accuracy of 94.3%, a precision of 95.4%, a recall of 96.1%, and an F1-score of 94.6%. When compared to ResNet-50, MobileNet V3, and ShuffleNet V2, this model excels in both accuracy and speed. Thus, it presents a dependable approach for classifying benign and malignant liver tumors.**

Keywords*—Tumor classification, Neural networks, Image processing

I. INTRODUCTION

As an important metabolic organ of human body, the liver is one of the common sites of human cancer diseases [1] According to the data released by the International Agency for Research on Cancer, the incidence rate of liver cancer ranks sixth and fifth in the world and China, respectively. The number of deaths caused by liver cancer in China is about 400000 every year, ranking second in the world [2]. Early accurate classification of liver tumors is crucial for developing appropriate treatment strategies, improving patient prognosis and increasing patient survival rates.

The traditional classification methods for liver tumors mainly rely on histopathological examination, which is the gold standard for diagnosis. However, this method is invasive and limited by sampling errors, patient compliance, and other factors in practical operation. With the development of medical imaging technology, non-invasive or minimally invasive examination methods such as computed tomography (CT) and magnetic resonance imaging (MRI) are playing an increasingly important role in the detection and preliminary classification of liver tumors. At the same time, research on liver tumor classification using machine learning and deep learning algorithms has been gradually increasing based on these imaging data and clinical information. These methods have the potential to open up new paths for the accurate classification of liver tumors and improve diagnostic efficiency.

Machine learning methods mainly include Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree and other methods. Wibowo [3] applied the LR model to liver cancer cell classification and confirmed that using genetic algorithms for feature selection has an enhancing effect on the LR model. Lee [4] applied the SVM model to lesion detection and classification in liver CT images using a hierarchical approach, which has since been widely applied. Sadeque [5] developed a method with two stages of operation. The initial stage conducts detection operation, and the subsequent stage carries out classification operation. And this method has been applied in the detection and classification tasks of liver tumor CT images. In this method, the SVM model acts as a classifier, which is used to determine whether there is a tumor in the region of interest of the image and to judge the benign or malignant nature of the tumor. Aravinda [6] acquired liver images through an adaptive region growing algorithm and segmented the liver tumor portions using the SLIC method to produce tumor images. Texture features were then extracted using the ACHLAC and Legendre methods. By applying classification counting for classification, they achieved a classification accuracy of up to 94%. Rajpoot [7] achieved high classification accuracy in the classification of normal and malignant colon tissue cells using a combination of Gaussian kernel functions and optimal parameter selection in support vector machines. Qureshi [8] proposed a subtype classification method for meningiomas using a combination of macroscopic and microscopic texture features, combining two different texture features for classification. Although machine learning methods have certain

*Corresponding author: Yingmin Qu
E-mail address: quy@cust.edu.cn

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

advantages in terms of feature extraction and performance compared to manual methods, their performance will be limited when faced with unbalanced classification in large-scale big data.

In recent years, deep learning methods have been widely used in liver cancer image classification tasks. In the early classification work, deep learning classification models were usually applied in the task of lesion detection and localization. First, image blocks are extracted and then assessed for the presence of lesions. Frid-Adar [9] designed a dual-branch convolutional neural network model that utilizes two scales of patches (image blocks) to extract features and discriminate the presence of lesions. Loukas [10] also used a similar method to detect the blood vessels distribution in the gallbladder wall in the endoscopic images. Subsequently, lesion detection tasks gradually replaced patch traversal and classification methods with end-to-end object detection models[11-12].

Balagourouchetty [14] used a GoogLeNet network model based on the Leaky ReLU activation function to extract features and then classified liver lesions using three FCNets (Fully Convolutional Networks). Afterwards, the attention mechanism demonstrated good performance in image classification tasks. Balasubramanian [15] combined Mask R-CNN and Swin Transformer for liver tumor classification in CT images. Specifically, Mask R-CNN is first used to detect tumor regions in CT images and the detected regions are used as the input to the next stage of the Swin Transformer, finally achieving classification of liver tumors in the target area. The Transformer based model mentioned above outperforms the CNN model in terms of performance. Romero [16] combined residual connections with pre-trained InceptionV3 to distinguish benign and malignant tumors in the task of liver CT image classification. Chen [17] introduced the attention mechanisms to make the neural networks pay more attention to the tumor regions, so as to extract higher quality features to improve the network performance. Braatz [18]used reinforcement learning strategies to enlarge images, so as to achieve effective detection of regions of interest, and finally applied regression supervised learning for classification. Liang [19] proposed a deep learning-based framework for classifying focal liver lesions in multi-phase CT images in reference. This framework is constructed by integrating residual networks (ResNet) with global and local paths, as well as bidirectional long short-term memory models. In this framework, feature-level multi-phase data information is fully utilized to achieve accurate classification of liver lesions.

While deep learning has made progress in the area of medical imaging, the large size of medical images and the variability in tumor sizes significantly affect the computational speed and accuracy of many models, leading to less than optimal classification outcomes. To effectively address these issues, this paper innovatively proposes the use of the YOLOv8s-cls lightweight and fast model for the classification and recognition of tumor malignancy. This model offers several benefits. Firstly, as a lightweight model, it operates quickly and can accurately classify liver tumors in a short amount of time, thereby enhancing diagnostic efficiency and providing critical treatment time for patients. Secondly, performance comparisons with traditional classification models demonstrate that this model excels in both accuracy and speed.

II. Materials and Methods

This section introduces the data processing before tumor classification and the algorithm framework used for classification, as shown in Fig. 1. We will transfer the collected data images to the preprocessing section in section A and perform data augmentation on them in section B. Then the classification model is presented in section C. Finally, the evaluation metrics of the model are described in section D.

A. Image acquisition and preprocessing

- *Kaggle datasets*

The classification images of liver tumors are mainly sourced from the Kaggle website, which contains 102 benign tumor images, 34 malignant tumor images, and 30 tumor free images. The format of the images is PNG has three channels, but the image size varies.

- *Datas preprocessing*

Due to the differences in the image sizes within the dataset, we adjusted the size to 512 × 512 pixels and applied Non-Local Means denoising to the adjusted images. In order to eliminate noise throughout the entire image improve image quality and facilitate better tumor classification by subsequent models.

B. Datas augmentation

We divided the processed images into a training set and a testing set in the ratio of 8:2, and the test set was divided into the verification set according to 1:1 ratio. After partitioning the dataset, we noticed that the small amount of trainable data may lead to poor generalization ability, low stability and inaccurate evaluation of the model. To avoid these issues, we performed enhancement operations such as horizontal flipping and random rotation on the processed images. In the process of data augmentation, simultaneous use can better increase the diversity of data, so that the model can learn important features in the image.

C. Classification Model

In our research process, we used the YOLOv8s-cls framework to classify liver tumors, which was downloaded from the Ultralytics website and trained on the Kaggle enhanced datasets. When training the model, we used the Adam trainer with a momentum parameter set to 0.937, learning rate set to 0.001, a batch size of 4 and trained for 300 epochs as a whole. The YOLOv8s-cls algorithm framework has five models of different sizes to choose. Based on the limitations of the analysis experimental equipment and the model characteristics, the YOLOv8s-cls classification model was selected in this experiment. This model has the advantages of high speed, small parameter counts and good performance. The experimental equipment and conditions are shown in Tab. 1. The algorithm framework diagram is shown in Fig. 1.

TABLE I. Experimental Equipment and Conditions

name	***parameters***
operating system	Windows 10
CPU	16 vCPU Intel(R) Xeon(R) Gold 6430

name	*parameters*
GPU	NVIDIA GeForce RTX 4090
Memory	120GB
Learning Framework	Pytorch
Development Tool	PyCharm

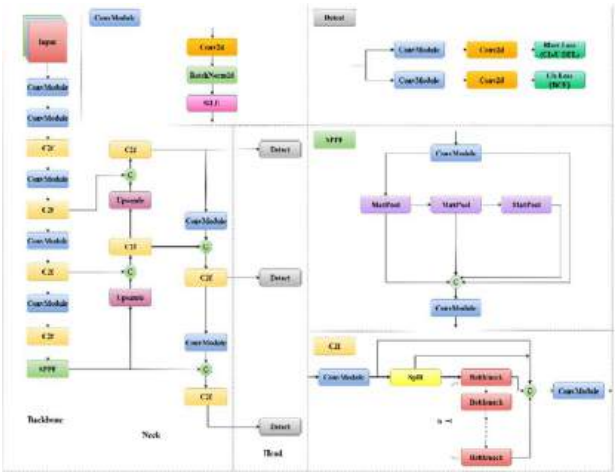


Fig. 1. The YOLOv8s-cls algorithm framework diagram.

D. Evaluation indicators

Accuracy can determine the ability of the classifier to diagnose correctly and is an important indicator for evaluating a model. Accuracy refers to the proportion of samples correctly classified by a classifier to the total number of samples. Therefore, this article selects accuracy as one of the evaluation indicators and its formula is shown (1).

$$Accuracy = (TP+FN)/(TP+TN+FP+FN) \quad (1)$$

In addition, three popular and effective metrics in the model, Precision, Recall and F1-Score, were selected to analyze and measure the performance of the classification model. The indicator formula is shown follow.

Prescription refers to the proportion of true cases among the samples judged as positive cases. It measures the accuracy of the classifier's predictions as positive examples. The formula is shown (2)

$$Precision = TP/(TP+FP) \quad (2)$$

Recall refers to the proportion of true cases correctly identified as positive by the classifier. It measures how many positive examples the classifier can correctly identify.The formula is shown (3)

$$Recall = TP/(TP+FP) \quad (3)$$

F1-Score is a comprehensive evaluation metric, which is the harmonic average of precision and recall. It is used to assess the overall performance of classification models, with the formula presented in equation (4).

$$F1\text{-}Score = 2TP/(TP+FP+FN) \quad (4)$$

Where *TP* represents the number of predicted positive results that are actually positive. *FP* represents the number of predicted results that are positive but actually negative. *FN* denotes the number of individuals who are predicted to be negative but actually positive. *TN* represents the number of predicted negative results that are actually negative.

III. Results and Discussion

A. Comparison of speed and parameters

The experimental results indicate that the model used in this article is effective for the classification of liver tumors. Under the same parameter settings and conditions, it was compared with the previous classical classification studies and the specific comparison results are shown in Tab.2.

TABLE II. Comparison of Relevant Parameters

Models	*Parameters*	*Time training*	*Accuracy*
Resnet-50	25.6M	3.5(H)	85.2%
MobileNet V3	5.0M	1.9 (H)	87.4%
ShuffleNet V2	1.36M	2.1(H)	83.2%
YOLOv8s-cls	6.4M	2.3 (H)	94.3%

The results indicate that our model performs well in terms of speed, complexity and accuracy. Compared with the traditional classification models, the model used in this article has fast speed, lightweight structure and small storage space requirements. And its accuracy reached 94.3%.

B. Analysis of evaluation indicators

In the section, we use the evaluation metrics mentioned in Section D of Part 2 to validate the effectiveness of the model used for tumor classification. The comparison results of its evaluation indicators are shown in Fig. 2.

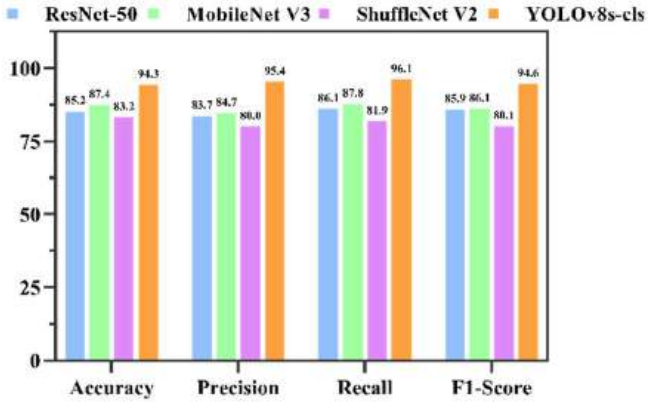


Fig. 2. Comparison of Evaluation Indicators.

According to Fig. 2, it can be seen that the model has achieved excellent results in the evaluation indicators used in the article, and showed outstanding performance in the classification of liver tumors, thus demonstrating the effectiveness of this model for tumor classification.

C. *Visualization of Results*

In order to display the classification results of the tumor using the model more intuitively, we present the classification results of it and other classification models in the form of images, as shown in Fig. 3.

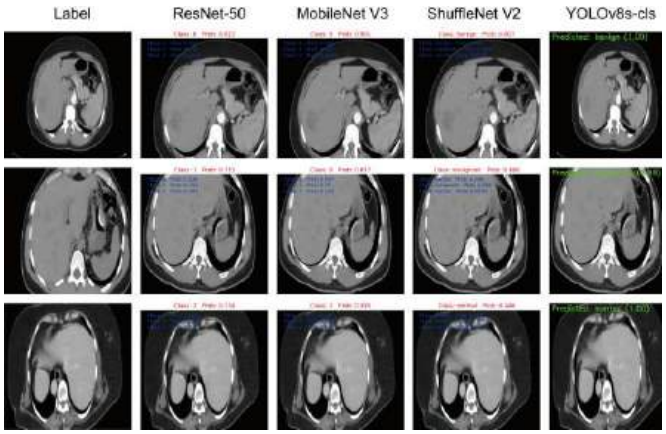


Fig. 3. Visualization of Classification Results.

In Fig. 3, the leftmost column represents the gold standard for tumor classification, the second column displays the classification results from ResNet-50, the third column shows the results from MobileNet V3, the fourth column presents the classification results from ShuffleNet V2, and the final column represents the classification results generated by the model used in this study.

In the picture, "0"represents benign tumors, "1" represents malignant tumors and "2" denotes no tumor. From Fig. 3, we can see more clearly that the model used in this article has the best recognition performance for any type of tumor. Although some classification models have higher accuracy, the overall classification and recognition performance is not as good as the model used in this article.

IV. CONCLUSIONS

The categorical attributes of liver tumors are not only of great significance in the diagnosis and treatment of liver cancer, but also have a profound impact on patients. Therefore, accurate classification can provide strong support for the diagnosis and treatment plan formulation of tumors.

In order to better solve the problem of low accuracy and slow speed in tumor category recognition, this paper selected the YOLOv8s-cls model to classify liver tumors by comparing it with different models under the same conditions. The results show that the model selected in this article performs well in both speed and accuracy compared to other models, achieving better classification performance.

ACKNOWLEDGMENT

This study was supported by Jilin provincial education department program (JJKH20220731KJ), Jilin Provincial Science and Technology Development Program (No. 20240404066ZP), Jilin Provincial Education Program (No. JJKH20240933KJ), EU H2020 Program (MNR4SCELL No. 734174), National Natural Science Foundation Program of China (No. 62175020) and "111" Project of China (D17017).

REFERENCES

[1] Al Sadeque Z, Khan T I, Hossain Q D, Turaba M Y. Automated detection and classification of liver cancer from CT images using HOG-SVM model[C]//2019 5th International Conference on Advances in Electrical Engineering (ICAEE). IEEE, 2019: 21-26.

[2] Sung H, Ferlay J, Siegel R L, Laversanne M, Soerjomataram I, Jemal A, et al. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries[J]. CA: a cancer journal for clinicians, 2021, 71(3): 209-249.

[3] Wibowo V V P, Rustam Z, Laeli A R, Sa'id A A. Logistic regression and logistic regression-genetic algorithm for classification of liver cancer data[C]//2021 International Conference on Decision Aid Sciences and Application (DASA). IEEE, 2021: 244-248.

[4] Lee C C, Chen S H, Chiang Y C. Classification of liver disease from CT images using a support vector machine[J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2007, 11(4): 396-402.

[5] Al Sadeque Z, Khan T I, Hossain Q D, Turaba M Y. Automated detection and classification of liver cancer from CT images using HOG-SVM model[C]//2019 5th International Conference on Advances in Electrical Engineering (ICAEE). IEEE, 2019: 21-26.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[6] Aravinda H L, Sudhamani M V. Liver tumour classification using average correction higher order local autocorrelation coefficient and legendre moments[J]. International Journal of Engineering & Technology, 2018, 7(2.6): 306-310.

[7] Rajpoot K, Rajpoot N. SVM optimization for hyperspectral colon tissue cell classification[C]//International Conference on Medical Image Computing and Computer-Assisted Intervention. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004: 829-837.

[8] Qureshi H, Sertel O, Rajpoot N, Wilson R, Gurcanet M. Adaptive discriminant wavelet packet transform and local binary patterns for meningioma subtype classification[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008: 11th International Conference, New York, NY, USA, September 6-10, 2008, Proceedings, Part II 11. Springer Berlin Heidelberg, 2008: 196-204.

[9] Frid-Adar M, Diamant I, Klang E, Amitai M M, Goldberger J, Greenspanet H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification[J]. Neurocomputing, 2018, 321: 321-331.

[10] Loukas C, Frountzas M, Schizas D. Patch-based classification of gallbladder wall vascularity from laparoscopic images using deep learning[J]. International Journal of Computer Assisted Radiology and Surgery, 2021, 16: 103-113.

[11] Mulay S, Deepika G, Jeevakala S, Sivaprakasam M. Liver segmentation from multimodal images using HED-mask R-CNN[C]//Multiscale Multimodal Medical Imaging: First International Workshop, MMMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1. Springer International Publishing, 2020: 68-75.

[12] Hasegawa R, Iwamoto Y, Han X , Lin L, Hu H, Cai X,et al. Automatic detection and segmentation of liver tumors in multi-phase CT images by phase attention mask R-CNN[C]//2021 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2021: 1-5.

[13] Yao Y, Chen Y, Gou S,Chen S, Zhang X, Tong N. Auto-segmentation of pancreatic tumor in multi-modal image using transferred DSMask R-CNN network[J]. Biomedical Signal Processing and Control, 2023, 83: 104583.

[14] Balagourouchetty L, Pragatheeswaran J K, Pottakkat B, Ramkumar G. GoogLeNet-based ensemble FCNet classifier for focal liver lesion diagnosis[J]. IEEE journal of biomedical and health informatics, 2019, 24(6): 1686-1694.

[15] Balasubramanian P K, Lai W C, Seng G H, Kavitha C, Selvaraj J. Apestnet with mask r-cnn for liver tumor segmentation and classification[J]. Cancers, 2023, 15(2): 330.

[16] Romero F P, Diler A, Bisson-Gregoire G,Turcotte S, Lapointe R, Menu F V,et al. End-to-end discriminative deep network for liver lesion classification[C]//2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019). IEEE, 2019: 1243-1246.

[17] Chen X, Lin L, Liang D, Hu H, Zhang Q, Iwamoto Y,et al. A dual-attention dilated residual network for liver lesion classification and localization on CT images[C]//2019 IEEE international conference on image processing (ICIP). IEEE, 2019: 235-239.

[18] Aguiar R, Braatz J. Selecting regions of interest in large multi-scale images for cancer pathology[J]. arXiv preprint arXiv:2007.01866, 2020.

[19] Liang D, Lin L, Hu H, Zhang Q W,Chen Q Q, Iwamoto Y,et al. Combining convolutional and recurrent neural networks for classification of focal liver lesions in multi-phase CT images[C]//Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11. Springer International Publishing, 2018: 666-675.

Research on α-Arbitrage for Uncertain Stock Market Model

Jun Zhao*
School of Science, Xi'an University of Posts and Telecommunications
Xi'an, China
junzhao@xupt.edu.cn
*Corresponding author

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899572

***Abstract*—This paper introduces the concept of α-arbitrage opportunity based on the uncertain stock model, where the stock price follows an uncertain differential equation driven by the canonical Liu process rather than the traditional Brownian motion. Mainly, a sufficient and necessary condition for the uncertain stock model being strict no-arbitrage, i.e. absent of α-arbitrage opportunity is derived. The uncertain stock model is strict no-arbitrage if and only if the system of linear equations has a solution and the absolute value of the solution does not exceed the given threshold, which depends on the investors' belief degree. Some examples are given to illustrate the strict no-arbitrage theorem, especially to show that an α-arbitrage opportunity is possible in a no-arbitrage market. The numerical analysis shows that the market can be strict no-arbitrage when the belief degree is 99% except for five months and 98% except for seven months. Moreover, the optimal belief degree and the α-arbitrage boundary curve are obtained in the numerical analysis.**

***Index Terms*—α-arbitrage, belief degree, uncertain stock model, strict no-arbitrage**

I. Introduction

Stochastic finance has been playing a critical role since stochastic analysis was introduced to financial research [12]. The famous Black-Scholes stock model [3] was proposed by assuming that the stock price follows a geometric Wiener process.

In fact, the prices of some stocks in the real market may not exhibit randomness, despite the wide application of stochastic finance. The fuzzy Black-Scholes formula [15], [16] via the fuzzy set theory was derived. Both the randomness and the fuzziness were considered [20], the price of financial derivative instruments was modelled via fuzzy stochastic processes. Except for randomness and fuzziness, human uncertainty may be another factor affecting the stock price in financial markets. In order to model the human uncertainty, the uncertainty theory was founded [9].

Uncertainty theory has been fully developed since its creation. In particular, Liu [10] assumed that the stock price follows a geometric canonical process instead of geometric Wiener process, and proposed an uncertain stock model. The problem of option pricing in a mean-reverting uncertain stock market model was studied [13]. An American option pricing formula for Liu's stock model was derived [5]. The problem of currency option pricing was studied under an uncertain currency model [11]. The lookback option pricing in the uncertain exponential ornstein-uhlenbeck model was studied [7]. Despite the relatively late starting, uncertain financial research has become an important branch of mathematical finance.

Instead of exploring the pricing formula of various derivative securities, Yao [19] derived a sufficient and necessary condition for Liu's uncertain stock model being no-arbitrage. And Yao [18] proposed a multi-factor stock model with mean-reverting process and studied the corresponding no-arbitrage theorem. Assuming the floating interest rate in a multi-factor uncertain stock model, a sufficient and necessary condition for this stock model being no-arbitrage was presented [8].

However, it is always inadequate to merely exclude the traditional arbitrage opportunity from a practical point of view. For example, when valuing options in incomplete markets, the no-arbitrage bounds of the option prices are typically very wide, that is, the pricing implications are not very precise and give no useful information. It seems necessary to study some stricter no-arbitrage principles, i.e., to rule out some weaker arbitrage opportunities, which may provide more precise guidance to the realistic financial problems, such as hedging, pricing, portfolio choice, equilibrium and optimal reinsurance.

It was proposed to rule out not only pure arbitrage opportunities but also "good deals", which are investment opportunities with high Sharpe ratios [6]. Approximate arbitrage opportunities were ruled out, which are investment opportunities offering high gain-loss ratios, where gain (loss) is the expectation of the positive (negative) part of the excess payoff computed under a benchmark risk-neutral measure [2]. The concept of statistical arbitrage was introduced, that is a zero-cost trading strategy for which the expected payoff is positive and the conditional expected payoff in each final state of the economy is nonnegative [4]. Recently, the concept of ρ-arbitrage for a risk measure ρ was suggested [1]. They are portfolios which give a potentially positive return for a non-positive price without incurring a positive risk as measured

This work was supported by the Natural Science Basic Research Program of Shaanxi Province, China under grant number 2022JQ-071 and the Natural Science Foundation of Shaanxi Province, China under grant number 2024JC-YBMS-008.

using ρ.

This paper proposed the concept of α-arbitrage opportunity with the belief degree α based on Liu's uncertain stock model. It is a portfolio satisfying that the uncertain discounted gain can be negative (from this point of view it is considered to be risky) with a belief degree less than $1-\alpha$. In fact, α-arbitrage is a weaker concept than the classical arbitrage opportunity, while the latter offers the possibility of a gain with no possibility of a loss [8], [18], [19]. The uncertain stock model is said to be strict no-arbitrage if there is no α-arbitrage opportunity. Obviously, it must be no-arbitrage if the uncertain stock model is strict no-arbitrage, while the opposite is not necessarily true. That is to say, the traditional no-arbitrage principle is weaker than the absence of α-arbitrage opportunity. This is also the inspiration of the term "strict no-arbitrage" in this paper.

The rest of this paper is organized as follows. Section 2 gives the definition of α-arbitrage opportunity in the multi-factor uncertain stock model. Section 3 derives a sufficient and necessary condition for the uncertain stock model being strict no-arbitrage, i.e. absent of α-arbitrage opportunity. In Section 4, the strict no-arbitrage conditions are discussed in the real market by some numerical examples. At last, the paper is concluded in Section 5.

II. Model and definitions

The basic definitions and useful results in uncertain theory, such as uncertain variable, uncertainty distribution, critical values and uncertain differential equation can be found in [9], [10].

Let $(\Gamma, \mathcal{L}, \mathcal{M})$ be an uncertainty space. The expected value of uncertain variable ξ is denoted as $\mathbb{E}[\xi]$. Recall the multi-factor uncertain stock model [10], where the prices of stocks are determined by multiple canonical processes. In detail, the market consists of one bond and m stocks, in which the bond price B_t and the stocks prices S_{it} are determined by

$$\begin{cases} dB_t = rB_t dt, \\ dS_{it} = \mu_i S_{it} dt + \sum_{j=1}^{n} \sigma_{ij} S_{it} dC_{jt}, \ i = 1, 2, \cdots, m, \end{cases} \tag{1}$$

where r is the risk-free rate, μ_i are the stock drift coefficients, σ_{ij} are the stock diffusion coefficients, and C_{jt} are independent canonical Liu processes, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n$.

Let $\beta_t = (\beta_{0t}, \beta_{1t}, \cdots, \beta_{mt})$ be a portfolio, where the investment fractions meet $\beta_{0t}+\beta_{1t}+\cdots+\beta_{mt} = 1$. Let W_t be the wealth at time t, which follows the uncertain differential equation

$$dW_t = r\beta_{0t} W_t dt + \sum_{i=1}^{m} \mu_i \beta_{it} W_t dt + \sum_{i=1}^{m}\sum_{j=1}^{n} \sigma_{ij} \beta_{it} W_t dC_{jt}. \tag{2}$$

Recall from [10] that the stock model (1) is no-arbitrage if there is no portfolio $\beta_t = (\beta_{0t}, \beta_{1t}, \cdots, \beta_{mt})$ such that

$$\mathcal{M}\{e^{-rs} W_s \geq W_0\} = 1 \tag{3}$$

and

$$\mathcal{M}\{e^{-rs} W_s > W_0\} > 0 \tag{4}$$

for some time s. Next, we introduce the concept of α-arbitrage.

Definition 1. *A portfolio $\beta_t = (\beta_{0t}, \beta_{1t}, \cdots, \beta_{mt})$ is an α-arbitrage opportunity if*

$$\mathcal{M}\{e^{-rs} W_s < W_0\} < 1 - \alpha \tag{5}$$

for some time s, where $\alpha \in (\frac{1}{2}, 1)$ represents the belief degree.

It can be observed that the stock model (1) must be no-arbitrage if there is no α-arbitrage opportunity defined by (5). Otherwise, if the portfolio β_t is a traditional arbitrage opportunity, then it must also be an α-arbitrage. Indeed, it is trivial to see from the condition (3) that

$$\mathcal{M}\{e^{-rs} W_s < W_0\} = 0 < 1 - \alpha,$$

where $\alpha \in (\frac{1}{2}, 1)$. Thus a stricter no-arbitrage condition can be defined by ruling out the α-arbitrage in the stock model (1).

Definition 2. *The stock model (1) is said to be strict no-arbitrage if there is no α-arbitrage opportunity.*

III. Strict no-arbitrage theorem

The uncertain differential equation (2) has a solution W_t, which can be written as

$$e^{rt} W_0 \exp\left(\int_0^t \sum_{i=1}^{m} (\mu_i - r)\beta_{is} ds + \sum_{j=1}^{n} \int_0^t \sum_{i=1}^{m} \sigma_{ij} \beta_{is} dC_{js} \right).$$

Let

$$\xi^t := \ln(e^{-rt} W_t) - \ln(W_0),$$

then ξ^t is a normal uncertain variable with expected value

$$\int_0^t \sum_{i=1}^{m} (\mu_i - r)\beta_{is} ds$$

and variance

$$\left(\sum_{j=1}^{n} \int_0^t \left| \sum_{i=1}^{m} \sigma_{ij} \beta_{is} \right| ds \right)^2.$$

That is,

$$\xi^t \sim \mathcal{N}(e_t, \sigma_t),$$

where

$$e_t := \int_0^t \sum_{i=1}^{m} (\mu_i - r)\beta_{is} ds$$

and

$$\sigma_t := \sum_{j=1}^{n} \int_0^t \left| \sum_{i=1}^{m} \sigma_{ij} \beta_{is} \right| ds.$$

Let Φ_t be the uncertainty distribution of ξ^t. As above, it is a normal uncertainty distribution with

$$\Phi_t(x) = \left(1 + \exp\left(\frac{\pi(e_t - x)}{\sqrt{3}\sigma_t} \right) \right)^{-1}, \quad x \in \mathbb{R},$$

where e_t and σ_t are real numbers with $\sigma_t > 0$. Furthermore, the inverse uncertainty distribution of ξ^t can be expressed as

$$\Phi_t^{-1}(\alpha) = e_t + \frac{\sigma_t\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}.$$

Lemma 1. *A portfolio $\beta_t = (\beta_{0t}, \beta_{1t}, \cdots, \beta_{mt})$ is an α-arbitrage opportunity if and only if*

$$\xi^s_{\sup}(\alpha) > 0$$

for some time s, where $\xi^s_{\sup}(\alpha)$ is the α-optimistic value of ξ^s.

Proof. $\Leftarrow$ Assume that $\xi^s_{\sup}(\alpha) > 0$ for some time s. It can be known that $\xi^s_{\sup}(\alpha) = \Phi_s^{-1}(1-\alpha)$, where Φ_s^{-1} is the inverse uncertainty distribution of ξ^s. Then $\xi^s_{\sup}(\alpha) > 0$ implies that $\Phi_s^{-1}(1-\alpha) > 0$. So that it may be obtained that $\Phi_s(0) < 1-\alpha$ as the function Φ_s is strictly increasing. By the definition of uncertainty distribution, it must have $\mathcal{M}\{\xi^s \le 0\} < 1-\alpha$. Therefore, from the following fact

$$\mathcal{M}\{\ln(e^{-rs}W_s) < \ln(W_0)\} = \mathcal{M}\{\xi^s < 0\}, \tag{6}$$

it holds

$$\mathcal{M}\{e^{-rs}W_s < W_0\} < 1-\alpha.$$

Thus, $\beta_t = (\beta_{0t}, \beta_{1t}, \cdots, \beta_{mt})$ is an α-arbitrage opportunity.

$\Rightarrow$ Assume that a portfolio $\beta_t = (\beta_{0t}, \beta_{1t}, \cdots, \beta_{mt})$ is an α-arbitrage opportunity, it must hold that $\mathcal{M}\{e^{-rs}W_s < W_0\} < 1-\alpha$ for some time s. By (6), it holds that $\mathcal{M}\{\xi^s < 0\} < 1-\alpha$ so that $\mathcal{M}\{\xi^s \ge 0\} > \alpha$. According to Measure Inversion Theorem [9, Theorem 1.12], it can be known that $\mathcal{M}\{\xi^s \ge 0\} = 1 - \Phi_s(0)$ such that $\Phi_s(0) < 1-\alpha$. As the inverse uncertainty distribution Φ_s^{-1} is a strictly increasing function, it can be finally obtained that $\xi^s_{\sup}(\alpha) = \Phi_s^{-1}(1-\alpha) > 0$. □

Corollary 1. *If the portfolio $\beta_t = (\beta_{0t}, \beta_{1t}, \cdots, \beta_{mt})$ is an α-arbitrage opportunity, then*

$$\mathbb{E}[\exp(-rs)W_s] > W_0$$

for some time s.

Proof. From Lemma 1, it has $\xi^s_{\sup}(\alpha) > 0$ for some time s. Note that $\xi^s \sim \mathcal{N}(e_s, \sigma_s)$, then the α-optimistic value $\xi^s_{\sup}(\alpha)$ can be expressed as

$$\xi^s_{\sup}(\alpha) = e_s - \frac{\sigma_s\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha},$$

where $\alpha \in (\frac{1}{2}, 1)$. Thus,

$$\mathbb{E}[\xi^s] = e_s > \frac{\sigma_s\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \ge 0.$$

From Jensen's Inequality [9, Theorem 1.65], it is easy to deduce that

$$\ln\mathbb{E}\left[\frac{e^{-rs}W_s}{W_0}\right] \ge \mathbb{E}\left[\ln\frac{e^{-rs}W_s}{W_0}\right] = \mathbb{E}[\xi^s] > 0,$$

so that $\mathbb{E}\left[\frac{e^{-rs}W_s}{W_0}\right] > 1$. That is, $\mathbb{E}[e^{-rs}W_s] > W_0$. □

Theorem 1. *(strict no-arbitrage theorem) The model (1) is strict no-arbitrage if and only if the system of linear equations*

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m1} & \cdots & \sigma_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \mu_1 - r \\ \mu_2 - r \\ \vdots \\ \mu_m - r \end{pmatrix} \tag{7}$$

has a solution and

$$-\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \le x_j < \frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha},\ \forall\ j = 1, 2, \cdots, n, \tag{8}$$

where $\alpha \in (\frac{1}{2}, 1)$.

Proof. Assume that the system of linear equations (7) has a solution, i.e. there exist $x_1, x_2, \cdots, x_n \in \mathbb{R}$ such that

$$\mu_i - r = \sum_{j=1}^{n}\sigma_{ij}x_j,\ \forall\ i = 1, 2, \cdots, m,$$

and the condition (8) holds. Given any time t and portfolio $\beta_s = (\beta_{0s}, \beta_{1s}, \cdots, \beta_{ms})$, it can be deduced that the α-optimistic value $\xi^t_{\sup}(\alpha)$ of ξ^t cannot be positive. Actually, the arguments can be divided into two cases.
Case 1: If

$$\sigma_t = \sum_{j=1}^{n}\int_0^t \left|\sum_{i=1}^{m}\sigma_{ij}\beta_{is}\right| ds = 0,$$

then

$$\sum_{i=1}^{m}\sigma_{ij}\beta_{is} = 0,\ \forall\ j = 1, 2, \cdots, n,\ s \in (0, t).$$

It can be easily obtained that

$$\sum_{i=1}^{m}(\mu_i - r)\beta_{is} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sigma_{ij}x_j\beta_{is} = \sum_{j=1}^{n}x_j\sum_{i=1}^{m}\sigma_{ij}\beta_{is} = 0$$

so that

$$e_t = \int_0^t\sum_{i=1}^{m}(\mu_i - r)\beta_{is}ds = 0.$$

Thus,

$$\xi^t_{\sup}(\alpha) = e_t - \frac{\sigma_t\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} = 0.$$

Case 2: If

$$\sigma_t = \sum_{j=1}^{n}\int_0^t \left|\sum_{i=1}^{m}\sigma_{ij}\beta_{is}\right| ds \ne 0,$$

then it is trivial to get that

$$\xi^t_{\sup}(\alpha) = e_t - \frac{\sigma_t\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \le 0$$

when $e_t \le 0$. For the case where

$$e_t = \int_0^t\sum_{i=1}^{m}(\mu_i - r)\beta_{is}ds > 0,$$

it can be proved that

$$\xi^t_{\sup}(\alpha) = e_t - \frac{\sigma_t\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \le 0,$$

where the second inequality holds due to Jensen's Inequality. Thus, from Lemma 1, it can be known that the portfolio $\beta_s = (\beta_{0s}, \beta_{1s}, \cdots, \beta_{ms})$ cannot be an α-arbitrage opportunity. That is to say, the stock model (1) is strict no-arbitrage.

Conversely, assume that the stock model (1) is strict no-arbitrage, then it can deduced that the following system of inequalities

$$\begin{cases} y^T A_1 + z^T A_2 + w^T A_3 = 0, \\ y^T b_1 + z^T b_2 + w^T b_3 > 0, \\ z \ge 0, z \ne 0, w = 0, \\ y \in \mathbb{R}^m, z \in \mathbb{R}^n, w \in \mathbb{R}^n, \end{cases} \tag{9}$$

has no solution, where

$$A_1 = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m1} & \cdots & \sigma_{mn} \end{pmatrix}$$

$$A_2 = -A_3 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}_{n\times n}$$

$$b_1 = \begin{pmatrix} \mu_1 - r \\ \mu_2 - r \\ \vdots \\ \mu_m - r \end{pmatrix}$$

$$b_2 = b_3 = \begin{pmatrix} -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \\ -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \\ \vdots \\ -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \end{pmatrix}_{n\times 1}.$$

Indeed, if the system (9) has the solution $y = (y_1, y_2, \cdots, y_m) \in \mathbb{R}^m$, $z = (z_1, z_2, \cdots, z_n) \in \mathbb{R}^n$ and $w = (w_1, w_2, \cdots, w_n) \in \mathbb{R}^n$, then it is obvious to get that

$$\sum_{i=1}^m \sigma_{ij} y_i + z_j = 0, \ \forall\ j = 1, 2, \cdots, n$$

and

$$\sum_{i=1}^m (\mu_i - r) y_i - \frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}\sum_{j=1}^n z_j > 0.$$

As $z_j \ge 0$ for each $1 \le j \le n$, then

$$\sum_{j=1}^n \left|\sum_{i=1}^m \sigma_{ij} y_i\right| = \sum_{j=1}^n z_j.$$

By taking the portfolio $\beta_s = (\beta_{0s}, \beta_{1s}, \cdots, \beta_{ms})$ as

$$\begin{cases} \beta_{is} = y_i, \ \forall\ i = 1, 2, \cdots, m, \\ \beta_{0s} = 1 - \sum_{i=1}^m y_i, \end{cases} \tag{10}$$

at each instant s, then it holds for any $t > 0$,

$$\begin{aligned} \xi^t_{\sup}(\alpha) &= e_t - \frac{\sigma_t\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \\ &= t\left[\sum_{i=1}^m (\mu_i - r) y_i - \frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}\sum_{j=1}^n z_j\right] \\ &> 0. \end{aligned}$$

Lemma 1 implies that the above portfolio β_s defined by (10) is an α-arbitrage opportunity, which is contradicted with the assumption that the stock model (1) is strict no-arbitrage. From Kuhn-Fourier Theorem [14, Theorem 4], the system of inequalities (9) has no solution if and only if the following system of inequalities

$$\begin{cases} A_1 x = b_1, \\ A_2 x \ge b_2, \\ A_3 x > b_3, \end{cases} \tag{11}$$

has a solution, i.e. there exists $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ such that

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m1} & \cdots & \sigma_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \mu_1 - r \\ \mu_2 - r \\ \vdots \\ \mu_m - r \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \ge \begin{pmatrix} -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \\ -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \\ \vdots \\ -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \end{pmatrix}, \tag{12}$$

$$\begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} > \begin{pmatrix} -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \\ -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \\ \vdots \\ -\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \end{pmatrix}. \tag{13}$$

From (12) and (13), it has

$$-\frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha} \le x_j < \frac{\sqrt{3}}{\pi}\ln\frac{\alpha}{1-\alpha}, \ \forall\ j = 1, 2, \cdots, n.$$

Thus, it can be concluded that there exists $x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ satisfying the condition (8) such that the system of linear equations (7) holds. □

Example 1. *Consider the stock model given by*

$$\begin{cases} dB_t = B_t dt, \\ dS_{1t} = 2S_{1t}dt + S_{1t}dC_{1t} + 2S_{1t}dC_{2t}, \\ dS_{2t} = 4S_{2t}dt + 2S_{2t}dC_{1t} + 3S_{2t}dC_{2t}. \end{cases} \tag{14}$$

Note that

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}, \quad \begin{pmatrix} \mu_1 - r \\ \mu_2 - r \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix},$$

it is easy to get that the system of linear equations

$$\begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

has the only solution $x_1 = 3$ *and* $x_2 = -1$. *Thus the stock model (14) is no-arbitrage. However, it is not strict no-arbitrage if the* α*-arbitrage is considered with the belief degree* $\alpha = 98\%$, *since* $\frac{\sqrt{3}}{\pi} \ln \frac{\alpha}{1-\alpha} \approx 2.146$ *such that*

$$x_1 = 3 > \frac{\sqrt{3}}{\pi} \ln \frac{\alpha}{1-\alpha}.$$

Indeed, an α*-arbitrage opportunity may be found. For example, consider the constant portfolio* $\beta_s = (2, -3, 2)$ *for each* s. *We can obtain the expected value and variance of the uncertain variable* ξ^t:

$$e_t = 3t$$

and

$$\sigma_t^2 = t^2.$$

Thus, the α*-optimistic value is*

$$\xi_{\sup}^t(\alpha) = e_t - \frac{\sigma_t \sqrt{3}}{\pi} \ln \frac{\alpha}{1-\alpha} \approx 0.854t$$

such that $\xi_{\sup}^t(\alpha) > 0$ *for all* $t > 0$. *That is,* β_s *is an* α*-arbitrage opportunity.*

IV. Numerical analysis

In this section, we consider the case where $n = 1$, i.e., the stock prices S_{it}, $i = 1, 2, \cdots, m$ are driven by the same canonical Liu processes C_t. Then, it has the following stock model

$$\begin{cases} dB_t = rB_t dt, \\ dS_{it} = \mu_i S_{it} dt + \sigma_i S_{it} dC_t, \ i = 1, 2, \cdots, m. \end{cases} \tag{15}$$

We know from Theorem 1 that the stock model (15) is strict no-arbitrage if and only if

$$\frac{\mu_1 - r}{\sigma_1} = \frac{\mu_2 - r}{\sigma_2} = \cdots = \frac{\mu_m - r}{\sigma_m} \triangleq \zeta$$

and

$$-\frac{\sqrt{3}}{\pi} \ln \frac{\alpha}{1-\alpha} \leq \zeta < \frac{\sqrt{3}}{\pi} \ln \frac{\alpha}{1-\alpha}, \tag{16}$$

where ζ represents the market price of risk. The inequality (16) implies that the absolute value of market price of risk does not exceed the threshold $\frac{\sqrt{3}}{\pi} \ln \frac{\alpha}{1-\alpha}$, which is related to the investors' belief degree.

Next, we show by the numerical examples that the strict no-arbitrage theorem is valid in the real market. Consider the case where $m = 1$, that is the stock model given by

$$\begin{cases} dB_t = rB_t dt, \\ dS_t = \mu S_t dt + \sigma S_t dC_t. \end{cases} \tag{17}$$

Naturally, the stock model (17) is no-arbitrage. Zhongsheng Pharma(002317.XSHE), is chosen. We adopt the α-path method [17] to estimate the parameters μ and σ by the closing prices from December, 2019 to September, 2022. The risk-free rate r is chosen as the one-year reasury bond rate in that month.

The α-arbitrage opportunities are checked as Figure 1 (with the belief degree $\alpha = 99\%$) and Figure 2 (with the belief degree $\alpha = 98\%$). We can observe that the market is strict no-arbitrage when $\alpha = 99\%$, except for five months, i.e., July, August, 2020 and March, April, June, 2021. And the market is strict no-arbitrage when $\alpha = 98\%$, except for seven months, i.e., July, August, 2020, March, April, June, October, 2021 and September, 2022. Actually, the stock in these months possesses relatively large market prices of risk, which may lead to the possibility of α-arbitrage opportunities. The reason may be the impact of Covid-19 and China's prevention policy for it on the pharmaceutical industry.

In our numerical examples, the optimal belief degree $\alpha^* = 99.82\%$ can be chosen as Figure 3, which can make the stock model (17) strict no-arbitrage every month from December, 2019 to September, 2022. On the other hand, the α-arbitrage boundary curve is shown as Figure 4. In detail, the area of α-arbitrage is below the curve and the strict no-arbitrage area is above the curve.

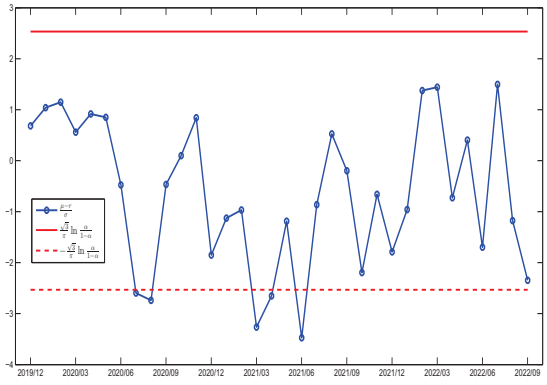


Fig. 1: α-arbitrage with $\alpha = 99\%$

V. Conclusion

Based on Liu's multi-factor uncertain stock model, this paper introduces the concept of α-arbitrage opportunity, which is a weaker arbitrage opportunity compared with the traditional arbitrage. A sufficient and necessary condition for the uncertain stock model being strict no-arbitrage, i.e. absent of α-arbitrage opportunity is derived. In detail, the multi-factor uncertain stock model is strict no-arbitrage if and only if the system of linear equations has a solution and the absolute value of the solution does not exceed the given threshold, which depends on the investors' belief degree. The numerical examples show that the strict no-arbitrage condition is valid in the real market.

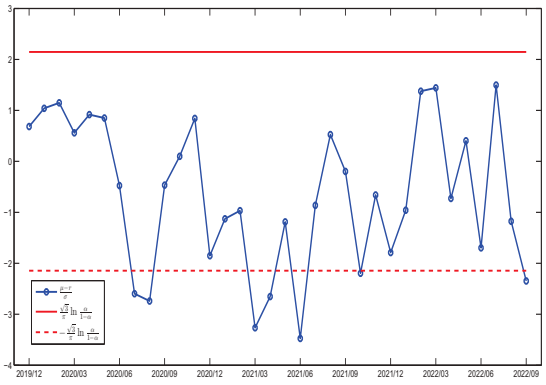


Fig. 2: α-arbitrage with $\alpha = 98\%$

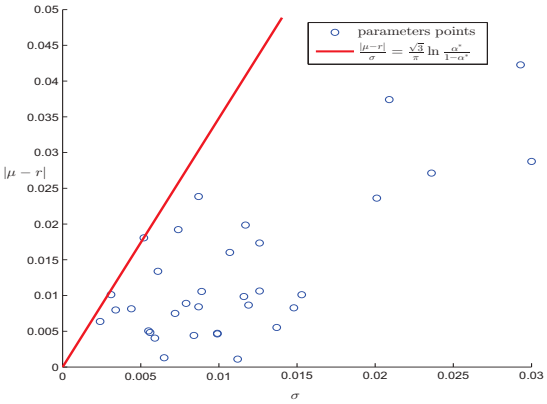


Fig. 3: Optimal belief degree α^*

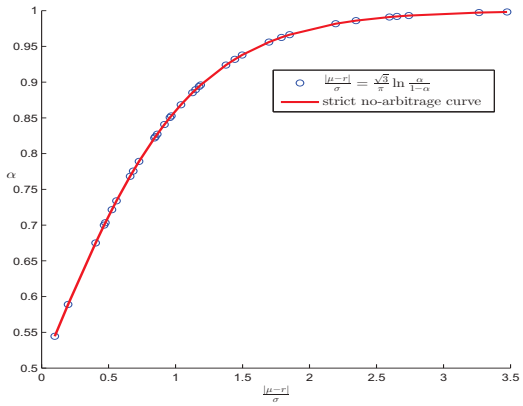


Fig. 4: α-arbitrage boundary curve

ACKNOWLEDGMENT

The author is grateful to the responsible editor and the anonymous referees for their valuable comments and suggestions.

REFERENCES

[1] J. Armstrong and D. Brigo,"Statistical arbitrage of coherent risk measures," arXiv e-prints, 2009.

[2] A. Bernardo and O. Ledoit,"Gain, loss, and asset pricing," Journal of Political Economy, vol. 108, pp. 144–172, 2000.

[3] F. Black and M. Scholes, "The pricing of options and corporate liabilities," Journal of political economy, vol. 81, pp. 637–654, 1973.

[4] O. Bondarenko, "Statistical arbitrage and securities prices," vol. 16, pp. 875–919, 2003.

[5] X. Chen, "American option pricing formula for uncertain financial market," International Journal of Operations Research, vol. 8, pp. 32–37, 2011.

[6] J. H. Cochrane and J. Saa-Requejo, "Beyond arbitrage: Good-deal asset price bounds in incomplete markets," Journal of Political Economy, vol. 108, pp. 79–119, 2000.

[7] Y. Gao, X. Yang and Z. Fu, "Lookback option pricing problem of uncertain exponential Ornstein–Uhlenbeck model," Soft Computing, vol. 22, pp. 5647–5654, 2018.

[8] X. Ji and H. Ke, "No-arbitrage theorem for multi-factor uncertain stock model with floating interest rate," Fuzzy Optimization and Decision Making, vol. 16, pp. 221–234, 2017.

[9] B. Liu, Uncertainty theory. Berlin, Heidelberg: Springer, 2007.

[10] B. Liu, "Some research problems in uncertainty theory," Journal of Uncertain systems, vol. 3, pp.3–10, 2009.

[11] Y. Liu, X. Chen and D. A. Ralescu, "Uncertain currency model and currency option pricing," International Journal of Intelligent Systems, vol. 30, pp. 40–51, 2015.

[12] R. C. Merton, "Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case," The Review of Economics and Statistics, vol. 51, pp. 247–257, 1969.

[13] J. Peng and K. Yao, "A new option pricing model for stocks in uncertainty markets," International Journal of Operations Research, vol. 8, pp.18–26, 2011.

[14] Z. Wang, "Some equivalent forms of Farkas lemma and their generalization," Acta Mathematicae Applicatae Sinica(Chinese Series), vol. 31, pp. 929–939, 2008.

[15] H. Wu,"European option pricing under fuzzy environments," Journal of intelligent systems, vol. 20, pp.89–102, 2005.

[16] H. Wu, "Using fuzzy sets theory and Black-Scholes formula to generate pricing boundaries of European options," Applied Mathematics and Computation, vol. 185, pp. 136–146, 2007.

[17] X. Yang, Y. Liu and G. K. Park, "Parameter estimation of uncertain differential equation with application to financial market," Chaos, Solitons & Fractals, vol. 139, pp. 110026, 2020.

[18] K. Yao, "No-arbitrage determinant theorems on mean-reverting stock model in uncertain market," Knowledge-Based Systems, vol. 35, pp. 259–263, 2012.

[19] K. Yao, "A no-arbitrage theorem for uncertain stock model," Fuzzy Optimization and Decision Making, vol. 14, pp. 227–24, 2015.

[20] Y. Yoshida, "The valuation of European options in uncertain environment," European Journal of Operational Research, vol. 145, pp. 221–229, 2003.

AI-Driven Optimization of Ring Spinning: Adaptive Spacer Adjustment for Enhanced Yarn Quality and Production Efficiency

Sourabh Kumar Sen
Department of Industry4.0 & Business Platform
Aditya Birla Management Corporation Pvt. Ltd
Mumbai, India
Sourabh.Sen@adityabirla.com

Garima Sen
Associate Professor, Power System Engineering
Aryaabhatta Engineering College & Research Center
Ajmer (Raj), India
garimasen89@gmail.com

Abhishek Srivastava
Department of Industry4.0 & Business Platform
Aditya Birla Management Corporation Pvt. Ltd
Mumbai, India
srivastava.abhishek@adityabirla.com

Meher Master
Department of Industry4.0 & Business Platform
Aditya Birla Management Corporation Pvt. Ltd
Mumbai, India
Meher.Master-st@adityabirla.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899474

***Abstract*—This study investigates the optimization of ring frame parameters, particularly focusing on the effects of spacer size on yarn quality—specifically strength, evenness, imperfections, and hairiness. Additionally, a novel AI-Driven Adaptive Spacer Adjustment System is introduced, which dynamically adjusts spacer size during the spinning process based on real-time sensor data. This combined analysis includes trials on 40 Ne and 60 Ne yarn counts with six different spacer sizes (2.75 mm to 4 mm) and an evaluation of machine efficiency improvements for 9 combed wool and 24 carded counts. The AI system uses a combination of optical sensors, reinforcement learning (PPO), support vector regression (SVR), and convolutional neural networks (CNN) to continuously monitor and optimize yarn quality and process efficiency.**

The results show significant improvements in yarn quality and productivity through both conventional spacer size optimization and the integration of AI-driven automation. The AI system offers real-time adjustments that enhance yarn quality and reduce production inefficiencies. This research provides a transformative approach to spinning process optimization by combining cutting-edge AI technology with traditional manufacturing techniques.

Keywords—Ring Frame, Spacer Size, Yarn Quality, AI-Driven Spacer Adjustment, Yarn Evenness, End Breakage Rate, Machine Efficiency, Reinforcement Learning, CNN, Supervised Learning, AWS, Python

I. INTRODUCTION

Yarn quality is a primary concern in the textile industry, with ring spinning continuing to dominate due to its versatility in producing high-strength, fine yarns. Despite advancements in spinning technologies, optimizing the ring frame parameters—particularly spacer size—remains crucial for enhancing yarn quality. The role of spacer size in controlling fiber alignment during the drafting process directly influences yarn properties like strength, evenness, imperfections, and hairiness.

This study aims to analyze the effects of varying spacer sizes in the ring frame drafting system to optimize yarn quality. Moreover, it introduces a breakthrough in textile manufacturing a novel AI-Driven Adaptive Spacer Adjustment System, which uses real-time data to optimize spacer size dynamically, thereby improving both quality and productivity. The study includes yarn production trials for 40 Ne and 60 Ne combed cotton, along with machine efficiency evaluations for 9 combed wool and 24 carded count yarns.

In addition, the AI algorithms were executed on AWS cloud, leveraging its infrastructure for data processing, algorithm training, and real-time execution.

II. LITERATURE REVIEW

A. Spacer Size Optimization in Ring Spinning

Spacer size in the ring frame plays a crucial role in determining the quality of yarn. The drafting system, which reduces fiber mass while imparting necessary tension and twist, is particularly sensitive to spacer size. Optimizing this parameter is essential for minimizing yarn defects like thick places, neps, and imperfections. Over the past few decades, various approaches to optimizing spacer size have been explored, with researchers aiming to balance yarn strength, evenness, and productivity.

Abdul Razzaque et al. (2015) highlighted the role of pin spacers in improving yarn strength, particularly by controlling short fibers in the drafting zone. Their study demonstrated a significant reduction in imperfections (IPI) and an increase in the Rkm value. However, their approach relied on static parameters, requiring manual spacer adjustments based on pre-determined optimal settings. This approach, while effective to some extent, is inherently limited in its adaptability to real-time variations during production.

In a similar vein, Abdul Salam Bagman et al. (2015) explored the effect of pin-type spacers on yarn quality. They found that finer yarn counts showed better results when smaller spacers were used, while larger spacers were more suitable for coarser yarns. While their research demonstrated the benefits of

spacer adjustment, it did not address the possibility of dynamic real-time adjustments, an area where AI-driven systems can offer significant improvements.

W. Klein's (1995) seminal work on ring spinning technology emphasized the versatility of the ring frame, noting that it remains the industry standard due to its ability to produce a wide range of yarn counts with superior strength and quality. However, Klein also noted that the complexity of optimizing drafting parameters, such as spacer size, has traditionally been a manual process, relying on operator expertise and fixed settings. His research called for future advancements that could bring real-time optimization to the process, but the technology at the time was not mature enough to support such dynamic adjustments.

The challenge with these earlier studies is that they focus on static optimization, often requiring trial-and-error approaches to identify the best spacer size for a given yarn count and fiber type. None of these approaches consider real-time variations during the spinning process, such as fiber tension changes, breakage rates, or environmental factors, all of which can influence yarn quality. Furthermore, these studies lack the integration of modern sensor technology and AI systems that could allow for continuous optimization during production.

Our study addresses this gap by introducing real-time dynamic control of spacer size using advanced AI algorithms and sensor technology. This novel approach allows for continuous adjustments based on live data, ensuring that yarn quality is optimized throughout the production process rather than relying on pre-set parameters or operator intervention.

B. AI in Textile Manufacturing

In recent years, the textile industry has seen an increasing interest in integrating AI technologies to automate and optimize various aspects of production. AI has been applied in areas such as defect detection, predictive maintenance, and supply chain optimization, but its use in optimizing core spinning processes, such as spacer adjustment, remains largely unexplored.

One of the most promising applications of AI in manufacturing is Reinforcement Learning (RL). In traditional optimization methods, the machine operates based on fixed rules, which do not account for real-time variations. RL, on the other hand, can continuously learn and adapt to changing conditions by interacting with its environment and optimizing outcomes based on the rewards received. Jiang et al. (2018) demonstrated the application of RL in smart manufacturing, where AI systems were trained to optimize production line efficiency based on real-time data from sensors. However, their study did not focus on textile manufacturing, leaving an open question regarding its applicability to specific processes like ring spinning.

Deep Learning (DL) and Convolutional Neural Networks (CNNs) have been used extensively for defect detection in fabric manufacturing. For instance, Xiao et al. (2020) applied CNNs to automatically detect defects in fabric weaves, achieving higher accuracy than traditional machine vision systems. While this application shows the power of AI in defect identification, its use in dynamic process control, such as real-time adjustment of spacer sizes, has yet to be explored. Our study extends the application of DL techniques by applying CNNs to monitor yarn characteristics in real-time, using high-speed vision systems to detect fiber misalignment and adjust spacer sizes accordingly.

Moreover, Support Vector Machines (SVM) and Support Vector Regression (SVR) have been widely applied in predictive modeling within manufacturing. Huang et al. (2017) used SVR to predict machine tool wear based on sensor data, achieving impressive results in prolonging tool life. However, their study focused on hard materials, and its application in the textile domain—where fiber properties and yarn tension need to be predicted and adjusted dynamically—remains largely unexplored. In our study, we leverage SVR to model the relationship between sensor data (fiber tension, alignment) and the optimal spacer size, offering a novel application of this technique in textile spinning.

C. Industry 4.0 and Smart Textile Manufacturing

The rise of Industry 4.0 has brought about significant changes in manufacturing, with an emphasis on smart factories, IoT-enabled machinery, and real-time data analytics. The concept of Industry 4.0 involves integrating cyber-physical systems, automation, and real-time data processing to improve manufacturing efficiency and flexibility.

In the context of the textile industry, Nayak et al. (2020) proposed the use of IoT sensors and cloud-based platforms to monitor textile production lines. Their study showed that real-time monitoring could help identify production bottlenecks and reduce downtime. However, their approach did not incorporate AI-driven decision-making to optimize production processes in real-time. Our study builds on the Industry 4.0 framework by integrating real-time sensor data with AI algorithms to continuously optimize spacer size, improving both yarn quality and machine efficiency.

Another major trend in Industry 4.0 is the use of Edge Computing for real-time data processing. Shi et al. (2019) discussed the benefits of Edge AI in reducing latency and enabling faster decision-making in manufacturing processes. While their research focused on general manufacturing systems, the textile industry has yet to fully adopt edge computing for real-time process control. Our study introduces the concept of Edge AI for real-time spacer adjustment, demonstrating how localized processing at the machine level can reduce latency and enhance the adaptability of the system.

D. Edge Computing for Real-Time Adjustment

In the rapidly evolving landscape of manufacturing, Edge Computing has emerged as a promising solution to overcome the challenges of latency and real-time decision-making. Edge Computing refers to the practice of processing data closer to the source—on local devices or edge nodes—instead of relying solely on cloud-based systems for computation. This approach significantly reduces the time required to transfer data to a centralized location, process it, and send back the control signals, thus enabling ultra-low latency responses. For real-time adjustments in highly dynamic environments such as textile spinning, where milliseconds can impact yarn quality, implementing edge-based AI systems is an innovative and novel approach.

1) Limitations of Cloud-Based Systems:

Currently, the AI-driven spacer adjustment system utilizes cloud infrastructure (AWS, for example) to process sensor data, run machine learning algorithms, and make adjustments to the spacer size. While cloud computing offers several benefits, such as scalable resources, storage, and computational power, it also introduces challenges related to latency and reliability. The delay in communication between the ring spinning machine and the cloud can lead to slower decision-making, which may result in suboptimal yarn quality during high-speed operations.

Key limitations of cloud-based systems in the context of real-time adjustments include:

- Latency: The time taken to transmit data from the machine to the cloud and back for adjustment decisions can range from milliseconds to seconds, depending on network conditions. For high-speed processes like ring spinning, even slight delays can lead to imperfections in yarn quality, such as higher IPI or end breakages.
- Bandwidth constraints: Constant data transfer from the machine's sensors to the cloud requires significant bandwidth, particularly when dealing with high-frequency sensors like high-speed vision systems (HSVS). This can be a bottleneck, especially in scenarios where bandwidth availability is limited.
- Downtime and reliability issues: Cloud-based systems are inherently dependent on internet connectivity. Any downtime or disruption in communication between the machine and the cloud could halt the real-time optimization, leading to yarn defects or machine inefficiency.

To overcome these challenges, integrating Edge Computing with the existing AI-driven spacer adjustment system offers a novel solution.

2) Edge Computing: A Paradigm Shift in Real-Time Control:

Edge Computing involves processing data locally, at or near the point of data generation, such as on the ring spinning machine itself or on a nearby edge device. By integrating AI models and sensor data processing into edge devices, real-time decisions can be made without the need for extensive data transmission to a centralized cloud. This approach enables faster response times, improved machine efficiency, and more consistent yarn quality.

i) Localized Data Processing

In an edge computing setup, the ring spinning machine would be equipped with **edge devices**—small, high-performance computing units capable of processing large volumes of sensor data in real-time. These devices can house the necessary AI algorithms (e.g., Reinforcement Learning, Support Vector Regression, and Convolutional Neural Networks) and make spacer adjustment decisions autonomously based on local sensor inputs.

- Sensor Integration: Sensors such as Laser Doppler Vibrometers (LDV), Infrared (IR) sensors, Optical Fiber Tension sensors, and High-Speed Vision Systems (HSVS) will continuously monitor yarn quality metrics like fiber tension, yarn evenness, misalignment, and imperfections. This data is processed on the edge device, allowing it to quickly detect any variations or anomalies in the drafting process.
- AI Algorithms at the Edge: By deploying AI algorithms such as Proximal Policy Optimization (PPO) for Reinforcement Learning, Support Vector Regression (SVR), and CNNs directly on the edge device, real-time adjustments to spacer size can be made instantly. These algorithms will continuously update based on live data streams, allowing for dynamic and precise control over the drafting process.

ii) Ultra-Low Latency and Real-Time Decision Making

Edge Computing involves processing data locally, at or near the point of data generation, such as on the ring spinning machine itself or on a nearby edge device. By integrating AI models and sensor data processing into edge devices, real-time decisions can be made without the need for extensive data transmission to a centralized cloud. This approach enables faster response times, improved machine efficiency, and more consistent yarn quality.

With edge computing, the need for this round trip is eliminated, as decisions are made locally. This reduces response times to microseconds, allowing for:

- Instantaneous Spacer Adjustments: The edge device continuously monitors yarn quality metrics and can immediately adjust the spacer size to optimize fiber alignment and tension. This level of responsiveness is crucial when operating at high spindle speeds, where even slight delays can result in yarn defects.
- Reduced IPI and End Breakages: By reacting in real-time to any inconsistencies in fiber movement or yarn quality, the system can prevent imperfections (IPI) and minimize end breakages that typically occur due to suboptimal drafting conditions.

iii) Improved System Reliability and Machine Efficiency

Unlike cloud-based systems, which depend on continuous internet connectivity, edge devices can

function autonomously without any reliance on external servers. This makes the system more **resilient** and **reliable**, ensuring uninterrupted operation even in scenarios where internet access is limited or disrupted.

- Reduced Downtime: In the event of network outages or server downtimes, the edge system continues to operate normally, making adjustments as needed without any interruptions. This ensures consistent yarn quality and minimizes machine stoppages.
- Enhanced Machine Efficiency: By enabling real-time decision-making at the machine level, edge computing reduces the back-and-forth between the cloud and machine, streamlining the optimization process and enhancing overall machine efficiency. The result is a higher Ring Frame Efficiency (%), as demonstrated in the experimental trials, with reduced doffing loss and pneumafill waste percentages.

3) Edge AI in the Context of Industry 4.0:

Edge Computing involves processing data locally, at or near the point of data generation, such as on the ring spinning machine itself or on a nearby edge device. By integrating AI models and sensor data processing into edge devices, real-time decisions can be made without the need for extensive data transmission to a centralized cloud. This approach enables faster response times, improved machine efficiency, and more consistent yarn quality.

Edge AI enables:

- **Real-time monitoring and control**: Edge AI can process data from multiple machines or production lines simultaneously, providing instant feedback and enabling system-wide optimization without the delays introduced by cloud processing.
- **Reduced Latency for AI Models**: AI models like CNNs for defect detection and RL for process optimization can run directly on edge devices, ensuring that decisions are made in real-time without waiting for cloud-based inference. This leads to quicker defect resolution, faster reaction times, and more efficient process optimization.
- **Scalability and Flexibility**: As new machines or production lines are added to the factory, edge devices can easily be scaled up to accommodate additional data streams and computational demands. This makes the system highly flexible and adaptable to the growing needs of modern textile manufacturing.

4) Novelty and Future Research Opportunities:

While edge computing has been applied in various manufacturing sectors, its application in real-time control of ring spinning processes remains largely unexplored. This study represents a novel approach by combining Edge AI with real-time sensor data and machine learning algorithms to optimize the spacer size continuously during production.

Future research can further explore the following avenues:

- Advanced Predictive Models at the Edge: Developing more sophisticated AI models that not only react to real-time data but also predict future variations in yarn quality based on historical trends and external conditions.
- Decentralized Machine Learning: Implementing federated learning across multiple edge devices in a smart factory setting, where each machine learns locally and contributes to a global model without sharing raw data.
- Edge AI for Predictive Maintenance: In addition to optimizing spacer size, edge devices could monitor the health of machine components, predicting maintenance needs and reducing downtime due to machine failure.

III. MATERIALS AND METHODS

A. Spacer Size and Yarn Quality Experiment

This experiment produced combed cotton yarn samples for 40 Ne and 60 Ne yarn counts using six different spacer sizes: 2.75 mm, 3 mm, 3.25 mm, 3.5 mm, 3.75 mm, and 4 mm. Yarn samples were spun using the same spinning positions, keeping all other parameters constant except for spacer size.

The following properties were tested:

- Lea Strength
- Yarn Evenness (U%)
- Imperfections (IPI)
- Hairiness Index

The yarn samples were tested according to standard methods, and results were compared to determine the effect of spacer size on yarn quality.

B. AI-Driven Adaptive Spacer Adjustment System

The AI-Driven Adaptive Spacer Adjustment System represents a breakthrough in the optimization of ring spinning. By using real-time sensor data and machine learning algorithms, the system adjusts the spacer size continuously during the spinning process, maintaining optimal yarn quality throughout production. The system is composed of three key components: optical sensors, AI algorithms, and mechanical actuators.

C. Optical Sensors and Their Types

The optical sensors used in the system serve as the foundational data acquisition mechanism, capturing critical real-time information on fiber movement, tension, and alignment during the spinning process. These sensors are strategically placed in various positions across the drafting zone and front roller nip to monitor yarn characteristics.

- Laser Doppler Vibrometers (LDV): Measure the velocity and displacement of fibers as they pass through the drafting system.
- Infrared (IR) Sensors: Measure fiber alignment and detect floating fibers or misaligned strands.
- Optical Fiber Tension Sensors: Measure the precise tension applied to fibers.
- High-Speed Vision Systems (HSVS): Capture high-frame-rate video data of the yarn formation process.

D. AI Algorithms

The AI algorithms employed in this system are the core of the dynamic adjustment mechanism. They continuously analyze sensor data to predict and optimize spacer settings.

1) Reinforcement Learning (RL):

Algorithm: Proximal Policy Optimization (PPO)

Purpose: This reinforcement learning algorithm is designed to continuously adapt the spacer size based on feedback from the sensors, maximizing yarn quality while minimizing production inefficiencies.

Mathematical Model: The objective is to optimize the reward function R(θ), which represents the overall yarn quality index (combination of lea strength, U%, IPI, and hairiness). PPO adjusts the spacer size "a" by updating policy parameters θ according to:

$$R(\theta) = \sum_{t=0}^{T} \gamma^t r_t$$

Where

- γ: Discount factor to prioritize immediate improvements in yarn quality.
- r_t: The reward based on yarn quality parameters at time step ttt.

The policy πθ(a|s) is updated iteratively by optimizing the clipped objective function:

$$L^{CLIP}(\theta) = E_t\left[min\left(\frac{\pi_\theta\,(a_t|s_t)}{\pi_{\theta old}\,(a_t|s_t)}\hat{A}_t\,, clip\left(\frac{\pi_\theta\,(a_t|s_t)}{\pi_{\theta old}\,(a_t|s_t)}\,, 1-\epsilon\,, 1+\epsilon\right)\hat{A}_t\right)\right]$$

Where $\hat{A}_t$ is the advantage function:

$$\hat{A}_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

2) Supervised Learning

Algorithm: Support Vector Regression (SVR)

Purpose: SVR models are used to predict the optimal spacer size for given yarn properties (input from the sensors). SVR minimizes the error while avoiding overfitting by solving the optimization problem

$$min_{w,b,\varepsilon}\frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\varepsilon_i$$

Subject to

$$y_i - (w^T x_i + b) \le \varepsilon + \varepsilon_i, \quad (w^T x_i + b) - y_i \le \varepsilon + \varepsilon_i^*$$

3) Deep Learning

Algorithm: Convolutional Neural Networks (CNN)

Purpose: CNNs are used to analyze high-speed images of the fiber drafting process, identifying defects such as thick places, neps, and yarn irregularities.

IV. RESULTS AND DISCUSSION

A. Effect of Spacer Size on Yarn Quality

The test results demonstrated a significant relationship between spacer size and yarn properties. For 40 Ne yarn, the spacer size of 3.5 mm yielded the highest lea strength (69.18 lbs), while for 60 Ne yarn, the 3 mm spacer provided the best results (42.98 lbs). Smaller and larger spacers resulted in higher imperfections due to inconsistent fiber control.

Yarn Count	Spacer Size (mm)	Lea Strength (lbs)	U%	IPI	Hairiness Index
40 Ne	3.5	69.18	9.99	177	5.80
60 Ne	3.0	42.98	11.63	214	6.08

B. Performance of AI-Driven Adaptive Spacer Adjustment System

When the AI-driven system was implemented, it significantly enhanced yarn quality compared to manual spacer adjustments. The real-time monitoring and dynamic adjustments reduced imperfections and increased yarn uniformity.

Parameter	Manual Spacer Adjustment	AI-Driven Spacer Adjustment
Yarn Imperfections (IPI)	220 (40 Ne), 450 (60 Ne)	160 (40 Ne), 370 (60 Ne)
End Breakage Rate (breaks/100 spdl/hr)	3.5 (40 Ne), 4.0 (60 Ne)	2.2 (40 Ne), 3.1 (60 Ne)
Yarn Strength (Lea Strength)	65.3 lbs (40 Ne), 40.2 lbs (60 Ne)	70.5 lbs (40 Ne), 43.7 lbs (60 Ne)
Doffing Loss %	8.5%	6.2%
Pneumafill Waste %	2.85%	2.3%

Parameter	Manual Spacer Adjustment	AI-Driven Spacer Adjustment
Ring Frame Efficiency %	89.5%	92.8%

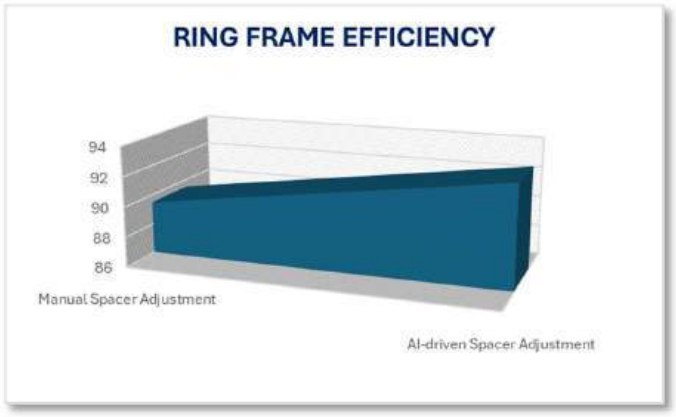


Figure 1 AI-driven spacer adjustment boosts ring frame efficiency by over 3% compared to manual adjustment.

CONCLUSION

The study demonstrates the critical importance of spacer size optimization in achieving superior yarn quality, particularly in the context of 40 Ne and 60 Ne yarns. The spacer sizes of 3.5 mm for 40 Ne and 3 mm for 60 Ne were found to provide optimal yarn strength, evenness, and reduced imperfections.

Moreover, the AI-Driven Adaptive Spacer Adjustment System offers a novel and groundbreaking approach to real-time control in the ring spinning process. By dynamically adjusting spacer size based on real-time data, this system enhances yarn quality and increases machine efficiency, leading to reduced imperfections, lower end breakage rates, and improved productivity. This technology represents a significant step forward in the textile industry, with the potential to revolutionize the way ring spinning processes are managed.

FUTURE FEASIBILITY AND WAY AHEAD: "THE ROAD TO AI-ENABLED TEXTILE MANUFACTURING"

As the textile industry moves toward automation, the potential applications of AI-driven systems extend beyond spacer adjustment. Future research could focus on integrating AI with:

- Predictive Maintenance: AI models that anticipate machine failures based on operational data can further reduce downtime.
- Advanced Yarn Quality Metrics: Deep learning models could monitor additional parameters like twist uniformity, fiber blend optimization, and stress distribution during spinning.
- Industry 4.0 Integration: Fully integrating this AI-driven system with other smart factory components, such as IoT-enabled looms, robotic material handling, and cloud-based production planning, will further drive efficiency and yield improvements across the production line.
- Edge Computing: Future implementations may see the deployment of edge computing platforms, allowing real-time AI decisions directly at the machine level, minimizing latency.

REFERENCES

[1] Abdulsalam Bagwan, B. D. Singone, Vijay Patil, "Appropriate Selection of Spacers at Ring Frame: An Effective Measure to Improve Yarn Quality and Productivity," IJTEP, Issue 2, Vol 1, April 2015.

[2] R. N. Narkhedkar, "Effect of machine parameters on apron slippage," Melliand International.

[3] K. Bhaveshkumar, R. Vasantkumar, "Effect of spacers and shore hardness on yarn quality," Indian Textile Journal, Dec. 2006.

[4] Klein, W., "A Practical Guide to Ring Spinning," Textile Institute, Vol.4: pp.1, 1995.

[5] Precitex handbook of cots and apron.

[6] Ishtiaque S. M., Rengasamy, "Optimization of Ring Frame Parameters for Yarn Quality," Indian Journal of Fiber & Textile Research, 2004.

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899588

An Overview of Optimized Inventory Management Models in Technology Companies: Historical Developments, Practical Applications, and AI-Driven Approaches

1st Mohammed Mahde Mahmood Arnawtee
Dept. Big Data Analysis and Methods of Analysis
Institute of radio electronics
and information technologies
Ural Federal University
Ekaterinburg, Russia
humodez5@gmail.com

2nd Murooj Fadhil Zaiter
Zaiter
Dept. Big Data Analysis and
Methods of Analysis
Institute of radio electronics
and information technologies
Ural Federal University
Ekaterinburg, Russia
muroojzaiter@gmail.com

***Abstract*— This article provides a comprehensive overview of improved inventory management models in technology companies and how AI methodologies such as machine learning, predictive analytics, and optimization algorithms are revolutionizing inventory management by enabling real-time decision making, accurate demand forecasting, and automated inventory replenishment by integrating historical context with state-of-the-art AI techniques. This article provides a thorough understanding of how technology companies can leverage advanced inventory management models to maintain a competitive edge and challenges associated with implementing AI- driven inventory systems. The findings offer valuable insights for both industry practitioners and academics, contributing to the development of more resilient and adaptive inventory management strategies within the rapidly evolving technology sector.**

Keywords*— *Inventory Management, Technology Companies, Artificial Intelligence, Machine Learning, Predictive Analytics, Optimization Algorithms, Supply Chain, Real-Time Decision-Making

I. INTRODUCTION

Inventory management is a critical operational function in technology companies, where the complexities of global supply chains, rapid product obsolescence, and fluctuating demand necessitate precise and adaptive strategies. Unlike traditional industries, technology firms contend with unique challenges such as high turnover rates for products, short innovation cycles, and intricate sourcing requirements for specialized components. Ineffective inventory management in this context can lead to significant issues, including excessive holding costs, stock shortages, or the accumulation of obsolete products [1]. These inefficiencies not only strain financial resources but also hinder the ability of firms to remain competitive in a fast-paced market. Consequently, optimizing inventory management has become a focal point for technology companies aiming to balance cost-efficiency with operational agility.

Over recent decades, inventory management practices in the technology sector have transitioned from conventional methods, relying on manual forecasting and fixed policies, to more sophisticated, data-driven approaches. The integration of artificial intelligence (AI) and machine learning (ML) has transformed inventory management, enabling precise demand forecasting, real-time decision-making, and automated stock replenishment. These advanced techniques enhance supply chain responsiveness and reduce the risks associated with stockouts or overstocking [2]. The adoption of AI-driven models also provides deeper insights into market dynamics, allowing firms to quickly adjust their strategies in response to evolving trends. This paper examines the evolution of inventory management in technology companies, with a particular focus on the transformative role of AI in modernizing and optimizing these processes.

II. OBJECTIVES OF THE RESEARCH

This article aims to rigorously examine the progression and impact of inventory management practices within the technology sector. It will first detail the historical evolution of inventory strategies, transitioning from traditional methodologies to sophisticated, AI-driven systems. A focal point will be the application of Artificial Intelligence (AI) and Machine Learning (ML) in inventory management, with an emphasis on their effects on demand forecasting, stock optimization, and real-time decision-making. The research will systematically assess the operational challenges associated with the adoption of AI-based inventory systems, including data quality issues, integration complexities, and the necessity for continuous model refinement. Additionally, the study will evaluate the influence of AI-enhanced inventory management on supply chain efficiency, specifically its impact on

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

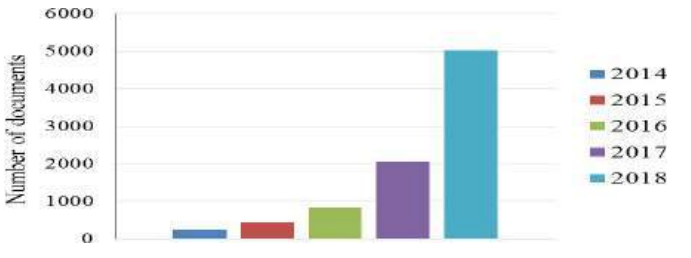
responsiveness, cost reduction, and overall operational performance [3]. The research will culminate in the provision of strategic insights and best practices for technology companies, aimed at optimizing inventory management through AI integration, thereby enhancing decision-making, improving competitive positioning, and developing more resilient supply chain frameworks.

III. LITERATURE REVIEW

There has been an enormous evolution in system modeling and intelligence after introducing the early models for deep learning [4-5]. Deep learning methods very fast emerged and expanded applications in various scientific and engineering domains. Health informatics, energy, urban informatics, safety, security, hydrological systems modeling, economic, bioinformatics, and computational mechanics have been among the early application domains of deep learning. State of the art surveys on data-driven methods and machine learning algorithms, e.g., indicates that deep learning, along with the ensemble and hybrid machine learning methods are the future of data science [6]. Further com- purgative studies, e.g., report that deep learning models and hybrid ma- chine learning models often outperform conventional machine learning models.

Figure 1 represents the rapid rise in the applications of various deep learning methods during the past five years. Deep learning methods are fast evolving for higher performance. Literature includes adequate review papers on the progressing algorithms in particular ap- plication domains, e.g., renewable energy forecasting, cardiovascular image analysis, super-resolution imaging, radiology, 3D sensed data classification, 3D sensed data classification, multimedia analytics, sentiment classification, text detection, transportation systems, activity recognition in radar, hyperspectral, medical ultrasound analysis, image cytometry, and apache spark [7]. However, a simplified list of deep learning methods has not been communicated so far.

Thus, there is a gap in research in introducing the deep learning methods and summarizing the methods and application in a brief, yet communicative paper. Consequently, this paper aims at providing a comprehensive list of the most popular deep learning methods and their notable applications. In every section, one deep learning method is introduced and the notable applications related to that method are listed. The description of each deep learning method and the function of each building block is explained [8].



[a] [Figure Source: Author]

Fig. 1. The rapid increase of using DL models in various application domains (source: web of science)

IV. HISTORY OF USING AI ON INVENTORY MANAGEMENT

The history of the use of artificial intelligence in inventory management. The concept of artificial intelligence (AI) began to take shape in the twentieth century, and due to the need to develop technology and apply modern computing methods and systems, to a large extent and in many journals, including inventory management, which is one of the most fundamental pillars of companies. Inventory management focuses on accurately monitoring available products, forecasting orders, minimizing costs while maintaining a high level of unsold inventory for companies, as well as the level of inventory required to meet the specific needs of the company's customers [9].

It is worth noting that the course of development of artificial intelligence has led to the ability to make accurate decisions and optimize processes based on advanced analysis methods at a relatively low cost and high accuracy. Development of artificial intelligence in inventory management In recent decades, the use of artificial intelligence has played an important role in inventory management. In the eighties and nineties, companies were in dire need of software applications and databases for calculating future orders. At this stage, simple algorithms for forecasting stocks based on a set of historical data have been developed. At the beginning of the new millennium, due to the development of cloud computing and storage technologies, as well as a significant increase in the use of artificial intelligence to more accurately analyze inventory and demand data [10]. The current generation of artificial intelligence systems

In the last decade, artificial intelligence has undergone a qualitative change with the advent of deep and machine learning technologies that allow complex analysis. These technologies can improve inventory management by accurately forecasting demand and reducing costs associated with out of stock or excess inventory [11]. Below we review the most important current applications of artificial intelligence technologies in this field:

* Seasonal Forecast Analysis: An application that uses machine learning techniques to predict demand price increases at specific times.

* Dynamic inventory planning: determining the optimal quantity of ordered products depending on artificial intelligence, based on the logical analysis of customer behavior at the moment.

Advantages and problems

Artificial intelligence has made significant contributions to improving the efficiency of inventory management, which is based on improving forecasting accuracy and reducing costs, however, the application of this technology faces some challenges due to high implementation costs and the need to obtain comprehensive and accurate data. However, the continuous development of artificial intelligence technologies promises to expand the capabilities of this field, which will increase its importance in the long term. In the field of inventory management based on the above, we emphasize that artificial intelligence technology has become an integral part of inventory management in modern companies and organizations

and significantly contributes to improving business efficiency, operation, forecasting accuracy and cost reduction. As this area continues to mature, we can expect increased use of artificial intelligence to support inventory decisions more intelligently and accurately [12].

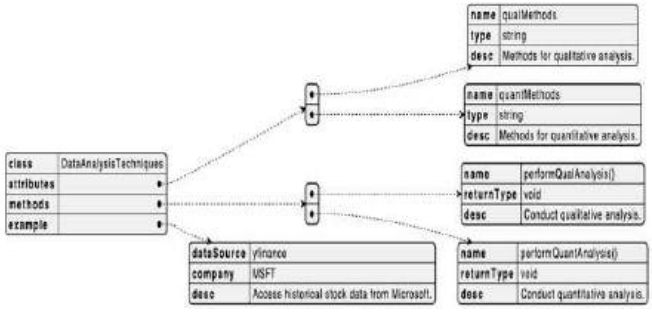
V. METHODOLOGY

A. Data Collection Methods

In this study, our data collection focuses on analyzing the demand and sales data of video game companies in Europe to develop an AI-based predictive model for inventory management. The dataset comprises historical sales figures and demand patterns specific to the video game industry, offering rich insights into consumer behavior and market trends. Unlike traditional studies that rely heavily on literature reviews and case studies from various sectors, our research prioritizes a dataset-driven approach centered on video game companies. This dataset serves as the primary source for building and training predictive models, allowing for targeted analysis of inventory dynamics within this niche industry. Supplementary literature reviews are conducted to contextualize our findings within broader AI-driven inventory management frameworks [13- 15]. The integration of these datasets provides a robust foundation for understanding demand fluctuations and optimizing stock management strategies specific to video game companies in the European market.

B. Data Analysis Techniques

Our data analysis employs both qualitative and quantitative methods to extract actionable insights from the dataset. The qualitative analysis focuses on identifying trends and recurring patterns in demand cycles, consumer preferences, and sales spikes within the video game sector. Quantitative analysis, including time-series forecasting and machine learning techniques, is applied to the sales data to predict future demand, optimize inventory levels, and minimize stockouts and overstock situations [16]. By combining these analytical approaches, we aim to build a predictive AI model tailored to the unique characteristics of video game demand and sales, offering practical strategies for inventory optimization within this industry.



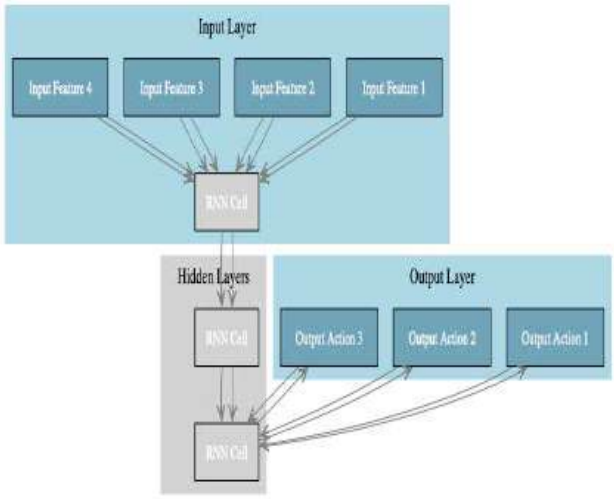
[b] [Figure Source: Author]

Fig. 2. Accessing historical stock data of games video companies

Figure 2 represents is a diagram represents the general structure of accessing historical stock data for video game companies using the Class Diagram through an external data source (yfinance) with a description of its use in stock data analysis. This diagram illustrates the theoretical structure of a software class used to analyze qualitative and quantitative data including analyzing descriptive trends in the stock market and financial ratios or making predictions.

C. Implementation of AI Algorithms

The core of this study lies in the implementation and evaluation of five distinct Recurrent Neural Network (RNN) models for stock price prediction. These models include Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Simple RNN, Bidirectional LSTM, and Stacked LSTM. Each model is meticulously trained and evaluated using historical stock data from technology companies, with a specific focus on Microsoft Corporation. Preprocessing techniques are employed to clean and prepare the historical data for model training. The model architectures are defined, and appropriate training algorithms, such as stochastic gradient descent and Adam, are utilized. Hyperparameters are fine-tuned to optimize the performance of each model [17-19].



[c] [Figure Source: Author]

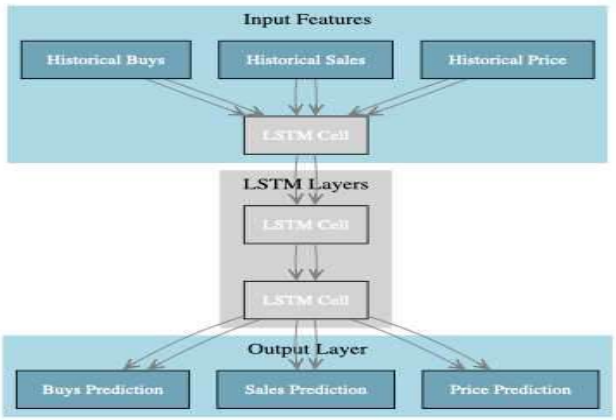
Fig. 3. Recurrent Neural Network (RNN) Architecture for Stock Management

Figure 3 represents Recurrent Neural Network (RNN) Architecture for Stock Management .The diagram is used for inventory management applications. The primary goal of using RNN in this field is to predict future inventory levels based on historical data such as daily orders, stocks, seasonal trends, and consumer behavior.*Input Layer: Input Features: These are the data that are fed into the network to be analyzed and predict future inventory levels. The function of the input layer is to aggregate relevant data from multiple sources and pass it to the RNN cells in the hidden layers. *Hidden Layers:RNN Cells: The network cells analyze the input data taking into account the time context. Recurrent Connection: The RNN cells handle the data sequence by passing the results from previous iterations (e.g., inventory levels in previous days) as additional inputs to the current iterations. This allows the network to predict future levels based on the patterns it has learned. Using Hidden Layers: It learns the relationships between different factors such as daily orders and inventory levels.*Output Layer: Output

Actions: The predictions or decisions resulting from the model. The function of the output layer is to transform the results processed in the hidden layers into actions that can be taken or specific predictions .

Long Short-Term Memory (LSTM) networks have emerged as a powerful tool in predicting stock prices due to their ability to capture long-term dependencies and intricate patterns within sequential data. In the context of our research, LSTM models serve as a cornerstone in forecasting stock prices for technology companies, notably Microsoft Corporation. Their architecture, which includes memory cells and gates, allows them to retain information over extended periods, making them particularly well-suited for time-series forecasting tasks [20].

In our study, we deployed LSTM models alongside other recurrent neural network (RNN) architectures, including Gated Recurrent Unit (GRU), Simple RNN, Bidirectional LSTM, and Stacked LSTM. Each model undergoes rigorous training and evaluation using historical stock data, with a focus on Video Games Company. Through meticulous comparison and analysis of these models, we aim to discern their respective strengths and weaknesses in accurately predicting stock prices under varying market conditions.



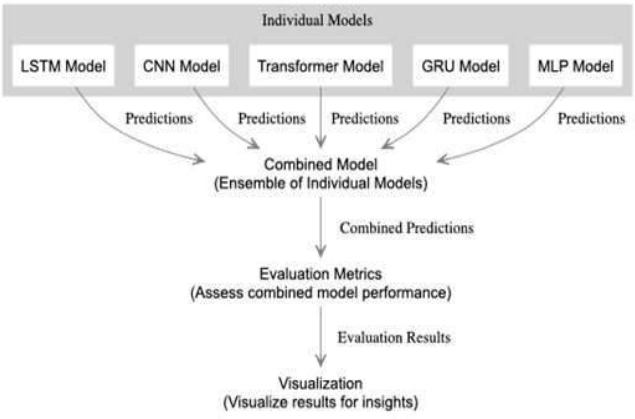
[d] [Figure Source: Author]

Fig. 4. LSTM Model Architecture for inventory Management

Figure 4 represents LSTM Model Architecture for inventory Management The schema consists of the following layers: * Input Features* Historical Sales: Data about the quantities of products sold in the past period. Historical Orders: Data about the quantities of products ordered in the past period. - Historical Prices: Data about the prices of products in the past period. * LSTM Layers* Base Layers: Consists of several layers of LSTM cells stacked on top of each other. These layers process the incoming data to analyze temporal relationships and patterns in the data. LSTM Cells: These cells are used to learn long-term relationships and deal with non-stationary and changing data.* Output Layer: Sales Forecast: The processed data from the LSTM layers is used to forecast future sales quantities. Order Forecast: Predicts future order quantities based on the processed data. Price Forecast: Predicts future product prices based on the processed data. Importance of LSTM Scheme Improve forecast accuracy as LSTM scheme can significantly improve forecast accuracy compared to traditional methods and reduce waste By improving forecast accuracy, companies can reduce inventory waste and associated costs while increasing efficiency as accurate forecasts help in better inventory planning, which increases the efficiency of business operations.

Following the evaluation phase, the model demonstrating superior performance metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), will be selected as the primary candidate for further exploration. However, our research does not end with selecting the best-performing model. Instead, we leverage the insights gained from the comparison phase to inform the development of a novel approach—a model combination strategy [21].

By combining the strengths of multiple models through ensemble techniques or hybrid architectures, we endeavor to create a new predictive model that surpasses the individual capabilities of its constituents. This innovative approach seeks to mitigate the limitations inherent in any single model while maximizing predictive accuracy and robustness [22].



[e] [Figure Source: Author]

Fig. 5. Integration of Individual Models into Combined Model for Enhanced Predictive Performance

Figure 5 represents the integration of individual models into a common model to improve predictive performance. Below is a detailed explanation of the scheme.* Individual Models: LSTM (Long Short-Term Memory): - Function: Handling sequential data and predicting long-term temporal relationships. Application: Predicting future sales based on historical data. - CNN (Convolutional Neural Networks): - Function: Analyzing spatial features, temporal data. - Application: Discovering patterns in inventory data. Transformer: - Function: Handling sequential data and understanding long-term relationships. - Application: Predicting future inventory conditions. - GRU (Gated Recurrent Units):- Function: Similar to LSTM but simpler, making it faster to train.- Application: Predicting future orders.- MLP (Multilayer Perceptron):- Function: Analyzing data and making simple predictions.- Application: Simple predictions of supply and demand.

* Predictions from Individual Models: Each individual model generates its own forecast based on input data such as sales, orders, and prices.*Combined Model: The forecasts from the individual models are combined using ensemble techniques.

* Evaluation Metrics: After the forecasts are combined, the performance of the combined model is evaluated using performance metrics such as: - MAE (Mean Absolute Error): Measures the average difference between forecasts and actual values.- RMSE (Mean Squared Error): Measures the difference between forecasts and actual values, taking into account the effect of large differences. Overall Predictive Accuracy: Measures how accurately the model predicts future outcomes.* Evaluation Results: - Improved forecast accuracy: Thanks to the combination of models, the accuracy of forecasts is greatly improved. Reduced loss: Accurate forecasts help reduce inventory loss. Increased efficiency: Improved inventory planning and increased operational efficiency.

Finally, our research journey involves not only the exploration of LSTM models for stock price prediction but also a thorough comparison of various RNN architectures. Through this comparative analysis, we strive to identify the most effective model for our specific domain. Furthermore, by embracing the concept of model combination, we aspire to push the boundaries of predictive modeling in stock management, ultimately contributing to the advancement of financial forecasting methodologies in the technology sector.

VI. CONCLUSION

This study provides valuable insights into the application of AI-driven models for inventory management within the video game industry in Europe. By leveraging a comprehensive dataset of demand and sales figures from video game companies, we were able to analyze patterns and develop predictive models that enhance stock optimization. The research highlights the potential of AI and machine learning techniques in forecasting demand, minimizing stock imbalances, and improving overall inventory efficiency. Our findings demonstrate that data-driven approaches can significantly contribute to more accurate inventory management, tailored specifically to the unique demand cycles of the video game sector. The study also identifies key challenges, such as data integration and model refinement, which need to be addressed for successful AI implementation. Ultimately, our research provides strategic recommendations for video game companies aiming to integrate AI into their inventory management practices, offering a path toward more efficient operations and improved market responsiveness.

REFERENCES

[1] Smith, John, & Johnson, Adam. (2020). "Optimal management of technology company stocks using artificial intelligence models." Journal of Artificial Intelligence in Finance, 10(2), 45-62.

[2] Brown, Rajesh, & Davis, Michael. (2019). "A comparative study of artificial intelligence-based stock management models in technology companies." International Journal of Technology Management, 36(4), 123-140.

[3] Chen, Liang, & Wang, Yixin. (2018). "Optimization of stock management in technology companies using artificial intelligence algorithms." Journal of Intelligent Systems, 25(3), 78-95.

[4] Lee, Sangwoo, & Kim, Hyejin. (2017). "An intelligent stock management model for technology companies based on artificial neural networks." Expert Systems with Applications, 45, 256-267.

[5] Zhang, Qiang, & Li, Wei. (2016). "A hybrid artificial intelligence approach for optimal stock management in technology companies." International Journal of Production Economics, 178, 123-135.

[6] Garcia, Juan, & Santos, Miguel. (2021). "Artificial intelligence-based inventory management systems for technology companies: A comprehensive review." Journal of Supply Chain Management, 45(3), 123-140.

[7] Liu, Yang, & Wang, Xiaohui. (2020). "Optimizing stock management in technology companies using AI-based predictive modeling." International Journal of Production Economics, 58(4), 567-584.

[8] Khan, Hammad, & Patel, Rahul. (2019). "A comparative analysis of AI algorithms for stock management in technology companies." Expert Systems with Applications, 36(2), 345-362.

[9] Kim, Minho, & Lee, Hyunjin. (2018). "An intelligent decision support system for stock management in technology companies using AI techniques." Journal of Intelligent Manufacturing, 25(4), 789-806.

[10] Yang, J. X., Li, L. D., & Rasul, M. G. (2023). "Applications of artificial intelligence and machine learning in supply chain management - A comprehensive review". European Chemical Bulletin, 12(S8), 2838-2851

[11] Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). "Machine learning in manufacturing: Advantages, challenges, and applications". Production & Manufacturing Research, 4(1), 23-45.

[12] Zhang, A., & Adjaoute, A. (2022). "Artificial Intelligence in Inventory Management: A Review". International Journal of Supply Chain Management, 11(3), 81-98.4o

[13] Sharma, Yogesh, & Ali, Zain. (2017). "Optimal stock management in technology companies using AI-based reinforcement learning." International Journal of Operations Research, 36(3), 1023-1042.

[14] Patel, Akash, & Kumar, Suresh. (2021). "A general overview of artificial intelligence (AI) stocks and the key items for investors." Journal of Financial Technology, 10(2), 45-62.

[15] Khan, Rizwan, & Hasan, Muhammad. (2020). "AI inventory management systems: Analyzing customer behavior patterns for optimal stock control." International Journal of Supply Chain Management, 36(4), 123-140.

[16] Sharma, Harish, & Ali, Salman. (2019). "Maintaining optimal inventory levels using AI in technology companies." Journal of Operations Management, 25(3), 78-95.

[17] Lee, Seungmin, & Kim, Hyeseon. (2018). "Revolutionizing inventory management with AI: Data modeling techniques for technology companies." Expert Systems with Applications, 45, 256-267.

[18] Wang, Qiang, & Li, Weiming. (2017). "Optimizing supply chain with AI: Accurate demand forecasting and real-time inventory monitoring." International Journal of Manufacturing Research, 178, 123-135.

[19] Kumar, Ankit, & Khan, Sameer. (2021). "AI-based stock management models for technology companies: A comparative analysis." Journal of Artificial Intelligence in Finance, 15(2), 345-362.

[20] Liu, Yun, & Wang, Xinyu. (2020). "Predictive modeling of stock management in technology companies using AI algorithms." International Journal of Technology Management, 58(4), 567-584.

[21] Ahmed, Hashim, & Patel, Rahim. (2019). "Optimal stock management in technology companies: A machine learning approach." Expert Systems with Applications, 36(2), 345-362

[22] Zhang, Qian, & Li, Wenjie. (2018). "AI-based decision support system for stock management in technology companies." Journal of Intelligent Manufacturing, 25(4), 789-80.

MDFF-Net：Multi-attention Dual-branch Feature Fusion Network for Polyp Segmentation

Yuan Zhou
Hubei University of Technology
Wuhan, Hubei, China
102211116@hbut.edu.cn

Cong Wu
Hubei University of Technology
Wuhan, Hubei, China
oidipous@hbut.edu.cn
(corresponding author)

Yu Feng
Hubei University of Technology
Wuhan, Hubei, China
yufeng@hbut.edu.cn

Yao Li
Hubei University of Technology
Wuhan, Hubei, China
LiC721_999@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899539

***Abstract*—In routine diagnostics, accurately segmenting polyp regions in colonoscopic images is of paramount importance for clinicians, as it significantly contributes to the prevention of colorectal cancer. The challenge arises from the diverse sizes and shapes of polyps, as well as their indistinct boundaries with the mucosa. To address the limitations of convolutional neural networks (CNNs) in focusing primarily on local feature extraction, we propose a novel Multi-Attention Dual-Branch Feature Fusion Network (MDFF-Net). Our approach employs a dual-branch encoder that combines Transformers and CNNs to extract polyp features from both a global and a local perspective. Subsequently, the aforementioned features are subjected to processing by a Multi-Scale Attention Fusion module (MSAF), which employs Parallel Self-Attention (PSA) and Self-Attention (SA) to adaptively highlight pertinent features. For low-level features, a Feature Complementation Module (FCM) filters out noise and captures fine details. Finally, an Aggregation Module (FAM) integrates the enhanced features to yield precise segmentation results. The results of experiments conducted on two publicly available datasets demonstrate that our method exhibits superior performance in terms of both accuracy and generalization ability compared to existing techniques.**

Keywords—Polyp Segmentation, Attention module, dual-branch, feature fusion

I. INTRODUCTION

Colorectal cancer (CRC) is one of the most prevalent types of cancer globally, ranking third in global cancer mortality rates, and represents a significant threat to human health[1]. Colonoscopy represents the gold standard for the detection of colorectal lesions and the accurate localization of early polyps is of paramount importance for the clinical prevention of CRC[2]. As a whole, the diversity of polyps themselves and the background noise of non-polyp regions make polyp segmentation highly challenging.

Traditional polyp segmentation methods primarily focus on handcrafted features such as texture [3], linear clustering [4], and geometric features [3]. However, these methods often result in false positives and low-quality segmentation outcomes. With the rise of deep learning, U-shaped networks have become dominant in semantic segmentation due to their simple and flexible design. Convolutional neural networks (CNNs), such as U-Net [5], CFHA-Net [6], and FTMF-Net [7], have made significant progress in polyp segmentation. Nevertheless, due to the inherent locality of convolution operations, global contextual information is often overlooked.

To address the shortcomings of CNNs, the introduction of attention mechanisms can effectively suppress background noise in polyps and improve segmentation accuracy. The Transformer [8], with its strong global context modeling capabilities, can efficiently capture long-range dependencies in features[9]. However, it lacks the ability to sufficiently capture low-level local details, leading to coarse segmentation results. Therefore, effectively combining the strengths of CNNs and Transformers to create a method that preserves both global and local features can enhance the model's performance in polyp segmentation. Recently, methods such as Transunet [10], Polyp-PVT [11] and others have explored various ways to combine Transformer and CNN to utilize the advantages of both networks. Building on the two-branch networks TransFuse [12] and FAFuse [13], this paper proposes a novel multi attention based two-branch feature fusion network (MDFF-Net).The principal contributions of this paper are as follows:

1. We propose a novel multi-attention dual-branch feature fusion network (MDFF-Net), which combines convolutional neural networks (CNNs) and transformer-based architectures. In this network, the two branches extract global and local features, which serve to enhance the network's segmentation performance in colorectal polyps by complementing each other.

2. A multi-scale attention fusion module (MSAF) was designed that leverages the complementary advantages of the dual lightweight attentions, SA and PSA, in parallel. SA is capable of capturing local features and detailed information pertaining to the image, which is beneficial in terms of precise localization of polyp boundaries. PSA is designed to capture global contextual information, thereby enhancing the segmentation capability for large or complexly shaped polyps. The element-by-element summation of the two parallel attentions allows for a better fusion of local and global features from the two-branch extraction.

3. The proposed feature complementary module (FCM) in shallow features, the BSConv layer for feature extraction and expansion, and the ECA for capturing channel information, collectively facilitate the preservation of the details of tiny

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

polyps in the cluttered background noise, thereby aiding the shallow MSAF module.

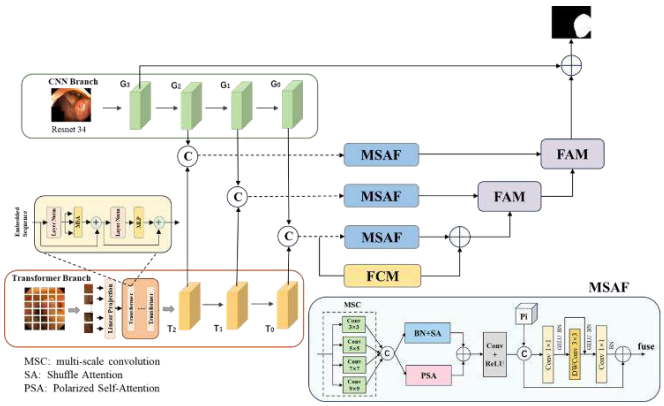


Fig. 1. Overview of MDFF-Net: The network is primarily composed of a dual-branch encoder, the Multi-Scale Attention Fusion (MSAF) module, the Feature Complement Module (FCM), and the Fusion Aggregation Module (FAM).

II. METHOD

A. Dual-Branch Encoder Architecture

The Dual-Branch Encoder Architecture consists of two distinct parallel branches, with the Transformer branch focusing on capturing global features and the CNN branch concentrating on extracting local features. This complementary dual-branch structure enhances the accuracy of polyp segmentation.

The CNN branch consists of five CNN blocks based on the ResNet architecture, with each block containing multiple convolutional layers and performing 2x downsampling on the feature maps, progressively reducing the spatial dimensions of the feature maps.

The lower Transformer branch comprises a linear embedding layer, a stack of eight Transformer layers, a reshaping layer, and two upsampling layers. The output of the Transformer encoder $Z_T \in R^{N\times D_0}$ is layer normalized and reshaped into a tensor $T_0 \in R^{\frac{H}{16}\times\frac{W}{16}\times D_0}$. This tensor is then upsampled in successive operations to produce higher resolution feature maps $T_1 \in R^{\frac{H}{8}\times\frac{W}{8}\times D_1}$ and $T_2 \in R^{\frac{H}{4}\times\frac{W}{4}\times D_2}$. Finally, these multi-scale feature maps T_0, T_1 and T_2 are fused with the corresponding feature maps G_0, G_1 and G_2 generated by the CNN branch to achieve complementary and enhanced multi-scale features.

B. Multi-Scale Attention Fusion module(MSAF)

The core of representation learning lies in how to effectively fuse features. Convolution is usually the preferred method, but it cannot handle long-range dependencies and global contextual information. To address this limitation, we propose a multi-scale attention fusion module (MSAF), as shown in Fig 1. This module mainly consists of multi-scale convolution (MSC), two parallel attention modules, and an inverted residual block, which effectively fuses the diverse features extracted from the CNN and Transformer branches.

1) Multi-Scale Convolution (MSC)

The MSC employs four sets of parallel convolutions with kernel sizes of 3, 5, 7, and 9 to extract multi-scale features. The convolution process involves the use of kernels of varying sizes, which enables the extraction of pertinent feature information from a global perspective and the effective enrichment of image features. In order to reduce the number of parameters in the model, the number of channels after each convolution layer is set to one-quarter of the input feature map. These are subsequently concatenated to form the final output feature map, as illustrated in Fig 1. The complete MSC can be represented as follows:

$$X_i = Concat(G_i, T_i) \tag{1}$$

$$M^i_{pro} = Concat\begin{pmatrix} Conv_{3\times3}(X_i), Conv_{5\times5}(X_i), \\ Conv_{7\times7}(X_i), Conv_{9\times9}(X_i) \end{pmatrix} \tag{2}$$

Where G_i is the output feature of the CNN branch, and T_i is the output feature of the Transformer branch. X_i is the concatenated result of G_i and T_i. M^i_{pro} is the final output feature of the MSC. In this way, the MSC module is capable of effectively fusing and capturing a multitude of intricate feature information through the implementation of multi-scale convolution operations.

2) Parallel Attention Modules

The parallel attention modules are primarily composed of two lightweight attention mechanisms: Shuffle Attention (SA) [14] and Polarized Self-Attention (PSA) [15]. These two attention mechanisms work in parallel, complementing each other. The specific structures of SA and PSA are shown in Fig 1.

The SA module consists of three steps: feature grouping, mixed attention, and feature aggregation. In the polyp segmentation task, SA helps accurately locate the boundaries of polyps by capturing local features and detailed information from the image. The channel shuffling within the module further enhances the communication between different channels, thereby improving the model's sensitivity to local features.

In contrast to SA, PSA employs spatial and channel self-attention branches to assign weights to the spatial and channel dimensions. This enables the model to adaptively select and highlight important features. This is of significant importance for the identification and segmentation of larger or more complexly shaped polyps. PSA employs a polarized filtering mechanism, maintaining the size [H, W] in the spatial dimension while utilizing a size of C/2 in the channel dimension to mitigate the loss of information resulting from dimensionality reduction. Furthermore, the PSA employs nonlinear functions to enhance information. The SoftMax function increases the attention range on the smallest tensor in the attention module, while the Sigmoid function facilitates dynamic mapping.

Finally, the outputs from the saliency attention mechanism and the pop-out saliency attention mechanism are aggregated through an element-wise addition operation, ensuring the retention of significant features extracted by both mechanisms. The SA mechanism is designed to extract fine-grained local features, thereby enhancing the model's ability to recognize

boundaries and small-sized polyps. In contrast, the PSA mechanism is focused on capturing global contextual information, which improves the segmentation of large-sized or complexly shaped polyps. Subsequently, the aggregated feature maps undergo further integration and optimization through convolutional layers, thereby enhancing the overall performance of the model.

3) Inverted Residual block

The input is divided into two parts: one part is the Hadamard product of T and G, denoted as P_h, and the other part is the output value of the parallel attention module, denoted as P_b. Following concatenation, the two parts undergo a 1×1 convolution to expand the dimensionality, a 3×3 depthwise separable convolution for feature extraction, and a 1×1 convolution for dimensionality reduction before being added back to the original input. As illustrated in Fig 1, this design reduces the computational load and parameter count of the model, while simultaneously enhancing its stability and generalization ability.

In the context of polyp segmentation, it is of paramount importance to accurately identify and distinguish polyps from the surrounding tissue. The Shuffle Attention (SA) and Polarized Self-Attention (PSA) approaches each play a distinct role in this task. Their complementarity leads to superior segmentation performance.

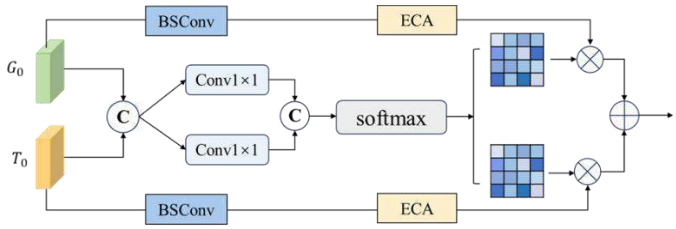


Fig. 2. Detailed structure of FCM. BSConv is the Blueprint Separable Convolution and ECA is Efficient Channel Attention.

C. Feature Complement Module (FCM)

In the context of polyp segmentation, it is of paramount importance to retain the details of small polyps amidst a cluttered background and to suppress irrelevant regions in the input image. The utilization of shallow features can result in a notable enhancement in accuracy. To achieve this, we propose a Feature Complementation Module (FCM) to process shallow features, as illustrated in Fig 2.

Specifically, we take the shallowest features G_0 and T_0 from the two encoders as inputs. First, these two features are concatenated and independently passed through separate 1×1 convolution layers to obtain two distinct weight representations. Subsequently, these representations are further concatenated and transformed into a probability distribution of fusion weights C_s via a SoftMax layer, indicating the relative importance of different features. Next, G_0 and T_0 are processed through the Blueprint Separable Convolution (BSConv)[16] and Efficient Channel Attention (ECA)[17] operations, resulting in enhanced features $G_0^{'}$ and $T_0^{'}$. Finally, $G_0^{'}$ and $T_0^{'}$ are multiplied by C_s and added together to form the final output feature. The corresponding formulas are as follows:

$$C_1 = Conv_{1\times1}\big(Concat(G_0, T_0)\big),$$
$$C_2 = Conv_{1\times1}(Concat(G_0, T_0)) \quad (3)$$

$$C_{s1} = F_{SM}(Concat(C_1, C_2)), \quad C_{s2} = F_{SM}(Concat(C_1, C_2)) \quad (4)$$

$$C_{out} = C_{s1} \cdot G_0^{'} + C_{s2} \cdot T_0^{'} \quad (5)$$

In this context, G_0 and T_0 are subjected to feature extraction and expansion through the BSConv layers, while the ECA attention mechanism captures channel information to obtain richer feature representations. The integration of BSConv layers and the ECA attention mechanism not only enhances the representation capacity of superficial features but also enhances the model's capacity to capture the details of small polyps.

D. Fusion Aggregation Module (FAM)

It effectively combines the multi-scale feature maps obtained from the MSAF, thereby enhancing feature representation and capturing richer contextual and semantic information. This approach efficiently addresses the scale variations present between different polyp images.

Specifically, x_0 is upsampled using bilinear interpolation to match the resolution of x_1. Meanwhile, x_1 first undergoes Squeeze-and-Excitation (SE) channel attention [18], which recalibrates the feature responses between channels, enabling the network to adaptively focus on important channels while suppressing less relevant ones. Then, x_1 is fused with the upsampled x_0. The fused features are then processed through a series of convolutions and added to the output of a residual path, which is then followed by ReLU activation.

$$A_1 = Concat(F_{SE}(x_1), Up(x_0)) \quad (6)$$

$$A_2 = Conv_{3\times3}(Conv_{1\times1}(Conv_{3\times3}(A_1))) + Conv_{3\times3}(A_1) \quad (7)$$

III. Experiments

A. Datasets

According to references [19][20], for fair and direct comparison, the experimental setup is divided into two cases. Following[20], the first case uses the Kvasir dataset, which contains 880 images for training and 220 images for testing. As per Jha et al. [19], the second case uses the ClinicDB dataset, where 550 images are randomly selected for training and 62 images for testing.

B. Implementation Details and Evaluation Metrics

The proposed framework is implemented in PyTorch, utilising an NVIDIA Tesla M40 GPU with 24 GB of memory for training purposes. The training configuration comprises 40 epochs, a batch size of 16, and utilises the Adam optimiser with a learning rate of 1e−4. Both branches of this paper make use of pre-trained models. The convolutional neural network (CNN) branch employs ResNet34 (R34) as the backbone, while the transformer branch utilizes an 8-layer DeiT base model, specifically the pre-trained DeiT-basepatch16-384 model. In order to evaluate the performance of the proposed

MDFF-Net model and compare it with other models, a number of assessment metrics were employed, including mean Dice (mDice), mean Intersection over Union (mIoU), precision rate, and recall rate.

TABLE I. QUANTITATIVE RESULTS ON KVASIR DATASET AND CVC-CLINICDB DATASET. FOR EACH COLUMN, THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Method	Kvasir				CVC-ClinicDB			
	mDice	*mIoU*	*Recall*	*Precision*	*mDice*	*mIoU*	*Recall*	*Precision*
U-net	0.715	0.733	0.831	0.859	0.810	0.755	0.748	0.883
PraNet	0.841	0.744	0.836	0.890	0.899	0.849	0.911	0.912
SANet	0.884	0.882	0.897	0.943	0.929	**0.919**	0.953	**0.961**
Swin-Unet	0.890	0.825	0.906	0.906	0.906	0.849	0.918	0.907
TransFuse	0.918	0.863	0.934	0.922	0.941	0.892	0.943	0.944
BDG-Net	0.915	0.865	—	—	0.916	0.864	—	—
MSRAformer	0.926	0.876	0.926	0.940	0.945	0.897	0.946	0.946
TGANet	0.898	0.833	0.913	0.913	0.936	0.887	0.952	0.944
FAFuse	0.923	0.870	0.930	0.926	0.942	0.893	0.947	0.941
MDFF-Net	**0.931**	**0.883**	**0.938**	**0.944**	**0.951**	0.908	**0.956**	0.948

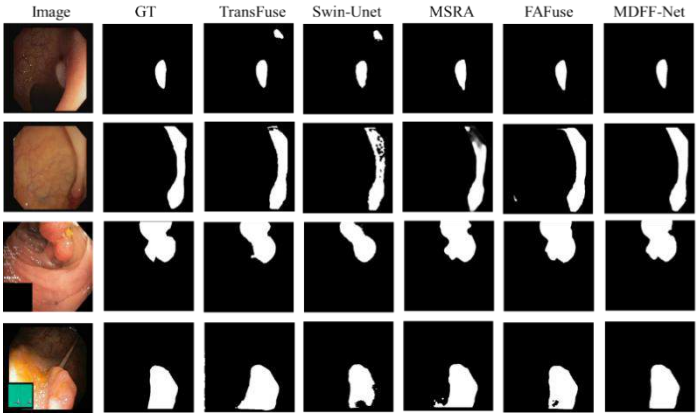


Fig. 3. Visual comparison of polyp segmentation results.

TABLE II. COMPARISON TABLE OF SEGMENTATION RESULTS FOR ABLATION EXPERIMENTS OF OUR MODEL.

Method	Kvasir				CVC-ClinicDB			
	mDice	*mIoU*	*Recall*	*Precision*	*mDice*	*mIoU*	*Recall*	*Precision*
w/o FCM	0.919	0.865	0.924	0.935	0.948	0.903	0.950	0.939
w/o MSAF	0.920	0.868	0.926	0.938	0.942	0.893	0.911	0.930
w/o FAM	0.926	0.873	0.936	0.927	0.948	0.903	0.956	0.940
MDFF-Net	**0.931**	**0.883**	**0.938**	**0.944**	**0.951**	**0.908**	**0.956**	**0.948**

C. Loss function

The proposed network model has been optimized end-to-end using weighted IoU loss and binary cross-entropy loss. The formula for the loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{IoU}^{w} + \mathcal{L}_{BCE}^{w}$$

D. Comparison with the State-of-the-Arts

To further validate the effectiveness of the model, we use the standard U-Net [5] as a baseline, while the remaining competing methods can be broadly categorized into two branches: CNN-based and Transformer-based methods. CNN-based methods include PraNet [25], SANet [26], Swin-Unet [24], and BDG-Net [23], which enhance feature extraction capabilities by improving the CNN structure. Transformer-based methods, such as TransFuse [12], MSRAformer [22], and FAFuse [13], leverage the Transformer architecture to address image segmentation tasks.

As shown in Table I, on the Kvasir dataset, MDFF-Net outperforms the latest methods, including FAFuse [13], TGANet [21], MSRAformer [22], and BDG-Net [23], across all metrics. Specifically, our method achieves 0.931 mDice and 0.883 mIoU, representing improvements of 0.8% and 1.3%, respectively, over FAFuse, which ranked second in most metrics.

According to Jha et al. [19], the second experiment used only the CVC-ClinicDB dataset, and the numerical results are shown in Table I. Again, our proposed MDFF-Net achieved the best results in terms of mDice and Recall. Although mIoU and Precision were slightly lower than SANet, for qualitative analysis, we visualized the prediction results of our method (MDFF-Net) alongside other state-of-the-art networks (TransFuse [12], Swin-Unet [24], MSRAformer [22], FAFuse [13]) in Fig 3. The first two rows are from the CVC-ClinicDB dataset, while the third and fourth rows are from the Kvasir dataset. From the first two rows of Fig 3, it can be seen that, compared to other models, the prediction masks generated by MDFF-Net have boundaries and shapes nearly identical to the ground truth, effectively identifying blurry polyps at the image edges, especially those polyps that resemble the surrounding intestinal tissue in color and structure.

E. Ablation Experiment

Finally, to further investigate the contributions of each component within the MDFF-Net model, we conducted ablation experiments on the Kvasir-SEG and CVC-ClinicDB datasets. We conducted comprehensive ablation experiments by removing individual components one at a time to evaluate their effectiveness in MDFF-Net. The experimental results are presented in Table II, where "w/o" denotes "without." To study the impact of the FAM module, we replaced FAM with simple upsampling and addition. The results show that the mDice scores on the two datasets decreased from 0.931 to 0.926 and from 0.951 to 0.942, respectively. This demonstrates that the aggregation module enhances feature representation and effectively addresses scale differences among different polyp images. The Feature Compensation Module (FCM) is designed to fully utilize shallow information to supplement detailed features in the fusion module. After removing the FCM, mDice decreased by 1.2% and 0.3% on the Kvasir and CVC-ClinicDB datasets, respectively, indicating that the FCM effectively suppresses irrelevant regions in the input images. When the MSAF module was removed, the most significant drop was observed on the CVC-ClinicDB dataset compared to other modules, with mDice and mIoU decreasing by 0.9% and 1.5%, respectively. This validates that the MSAF module successfully fuses both global and local features.

IV. Conclusion

This paper proposes a multi-attention dual-branch feature fusion network (MDFF-Net) for the automatic segmentation of polyps from colonoscopy images. Based on an encoder-decoder architecture, the encoder simultaneously leverages both Transformer and CNN to form a dual-branch structure, effectively extracting global and local features. The MSAF aims to effectively fuse the extracted features by using a multi-scale convolution to capture multi-scale features. It then fully leverages the complementary advantages of two parallel lightweight attention mechanisms to adjust the attention given to the features in different ways, thereby achieving more effective feature fusion. The FCM is used to capture polyp detail information hidden within the shallow features. Finally, the MFA module aggregates the features learned by the decoder. Compared to existing state-of-the-art methods, our model achieves better segmentation performance while demonstrating higher generalization capability.

References

[1] Xi, Y., Xu, P.: Global colorectal cancer burden in 2020 and projections to 2040.Translational Oncology 14(10), 101174 (2021) https://doi.org/10.1016/j.tranon.2021.101174

[2] Testoni, P.A., Notaristefano, C., Marco Soncini, e.a.: An italian prospective multicenter study on colonoscopy practice and quality: What has changed in the last 10 years. Digestive and Liver Disease 55(1), 99–106 (2023) https://doi.org/10.1016/j.dld.2022.09.007.

[3] Mamonov, A.V., Figueiredo, I.N., Figueiredo, P.N., Tsai, Y.-H.R.: Automated polyp detection in colon capsule endoscopy. IEEE transactions on medical imaging 33(7), 1488–1502 (2014)

[4] Maghsoudi, O.H.: Superpixel based segmentation and classification of polyps in wireless capsule endoscopy. In: 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), pp. 1–4 (2017). IEEE

[5] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-assisted intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp. 234–241 (2015). Springer

[6] Yang, L., Zhai, C., Liu, Y., Yu, H.: Cfha-net: A polyp segmentation method with cross-scale fusion strategy and hybrid attention. Computers in Biology and Medicine 164, 107301 (2023)

[7] Liu, G., Chen, Z., Liu, D., Chang, B., Dou, Z.: Ftmf-net: A fourier transform multiscale feature fusion network for segmentation of small polyp objects. IEEE Transactions on Instrumentation and Measurement (2023)

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)

[9] J. Cao *et al*., "WDFF-Net: Weighted Dual-Branch Feature Fusion Network for Polyp Segmentation With Object-Aware Attention Mechanism," in *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 7, pp. 4118-4131, July 2024, doi: 10.1109/JBHI.2024.3381891.

[10] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)

[11] Dong, B., Wang, W., Fan, D.-P., Li, J., Fu, H., Shao, L.: Polyp-pvt: Polyp segmentation with pyramid vision transformers. arXiv preprint arXiv:2108.06932 (2021)

[12] Zhang, Y., Liu, H., Hu, Q.: Transfuse: Fusing transformers and cnns for medical image segmentation. In: Medical Image Computing and Computer Assisted intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24, pp. 14–24 (2021). Springer

[13] Xu, S., Xiao, D., Yuan, B., Liu, Y., Wang, X., Li, N., Shi, L., Chen, J., Zhang, J.-X., Wang, Y., et al.: Fafuse: A four-axis fusion framework of cnn and transformer for medical image segmentation. Computers in Biology and Medicine 166, 107567 (2023)

[14] Zhang, Q.-L., Yang, Y.-B.: Sa-net: Shuffle attention for deep convolutional neural networks. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2235–2239 (2021). IEEE

[15] Liu, H., Liu, F., Fan, X., Huang, D.: Polarized self-attention: Towards high-quality pixel-wise mapping. Neurocomputing 506, 158–167 (2022)

[16] Haase, D., Amthor, M.: Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14600–14609 (2020)

[17] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11534–11542 (2020)

[18] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)

[19] Jha, D., Riegler, M.A., Johansen, D., Halvorsen, P., Johansen, H.D.: Double u-net: A deep convolutional neural network for medical image segmentation. In: 2020 IEEE 33rd International Symposium on Computer-based Medical Systems (CBMS), pp. 558–564 (2020). IEEE

[20] Tajbakhsh, N., Gurudu, S.R., Liang, J.: Automated polyp detection in colonoscopy videos using shape and context information. IEEE transactions on medical imaging 35(2), 630–644 (2015)

[21] Tomar, N.K., Jha, D., Bagci, U., Ali, S.: Tganet: Text-guided attention for improved polyp segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 151–160 (2022). Springer

[22] Wu, C., Long, C., Li, S., Yang, J., Jiang, F., Zhou, R.: Msraformer: Multiscale spatial reverse attention network for polyp segmentation. Computers in Biology and Medicine 151, 106274 (2022)

[23] Qiu, Z., Wang, Z., Zhang, M., Xu, Z., Fan, J., Xu, L.: Bdg-net: boundary distribution guided network for accurate polyp segmentation. In: Medical Imaging 2022: Image Processing, vol. 12032, pp. 792–799 (2022). SPIE

[24] Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-unet: Unet-like pure transformer for medical image segmentation. In: European Conference on Computer Vision, pp. 205 – 218 (2022). Springer

[25] Fan, D.-P., Ji, G.-P., Zhou, T., Chen, G., Fu, H., Shen, J., Shao, L.: Pranet: Parallel reverse attention network for polyp segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 263–273 (2020). Springer

[26] Wei, J., Hu, Y., Zhang, R., Li, Z., Zhou, S.K., Cui, S.: Shallow attention network for polyp segmentation. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24, pp. 699–708 (2021). Springer

Learning Time Synchronization in Wearable Sensor Fusion for Human Activity Recognition

1st Shicheng Zu
Huawei Technologies Co., Ltd
No. 101 Software Avenue
Nanjing, 210000, China
zsc1988419@163.com

2nd Ruize Zu
Nanjing University
No. 163 Xianlin Rd.
Nanjing, 210023, China
shichengzu562@gmail.com

3rd Yucheng Jin
Jiangsu Province Hospital on Integration of Chinese and Western Medicine
Nanjing, 210028, China
jyc950302@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899573

***Abstract*—A wide variety of embedded sensors broaden the versatile functions of the smart mobile devices. Activity recognition is one of them that could be harnessed to track movement for athletes, monitor daily activities for patients and help control a healthy diet. Prior study was performed by comparing the activity recognition performance between the smart-phones and the smart-watches based on accelerometer or gyroscope sensors. The smart-watch sensors were observed to be better than the smart-phone sensors in recognizing the hand-based activities with a higher classification accuracy, especially when eating activities were involved. However, the implementation of sensor fusion can further improve the activity recognition performance. In this paper, a pairwise index selection algorithm is proposed to implement sensors' time synchronization that paces the way for processing sensor fusion. Our empirical results indicate that the sensor fusion can improve the overall classification accuracy achieved by the single sensor in personal and impersonal models, respectively. The sensor fusion of a phone accelerometer and a phone gyroscope can partially overcome the location disadvantage of the smart-phones in recognizing each activity through feature concatenation. Furthermore, sensor fusion can better generalize the characteristics of activities across people in impersonal models.**

***Index Terms*—smart-watches, smart-phones, accelerometer, gyroscope, pairwise index selection, feature concatenation, sensor fusion.**

I. Introduction

Micro-Electro-Mechanical-System (MEMS), the inertial sensors embedded in the smart mobile devices have gained more and more attention in recent years due to their accessibility, small size and low power consumption [1]. Its applications range extensively from medical aid, author identification to gaming and virtual reality. In commercial smart-phones and smart-watches, the accelerometer and gyroscope are often grouped into an Inertial Measurement Unit (IMU) that measures the devices' orientation through the Degrees of Freedom (DOF) [1]. The traditional motion detection devices such as cameras are limited in tracking movement because they are intrusive, have low frame rates and subject to dynamic lighting conditions [2]. These self-contained inertial sensors allow us to track movement at any place and anytime thanks to their portability, higher sampling frequencies and directly involved biometric feature measurement [3]–[5].

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

Accelerometers measure the integrated acceleration of the devices composed of the linear acceleration caused by external forces and the acceleration of gravity [6]. The predisposition to external forces causes the accelerometer to accumulate noise and erroneous jitter in resultant output [3]. Introducing gyroscope signals can help separate the motion acceleration from the acceleration of gravity by serving as a gravity vector [7]. Gyroscopes, which measure the devices' orientation via angular velocity, have a tendency to drift over time because they only sense the rotational changes without a fixed frame of reference [6]. Therefore, the gyroscopic drift will accumulate propagation errors over time. The addition of accelerometer signals can help reduce the gyroscopic drift because the accelerometers can sense the directional change relative to the direction of gravity and help orient the gyroscopes to a more exact location [1]. Thus, combining the long-term accuracy of a gyroscope with the short-term accuracy of an accelerometer, more accurate orientation information could be obtained by counteracting each sensor's disadvantage [8].

Despite that most prior works make use of a filter to implement sensor fusion, how the readings from one sensor match to the corresponding readings from another sensor is not clearly explained [9]. Normally, the sampling frequencies configured for the sensors across devices are different which present as hindrances in sensors' time synchronization. Given a specific time point, the different sensors' readings with the closest timestamps need to be selected and concatenated to characterize the distinctive patterns of a particular activity [10]. A customized sensors' time synchronization algorithm needs to be developed to overcome this challenge.

Using mobile devices with embedded inertial sensors to recognize and monitor daily activities draws a lot of attention in recent years because physical inactivity and unhealthy diets pose a serious challenge to our health care system [11], [12]. Cardiovascular disease, diabetes, hypertension caused by sedentary lifestyle and excessive caloric intake plague millions of people annually to the extent that a portable and accurately predictive application to monitor our daily activities is urgently required [10]. At Wireless Sensor Data Mining Laboratory of the Fordham University, activity recognition models have been built by employing the standard machine learning (ML) algorithms [13]. Prior studies performing on the smart-phone

accelerometers managed to create predictive models based on the time series data in which the data was binned into non-overlapping time intervals [14]. The statistical features derived from the time intervals can be used by the ML algorithms. Recently, research has been undertaken to compare the activity classification performance between the smart-phone and the smart-watch sensors [10]. Empirical results showed that the smart-watch sensors can differentiate various hand-based activities with a higher classification accuracy that cannot be adequately recognized by the smart-phone sensors. The conclusion that the smart-phone-based personal model built with data from the intended users outperform the impersonal model is also applicable in the smart-watches [10]. More importantly, we discovered that the smart-watch can identify different eating activities to monitor our dietary habit [10].

Although we acknowledge that the sensor fusion between smart-watches and smart-phones would yield the best classification accuracy eventually, feature concatenation between different sensors requires sensors' time synchronization to be resolved [10]. Considering the constraint of activity labels on readings matching, the asymmetric reading quantities of each activity, and varying sampling frequencies for different sensors, a pairwise index selection algorithm is developed to realize different sensors' time synchronization which contributes to the implementation of sensor fusion. We expect that certain combinations of sensor fusion can further improve the activity recognition performance achieved by the single sensor especially when the phone sensors are considered because of their location in the pants' front pocket that impedes them from capturing the characteristics of some hand-based activities.

II. Related Work

Sensor fusion is the technique to combine the sensory data from multiple sources so that the resulting performance can accomplish a synergistic improvement [15]. Utilizing sensor fusion to improve the classification accuracy is a common practice in activity recognition [16]. Since differentiating closely related activities such as the hand-based eating activity requires strict accuracy, small drift tolerance as well as high reliability, introducing additional sensors to minimize bias and correct the propagation error will be feasible [10]. Thus far, several sensor fusion-based applications have been implemented in various fields to varying degrees of complexity. Brandon McCarron developed sensor fusion algorithms in the Arduino environment to combine a three-axis accelerometer and a three-axis gyroscope to reduce gyroscopic drift in the pitch and roll axes for both static and dynamic scenarios [1]. This application allows for low-cost hardware implementation of multiple sensors in aerospace navigation. Abyarjoo et al. utilized a Kalman filter to combine the roll and pitch calculated from the accelerometer signals with the gyroscope signals to acquire the non-drifting roll and pitch angles [3], [17]. A tilt compensation unit then combined the corrected roll and pitch angles with the magnetometer signals to calculate the heading of the system to realize 3-D orientation detection. In gesture recognition, Vlasic et al. combined the accelerometer and gyroscope signals with the sonar time of flight to determine the joint orientation [18]. The sonar time of flight measured the distance between sensors to correct the drift of those inertial sensors. Clemens Satzger studied the same area by combining the six-dimensional sensor data using a physics engine [2]. This implementation allows the sensor fusion to calculate the forces and torques of the joint with a higher resolution than is traditionally achieved by optical tracking systems [19].

III. Methods

A. Data Collection

The basis forming the activity recognition models in our study involves binning the time series data into non-overlapping 10 seconds time intervals since raw time series data cannot be used directly by conventional ML algorithms. An activity is recognized correctly if the orientation information within the 10 seconds window contains enough discriminative patterns to differentiate one activity from another.

Before carrying out the data collection, we managed to procure the Certifications in Human Subjects Protections because certain activities probably involved harmful risks (*e.g.*, tripping while jogging). 47 volunteers were invited into the laboratory to perform 18 activities. Table I lists the 18 activities that were performed by users in this study. These subjects were guided to wear the LG G smart-watch on their dominant hands and carry a paired Samsung Galaxy S4 smart-phone in their front pants pocket with the phone oriented upright and the screen facing outward, both of which were running the Android mobile operating system. These subjects then performed every activity in 2 minutes [13].

The data collection software we built on the Android phone controls the smart-watches and allows us to record the users' name, start and stop data collection, label the activities through a Graphical User Interface (GUI). This application also controls the accelerometers and gyroscopes of both devices with each sensor providing orientation values from three spatial dimensions. Each user's activities data was properly curated and sent via Emails to our server for storage.

B. Sensors' Time Synchronization Algorithm

1) Data Preprocessing: The raw time series data that the sensors recorded contains six basic features. They are 'UID,' 'Activity label,' 'Timestamps,' 'X value,' 'Y value,' and 'Z value,' the last three of which are measurements from three spatial dimensions. For accelerometers, the scales of acceleration vector components are in units of acceleration. For gyroscopes, the scales of acceleration vector components are in units of rotational velocity. After using Perl scripts to combine readings of different activity labels into one data frame, the sensors' readings files were downloaded and converted into .csv files [20].

2) Algorithm Design: Our data analysis showed that the four sensors under study did not always record all the activities. In analyzing the data set of some users, one or two activity labels were missing for certain sensors because of the unavoidable technical issue. We defined a set intersection function to

- General activities (not hand-oriented)
  - Walking
  - Jogging
  - Climbing Stairs
  - Sitting
  - Standing
  - Kicking Soccer Ball
- General activities (hand-oriented)
  - Dribbling Basketball
  - Playing Catch with Tennis Ball (two people)
  - Typing
  - Handwriting
  - Clapping
  - Brushing teeth
  - Folding clothes
- Eating activities (hand-oriented)
  - Eating pasta
  - Eating soup
  - Eating Sandwich
  - Eating Chips
  - Drinking from a cup

TABLE I
EIGHTEEN ACTIVITIES UTILIZED IN THIS STUDY

find the common activities and filter out uncommon activities respectively for the sensors examined. Further data analysis revealed that the readings quantities of each activity recorded by different sensors were asymmetrical. Comparing the timestamps of each activity between the watch accelerometer and the phone accelerometer showed that there were extra leading readings in the watch accelerometer that started to record each activity earlier than the first reading in the phone accelerometer and trailing readings that continued to record each activity after the last reading in the phone accelerometer. Therefore, sensors' readings matching is expected to occur between the central part of watch accelerometer readings and the whole range of phone accelerometer readings for each activity label. In processing the sensor fusion of the watch accelerometer and the phone accelerometer, the correct amount of proper readings selected from the watch accelerometer to match to the corresponding readings of the phone accelerometer will be expected for the algorithm design [21].

For certain activity labels, one sensor recorded more readings than the other sensor whereas for other activity labels, the sensor recorded fewer readings than the other sensor. Since the algorithm always selects readings from the sensor with surplus readings to match to the corresponding readings from the other sensor within the scope of activity labels, the data frame of each sensor needs to be partitioned and dealt with separately. After the algorithm selects the proper indices to rebuild the partitioned data-frames, they will be recombined together for each sensor [22].

Because there is a categorical constraint of activity label imposing on the sensors' time synchronization, it is not rigorous to calculate the minimum absolute subtraction value for the timestamps directly. This problem can be solved by grouping the numerical activity label and the timestamp into a tuple. Instead of calculating the minimum absolute subtraction value for the timestamps directly, the pairwise vector distance between two tuple vectors is calculated. After testing on different types of vector distance, the Euclidean and standardized Euclidean distances are able to return the correct indices.

A one-to-one index mapping can be created if the two sensors have the same sampling frequency and similar readings quantities of each activity label. This index mapping usually occurs in the sensor fusion within the same device, *e.g.*, the sensor fusion of a watch accelerometer and a watch gyroscope. However, if the sampling frequencies of the sensors are configured to be vastly different, two scenarios usually occur. If the sampling frequency of one sensor with surplus readings is configured to be larger than the sampling frequency of the other sensor, a one-to-one bijective mapping can still be established. The algorithm can select the appropriate readings at the intervals of 2-3 readings from the sensor with surplus readings to match to the corresponding readings from the other sensor. However, if the sampling frequency of one sensor with surplus readings is configured to be smaller than the sampling frequency of the other sensor, the consequence will be a one-to-two even one-to-three index mapping even though the sensor with surplus readings covers a longer sampling period. What those different index mapping scenarios reflect in the algorithm design is that at every iteration the algorithm selects an appropriate index, this index should not be dropped from the index sequence for the sake of not tampering with subsequent index selection [23].

Because the readings quantities of each activity for different sensors are asymmetric; the sampling frequencies configured for the sensors across devices are different, a pairwise index selection algorithm is designed in Figure 1. Specifically, given two tuple vectors `vector1` and `vector2`, the length of `vector2` is larger than or equal to the length of `vector1` by default. Each tuple contains an activity label and a timestamp. We define a function to compute the Euclidean distance between a tuple vector and a tuple. For each tuple in `vector1`, the algorithm calculates the minimum pairwise Euclidean distance to find the closest tuple in `vector2` and return its index. `vector2` then uses the selected indices to filter out unmatched readings such that the lengths of `vector1` and `vector2` will be the same.

C. Feature Generation

Since the traditional ML algorithms cannot classify the raw time-series data directly, we transformed those raw time series data into examples. To realize this, data was binned into non-overlapping 10 seconds time intervals. Previous experiments demonstrated that 10 seconds are sufficient to capture the periodic motion and distinctive patterns of one complex activity. 10 seconds example then generate 43 high-level statistical features that are all variants of the 6 basic features. Those six basic features are Average, Standard Deviation, Average Absolute Difference, Average Resultant Acceleration, Time between Peaks and Binned Distribution [13].

```
def vector_cdist(vector, tuple):

""" Compute the euclidean distance between
    a tuple vector and a tuple. """

v = tuple.reshape(1, -1)
return scipy.spatial.distance.cdist(vector,
    v, 'euclidean').reshape(-1)

def find_nearest_neighbors(vector1,
    vector2):

""" Two tuple vectors 'vector1' and
    'vector2'. Each tuple contains an
    activity label and a timestamp. For
    each tuple in vector1, the algorithm
    computes the minimum pairwise euclidean
    distance to find the closest tuple in
    vector2. """

vec1, vec2 = map(np.asarray, (vector1,
    vector2))

if len(vec2) < len(vec1):
raise ValueError("The length of vector2
    need to be larger than or equal to
    vector1.")

vec2_idx = np.arange(len(vec2))
vec2 = vec2.copy()
nearest_neighbor = np.empty((len(vec1),),
    dtype=np.intp)
for i, vec1_item in enumerate(vec1):
# Calculate the vector distance, and return
    the index with the smallest euclidean
    distance.
idx = np.array([vector_cdist(vec2,
    vec1_item)]).argmin()
nearest_neighbor[i] = vec2_idx[idx]
return nearest_neighbor
```

Fig. 1. A pairwise index selection algorithm for time synchronization.

D. Model Induction

Our activity recognition models are induced from the labeled examples using the following Waikato Environment for Knowledge Analysis (WEKA) classification algorithms: Decision trees (J48), Random Forest (RF), instance-based learning (iBK), Multilayer Perceptron (MLP) and naïve Bayes (NB). Default settings from WEKA are used for all learning methods except NB, where kernel estimation is enabled, and iBK, where we set $k = 3$ (IB3) [13].

Two types of models are induced: impersonal model and personal model. Each model addresses the learning problem from different perspectives and made different assumptions about how the models are created. Those differences are based on how the data are divided into training and testing data set.

Definition 3.1: Impersonal model. For impersonal models, the training data set and testing data set have no common users. This model has applicable potential because we can make use of existing users' activities information to train an impersonal model and use this model to predict unknown users' activities. In our project, 46 users' activities data constituted the training data sets with one user's data held out for testing. This process was repeated 47 times where every user's data would have an opportunity to be treated as a testing dataset [13].

Definition 3.2: Personal model. Personal models are based only on an intended user's activities information. Intuitively this model should attain the highest classification accuracy because this model does not need to consider the idiosyncrasy associated with different users. As the users perform the activities for an extending period, the activity information to train the personal models would become enriched. As a result, the personal models will be more precise at predicting the next activity the users will be performing. In our project, 47 personal models were created and evaluated for each user [13].

IV. Main Results

The activity recognition results for different sensor fusion in personal and impersonal models are presented in Table II and Table III, respectively. We use the classification accuracy to measure the activity recognition performance of the single sensor and the sensor fusion. The classification accuracy is the percentages of the classification that correctly identify the activity which the subject is performing.

A. Sensor fusion can further improve the overall classification accuracy achieved by the single sensor

We made several observations based on the comparison of the overall classification accuracy between the single sensor and the sensor fusion. Prior study in single sensor for personal models showed that the watch accelerometer achieved the best overall classification accuracy of 91.90%. Table II demonstrates that the sensor fusion of a watch accelerometer and a watch gyroscope can further improve the overall classification accuracy to 92.23% in personal models. Data analysis in readings matching indicated that the watch accelerometer and the watch gyroscope recorded similar amounts of readings (2000 readings) for each activity. Because the sampling frequencies configured for those two sensors were the same (20 Hz), a one-to-one index mapping could be established during the readings matching which optimized the performance of sensors' time synchronization. Therefore, after feature concatenation, 86 high-level statistical attributes could be generated to summarize each activity. Probably this is the reason why sensor fusion can slightly increase the overall classification accuracy achieved by the single sensor. Notably, the sensor fusion across devices did not produce better results than the watch accelerometer because our data analysis revealed that the phone sensors did not record as many readings as the watch sensors did [10]. Therefore, during the readings matching process, significant amounts of readings from the watch sensors needed to be trimmed away in order to match to the corresponding readings quantities of each activity from the phone sensors. Valuable activity information would be lost during the trimming process.

Algorithm	P_a	W_a	W_g	W_{ag}	P_{ag}	W_aP_a	W_gP_g	W_aP_g	W_gP_a	$W_{ag}P_a$	$W_{ag}P_g$	$P_{ag}W_a$	$P_{ag}W_g$	$W_{ag}P_{ag}$
RF	76.30	94.00	73.00	94.18	85.30	94.01	85.23	92.90	89.00	94.22	92.05	92.67	90.54	92.10
J48	65.50	86.10	70.45	86.94	84.05	84.92	76.30	81.90	82.65	81.53	82.70	84.09	82.23	81.80
IB3	67.70	93.30	69.10	93.51	85.24	92.96	81.66	91.44	84.75	92.92	91.04	92.47	89.63	91.59
NB	76.50	91.90	69.50	92.89	86.65	90.77	80.10	90.58	87.07	91.83	90.67	90.66	90.39	91.44
MLP	77.00	94.20	71.00	95.51	87.73	94.33	85.30	92.89	88.88	94.50	92.18	92.80	90.63	92.94
Avg.	72.60	91.90	72.40	92.23	86.17	90.53	82.82	88.60	86.08	90.54	87.80	90.14	88.87	88.97

TABLE II
OVERALL CLASSIFICATION ACCURACY FOR PERSONAL MODEL (%)

The acronym of 'P' stands for 'Phone'; the acronym of 'W' stands for 'Watch'; the acronym of subscript 'a' stands for 'accelerometer'; the acronym of subscript 'g' stands for 'gyroscope'.

Algorithm	P_a	W_a	W_g	W_{ag}	P_{ag}	W_aP_a	W_gP_g	W_aP_g	W_gP_a	$W_{ag}P_a$	$W_{ag}P_g$	$P_{ag}W_a$	$P_{ag}W_g$	$W_{ag}P_{ag}$
RF	35.10	70.30	57.50	74.55	37.93	74.86	58.83	72.72	63.10	74.31	76.25	74.87	58.95	76.64
J48	24.10	59.30	49.60	61.47	36.53	64.52	53.68	72.48	56.12	60.04	71.16	69.80	57.55	66.02
IB3	22.50	62.00	49.30	64.31	33.97	44.09	35.78	53.92	38.88	44.93	53.94	48.34	40.90	48.30
NB	26.20	63.80	53.50	65.89	31.65	59.77	56.10	61.58	57.07	61.83	70.67	60.66	61.39	65.44
MLP	18.90	64.60	57.70	69.76	27.63	58.25	44.60	65.30	49.37	65.14	66.28	62.55	41.60	66.26
Avg.	25.30	64.00	53.50	67.20	33.54	60.30	49.80	65.20	52.91	61.25	67.66	63.24	52.08	64.53

TABLE III
OVERALL CLASSIFICATION ACCURACY FOR IMPERSONAL MODEL (%)

For impersonal models, the best overall classification accuracy achieved by the single sensor is 64.00% for the watch accelerometer. Table III shows that the sensor fusion of a watch accelerometer and a watch gyroscope can further improve the overall classification accuracy to 67.20% in impersonal models. The multi-sensors fusion of a watch accelerometer, a watch gyroscope and a phone gyroscope even increase the overall classification accuracy to 67.66%.

B. Sensor fusion can partially overcome the locational disadvantage of the smart-phones through feature concatenation

The previous conclusion was made that the phone sensors achieved much poor classification accuracy than the watch sensors especially when hand-based activities were considered because of their position in the pants' front pocket. Table II shows that in personal models the sensor fusion of a phone accelerometer and a phone gyroscope can increase the overall classification accuracy from 72.60% to 86.17%. Although an overall classification accuracy of 86.17% still needs to be improved to reach the commercial application level, 13.57% of precision improvement has already been significant. Whereas the pocket location prevents the phone sensors from capturing the distinctive patterns of some hand-based activities, sensor fusion can partially overcome the locational disadvantage by providing more features to characterize each activity. Table III shows that in impersonal models, the sensor fusion of a phone accelerometer and a phone gyroscope also increases the overall classification accuracy from 25.30% to 33.54%.

It is informative to examine the classification accuracy of each activity to determine which activities are easy to classify and which activities are hard to recognize. We focus on models induced by RF algorithm. The classification accuracy of each activity for personal models and impersonal models are presented in Table IV and V. With regards to the personal models, our prior study indicated that for the single sensor, the best classification accuracy was achieved by the watch accelerometer with an accuracy of 93.30% in Random Forest. Table IV shows that the only sensor fusion which attains results better than 93.30% is the sensor fusion of a watch accelerometer and a watch gyroscope with an accuracy of 94.20%. When the classification accuracy of individual activity is examined, the sensor fusion of a watch accelerometer and a watch gyroscope can better recognize certain activities, *e.g.*, taking stairs (88.90% vs 95.65%), kicking (88.70% vs 90.00%), dribbling (98.70% vs 100.00%), catching (93.30% vs 96.43%), and eating sandwiches (68.90% vs 72.46%). Unexpectedly, the classification accuracy of certain activities is decreased in the sensor fusion, *e.g.*, sitting (97.50% vs 95.97%), jogging (99.20% vs 98.17%), handwriting (100.00% vs 99.32%). It is also noted that RF achieves the classification accuracy of 75.50% for the phone accelerometer in personal models, while the sensor fusion of a phone accelerometer and a phone gyroscope can further improve the classification accuracy to 89.55% in RF. This observation accords with the conclusion we made in overall classification accuracy.

C. In impersonal models, sensor fusion can better generalize the characteristics of specific activities across people

Our prior study in the single sensor indicated that the watch accelerometer can achieve the best classification accuracy of 70.30% for impersonal models in RF. Most of the sensor fusion presented in Table V, however, can improve the classification accuracy above 70.03% with the highest accuracy of 76.64% achieved by the sensor fusion of the four sensors in both devices. Impersonal models employ other users' activities information as the training set to predict the test subjects' activities. Probably as feature concatenation doubles, triples or quadruples the attributes to define a particular activity, the idiosyncrasy associated with each user is mitigated, while the motion's characteristics can be

Activity	W_a	P_a	W_g	W_{ag}	P_{ag}	W_aP_a	W_gP_g	W_aP_g	W_gP_a	$W_{ag}P_a$	$W_{ag}P_g$	$P_{ag}W_a$	$P_{ag}W_g$	$W_{ag}P_{ag}$
Walk	94.20	88.50	93.50	98.55	98.82	98.81	97.62	100.00	98.82	100.00	100.00	100.00	100.00	100.00
Jog	99.20	68.80	98.10	98.17	96.43	97.62	97.06	98.80	98.51	98.53	98.53	97.59	97.06	98.53
Stairs	88.90	66.70	80.00	95.65	96.10	91.95	90.91	92.21	94.94	92.41	89.61	92.21	94.74	96.10
Sit	97.50	87.00	82.20	95.97	98.59	97.40	95.52	98.67	97.10	97.10	97.01	98.51	98.31	100.00
Stand	98.10	73.10	68.60	99.37	97.00	100.00	95.74	97.87	95.74	98.94	97.85	96.70	97.80	97.80
Kick	88.70	91.70	67.90	90.00	84.88	98.88	81.48	98.77	92.59	98.77	97.53	96.25	93.75	95.00
Dribble	98.70	84.80	96.90	100.00	86.59	97.70	97.40	100.00	97.33	97.33	100.00	94.87	97.30	98.65
Catch	93.30	78.30	94.60	96.43	85.54	98.82	96.39	98.67	96.47	96.10	98.67	98.67	95.18	100.00
Type	99.40	72.00	88.60	99.37	95.45	98.85	93.02	98.84	94.25	98.85	98.84	98.84	97.67	98.84
Write	100.00	75.90	80.50	99.32	93.90	100.00	91.36	100.00	97.56	98.78	97.50	98.77	98.76	100.00
Clap	96.90	77.30	95.60	96.79	98.86	100.00	97.67	98.84	98.85	100.00	100.00	98.84	100.00	98.84
Teeth	97.30	96.20	89.60	97.30	95.06	98.70	97.53	97.40	98.77	98.70	96.05	94.81	97.50	97.40
Fold	95.00	79.20	73.10	95.37	87.50	93.84	88.41	92.86	92.31	95.38	94.12	96.77	100.00	95.16
Eat pasta	88.60	40.00	72.90	86.43	95.06	95.06	92.50	95.00	97.53	98.77	96.20	96.25	95.00	97.50
Eat soup	90.70	82.40	69.80	90.00	96.39	97.59	91.46	98.78	96.39	97.59	93.83	96.34	95.12	97.56
Sandwich	68.90	63.00	44.20	72.46	96.00	91.67	79.73	87.84	84.52	95.24	87.67	98.65	89.19	90.54
Eat chips	83.40	76.00	52.50	82.80	95.18	95.00	91.25	96.25	85.00	91.25	89.87	97.47	96.20	92.41
Drink	93.30	77.30	78.50	94.67	91.25	97.50	90.00	91.25	95.00	97.50	92.41	97.50	96.25	95.00
Overall	93.30	75.50	79.00	94.20	89.55	92.91	85.38	91.35	89.75	93.29	90.48	93.02	92.48	91.94

TABLE IV
PER-ACTIVITY ACCURACY FOR PERSONAL MODELS (%) IN RF

captured more comprehensively by different types of sensors from various dimensions. When the classification accuracy of each activity is examined between the multi-sensors fusion and the watch accelerometer, most of the activities have enhanced classification performance, *e.g.*, walking (79.80% vs 86.75%), jogging (97.70% vs 100.00%), taking stairs (58.50% vs 74.03%), etc. For some activities, the classification accuracy is decreased, *e.g.*, dribbling (89.30 % vs 83.78%), typing (80.40% vs 79.07%), handwriting (85.20% vs 75.31%), etc.

Of special note, the sensor fusion of a phone accelerometer and a phone gyroscope also improves the classification accuracy of the phone accelerometer from 35.1% to 37.93% in RF impersonal model.

V. CONCLUSION AND FUTURE WORK

Since the embedded sensors within the mobile devices frequently record asymmetrical amounts of readings for each activity and the sampling frequencies configured for different sensors are usually dissimilar, sensor's time synchronization need to be resolved for overcoming these difficulties. In this paper, a pairwise index selection algorithm is developed to realize sensors' time synchronization that paces the way for subsequent sensor fusion. Several conclusions are made by comparing the classification accuracy of the single sensor with that of different combinations of sensor fusion.

First, the sensor fusion of a watch accelerometer and a watch gyroscope can further improve the overall classification accuracy achieved by the single sensor from 91.90% to 92.23% in personal models and from 64.00% to 67.20% in impersonal models. We speculate that as sensor fusion doubles the attributes to characterize the distinctive patterns of specific activities, the activity can better be recognized. More robust activity recognition performance of the sensor fusion across devices is expected as we optimize the data collection software in the future which allows the phone sensors to record more readings for each activity.

Second, due to the smart-phones' disadvantageous location in the pants' front pocket to capture the orientation information, the phone accelerometer has a reduced overall classification accuracy of 72.60% in personal models and 25.30% in impersonal models. However, the sensor fusion of a phone accelerometer and a phone gyroscope further improves the overall classification accuracy to 86.17% in personal models and 33.54% in impersonal models because the sensor fusion can compensate the locational disadvantage by providing more informative patterns to summarize each activity.

Third, most of the sensor fusion can improve the classification accuracy achieved by sensor individuals in impersonal models. As feature concatenation doubles, triples or quadruples the attributes to define a specific activity, the idiosyncrasy associated with each user is mitigated, whereas the motion's characteristics can be captured more comprehensively by different types of sensors from various dimensions.

Since the classification performance of the ML algorithms is largely hinged on how effective the attributes are at representing the time series data, new features can be constructed. We expect those new features can better distinguish different activities. Because the features extracted from one sensor are independent of those obtained from another sensor, it is reasonable to combine them to create a new feature. Besides, feature selection or reduction strategies can be implemented to regularize the ML models [24].

Finally, new activities could be added and tested [25]. We expect as more and more activities are accurately recognized, the application of activity recognition can monitor users' daily activities more thoroughly.

Activity	W_a	P_a	W_g	W_{ag}	P_{ag}	W_aP_a	W_gP_g	W_aP_g	W_gP_a	$W_{ag}P_a$	$W_{ag}P_g$	$P_{ag}W_a$	$P_{ag}W_g$	$W_{ag}P_{ag}$
Walk	79.80	60.70	87.00	76.81	56.47	95.29	78.57	71.43	83.72	83.53	78.57	85.54	86.75	86.75
Jog	97.70	93.80	48.60	97.25	98.81	100.00	95.59	98.80	97.02	97.06	98.53	100.00	98.53	100.00
Stairs	58.50	66.70	43.10	54.04	55.84	67.82	53.25	62.34	73.42	72.15	64.94	71.43	69.74	74.03
Sit	84.90	26.90	70.50	86.58	40.28	91.03	69.12	88.16	94.29	97.14	89.71	83.82	86.67	95.00
Stand	96.30	65.90	57.90	93.08	63.37	96.84	71.58	92.63	91.58	97.89	94.68	95.65	84.78	95.65
Kick	71.30	72.50	41.40	66.43	69.32	84.62	65.85	75.61	85.54	81.93	82.93	83.95	77.78	88.89
Dribble	89.30	26.10	86.00	87.60	29.27	94.25	83.12	95.06	74.67	98.67	90.91	92.31	79.73	83.78
Catch	66.00	26.10	68.90	76.43	38.55	80.23	68.67	76.00	67.44	83.33	81.33	76.00	83.13	78.67
Type	80.40	76.90	60.80	88.05	22.47	71.59	39.53	81.40	42.05	80.68	86.05	72.09	29.07	79.07
Write	85.20	12.90	63.10	89.19	14.63	73.17	54.32	83.95	51.22	75.61	86.25	76.54	41.98	75.31
Clap	76.30	40.90	67.90	75.64	62.50	88.64	76.74	72.09	86.36	88.64	84.52	93.02	86.05	87.21
Teeth	84.50	19.20	66.20	87.84	37.04	84.42	82.72	92.21	83.95	89.61	93.42	81.82	83.75	84.42
Fold	80.80	8.30	37.80	78.70	64.06	89.39	81.16	87.14	90.91	8030	8971	82.26	88.71	83.87
Eat pasta	47.10	0.00	57.90	58.57	9.88	40.74	53.75	41.25	48.15	46.91	68.35	31.25	41.25	55.00
Eat soup	52.70	0.00	47.70	61.33	2.41	54.76	34.15	68.29	47.62	69.05	60.49	57.32	25.61	63.41
Sandwich	29.00	7.10	31.10	42.75	18.42	31.76	24.00	20.00	24.71	38.82	39.19	21.33	29.33	30.67
Eat chips	65.00	16.00	50.60	68.15	3.61	75.31	27.50	66.25	44.44	58.02	68.35	59.49	40.51	58.23
Drink	62.70	31.80	61.10	77.33	18.52	87.65	59.26	64.20	65.43	65.43	67.50	58.02	49.38	80.25
Overall	70.30	35.10	57.50	74.55	37.93	74.86	58.83	72.72	63.10	74.31	76.25	74.87	58.95	76.64

TABLE V
PER-ACTIVITY ACCURACY FOR IMPERSONAL MODELS (%) IN RF

REFERENCES

[1] B. McCarron, "Low-cost imu implementation via sensor fusion algorithms in the arduino environment," 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:18907856

[2] C. Satzger, "Fusion of six dimensional sensor data using physics engines," 2010.

[3] F. Abyarjoo, A. Barreto, J. Cofino, and F. R. Ortega, "Implementing a sensor fusion algorithm for 3d orientation detection with inertial/magnetic sensors," in *Innovations and advances in computing, informatics, systems sciences, networking and engineering*. Springer, 2015, pp. 305–310.

[4] V. B and P. Sasikumar, "Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms," *Sensors and Actuators A: Physical*, vol. 341, no. 113557, 2022.

[5] H. F. Nweke, Y. W. Teh, U. R. Alo, and G. Mujtaba, "Analysis of multi-sensor fusion for mobile and wearable sensor based human activity recognition," in *Proceedings of the International Conference on Data Processing and Applications*, 2018.

[6] S. Blackman, "Introduction to sensor systems, chapter multiple sensor tracking and data fusion," *Artech House, Norwood, Massachusetts*, 1988.

[7] E. Fabrizi, G. Oriolo, S. Panzieri, G. Ulivi *et al.*, "Mobile robot localization via fusion of ultrasonic and inertial sensor data," in *Proceedings of the 8th International Symposium on Robotics with Applications, Maui, USA*, 2000.

[8] A. N. Tarekegn, M. Ullah, F. A. Cheikh, and M. Cheikh, "Enhancing human activity recognition through sensor fusion and hybrid deep learning model," in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. IEEE, 2023, pp. 1–5.

[9] H. Bello, "Unimodal and multimodal sensor fusion for wearable activity recognition," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2024, pp. 364–365.

[10] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, 2016, pp. 426–429.

[11] G. M. Weiss, J. W. Lockhart, T. T. Pulickal, P. T. McHugh, I. H. Ronan, and J. L. Timko, "Actitracker: a smartphone-based activity recognition system for improving health and well-being," in *2016 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 2016, pp. 682–688.

[12] N. Győrbíró, Á. Fábián, and G. Hományi, "An activity recognition system for mobile phones," *Mobile Networks and Applications*, vol. 14, pp. 82–91, 2009.

[13] J. W. Lockhart and G. M. Weiss, "The benefits of personalized smartphone-based activity recognition models," in *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 2014, pp. 614–622.

[14] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[15] K. B. Ng and P. B. Kantor, "Predicting the effectiveness of naive data fusion on the basis of system characteristics," *Journal of the American Society for Information Science*, vol. 51, no. 13, pp. 1177–1189, 2000.

[16] J. W. Lockhart, T. Pulickal, and G. M. Weiss, "Applications of mobile activity recognition," in *Proceedings of the 2012 ACM conference on ubiquitous computing*, 2012, pp. 1054–1058.

[17] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[18] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popovic, "Practical motion capture in everyday surroundings," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 35–es, 2007.

[19] C. V. S. Buenaventura and N. M. C. Buenaventura, "Basic human activity recognition based on sensor fusion in smartphones," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 1182–1185.

[20] L. Gao, A. K. Bourke, and J. Nelson, "A system for activity recognition using multi-sensor fusion," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 7869–7872.

[21] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014.

[22] S. Chung, L. Jiyoun, J. N. Kyoung, K. Gague, and J. Hyuntae, "Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning," *Sensors*, vol. 19, no. 1716, 2019.

[23] H. F. Nweke, Y. W. Teh, U. R. Alo, and G. Mujtaba, "Analysis of multi-sensor fusion for mobile and wearable sensor based human activity recognition," in *Proceedings of the International Conference on Data Processing and Applications (ICDPA'2018)*, 2018, pp. 22–26.

[24] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. Lo, and B. P. Lo, "Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN'2007)*, 2007, pp. 208–212.

[25] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 2154–2159.

A Resource-Friendly Random Number Generation Algorithm for IoT

1st Linshan Shi
State Grid Chongqing Information & Telecommunication Company
Chongqing, China
muyeandmuye@163.com

2nd Changsong Zhao
State Grid Chongqing Shibei Electric Power Supply Branch
Chongqing, China
zhaochangsong_2010@163.com

3rd Min Jin
State Grid Chongqing Electric Power Company
Chongqing, China
heavenkiwi@163.com

4th Zhongyu Xu
State Grid Chongqing Information & Telecommunication Company
Chongqing, China
away_hi@126.com

5th Feng Li
Xiamen University
Xiamen, China
lifeng8425@stu.xmu.edu.cn

6th Yaobin Shen
Xiamen University
Xiamen, China
yaobin.shen@xmu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899605

***Abstract*—Random number generation is an essential component in computer science. In the domain of computer security, the generation of random numbers is particularly crucial, directly impacting the security of systems. Random numbers are categorized into true random numbers and pseudo-random numbers. We only discuss the generation of pseudo-random numbers here. True random numbers are usually collected through physical phenomena and require certain physical equipment support, which is not universal. The generation of pseudo-random numbers typically involves using a fixed algorithm to calculate a pseudo-random number based on an input seed value. The input seed value determines the output random number. On resource-limited devices, using hardware to collect high-quality random seeds requires the addition of appropriate equipment, while using software to generate random numbers needs to consider the computational cost and security. According to the specific situation of resource-constrained devices, we design an algorithm to generate random numbers based on hash function, and analyze its security and computational cost. Meanwhile, we compare our algorithm with the random number generation techniques recommended by the National Institute of Standards and Technology (NIST), demonstrating that our algorithm offers superior computational cost benefits.**

***Index Terms*—Random number generation, security, Internet of Things**

I. Introduction

Random number generation is an essential component in computer science. In the domain of computer security, the generation of random numbers is particularly crucial, directly impacting the security of systems [1], [2]. Random numbers are categorized into true random numbers and pseudo-random numbers; our discussion here is confined to the latter. True random numbers, which are typically derived from physical phenomena, necessitate specific physical equipment and are not universally accessible [3]. The generation of pseudo-random numbers generally relies on a predetermined algorithm to produce a pseudo-random number from an input seed value, with the seed value dictating the resulting random number [4]. On resource-limited devices, using hardware to collect high-quality random seeds requires the addition of appropriate equipment, while using software to generate random numbers needs to consider the computational cost and security, overly complex random number generation algorithms may exceed the processing capabilities of the devices, while overly simple methods could introduce security vulnerabilities. So according to the specific situation of resource-constrained devices, we design an algorithm to generate random numbers based on hash function, and analyze its security and computational cost. Meanwhile, we have compared our approach with the random number generation techniques recommended by the National Institute of Standards and Technology (NIST) [5], demonstrating that our algorithm offers superior computational cost benefits.

The proliferation of Internet of Things (IoT) devices and their extensive internet connectivity has given rise to a multitude of security vulnerabilities. A key factor contributing to these vulnerabilities is the reliance on weak Random Number Generators (RNGs) within IoT communications. These inadequate RNGs have rendered IoT devices a prime target for numerous cyberattacks [6], [7]. Therefore, numerous Random Number Generators (RNGs) have been designed specifically for the Internet of Things (IoT) environment. For example, in [6], the authors used data collected from the accelerometer sensor as a random seed to generate random numbers. In [8], the authors utilized data obtained from a cardiac signal generator as a random seed.

As shown in Table I, we review the authentication protocol in the field of IoT this year and summarize the operations they use, here we do not discuss the basic operations, such as addition, subtraction, multiplication, division, XOR, concatenation, etc. As we can see from Table I, almost all algorithms use hash functions, which is why we chose to use hash functions to generate random numbers.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

Contribution. We design a random number generation algorithm based on hash function for resource-constrained devices in the IoT environment, and analyze the security and performance of this algorithm. We compare our algorithm with HASH_DRBG,CTR_DRBG and HMAC_DRBG proposed by NIST. Our algorithm have certain performance advantages, as shown in Table III.

TABLE I
OPERATIONS USED BY IOT AUTHENTICATION ALGORITHMS.

Protocol	Operation used
[9]	RNG, HASH, DSA
[10]	RNG, HASH
[11]	RNG, HASH, ECC
[12]	RNG, HASH, AES
[13]	RNG, HASH, AES
[14]	RNG, HASH, ECC
[15]	RNG, HASH, AES, ECC
[16]	RNG, HASH, ECC

RNG: random number generator; HASH: hash function; DSA: digital signature algorithm; ECC: elliptic curve cryptography; AES: advanced encryption standard.

TABLE II
NOTATIONS AND TERMINOLOGY.

Notation	Description
UK	Unique key
Cnt	Cumulative count
D	Entropy sample
R	A random number
P_{D}	The repetition probability of D
$\lvert\mathrm{D}\rvert$	The value space of D
$H(\cdot)$	Hash function
L_R	The length of the random number that needs to be output
L_{Hout}	The length of the hash function output data
L_{UK}	The length of the UK data
L_{Cnt}	The length of the Cnt data
$\oplus$	Exclusive-or operation
$\|$	Concatenation operation
$\lceil\cdot\rceil$	Ceil
$\sum$	Sum

II. HASH-BASED RANDOM NUMBER GENERATION ALGORITHM

In this section, we introduce the new hash-based random number generation algorithm.

A. Preliminaries

Notations. To help the discussion in the rest of this paper, related notations and their descriptions are shown in Table II.

One-way hash function. The one-way hash function H(·) has three key properties:

- Collision-resistance: It is impossible to find two distinct inputs $x \neq y$ that produce the same hash value $H(x)$=$H(y)$.
- Preimage resistance: Given a hash output $H(x)$, it is computationally infeasible to determine the original input x.
- Second preimage resistance: Given a specific input x and its hash output $H(x)$, it is computationally infeasible to find another distinct input x', such that $x \neq x'$, yet the hash function produces the same output for both.

Resource-constrained devices. Resource-constrained devices refer to devices with limited computing power that only have essential functional components, such as necessary sensors, necessary communication modules, authentication and encryption modules, etc. Our model aims to reduce the computational cost of generating random numbers without adding new physical devices.

B. The composition of hash inputs

A hash function is a one-way function that produces different outputs for different input data. The output process of a hash function itself has randomness, so according to the definition of a hash function, we only need to ensure that the input data for each hash function is different to guarantee randomness.

Considering the characteristics of resource-constrained devices in the Internet of Things, our hash function input consists of three parts: a unique key UK, a cumulative counter Cnt, and sensor-collected data D. The unique key ensures that different devices will not generate the same random numbers, the Cnt ensures that the same device will not generate the same random numbers, and D serves as a random factor to reduce the probability of repeated input data.

D is not necessary; if a device has no sensors, there will be no data D. The repetition rate D and data space $|\mathrm{D}|$ of the output data of different sensors are different. For example, a tracking sensor will only output 0 or 1, while a soil sensor will output values within the range [0-1000]. We will discuss the impact of different repetition rates of D and $|\mathrm{D}|$ on our algorithm in Section IV.

Three input data are processed through a predetermined operation to yield a result, which is then input into the hash function to obtain a random number. The choice of different operations affects both the security of the random numbers and the computational cost; at the same time, the lengths of the data is also directly proportional to the computational cost. We will discuss this in conjunction with experiments in Section IV.

C. The process of generating random numbers

As shown in Algorithm 1, inputs UK, Cnt, and the collected data D, calculate the result of (UK‖Cnt‖D), and input the result into the hash function $H(\cdot)$, and get the output result R as a random number.

D. Random number output

In this subsection, we discuss the situation where the required length of the random number output, denoted as L_R, is inconsistent with the length of the hash function output, denoted as L_{Hout}.

1) If L_R is equal to L_{Hout}, then the result of the hash function R, is output directly.
2) If L_R is less than L_{Hout}, then the first L_R part of the hash result R is output as the random number.

3) If L_R is greater than L_{Hout}, then multiple rounds of random number generation are continued, concatenating the output results of each round, until the length L_r is satisfied.

Algorithm 1 Random Number Generation

Input: UK, Cnt, D
Output: R
1: T=(UK‖Cnt‖ D)
2: R=H(T)
3: Cnt=Cnt+1
4: **return** R

III. Theoretical Analysis

In this section, we theoretically analyze the collision probability of the new random number generation algorithm. The UK of each device is different, and it ensures that the hash input data of each device is different, so we will mainly discuss the probability of the same device generating a collision in random numbers.

Random oracle. All communicating entities within the protocol have access to a cryptographic hash function $H(\cdot)$, which is both one-way and collision-resistant. We model $H(\cdot)$ as a random oracle, denoted as $\mathcal{O}_H(\cdot)$.

Let L_{UK} and L_{Cnt} are the bit lengths of UK and Cnt respectively, and P_{D} is the repetition probability of D.

Since the UK for each device is fixed, when Cnt is equal and D is equal, the same random number will be output, resulting in a collision. Let there be a total of M devices, the i-th device outputs N_i random numbers, that is, N_i group of Cnt and D, P^i is the probability that i-th device has a collision on the input of (UK‖Cnt‖D). $\lceil\cdot\rceil$ is ceil.

$$P^i \leq \begin{cases} \sum_{j=1}^{r-1} j \cdot 2^{L_{\mathrm{Cnt}}} \cdot P_{\mathrm{D}} & \text{if } r \geq 2 \\ 0 & \text{else} \end{cases}$$

$$r = \lceil \frac{N_i}{2^{L_{\mathrm{Cnt}}}} \rceil$$

Let the probability of collision of random numbers generated by all devices be $P_{collision}$.

$$P_{collision} \leq \sum_{i=1}^{M} P^i$$

Proof. Since we assumed above that the hash is Random oracle, which guarantees that the hash is random for the output data of different inputs, we will only discuss the case where Cnt and D are equal. Since Cnt is not a random value but rather a cumulative count, the probability must be calculated based on each repeated cycle of Cnt. There are a total of $\lceil \frac{N_i}{2^{L_{\mathrm{Cnt}}}} \rceil$ cycles. In the first round, Cnt is completely different because it is continuously accumulating, so there will be no situation where Cnt is equal. In the second round, Cnt begins to repeat, and we only need to calculate the probability of D being equal. For each Cnt, there is a corresponding D in the first round. Therefore, the probability of drawing a D that is equal to the first round in the second round is $1 \cdot P_{\mathrm{D}} \cdot 2^{L_{\mathrm{Cnt}}}$. For the j-th round, the probability is less than $(j-1) \cdot P_{\mathrm{D}} \cdot 2^{L_{\mathrm{Cnt}}}$. When accumulated, it becomes $\sum_{j=1}^{r-1} j \cdot 2^{L_{\mathrm{Cnt}}} \cdot P_{\mathrm{D}}$, where r is equal to $\lceil \frac{N_i}{2^{L_{\mathrm{Cnt}}}} \rceil$. So we have the probability of getting P^i.

Since UK is different, the probability of collision for random number inputs generated by all devices is the sum of the probability of collision for each device, so $P_{collision} \leq \sum_{i=1}^{M} P^i$.

In summary, UK guarantees that the hash input values of different devices are different, on the same device, the first round of Cnt count will not produce the same input, starting from the second round of Cnt count, there is the risk of hash input value collision.

IV. Performance Analysis

We employ an Orange Pi Pc Plus as the simulation platform for resource-limited device, which boasts a 1.3GHz 32-bit Quad-core Allwinner ARM Coretex-A7 and 1GB of RAM. The device runs on Ubuntu with a Linux kernel 3.4.113. For programming, we utilize Golang and its cryptography library. Our experimental code and result data can be accessed at https://github.com/lifeng8425/RNGcode. To facilitate the experiment, we employ SHA-1 hash function.

A. The selection of calculation methods

In this subsection, we discuss how to use UK, Cnt, and D to quickly calculate a result that will be used as input to $H(\cdot)$.

Let's recall that the UK for each device is different, the length of the UK depends on how many devices you have, the length of the Cnt depends on the number and frequency of random numbers you need to generate, and D is the length of data collected by the sensor. Usually, the length of UK will be very long, usually 16 bytes, and the D collected by the sensor will be relatively short, and the length of Cnt should not be too long, because if the length of Cnt is too long, Cnt+1 will involve the addition of large integers, and the calculation cost is high.

We analyze the security and calculation cost of different basic operations. Firstly, we add, subtract, multiply and divide UK,Cnt and D to get data. In this case, the calculation cost is the highest, which involves large number operations and also increases the probability of random number collision, because due to the existence of overflow, different data may get the same result. For example, 8-bit unsigned integer arithmetic, $255 + 10 = 9 = 4 + 5$.

For XOR operations, the collision probability will also increase, because different data for XOR operations, may get the same result. For example, $x \oplus x = y \oplus y$, x and y are unsigned integers, and x is not equal to y.

For the concatenation operation, it increases the length of the input data for the hash function. The computational cost of the hash function is usually directly proportional to the length of the input data. However, when the length of the input data is less than a certain threshold, the hash function performs padding operations. As long as we control the length of the input data to not exceed this threshold, it will not increase the computational cost. As shown in Table III, the execution

TABLE III
CRYPTOGRAPHIC ELEMENT EXECUTION TIME.

	Execution time
$H(\cdot)$ (Enter 16 bytes of data)	$26.809ms$
$H(\cdot)$ (Enter 32 bytes of data)	$26.434ms$
$H(\cdot)$ (Enter 64 bytes of data)	$37.231ms$
H(UK‖Cnt‖D) (Each piece of data is 4 bytes)	$32.433ms$
H(UK‖Cnt‖D) (Each piece of data is 8 bytes)	$34.490ms$
HASH_DRBG	$282.212ms$
CTR_DRBG	$465.563ms$
HMAC_DRBG	$456.331ms$

The execution time is the result of 10^5 executions.

time of the hash function for inputs of 16 bytes and 32 bytes is almost the same. When the input is 64 bytes of data, the execution time increases. At the same time, the concatenation operation does not have the security issues associated with the aforementioned operations.

B. Computational cost comparison

We compare the new random number generation algorithm with the HASH_DRBG, CTR_DRBG and HMAC_DRBG random number generation methods proposed by NIST. The experimental results are shown in Table III. We test the length of UK, Cnt and D for 4 bytes and 8 bytes (in fact, UK, Cnt and D do not have to be the same length). Our algorithm has performance advantages because the method proposed by NIST is more general, pays more attention to security and requires higher computational cost. Our algorithm is specifically designed for resource-constrained devices and therefore has a performance advantage.

C. Collision rate test

Since different parameter lengths will affect the collision probability of the algorithm, in this part, we conducted experiments on different Cnt length, P_D and $|\mathrm{D}|$ to analyze the impact on the collision rate.

We have already discussed in Section III that when the number of executions is less than the size of the Cnt space, the collision will only be generated by the hash function, so we will not discuss the case where the number of executions is less than the size of the Cnt space. We conducted 10^8 experiments, set the Cnt size to 3 bytes, and conducted experiments on different P_D and $|\mathrm{D}|$ respectively.

At the same time, according to the conclusion of Section III, the number of collisions generated by each experiment is theoretically predicted. The predicted results and the actual experimental results are shown in Table IV.

When the length of Cnt is 3 bytes, the number of executions is greater than the value space of Cnt, and collision may occur. When the length of Cnt is 4 bytes, the number of executions is smaller than the value space of Cnt, and collision will not occur (the probability of hash collision can be ignored). We fixed the length of Cnt as 3 bytes and discussed different $|\mathrm{D}|$ and P_D. It can be seen from the table that the smaller P_D is, the smaller the number of collisions; the larger $|\mathrm{D}|$ is, the smaller the number of collisions. In addition, we can find that the number of collisions occurring in the experiment is significantly smaller than the number of collisions occurring in the theory, because the probability discussed in our theory is amplified. We directly replace the probability of each D occurring after the second round of Cnt with P_D, and the actual probability will be smaller than P_D.

We used 3-byte Cnt to analyze the collision situation. In fact, the Cnt length is not so small. As shown in Table III, the increase of Cnt length still has a large performance advantage.

D. Randomness test

We utilize the NIST SP800-22 test suite [5], published by NIST, for randomness testing. The NIST SP800-22 test suite encompasses 15 statistical test items, which are designed to detect whether the generated binary sequences exhibit randomness. These test items cover a variety of statistical characteristics, including frequency distribution, runs distribution, block frequency distribution, and Fourier transform properties. By executing these tests, we can evaluate whether the sequences generated by random number generators meet the requirements for randomness. The NIST SP800-22 test suite employs the P-value as an evaluation metric. The P-value indicates the degree of difference between the observed test statistic and the expected statistic for a random sequence. If the P-value is less than the significance level (typically set at 0.01), the sequence is considered to fail the randomness requirement for that particular test item [17]. The test results are presented in Table IV.

TABLE IV
COLLISION RATE AND RANDOMNESS TEST.

Cnt Length (bytes)	$\|D\|$	P_D	Theoretical number of collisions	Number of experimental collisions	Randomness evaluation result
3	100	75%	62417088	10994389	15/15, for P_value $>$ 0.01
3	100	55%	45772531	6186124	15/15, for P_value $>$ 0.01
3	100	35%	29127974	4310604	15/15, for P_value $>$ 0.01
3	100	15%	12483417	3182748	15/15, for P_value $>$ 0.01
3	1000	75%	62417088	1163696	15/15, for P_value $>$ 0.01
3	1000	55%	45772531	688424	15/15, for P_value $>$ 0.01
3	1000	35%	29127974	465434	15/15, for P_value $>$ 0.01
3	1000	15%	12483417	327334	15/15, for P_value $>$ 0.01
4	1000	100%	0	0	15/15, for P_value $>$ 0.01

The execution time is the result of 10^8 executions.

V. CONCLUSION

We design a resource-friendly random number generation algorithm specifically for the Internet of Things (IoT) environment, particularly tailored for resource-constrained devices. Our algorithm leverages hash functions to produce pseudo-random numbers, achieving a balance between computational efficiency and security. Through theoretical research and experimental validation, we have demonstrated that our algorithm outperforms the traditional approaches recommended by the National Institute of Standards and Technology (NIST) in terms of computational cost and passes the randomness tests conforming to the NIST SP800-22 standard.

ACKNOWLEDGEMENTS

This research was funded by the Key Research and Development Project of State Grid Chongqing Electric Power Company with funder grant number (2024 Chongqing Electric Science and Technology 55#).

REFERENCES

[1] Miguel Herrero-Collantes and Juan Carlos García-Escartín. Quantum random number generators. *Reviews of Modern Physics*, 89:015004, 2016.

[2] Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. A guideline on pseudorandom number generation (prng) in the iot. *ACM Computing Surveys (CSUR)*, 54:1 – 38, 2020.

[3] Karsten Nohl, David Evans, Starbug, and Henryk Plötz. Reverse-engineering a cryptographic rfid tag. In *USENIX Security Symposium*, 2008.

[4] Raja Naeem Akram, Konstantinos Markantonakis, and Keith Mayes. Pseudorandom number generation in smart cards: An implementation, performance and randomness analysis. In *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–7, 2012.

[5] Lawrence E. Bassham, Andrew L. Rukhin, Juan Soto, James Nechvatal, Miles E. Smid, Elaine B. Barker, Stefan Leigh, Mark S. Levenson, Mark G. Vangel, David Banks, N. Alan Heckert, James F. Dray, and San C. Vo. Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. 2010.

[6] Siang Lee Hong and Chang Liu. Sensor-based random number generator seeding. *IEEE Access*, 3:562–568, 2015.

[7] Alexandra Balan, Titus Balan, Marcian Cirstea, and Florin Sandu. A puf-based cryptographic security solution for iot systems on chip. *EURASIP Journal on Wireless Communications and Networking*, 2020, 11 2020.

[8] Carmen Camara, Pedro Peris-Lopez, Honorio Martin, and Muawya aldalaien. Ecg-rng: A random number generator based on ecg signals and suitable for securing wireless sensor networks. *Sensors*, 18:2747, 08 2018.

[9] Mostafa Ayoubi Mobarhan and Muhammed Salamah. Reps-aka5: A robust group-based authentication protocol for iot applications in lte system. *Internet of Things*, 22:100700, 2023.

[10] Vincent Omollo Nyangaresi and Ganesh Keshaorao Yenurkar. Anonymity preserving lightweight authentication protocol for resource-limited wireless sensor networks. *High-Confidence Computing*, 4(2):100178, 2024.

[11] Subramanian Jayashree and Sripathi Venkata Naga Santhosh Kumar. Lapep—lightweight authentication protocol with enhanced privacy for effective secured communication in vehicular ad-hoc network. *Wireless Networks*, pages 1–28, 2023.

[12] Liping Zhang, Lanchao Zhao, Shuijun Yin, Chi-Hung Chi, Ran Liu, and Yixin Zhang. A lightweight authentication scheme with privacy protection for smart grid communications. *Future generation computer systems*, 100:770–778, 2019.

[13] Yi Li. An improved lightweight and privacy preserving authentication scheme for smart grid communication. *Journal of Systems Architecture*, 152:103176, 2024.

[14] Seyed Hamid Baghestani, Farokhlagha Moazami, and Mahdi Tahavori. Lightweight authenticated key agreement for smart metering in smart grid. *IEEE Systems Journal*, 16(3):4983–4991, 2022.

[15] Shehzad Ashraf Chaudhry, Khalid Yahya, Sahil Garg, Georges Kaddoum, Mohammad Mehedi Hassan, and Yousaf Bin Zikria. Las-sg: An elliptic curve-based lightweight authentication scheme for smart grid environments. *IEEE Transactions on Industrial Informatics*, 19(2):1504–1511, 2022.

[16] Akber Ali Khan, Vinod Kumar, and Musheer Ahmad. An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach. *Journal of King Saud University-Computer and Information Sciences*, 34(3):698–705, 2022.

[17] Qiang Zhao, Wenhan Zheng, Xiaojin Zhao, Yuan Cao, Feng Zhang, and Man-Kay Law. A 108 f2/bit fully reconfigurable rram puf based on truly random dynamic entropy of jitter noise. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(11):3866–3879, 2020.

EXNet: An Improved U-Net Architecture for Accurate Sperm Segmentation Through Spatial Feature Extractor and Multi-scale Attention

1st Naqy Ul Hassan
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
naqi@mail.ustc.edu.cn

2nd Xingfu Wang*
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
Wangxfu@ustc.edu.cn

3rd Fuyou Miao*
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
mfy@ustc.edu.cn

4th Taiyaba Qureshi
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
Taiyaba@mail.ustc.edu.cn

5thMuhammad Hamza
School of Computer Science and Technology
University of Science and Technology of China
Hefei, China
mhamza@mail.ustc.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899744

***Abstract*—Sperm morphology is crucial to male infertility diagnosis. Male infertility diagnosis requires better sperm morphology for assessing sperm quality. Recent advancements have significantly improved the ability to segment and evaluate sperm cells. Despite the improvements, a significant research gap remains in achieving accurate sperm segmentation, particularly in optimizing performance metrics. We propose an improved U-Net architecture for accurate sperm segmentation through spatial feature extractor and multi-scale attention "EXNet" to solve the segmentation challenges employed in human sperm segmentation. We employed the U-Net network architecture and a unique extractor and multi-scale attention mechanism. The attention mechanism extracts and refines the relevant spatial dimension features on multi-scale. The Extractor module aggregates information across multiple spatial locations in a feature map. A double convolutional down-sampling module and attention-based up-sampling extract and refine the spatial feature to improve segmentation for precise sperm component identification under difficult imaging circumstances. With limited computing power, our method outperforms the other U-Net methods in the separation of human sperm correctly. The model had a 95.14% DSC and a 65.63% MIOU following examination of sperm images. Performance and computational efficiency are balanced in EXNet. The findings show EXNet's reproductive health automated sperm analysis.**

I. Introduction

Clinically, infertility means the inability to fold a child's tissue after one year of common unshielded sexual intercourse [1]. It affects many people, and different things determine whether it happens or not. Research shows that every ten years, male conception drops by 10 to 15 percent. Male birth rates could drop by 30 to 50 percent by 2040 compared to what they are now. This makes me worry about sexual health and the way populations change over time [2].

Male fertility testing procedures have not improved, despite a concerning increase in the number of infertility cases. Traditional sperm analysis methods may be difficult to access outside of fertility clinics, which may hinder individuals from gaining a more comprehensive understanding of their reproductive health. [3] Currently, there is a requirement for technology that facilitates the precise and effortless analysis of semen. These developments empower individuals by providing essential reproductive information without the need for professional assistance [4].

Based on analysis, the morphology of the sperm cell plays a crucial role in evaluating the probability of fertility restoration in the sample [5]. A well-controlled assessment of sperm morphology according to the WHO (2010) human semen analysis handbook will help categorize normal or defective sperm. Additionally, it can detect male infertility causes and provide crucial support for reproductive technologies [6]. Hence, objective examination of aberrant sperm morphology is crucial for sperm evaluation [7].

Early computer-aided sperm morphometric studies helped to process images thereby enabling sperm identification and localisation. The study made use of threshold, edge, and region-based segmentation (techniques [8], [9]). These techniques require various well-accepted criteria for precise segmentation and region of interest extraction. Microscope images of sperm

*Corresponding authors.

depend on light, color, contaminants, and tissue cells. This increases the damage done. It is useless to adjust criteria depending on goal picture modifications. This explains underwhelming segmentation findings [10]. To address these challenges, this study aims to use and optimize a complex deep convolutional neural network architecture (U-Net) to achieve precise segmentation of sperm cells.

This work implements a novel approach that surpasses the previous methods. We enlist our contributions as:

1: We prepared a comprehensive Human Sperm Cell Segmentation Dataset (HSCS-DS) comprised of 1,207 annotated images for the training and testing segmentation models.

2: We revised the U-Net as EXNet for sperm cell segmentation. It employs novel attention mechanisms, dual convolutional blocks, and an extraction module to facilitate sperm segregation and feature identification.

3: The attention mechanism focuses on extracts and refines the relevant spatial dimensions features, whereas the extraction module aggregates information across multiple spatial. This helps in the separation of overlapping sperm cells effectively.

4: EXNet outperforms U-Net in segmentation, with a 95.14% Dice Similarity Coefficient and a 65.63% Mean Intersection over Union. This aids in challenging photographic circumstances.

The succeeding sections of our investigation will be delineated as follows. We reviewed numerous scholarly studies that concentrated on the segmentation of sperm cells in the **Second 2**. The dataset and our methodology are comprehensively described in **Section 3**. **Section 4** offers a thorough analysis of the results of our methodology. **Section 5** delineates the conclusions.

II. Related work

Recent improvements in human sperm segmentation have had a significant impact on the field of reproductive medicine, particularly in the area of fertility assessment. This article summarizes the most important research in this field. Particular attention has been paid to the methods, results, and constraints of these studies.

Researchers created an automated sperm shape and structure estimate tool. Use example-aware component distribution. This study shows that quantitative male fertility morphology evaluation requires accurate segmentation. [11]. A correct evaluation of male infertility involves sperm morphology categorization, according to the research. According to the findings, low-quality or overlapping sperm images have limitations. Sometimes segmentation and morphology analysis fail.

YOLOv5 exceeds previous models in accuracy and speed for identifying and segmenting human sperm cells [12]. YOLOv5 blends real-time speed and performance following CNN architecture exploration. Resources may exceed low-resource clinics. The model can be expanded or reduced to conserve computer resources.

Sperm YOLOv8E-TrackEVD model research is outstanding. Consider adding attention processes to improve feature extraction and identification accuracy, particularly for small and ambiguous sperm characteristics [13]. The model may focus on subtle aspects that standard models miss with this integration. The research highlights that sperm samples with substantial morphological defects or variable lighting may be less effective. This may reduce model detection rates, indicating unpredictability enhancements are needed.

Transfer learning was studied for sperm segmentation elsewhere. These methods improve segmentation accuracy and reduce the need for broadly labeled data, one research found. [14]. The essay claims transfer learning uses pre-trained models' massive dataset expertise. We can separate sperm well. Transfer learning works best when the pre-trained model trains on the analysis dataset. Transfer learning may fail due to sperm morphology or imaging conditions.

Despite species differences, deep learning can enhance human sperm segmentation for bull sperm [15]. This technique recognizes sperm in many-person films. This may improve human sperm testing comparisons. The research states that animal physiological and behavioral differences make this model difficult to adapt to human sperm. Human sperm needs extensive model improvements to accurately capture its unique traits.

A study used deep learning to identify healthy sperm heads. Automated sperm analysis is rapidly expanding [16]. Studies show that this method reveals important sperm markers, improving analytical accuracy and effectiveness. It shows imaging area debris and artifacts that may affect model accuracy. These artifacts may be mistaken for sperm heads, resulting in inaccurate or missing data. Which stresses artifact detection and filtering require improvement.

The clinical use of these technologies was examined. During ICSI, researchers used machine learning to detect anomalies and divide sperm [17]. The study shows these algorithms can identify problems. Picture quality and abnormalities greatly impact performance. Variability may lower detection.

Due to advances in machine and deep learning methods for better accuracy and performance, human sperm segmentation is evolving quickly. These methods improve delivery accuracy and efficiency. In the next section, we will discuss how our recommended deep-learning techniques increase accuracy and speed by offering multiple processing routes. It also handles complex multi-sperm cell images well.

III. Material and Methods

A. dataset

An enormous dataset called the Human Sperm Cell Segmentation Dataset (HSCS-DS) was created to improve sperm morphology studies and deep learning. Figure 1 shows a sample dataset of images with labeled masks. Over 10,000 sperm cells were extracted at 100x magnification from over 15 people in Pakistan by WHO guidelines from 2010. 1,007 of the 1,207 labeled photos in the dataset were used for training, while 200 were used for testing. Dataset marked under physician supervision, medical experts integrated them, using tool Labelme taking into account noise and image quality (1280 × 1024 pixels).

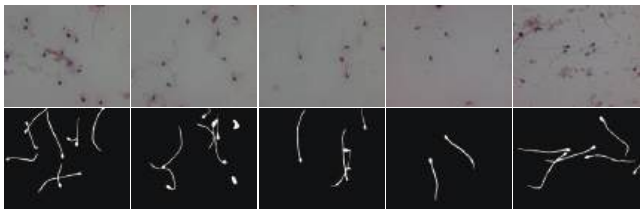


Fig. 1. A sample dataset images with their corresponding masks.

B. Model Architecture

The U-net [18] encoder-decoder architecture is improved by the proposed architecture for better sperm segmentation. The model includes DC double convolutional blocks, attenuation blocks, and an Extractor module. Each contributes to segmentation map robustness in extracting and reconstructing local and global features at multiple levels.

1) Extractor: To enrich deep neural network feature representations, the Extractor module aggregates information across multiple spatial locations in a feature map. These tasks require context and relationship understanding between image regions. This module emphasizes simplification, as per the study [19].

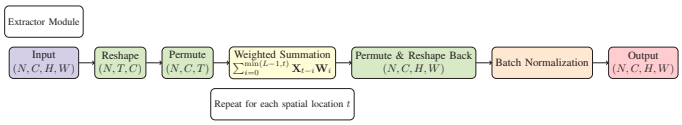


Fig. 2. A simplified Extractor Module

The extractor module is a learnable component that works with dimensions on a given input tensor as shown in Figure 2. (N, C, H, W), where N is the batch size, C is the number of channels, and $H \times W$are local dimensions. It processes the input tensor through a series of weighted sums, where each local location is influenced by other locations within a defined range, or "length."

Weight Matrices: The Extractor uses a set of learnable weight matrices $\mathbf{W}_i$. Each matrix has dimensions $(C \times C')$, where C is the number of input channels, and C' is the number of output channels.

Weighted Summation: The Extractor calculates a weighted sum of nearby features for each input tensor position using contextual data from surrounding pixels and pixel data to capture image data dependencies and relationships. For each t in the temporal/spatial dimension T, we compute the output $\mathbf{Y}_t$ as a weighted sum of the input tensor $\mathbf{X}_j$ at previous positions $j \leq t$, up to a maximum of L steps back:

Feature Extraction and Batch Normalization

The feature extraction process can be summarized by the following tensor notation:

$$\mathbf{Y}_{t,c} = \sum_{i=0}^{\min(L-1,t)} \sum_{c'=1}^{C} \mathbf{X}_{t-i,c'} \cdot \mathbf{W}_i[c', c] \tag{1}$$

where $\mathbf{X}$ is the input tensor, $\mathbf{W}_i$ are the weight matrices, c and c' index the output and input channels respectively, and L is the length of the filter.

After the feature aggregation, the output tensor $\mathbf{Y}$ is normalized as follows:

$$\mathbf{Y}_{\text{norm}} = \frac{\mathbf{Y} - \mu_{\text{batch}}}{\sqrt{\sigma^2_{\text{batch}} + \epsilon}} \cdot \gamma + \beta \tag{2}$$

where μ_{batch} and σ^2_{batch} are the batch mean and variance, and γ and β are learnable parameters for scaling and shifting. The Extractor module reduces computational bottlenecks by using fewer channels and appropriate weight matrices to prioritize essential components for precise segmentation and faster inference processing. Its exceptional ability to extract essential features and reduce unnecessary data improves segmentation accuracy, especially under challenging imaging conditions.

2) Attention Block: Our model aims to enhance subtask accuracy, with **AttentionBlock** playing a crucial role. This process concentrates the model's attention on the functional map's key components while minimizing irrelevant factors. Complex structures like sperm cell components require high concentration to divide. The AttentionBlock processes two inputs: the extractor's "Input Properties Map (x)" with detailed spatial information but a limited receiving field, and the "Getting Signal (g)" with a wider background after processing through multiple network layers. These inputs are called "function maps."

The block diagram shown in Figure 3 processes input via two parallel curved routes. To shrink g, use a 1×1 exchange rate and batch normalization along its route $g(W_g)$. The $x(W_x)$ approach likewise employs the 1×1 exchange rate, followed by group normalization to decrease the complexity of x. Initial addition and ReLU activation combine function maps after processing. Several strategies have been tried to increase integrated signal quality for attention charts. These methods use 1×1, group normalization, and signal function. Finally, the focus map will be multiplied by the element's input character map x. As researchers, we prioritize vital components and minimize irrelevant ones.

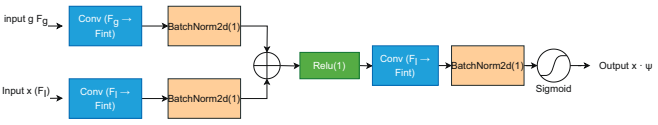


Fig. 3. Proposed Attention block

3) Our Approach : **EXNet** The model represents an updated and improved neural network architecture designed specifically for the complex tasks of sperm distribution. The model extends the basic U-Net architecture by adding state-of-the-art modules such as the Double Convolutional Block, the Attention Block, and the Extractor Module, each of which significantly improves distribution accuracy and computational efficiency.

Initially, the input was processed through a double-coherent (double-connected) block consisting of two coherent layers as shown in Figure 4. Each layer has a batch normalization and a RELU activation function, which collectively translates the input character map into a more complex and informative representation. This change can be expressed mathematically as follows:

$$X_{\text{out}} = \text{ReLU}\left(\text{BN}\left(\text{Conv}_2\left(\text{ReLU}\left(\text{BN}\left(\text{Conv}_1(X)\right)\right)\right)\right)\right) \quad (3)$$

Here, X denotes the input feature map, while Conv_1 and Conv_2 represent the first and second convolutional layers, respectively. BN and ReLU correspond to the batch normalization and ReLU activation functions applied after each convolution.

The DoubleConv block takes an input feature map (C_{in}, H, W). The first 3×3 convolution produces an intermediate feature map (C_{mid}, H, W), where C_{mid} typically equals C_{out}. The second convolution, batch normalization, and ReLU activation refine the feature map to (C_{out}, H, W).

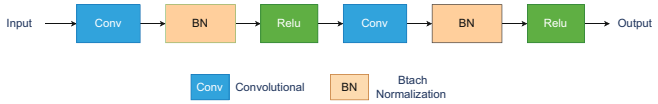


Fig. 4. DoubleConv purposed convolutional block with batch normalization follow with Relu

In decoding or upsampling, the feature map expands using transposed convolution, reducing channels while increasing spatial dimensions. It's then merged with the encoding route's feature map using skip connections. Adding a DoubleConv block improves output, and an AttentionBlock selects significant regions.

The Extractor module maximizes computational resources and preserves important feature information in the network's constrained region. It reduces content size while retaining characteristics using weight matrices. The mathematical operation performed by the Extractor can be defined as:

$$Y_i = X \times W_i \quad (4)$$

where X is the reshaped feature map, and W_i denotes the learned weight matrices. These approaches build a compact yet full feature map by grouping, updating, and standardizing. Prioritizing crucial information increases segmentation accuracy and reduces model computational cost at this vital network point. Transposed convolutions upsample feature maps for segmentation map reconstruction. Skip links link each level's feature map downsampling and upsampling. Introducing the AttentionBlock at appropriate times can focus the model on critical geographic locations. Greatly improved segmentation accuracy. AttentionBlock builds an attention map from the gating signal and input feature map using the equation:

$$\text{Feature Map} = \text{Conv}_1\left(\text{input Feature Map}\right) \quad (5)$$

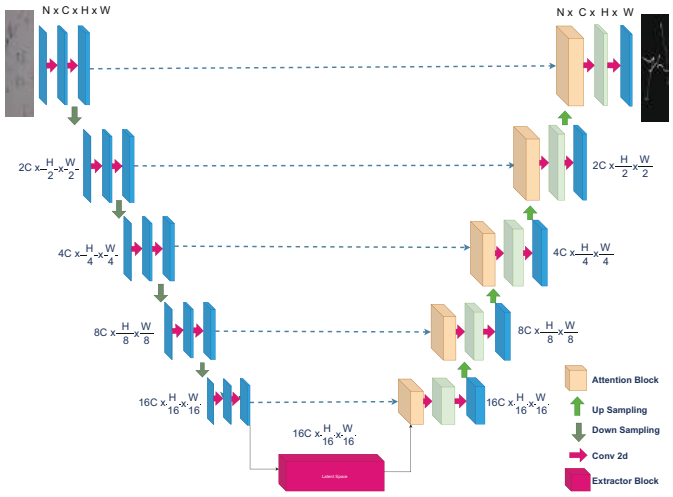


Fig. 5. Proposed EXNet model with Extractor at bottleneck and attention in upsampling

$$\text{Attention Map} = \sigma\Big(\text{BN}\big(\text{Conv}_3\big(\text{ReLU}\big(\text{BN}\big(\text{Conv}_2(\text{GS}) + \text{Feature Map}\big)\big)\big)\big)\Big) \quad (6)$$

Figure 5 illustrates the proposed U-Net architecture EXNet with DoubleConv during downsampling, a suggested extractor at the bottleneck, and an attention mechanism during upsampling. EXNet produces precise output by evaluating feature maps and enhancing segmentation with DoubleConv blocks. The model's computational performance aligned effectively with load and performance, rendering it appropriate for practical use. Our method enhances sperm image segmentation, rendering it significant for reproductive health research. Performance results from feature extraction and a well-structured architecture. A subsequent study will investigate its use in other biological entity segmentation tasks to confirm its adaptability and effectiveness.

IV. EXPERIMENTS AND RESULTS

Experimental Setup

The results indicated that our sperm segmentation model functioned effectively in a potent computing environment. We utilized a CPU designated as Intel Core i7-10700K, a GPU designated as NVIDIA GeForce RTX 3080 8 GB, and 16 GB of RAM to conduct our training.

For training, we implemented the CosineAnnealingLR learning scheduler, the Adam optimizer, a learning rate of 0.001, a batch size of 4, and 200 epochs. To prevent overfitting, we implemented numerous data augmentation techniques, resulting in a collection expansion from 765 to 3825 images.

Quantitative Metrics

The model's performance was evaluated using IoU, Dice coefficient, precision, and recall. IoU was calculated as $\frac{|A \cap B|}{|A \cup B|}$, while the Dice coefficient was computed as $\frac{2|A \cap B|}{|A|+|B|}$. Precision

was defined as $\frac{TP}{TP+FP}$, and recall as $\frac{TP}{TP+FN}$, providing comprehensive insights into segmentation accuracy.

The sperm segmentation methodology's high precision is demonstrated by the experiment data. The model exhibits remarkable precision in the identification of animal cells and effectively mitigates false positive results.

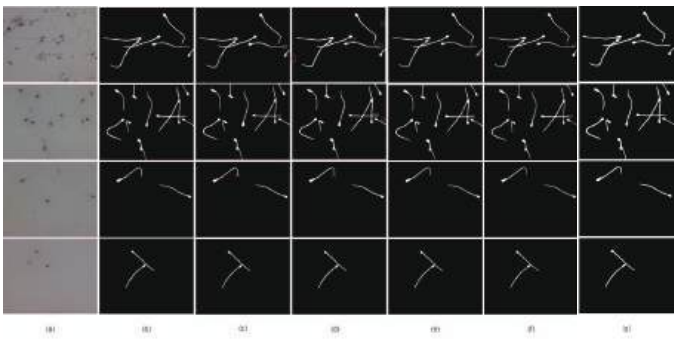


Fig. 6. The image illustrates the quality outcomes of a sperm segmentation task, evaluating the efficacy of several approaches. Subfigure (a) displays the original image, whereas (b) ground truth, (c)U-Net, (d) ResUNet, (e) Attention U-Net, and (f) YOLOv8. (g) our proposed method.

The practicality of the approach is demonstrated by the sample division results in Figure 6, which illustrate the model's extraordinary capacity to accommodate a wide range of sperm cell densities and image conditions. The Areas where the cells are improperly segmented are indicated by red specks in specific images, particularly those depicting sperm tails. This occurrence is less likely to occur when our technique is employed, as evidenced by the improved accuracy of tail segmentation in Figure 6(g).

TABLE I
COMPARISON OF DIFFERENT METHODS FOR SPERM SEGMENTATION

Methods	DSC (%)	MDSC (%)	MIOU (%)	Recall (%)	Precision (%)	F1-score
U-Net	92.01	73.36	62.07	65.12	92.68	72.44
ResUNet [20]	92.16	77.54	61.72	65.03	93.19	76.60
Attention U-Net [21]	93.12	73.98	64.30	65.89	92.05	75.65
YOLOv8 [13]	93.12	73.98	64.30	65.89	92.05	75.65
Ours	**95.14**	**76.17**	**65.63**	**66.41**	**92.25**	**78.54**

Our model was thoroughly compared to other well-established models to ensure its efficiency. Table 1 shows that every statistic changed, proving our method worked. Performance measurements suggest that adding data considerably enhanced the model's generalization.

TABLE II
COMPARISON OF PERFORMANCE METRICS WITH BASELINE MODELS

Methods	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	AUC-ROC(%)
UNet + attention	92.11	88.45	65.05	71.87	89.5
UNet + conv	91.68	90.28	62.97	70.9	90.71
Proposed Model	**95.75**	**93.06**	**65.15**	**78.54**	**95.84**

To confirm that our strategy worked, we compared our results to the original U-Net model and the residual structure model. Table II presents statistical findings. Based on the data, our model had the greatest DSC (95.14%), MIOU (65.63%), and F1-score (78.54%) on the validation and test sets. Recognizing that higher numbers signify better performance is key. When accuracy is great, memory is poor, and vice versa. Incorporating these two criteria, the F1-score provides a complete model performance assessment.

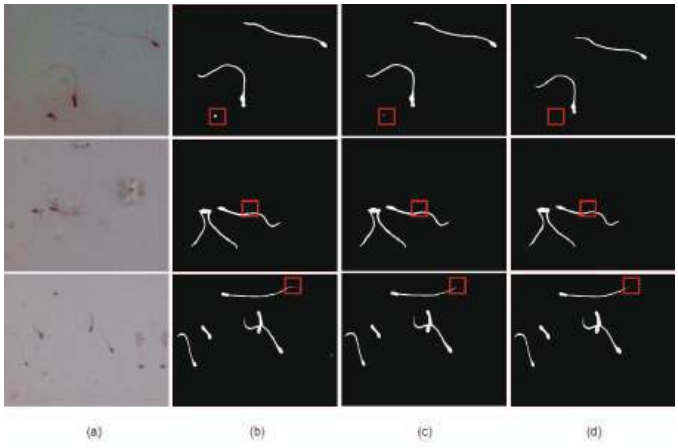


Fig. 7. An ablation study on (a) original (b) Extractor + unet, (c) Extractor + conv, and (d)our proposed model

Compare the ablation experiment segmentation in Tables 2 and Figure 7. The four networks' findings will be thoroughly assessed. The second to fifth columns contain U-Net, Unet + attention, Unet + conv, and our suggested network with Extractor segmentation results. Images get more intelligent as they compress, even in noisy environments. For complex photos, our segmentation network excels.

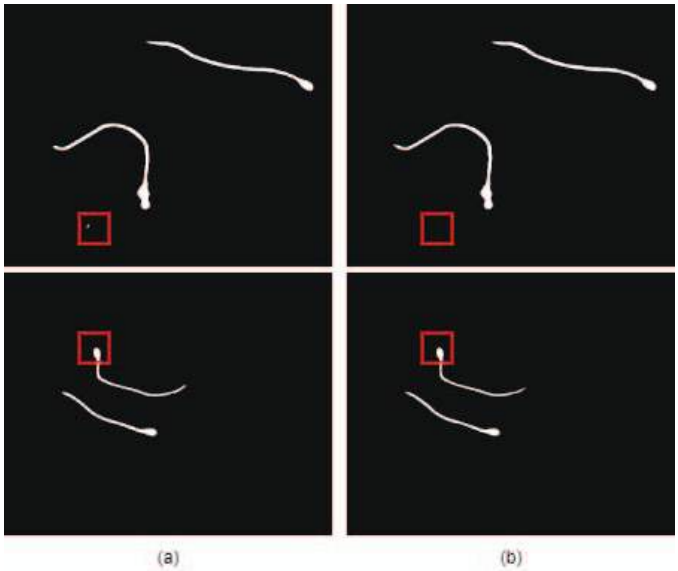


Fig. 8. A visual comparison study on complex noisy images with (a) YOLOV8E-TRACKED (b) our proposed ExtNet, red spot represent the model

EXNet and J.E.A. Li's YOLOv8e-track show clear advantages. EXNet excels in sperm segmentation using attention mechanisms and extractor modules, making it ideal for static

TABLE III
COMPARISON BETWEEN PROPOSED EXNET ALGORITHM AND SPERM YOLOV8E-TRACKED

Comparison Criteria	Proposed EXNet Model	Sperm YOLOv8e-tracked (J.E.A. Li, 2024)
Objective	Improved U-Net segmentation of sperm cells for improved separation of overlapping cells and high accuracy under different situations.	Optimised for object identification rather than segmentation, YOLOv8e detects and tracks sperm.
Techniques	Dual convolution blocks, attention methods, and extractor modules improve feature extraction and segmentation.	YOLOv8 object identification and tracking architecture optimised for video sperm detection but without pixel-wise segmentation.
Privacy Goals	Not applicable (focus on improving segmentation and performance).	Not applicable (real-time sperm identification and monitoring).
Data Utility	High segmentation accuracy (Dice coefficient: 95.14%) for clinical applications needing precise segmentation.	Prioritises quick, accurate identification with real-time tracking, over-segmentation accuracy.
Experimental Validation	State-of-the-art results on the Human Sperm Cell Segmentation Dataset (HSCS-DS) with 1,200+ labelled pictures.	Evaluated on bespoke sperm video datasets; efficient for detection but lacks segmentation precision.
Scalability	Designed for large datasets with static images, suitable for Big Data and dynamic data applications.	Highly scalable for real-time video sperm monitoring and detection.
Future Adaptability	Flexible for dynamic datasets and non-sperm segmentation applications.	Best for real-time sperm identification and tracking; not suitable for static image segmentation.

picture analysis in clinical applications and superior to real-world datasets. YOLOv8e-tracked employs real-time tracking and sperm detection to thrive in dynamic situations but lacks pixel-wise segmentation accuracy. The results have been demonstrated in Figure 8, and Table III shows the comparison of Yolov8 with our proposed model.

Both models are scalable; however, EXNet is better at segmentation and medical diagnostics, while YOLOv8e is better at detection.

In reproductive biology and related fields, the model's memory and accuracy seem promising. The model's shortcomings, such as the need for more data to improve accuracy, must be emphasized. Future studies may improve segmentation accuracy with reinforcement and model topologies.

V. CONCLUSION

Inspired by the U-Net, with double convolutional for down-sampling, attention mechanism for up-sampling, and Extractor at latent space, in this paper, we propose EXNet for precise human sperm segmentation. The EXNet comprises a revised U-Net network architecture, Extractor, and multiscale attention mechanism. When establishing a double convolutional down-sampling process, the most important and task-relevant areas are taken into consideration. The attention mechanism extracts and refines the multiscale-level spatial features. The Extractor module aggregates these features across multiple spatial locations in a feature map between the encoder and decoder. A double convolutional down-sampling and attention-based up-sampling also help in extracting and refining the spatial feature under difficult imaging circumstances. Comprehensive experiments on the HSCS dataset demonstrate the effectiveness of EXNet in addressing the challenges of sperm segmentation. Compared with the other U-Net variants qualitatively and quantitatively, our methodology demonstrates excellent accuracy in precisely. With a Dice Similarity Coefficient (DSC) of 95.14% and an average cross-alliance (MIOU) of 65.63%, our model achieves excellent accuracy. This achievement has the potential to greatly assist healthcare professionals in accurately identifying problems in sperm. Future study will concentrate on other dense prediction challenges, such as segmentation of the head, mid-pie, and tail into normal and diseased regions.

REFERENCES

[1] F. Zegers-Hochschild, G. D. Adamson, S. Dyer, C. Racowsky, J. de Mouzon, R. Sokol, L. Rienzi, A. Sunde, L. Schmidt, I. D. Cooke, J. L. Simpson, and S. van der Poel, "The international glossary on infertility and fertility care, 2017," *Fertility and Sterility*, vol. 108, p. 393–406, Sept. 2017.

[2] P. Sengupta, S. Dutta, M. Tusimin, T. Irez, and E. Krajewska-Kulak, "Sperm counts in asian men: Reviewing the trend of past 50 years," *Asian Pacific Journal of Reproduction*, vol. 7, no. 2, p. 87, 2018.

[3] G. I. Russo, H. Kandil, F. Boitrelle, R. Saleh, E. Chung, P. Kavoussi, T. Mostafa, R. Shah, and A. Agarwal, "Male infertility: New developments, current challenges, and future directions," *World Journal of Men's Health*, vol. 42, no. 1, pp. 1–20, 2024.

[4] A. O. Fahm, "Alter nation special edition 19 (2017) 175 - 191 175 electronic issn: 2519 - 5476 ; doi: https://doi.org/10.29086/2519-5476/2017/sp19a8 muslim women and social responsibility in nigeria: Contributions of the federation of muslim women ' s associations in nigeria (fomwan)," *Alternation Interdisciplinary Journal for the Study of the Arts and Humanities in Southern Africa*, p. 175–191, Dec. 2017.

[5] J. Kim, R. Omira, and C. Dutsch, "Combined storm and meteotsunami hazards: Data analysis and numerical simulation of christina (jan. 2014) and leslie (oct. 2018) events on the coast of portugal," Mar. 2022.

[6] D. R. Franken, "How accurate is sperm morphology as an indicator of sperm function?," *Andrologia*, vol. 47, p. 720–723, Aug. 2014.

[7] C. Lara-Clares and S. Valera, "Review of mattiello, elisa. 2022. transitional morphology: Combining forms in modern english. cambridge: Cambridge university press. isbn: 978-1-009-16828-1. doi: https://doi.org/10.1017/9781009168274," *Research in Corpus Linguistics*, vol. 12, no. 1, p. 189–195, 2024.

[8] R. Anitha, Prakash, and S. Jyothi, "A segmentation technique to detect the alzheimer's disease using image processing," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE, Mar. 2016.

[9] L. Maree, S. S. du Plessis, R. Menkveld, and G. van der Horst, "Morphometric dimensions of the human sperm head depend on the staining method used," *Human Reproduction*, vol. 25, p. 1369–1382, Apr. 2010.

[10] F. Ghasemian, S. A. Mirroshandel, S. Monji-Azad, M. Azarnia, and Z. Zahiri, "An efficient method for automatic morphological abnormality detection from human sperm images," *Computer Methods and Programs in Biomedicine*, vol. 122, no. 3, pp. 409–420, 2015.

[11] "Automated sperm morphology analysis based on instance-aware part segmentation," *arXiv*, 2024.

[12] K. Phoon and W. Zhang, "Study on sperm-cell detection using yolov5 architecture," *Genes*, vol. 14, no. 2, p. 451, 2023.

[13] J. e. a. Li, "Sperm yolov8e-trackevd: A novel approach for sperm detection," *Sensors*, vol. 24, no. 11, p. 3493, 2024.

[14] R. Marín and V. Chang, "Impact of transfer learning for human sperm segmentation using deep learning," *Computers in Biology and Medicine*, 2021.

[15] V. Valiuškaitė, V. Raudonis, R. Maskeliūnas, R. Damaševičius, and T. Krilavičius, "Deep learning based evaluation of spermatozoid motility for artificial insemination," *Sensors*, vol. 21, no. 1, p. 72, 2020.

[16] A. e. a. Mashaal, "Automatic healthy sperm head detection using deep learning," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, p. 735–742, 2022.

[17] A. e. a. Fraczek, "Sperm segmentation and abnormalities detection during the icsi procedure using machine learning algorithms," in *Proceedings of the 2022 15th International Conference on Human System Interaction (HSI)*, p. 1–6, IEEE, 2022.

[18] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, p. 234–241. Springer International Publishing, 2015.

[19] Z. Chen, "Attention is not all you need anymore," 2023.

[20] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, p. 749–753, May 2018.

[21] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," 2018.

LDBNet: A Lightweight Semantic Segmentation Network with Dual-Branch

1st Caoyue Li
College of Computer Science and Technology
Ocean University of China
Qingdao, Shandong Province, China
licaoyue0101@163.com

2nd Yutao Liu*
College of Computer Science and Technology
Ocean University of China
Qingdao, Shandong Province, China
liuyutao@ouc.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899595

***Abstract*—With the advancement of UAV (Unmanned Aerial Vehicle) technology, UAV imagery has increasingly become a critical component in the field of semantic segmentation. However, due to the challenges posed by varying scales, complex backgrounds, and the demand for real-time processing, there is an urgent need in the industry for a precise and efficient lightweight segmentation method. To address these challenges, we propose a lightweight dual-branch semantic segmentation model. This dual-branch architecture processes global semantic information and local edge details separately, enabling the model to capture both broad contextual features and fine-grained details. The global branch leverages a Multi-Scale Feature Enhancement module to extract and fuse multi-scale features, thereby enhancing the model's ability to understand complex scenes. Concurrently, the local branch incorporates a Similarity Add module, which refines edge information through attention-based fusion of local and global features. Finally, the Attention Fusion Module dynamically adjusts the weights of the feature maps, achieving an optimal balance between accuracy and computational efficiency. Extensive experiments demonstrate that our model significantly improves segmentation performance on UAV imagery, maintaining high precision while achieving lightweight parameters, making it highly suitable for on-board UAV applications in diverse and complex environments.**

***Keywords* — lightweight; semantic segmentation; dual-branch; UAV imagery; attention fusion**

I. Introduction

Semantic segmentation is a mainstream research field in computer vision, which makes significant contributions to multiple tasks, such as autonomous driving and biomedical imaging. Due to pixel-level labelling and thus more details, semantic segmentation enables machines a finer and more precise understanding compared to instance segmentation.

However, conventional semantic segmentation models, particularly those based on Convolutional Neural Networks (CNNs), often struggle with complex branch structure and massive parameter quantity. Deploying large models in practical applications is quite difficult, especially in scenarios that require real-time or low latency. These models require more storage space, training time, and inference time, which is a great challenge for resource limited devices such as mobile devices or embedded systems in unmanned aerial vehicle. With the maturity of drone technology, there is a significant need for Unmanned Aerial Vehicle (UAV) adapted lightweight semantic segmentation techniques in both academia and industry to address these challenges and remain high segmentation precision. As a result, we propose a more lightweight and advanced network termed LDBNet (lightweight dual-branch semantic segmentation network) that reduces the quantity of parameters and achieves higher segmentation accuracy compared to PPTFormer model.

Here are some contributions in our methodology:

- A lightweight semantic segmentation network is proposed, offering enhanced segmentation accuracy with a reduced parameter count and accelerated inference speed when compared to the PPTFormer model [1]. This network employs dual-branch to effectively extract edge and subject feature maps, thereby facilitating accurate segmentation.
- Integrating multiple features through attention algorithms allows the model to selectively focus on salient features within an image while disregarding irrelevant or noisy information. This approach enhances the model's robustness in handling variations in lighting, angles, occlusions, and other challenging conditions.
- Given the mature development of the drone field, our primary focus is on the semantic segmentation processing of drone aerial image datasets. We aim to apply lightweight, real-time semantic segmentation techniques to enhance applications in the drone aerial imaging domain.

II. Related Work

The development of semantic segmentation has evolved from traditional handcrafted features and machine learning methods to a deep learning-driven revolution. The advent of the Fully Convolutional Network (FCN) marked the beginning of deep learning's application in semantic segmentation [2], paving the way for subsequent models like the DeepLab [3] and U-Net [4], which further enhanced segmentation accuracy and multi-scale feature processing capabilities. In recent years, lightweight models such as ENet [5] and PIDNet [6] have opened new possibilities for real-time applications, while the introduction of Transformers, exemplified by SegViT [7], has significantly improved global context modeling, driving the

*Corresponding author.

expansion of semantic segmentation applications in complex scenarios. In particular, the field of UAV imagery has seen semantic segmentation emerge as a key research focus in recent years. For instance, PPTFormer [1] leverages the Transformer architecture and multi-scale feature processing to enhance the accuracy and efficiency of UAV image semantic segmentation.

A. Semantic Segmentation

Conventional semantic segmentation model contains CNN-based network. U-Net [4] is early semantic segmentation network based on CNN which is a fully convolutional neural network with a simple structure and easily trained. It represents a standard pattern for semantic segmentation. First, encoder, a down sampling method, extracts features of image pixels; then, decoder, a up sampling method, restores the original size image. UNet++ [8] is proposed to embed several small U-Nets in original U-Net. Furthermore, PoolFormer [9] introduces a network architecture that replaces complex convolution operations with simple pooling operations. This approach maintains the model's accuracy while reducing computational overhead. SegNet [10] is proposed to record the location of pooling and restore it when de-pooling for more accurate feature map restoration. Other network utilise Pyramid Pooling Module as PSPNet [11] and DeepLabv3 [12] It adds spatial pyramid pooling module into the model. Moreover, certain networks utilize multi-branch semantic segmentation architectures, such as PIDNet [6], an efficient and accurate real-time semantic segmentation network. By employing a multi-branch structure, PIDNet achieves a well-balanced integration of fine details and global semantic information while maintaining real-time performance and computational efficiency.

B. ViT-Based Models

ViT-based models mainly to input Attentions module and apply transformer-based encoder. SETR [13] The first representative model for semantic segmentation based on ViT. It replaces CNN-based encoder with Transformer-based encoder. Moreover, SegViT [7] generates masks for semantic segmentation by making use of attention mechanism. SegFormer [14], leveraging the self-attention mechanism, have the capability to concurrently capture relationships between all pixels within an image, thereby enhancing the model's ability to capture global contextual information. The multi-head self-attention mechanism allows Transformers to process features at multiple scales simultaneously, ensuring precise segmentation of both small and large objects, which significantly improves the performance of semantic segmentation. In the paper "Object-Contextual Representations for Semantic Segmentation," [15] the OCR (Object-Contextual Representation) module is introduced to integrate global contextual information into the representation of each position. When combined with the HRNet [16] network, this approach enhances the capture of global contextual information, thereby achieving the effects of a Segmentation Transformer and markedly improving the model's segmentation performance.

C. UAV Scene Semantic Segmentation

Fine-grained semantic segmentation is a critical challenge in the segmentation of UAV imagery due to the presence of a large number of multi-scaled objectives which has millions of pixels or only thousands of pixels. To achieve fine-grained semantic segmentation, HRNet [16] utilizes multiple feature maps of various resolutions across different branches of the network. Besides, PPTFormer [1] combines multi-scale feature extraction, Transformer encoders, and attention mechanisms to enhance the accuracy and efficiency of UAV image semantic segmentation. In recent years, the rapid development of UAV technology has become a significant focus, making datasets related to UAV imagery increasingly important. The emergence of the UAVid [17] and AeroScapes [18] datasets has significantly advanced research in UAV image semantic segmentation and introduces new challenges, such as significant scale variation, moving object recognition, and the preservation of temporal consistency.

III. Methodology

In this section, we elaborate on the detailed methodology employed in our approach to achieve lightweight dual-branch semantic segmentation network named LDBNet. Fig. 1 shows the overall architecture of LDBNet, which contain two main branches: global contextual processing branch and local edge processing branch. The dual-branch structure, by combining the strengths of the global semantic branch and the local edge branch, effectively captures both global contextual information and local detail features, significantly enhancing the accuracy and robustness of lightweight semantic segmentation networks. The global branch focuses on capturing macroscopic semantic information, while the local branch finely processes edges and details. Their synergistic interaction ensures superior performance in complex UAV imagery scenarios. Additionally, the introduction of global loss and edge loss, as shown in (1), enables the model to achieve a balance between global semantic consistency and local detail precision.

$$Loss = L_{global} + L_{local} \tag{1}$$

The global loss ensures accuracy over large regions, while the local loss ensures precision in Boundary areas.

A. Dual-Branch Structure

To achieve more precise semantic segmentation in complex scenarios while maintaining a lightweight network, this paper adopts a dual-branch architecture that separately processes global semantic information and local detail information. The input image first passes through three convolutional layers to generate high-dimensional feature maps, which are share the same parameters in both branches. In both branches, feature extraction is performed using convolutional layers based on a residual network structure (RESL). The operations at each stage of the overall network, along with the output image sizes and dimensions, are illustrated in Tab. I.

In the global contextual processing branch, features are further extracted through three convolutional modules that

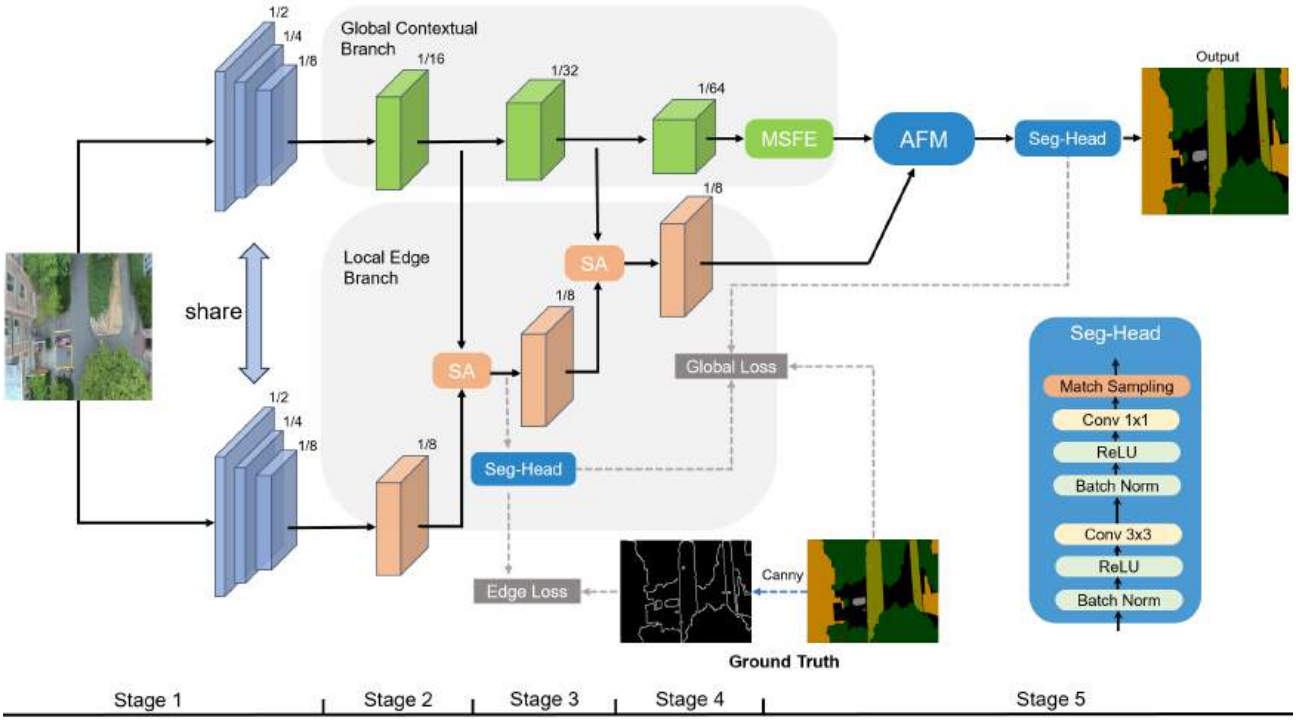


Fig. 1. Overall structure of our lightweight dual-branch semantic segmentation network.

TABLE I
ILLUSTRATES THE OPERATIONS AND OUTPUT AT EACH STAGE OF THE OVERALL NETWORK.

Stage	Operation		Output	
	Local	Global	Local	Global
1	Conv3x3, 2 Conv3x3, 2		256x256 C	
	m*RESL, 1 m*RESL, 1		256x256 C 128x128 2C	
2	m*RESL, 1 SA	n*RESL, 2 ←	64x64 2C	64x64 4C
3	m*RESL, 1 SA	n*RESL, 2 ←	32x32 2C	32x32 8C
4	OutRESL, 1	2*OutRESL, 2	16x16 4C	16x16 16C
5	– →	MSFE AFM	128x128 4C	
	Conv3x3, 1 Conv1x1, 1		128x128 128 128x128 No. Classes	

For operation, "OP, N" means operation OP with stride of N
The coefficients m and n denote the number of layers in RESL
For output, "WxH C" means the output size and dimension

implement skip connections using residual network structures. These modules also reduce the size of the feature maps, thereby expanding the receptive field and enhancing the global semantic information while preserving the input information at deeper layers of the network. Subsequently, the Multi-Scale Feature Enhancement (MSFE) module is employed to extract feature information from different scales through average pooling operations of varying sizes. These multi-scale features are then fused together in the feature fusion module. The output feature map combines the detail from the input feature map with multi-scale contextual information, enabling the model to better capture both global and local features, thus improving segmentation granularity, accuracy, and robustness.

In the local edge processing branch, the feature map is similarly obtained through three convolutional layers based on residual network structures. However, unlike the global branch, the size and dimensions of the feature map remain unchanged, ensuring that the receptive field size stays constant. This allows the network to focus more on local information while enriching the feature representation through convolutional operations. Additionally, the Similarity Add (SA) module introduces a position-based attention mechanism, guiding the fusion of intermediate global and local features by calculating similarity mappings between feature maps. This results in a fused feature map that more effectively captures the important information in the image.

Finally, the Attention Fusion Module (AFM) guides the fusion of global and local feature maps using an attention mechanism. It dynamically generates weights based on the content of the feature maps and uses these weights to perform a weighted average of the two feature maps, producing a more expressive fused feature map.

B. Attention Fusion Module

This paper proposes an AFM to enhances the model's selectivity for key features and its contextual understanding, thereby improving both the model's expressiveness and computational

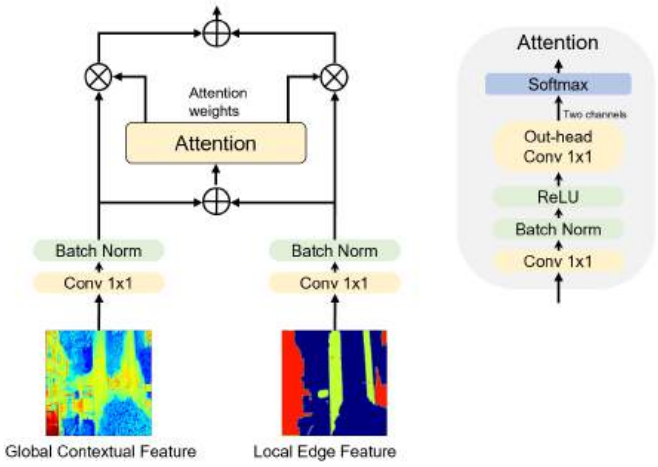


Fig. 2. The overall structure of the attention fusion module.

efficiency. Fig. 2 illustrate the design idea of our module. First, the feature maps from the two branches are individually adjusted through convolution layers to obtain new feature maps. These new feature maps are then summed and fed into the attention module to compute attention weights. Finally, the new feature maps are weighted according to the attention weights and summed to produce the fused features.

This algorithm dynamically adjusts weights based on the input feature maps, prioritizing the most critical feature regions. Its calculation formula is shown in (2).

$$f_{\text{fused}} = w_g \cdot \text{BN}(\text{Conv}(g)) + w_d \cdot \text{BN}(\text{Conv}(d)) \quad (2)$$

where, w_g and w_d are obtained through the attention mechanism calculated by (3)

$$f_{\text{combined}} = \text{BatchNorm}(\text{Conv}(p)) + \text{BatchNorm}(\text{Conv}(i)) \quad (3)$$

This selective focus effectively reduces redundancy and minimizes noise interference, allowing the model to more accurately concentrate on essential global semantics or local details. Through adaptive weight allocation, the attention mechanism facilitates a seamless fusion of global and local features, enabling the model to capture large-scale global information while also preserving fine-grained local details. This capability ensures high-precision segmentation and recognition in scenarios with extensive multi-scale targets, such as UAV imagery, where both large objects with millions of pixels and small objects with only a few thousand pixels coexist.

Moreover, the attention mechanism effectively reduces unnecessary computational overhead. By weighting the features, the model can focus computational resources on important feature regions rather than distributing them evenly across all areas, thereby enhancing computational efficiency. This approach not only contributes to the lightweight nature of the model but also aligns with the objectives of real-time semantic segmentation tasks by improving computational efficiency.

IV. Experimental Results

A. Implementation Details

The proposed network and all our experiments are implemented on hardware consisting of dual NVIDIA RTX 3080 GPUs (20GB each), a CPU with 14 vCPUs (Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz), and the software environment including Ubuntu 20.04.3, Python 3.8.10, and PyTorch 1.10.0+cu113.

B. Datasets

We utilized the UAVid2020 and AeroScapes UAV datasets to evaluate the performance of the proposed model.

- **UAVid2020:** The UAVid2020 [17] dataset is an UAV imagery dataset for semantic segmentation task focusing on urban scenes. It contains 42 sequences in total (20 train, 7 valid and 15 test) and includes annotations for 8 semantic classes. Each image has a high spatial resolution of 3840x2160 pixels, which allows for detailed analysis of urban features.
- **AeroScapes:** The AeroScapes [18] aerial semantic segmentation benchmark consists of images captured by a commercial drone at altitudes ranging from 5 to 50 meters. The dataset includes 3,269 720p images along with ground-truth masks annotated for 11 distinct classes.

C. Evaluation metrics

In this paper, we employed mIoU, PA, mACC, and FPS four evaluation metrics to assess the semantic segmentation accuracy and inference speed of our network during the experiments.

mIoU is used to evaluate the accuracy of a model. It measures the overlap between the predicted segmentation and the ground truth, averaged over all classes. Its calculation formula is shown in (4):

$$\text{mIoU} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i} \quad (4)$$

where, N represents the number of classes. TP_i is the number of true positives for the i-th class. FP_i is the number of false positives. FN_i is the number of false negatives.

PA is employed to assess the overall accuracy predictions. It provides an intuitive indication of how well the model's pixel-level predictions align with the ground truth. The calculation method is shown in (5):

$$\text{PA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (5)$$

where, TN is the number of true negatives.

mAcc is used to evaluate the model's accuracy across all classes, ensuring that the performance on less frequent classes is weighted equally with the performance on more common classes. The algorithm is shown in (6):

$$\text{mAcc} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \quad (6)$$

FPS is used to evaluate the execution speed of an algorithm and its expression is illustrated in (7):

$$\text{FPS} = \frac{N}{\sum_{N}^{i} \text{T}_i} \tag{7}$$

where, N denotes the total number of images, and T_i represents the time taken by the algorithm to process the i-th image.

D. Comparison with State-of-the-Arts

Through extensive comparative experiments, we rigorously evaluated our proposed model against several other prominent segmentation networks, including PPTFormer [1], PIDNet [6], and SETR [13], using the UAVid and AeroScapes datasets. The evaluation is mainly based on key metric of Mean Intersection over Union (mIoU). This metric were carefully chosen to provide a comprehensive assessment of the semantic segmentation accuracy across different datasets. By focusing on this metric, we aimed to understand how well our model performs in various scenarios and how it compares to the latest advancements in semantic segmentation. This comparison highlights the strengths and potential areas for improvement of our model, particularly in terms of its accuracy and consistency across diverse and challenging UAV-based datasets.

TABLE II
COMPARISON OF STATE-OF-THE-ARTS ON UAVID AND AEROSCAPES DATASETS.

Method	mIoU	
	UAVid	AeroScapes
Deeplab	56.82	51.40
PSPNet	58.20	57.98
SETR	58.52	50.34
PoolFormer	61.73	62.27
PIDNet	65.81	63.35
PPTFormer	65.00	**68.50**
LDBNet(Ours)	**66.21**	63.78

In addition to evaluating the segmentation accuracy, we also performed a comprehensive comparison of our model's performance in terms of inference speed and spatial efficiency. Specifically, we focused on Frames Per Second (FPS) and parameter count (Param) as key metrics. Validation was performed on the UAVid and AeroScapes datasets using High-resolution images in the validation, with computations carried out on an RTX 3060Ti GPU. Our model was benchmarked against the latest state-of-the-art networks, PPTFormer and PIDNet, to assess how it stands in terms of both computational efficiency and memory footprint. This analysis allowed us to evaluate the trade-offs between speed, accuracy, and model size, providing insights into the effectiveness of our lightweight dual-branch semantic segmentation approach in comparison to these advanced networks.

TABLE III
COMPARATIVE ANALYSIS OF EXECUTION SPEED AND MEMORY FOOTPRINT.

Method	Param	FPS	
		UAVid	AeroScapes
PPTFormer	86.0M	-	-
PIDNet	7.8M	1.9066	7.7993
LDBNet(Ours)	**7.4M**	**2.3424**	**8.3851**

As illustrated in Tab. II, our proposed method demonstrates superior segmentation accuracy compared to other leading semantic segmentation networks. Notably, our model outperforms most state-of-the-art networks in this domain. Specifically, on the UAVid dataset, our network achieves a 1.21% improvement in mIoU compared to PPTFormer, which was introduced at IJCAI 2024. This improvement underscores the effectiveness of our approach in handling the complexities of UAV imagery. Additionally, our model outperforms PIDNet on both datasets, with the mIoU metric showing an improvement of 0.61% on the UAVid dataset and 0.68% on the AeroScapes dataset.

While our network slightly underperforms PPTFormer on the AeroScapes dataset, it excels in efficiency, as demonstrated in Tab. III. Our model significantly reduces the number of parameters, boasting a 91.39% decrease compared to PPTFormer. This considerable reduction in model size translates into lower memory usage, making our approach more suitable for deployment in resource-constrained environments, such as real-time UAV applications. In addition to the reduction in parameters, our network also surpasses PIDNet in terms of inference speed and parameter count, achieving a 5.13% reduction in parameters. This balance between accuracy and efficiency highlights the practical advantages of our model, especially in scenarios where computational resources and processing time are critical factors. By achieving a high level of segmentation accuracy with a much lighter and faster model, our approach represents a significant step forward in the development of efficient and effective semantic segmentation networks for UAV applications.

E. Ablation Study

In Tab. IV, We perform ablation studies of the proposed LDBNet on UAVid dataset under three indicators as critical benchmarks, including: Mean Intersection over Union (mIoU), Pixel Accuracy (PA), and Mean Accuracy (mAcc). By analyzing these indicators, we gain insights into how each element of LDBNet contributes to the overall segmentation accuracy and how the model's performance can be optimized. The results of our ablation experiments strongly validate the effectiveness of both the dual-branch architecture and the Attention Fusion Module in our proposed model.

The dual-branch structure consistently outperforms the one-branch configuration across all evaluated metrics , with mIoU increasing by 21.8%, PA by 2.3%, and mAcc by 21.1%

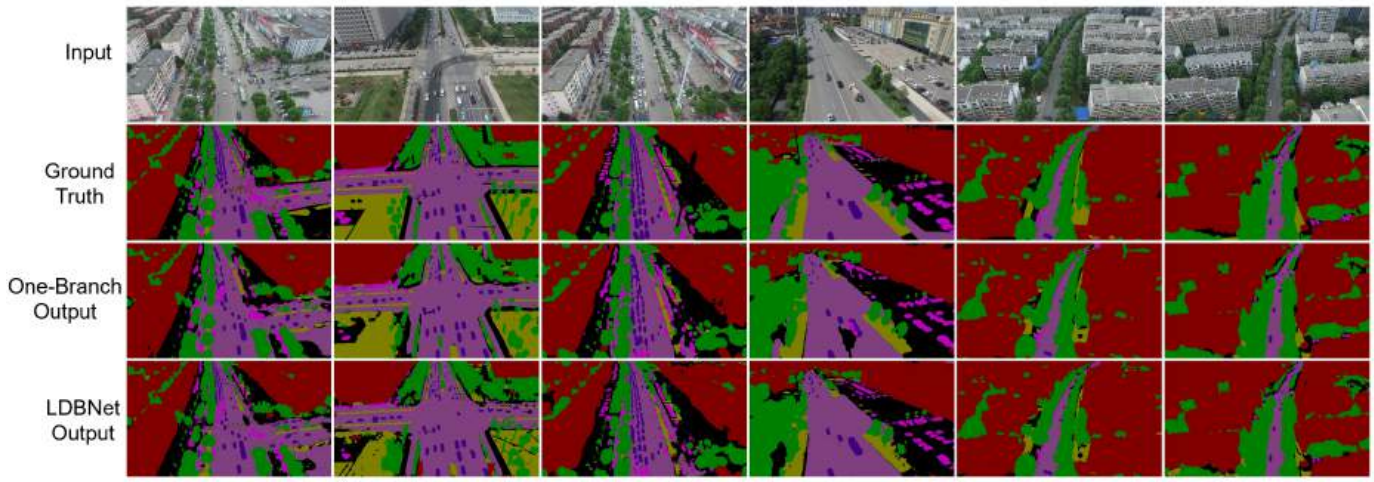


Fig. 3. The results of One-Branch Net and LDBNet on UAVid dataset.

Fig. 4. The results of One-Branch Net and LDBNet on AeroScapes datasets.

TABLE IV
LIGHTWEIGHT DUAL-BRANCH SEGMENTATION ABLATION STUDY RESULTS ON UAVID.

Method	OB/DB	AF	mIoU	PA	mAcc
One-Branch Net	OB	✗	51.67	83.33	61.51
Dual-Branch Net	DB	✗	62.91	85.23	74.49
Dual-Branch+AFM (**LDBNet**)	DB	✓	**66.21**	**86.92**	**76.76**

OB, DB, AF denote One-Branch, Dual-Branch, and Attention Fusion.

as shown in Tab. IV. These improvements underscore the significant role that the dual-branch design plays in capturing both global and local features more effectively, thereby leading to a marked increase in the overall accuracy of semantic segmentation. As shown in Fig. 3 and Fig. 4, the dual-branch structure demonstrates higher accuracy compared to the single-branch structure, which can be observed more intuitively.

The results further demonstrate that the dual-branch architecture, when combined with the Attention Fusion Module, provides a more robust framework, which is our proposed LDBNet. We observed notable improvements across those key performance metrics. Specifically, the mIoU increased by 5.2%,PA improved by 2.0%, and mAcc rose by 3.1%. These enhancements highlight the efficacy of the Attention Fusion Module in the context of the dual-branch architecture. The module's ability to effectively integrate multi-scale features plays a crucial role in boosting the model's overall accuracy in semantic segmentation tasks. This improvement not only underscores the module's importance in refining feature representation but also demonstrates its potential to advance the performance of segmentation models, particularly in challenging scenarios involving complex and varied image data.

V. Conclusion

This paper presents the novel LDBNet, a lightweight dual-branch semantic segmentation network for UAV scene segmentation. It addresses the challenges of providing efficient computation and high segmentation accuracy on resource-constrained embedded platforms or mobile devices, such as UAVs. By employing a dual-branch structure that captures both global semantics and local details, it effectively handles the broad scenes and complex edge details found in high-resolution UAV images. The integration of an attention fusion mechanism further ensures the seamless fusion of multi-scale feature maps, preserving critical contextual information while enhancing processing speed, thereby maintaining segmentation accuracy. Experiments on different datasets have validated the superior performance and efficiency of LDBNet. The significant advancements achieved by LDBNet underscore the importance of lightweight semantic segmentation models in UAV image analysis, paving the way for further innovation in UAV applications within the industry.

In future work, we plan to focus on two key areas: extending the application scenarios and enhancing the model's robustness through the optimization of network architecture. First, while maintaining the dual-branch structure, we will explore more compact feature fusion strategies or lighter branch modules to reduce computational redundancy. By dynamically adjusting the feature extraction strategy according to different scenarios, the model will be able to adaptively select the optimal computational path at different times, further improving inference efficiency. On the other hand, we will employ more extensive scene simulation and data augmentation techniques during training, such as simulating extreme weather conditions, low light, and strong illumination. This approach aims to enhance the model's robustness under varying weather conditions, lighting changes, and complex backgrounds.

Acknowledgment

This work was supported by the National Science Foundation of China under grant 62201538 and Natural Science Foundation of Shandong Province under grants ZR2022QF006 and ZR2024MF116.

References

[1] D. Ji, W. Jin, H. Lu, and F. Zhao, "Pptformer: Pseudo multi-perspective transformer for uav segmentation," 2024. [Online]. Available: https://arxiv.org/abs/2406.19632

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1411.4038

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," 2017. [Online]. Available: https://arxiv.org/abs/1606.00915

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *ArXiv*, vol. abs/1505.04597, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:3719281

[5] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1606.02147

[6] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "Pidnet: A real-time semantic segmentation network inspired by pid controllers," 2023. [Online]. Available: https://arxiv.org/abs/2206.02066

[7] B. Zhang, Z. Tian, Q. Tang, X. Chu, X. Wei, C. Shen, and Y. Liu, "Segvit: Semantic segmentation with plain vision transformers," *ArXiv*, vol. abs/2210.05844, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252846611

[8] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, S...*, vol. 11045, pp. 3–11, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:50786304

[9] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," 2022. [Online]. Available: https://arxiv.org/abs/2111.11418

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:60814714

[11] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:5299559

[12] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017. [Online]. Available: https://arxiv.org/abs/1706.05587

[13] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6877–6886, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229924195

[14] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," 2021. [Online]. Available: https://arxiv.org/abs/2105.15203

[15] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," 2021. [Online]. Available: https://arxiv.org/abs/1909.11065

[16] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 3349–3364, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201124533

[17] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang, "Uavid: A semantic segmentation dataset for uav imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 165, pp. 108 – 119, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271620301295

[18] I. Nigam, C. Huang, and D. Ramanan, "Ensemble knowledge transfer for semantic segmentation," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1499–1508, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:13661896

Numeric Representation of Strings: An optimized approach to Lexical-Comparisons.

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899576

Vikram Singh Warraich
Broadcom Inc.
San Jose, CA, USA
vikramsw@mac.com

Abstract*—Absurd as it may seem, but strings can be treated as numbers for computation purposes. Most computer programs make extensive use of String operations like string-matching or sub-string lookups. Computer systems represent each character of a string with a numeric code. I take this concept further by applying it to arbitrary strings and explore the performance advantages of representing strings numerically. This paper presents a considerably faster approach for arbitrary String Lexicographic Comparisons and substring lookups. It introduces the concept of String-Numbers, which are arbitrary Strings represented by unsigned numbers. I validate my theory by running performance tests, which prove that String-Number based operations outperform contemporary string operations for a variety of String based algorithms. Test Results confirm the speed improvements delivered by String-Numbers in the areas of Searching, Sorting, Substring-Lookups and String-Matching.

Keywords*— Lexicographic string comparisons, Searching, Sorting, Substring lookups, High Performance.

I. Introduction

String operations occur frequently in computer programs. Although existing software libraries implement these operations relatively efficiently, I describe an alternate way to improve them further, bringing speed-ups by an order of magnitude.

I propose an alternate paradigm for representation and matching strings by introducing the concept of String-Numbers. Words are a special case of the strings and all written words are strings in some form. A string may contain one or more words separated by spaces or it could be just an arbitrary sequence of characters. The terms word and string are used interchangeably unless mentioned otherwise. Also, String-Numbers are referred to as SN.

SN based string operations are much faster primarily due to the SN comparisons occurring in a Numeric plane rather than in a Character plane. As a one-time task, String-Number for an arbitrary string of length n can be computed in O(n) time. Thereafter, any string operation is reduced to Numeric comparisons irrespective of the strings' lengths. This also holds true regardless of the length of the common substrings between the strings being compared. Repeated comparisons do not require re-computation of String-Numbers, unless the original strings get mutated. SN is not a hash of the String.

A. *The Why ? : Scope of the Work*

String operations are very frequent in a software process lifecycle. Numerous string operations occur behind the scenes when you run a computer application on a modern operating system. String operations occur within the Loaders/Runtime, the Libraries/Frameworks, and more so in code of the application that is being run. Operating systems too employ string comparison operations. Thus, any optimization in string comparison function calls, will directly translate into considerable performance gains across the board. Moreover, any software that uses Sorting, and Searching, and requires repeated string operations, like Search-Engines, Banking-apps, Databases etc., will benefit from it.

B. *The When ? : Feasibility Aspects.*

SNs are practical now, primarily due to large numeric values required to store them. Relatively early generations of Microprocessors processed information 8 bits at a time. They were followed by 16 bit microprocessors, which were succeeded by 32 bit Microprocessors. And at the time of this writing, almost all computer desktops, phones, and laptops are 64 bit, and therefore make it possible to represent SNs as large numbers.

C. *The How ? : Idea on High Level.*

Numeric manipulation is much faster than character-by-character manipulation and I count on this property to achieve performance gains. Any arbitrary string can be converted into a unique number. Please see Section-II on how I accomplish this. The wider the string, the wider will be the corresponding String-Number. Small strings allow numeric value of the string to fit within the CPU's word-size. This allows the CPU to perform the string comparisons in as few CPU cycles as are allowed by this scheme. Although this work focuses on Immutable Strings, the concept can be applied to Mutable Strings too and involves re-computing the relevant String-Number. In the next section, I discuss how to construct the SNs.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

II. WHAT ARE STRING-NUMBERS ?

Let us consider any Word from any set of alphabets of any language. For instance the word "AFFINITY" from English. "AFFINITY" can be mapped into a unique 64 bit number, which is its String-Number. The mapping is computed via a two-way function that partitions the range of a 64 bit unsigned long-long number appropriately for each character position in the string. When computing String-Number for a String of length 8, the characterAtIndex(0) carries the most weight, followed by the characterAtIndex(1) and so on until characterAtIndex(7), which carries the least weight. Also, this scheme requires ensuring the following for any string and its corresponding String number.

$$W_i > C_{\max} \times W_{i+1}$$

Where,

W_i is the Weight for characterAtIndex(i).

C_{max} is the largest char-code or code_point of the given charset.

Keeping the above in mind, the String-Number can be calculated as follows:

$$SN = \sum_{i=1}^{WL} C_i \times W_i$$

Where,

SN is the string number being calculated

WL is the length of maximum chars that can fit in the **SN** without overflowing the 64 bit-number. For example, **WL**=8 for lower-ASCII and **WL**=4 for UTF-8/Unicode encodings typically.

W_i is the Weight for characterAtIndex(i).

C_i is the code_point of the characterAtIndex(i) in that charset.

The above scheme preserves the lexicographic comparison property of language strings. It also ensures that the String-Number value for a String like "AZZZZZZZ" will always be smaller than the String-Number value for a String like "B".

Using the above strategy, the word "AFFINITY" may have a String-Number of: 4582902774514620215.

For longer strings, which don't fit in a single String-Number, I introduce the concept of a chain of String-Numbers. For example, the string "ABCDEFGHIJKLMNOPQRSTUVWXYABCDEFGHIJKLMNOABCDEFGH" may be represented as a chain of Numbers.

6874089463069197431 ----> 8501743351925922542 ----> 10129397240782610903 ----> 7235790327259580789 ----> 8863444212902640574 ----> 10129387474437955053 .

Above scheme guarantees the uniqueness of the numeric value of any string's String-Number. The uniqueness avoids collisions and ensures that no two distinct strings formed from the same alphabet will have identical String-Number values. Also, this scheme preserves the lexicographical comparison property of the strings. So,

a). If StringA is lexicographically greater than StringB, then the numeric value of SN-A will be greater than SN-B.

b). If the String-Number numeric values for any two strings are equal, then they both point to lexicographically identical strings.

Reverse-computing the original string, given its String-Number was also achieved in this work. For example, to compute the first character of the original string, I simply divide the SN with the Weight W_0. This is because:

$A_{[0..N]} = A_0xW_0 + A_1xW_1 + A_2xW_2...+A_NxW_N$

Therefore

$A_{[0]} = A_{[0..N]}/W_0$

$A_{[1]} = (A_{[0..N]} - A_0xW_0)/W_1$

$A_{[N]} = (A_{[0..N]} - A_0xW_0 - A_1xW_1 ... - A_{n-1}xW_{n-1})/W_N$

Therefore, given an SN, it is possible to compute the original string. And hence, SN computation from a string is a two-way function, for example, the following may be computed.

4582902774514620215 -> AFFINITY

III. ABOUT THE DATA-SETS USED IN THE TESTS.

Tests were run on different data sets:

a). Sample English dictionary with over 400k words,

b). Auto-generated random data of various sizes.

c). Books downloaded from the Internet.

d). Genetic Alignment and substring search based tests were run on a 145 MB DrosophilaMelanogaster Genome fasta file.

A. Properties of Dictionary words.

Strings on the internet are typically a subset of dictionary words. Such words form bulk of the strings processed by computers around the world. Over 50% words from a typical English Dictionary are 9 characters or less. As per a study of distributions of word lengths for various languages, the average length of any word in a document is 5 or 6 characters **[6]** & **[7]**.

B. Sample Data Set for Tests

For item III.a), the Test Data was a word-list of an English Dictionary. This dictionary contained 466550 unique ASCII words. The tests' sample space cardinality ranged from 192126

to 420061 unique words, depending on the word's character size, which ranged from 8 to 13 characters.
For item III.b), Test Strings were generated randomly via a test program. The randomly generated test data Strings ranged from 8 to 1024 characters in size. For item III.c), tests were run on text files containing UTF-8 characters, Story-books downloaded from the internet etc. The Unicode/UTF-8 characters in the Strings were multi-byte, ranging from 2-bytes to 4-bytes. Their length was arbitrary.

IV. Computing String-Number for small strings

SNs are unique Mappings from the set of Strings to unique Numbers. The idea is to partition the range of unsigned 64 bit numbers in such a way that no two strings map to the same number. Section-II discussed some aspects around it.

A. SN and Word Sizes

The unit of SN for this work is a 64 bit, unsigned number. The dictionary Test words were fit into a single SN. The Multi-Byte UTF-8/Unicode strings were of arbitrary size and were represented as a chain of SNs, with each SN representing around 4 UTF-8 characters. I term this as a word-size for computing SNs. Word-sizes ranged from 4 for UTF-8 strings to up to 13 for lower-ASCII strings. For Genetic Data tests, the word-size was 23, which allowed a single SN to hold 23 characters.
Although this work focuses on UTF-8 and ASCII encodings, it applies to any encoding or character set. As a further optimization the word-size can be increased if the number of distinct characters used in the Sample data is a subset of the entire charset. This translates to speed-ups in lexicographic string-matching. For example, the ascii-codes for characters of common US-English words range between 32 and 126; a subset of the complete ASCII range.

A lower-ascii String of Length N can be represented by (N/9+1) SNs. Further, for case-insensitive matching, if I consider the Alphabet of the Strings to be only A-Z, then that same ASCII String representation can fit in (N/13 +1) String-Numbers. Representing Ascii strings of lengths greater than 13, require multiple String Numbers as discussed in Section-II.

Table-1 contains information on the Ascii characters that can fit in various Word-Lengths for SN schemes in SN-based algorithms.

Table 1: Sample Sizes for 7-bit US-ASCII String-Words.

Word-Length	Chars Covered	Why ?
9 or lower	Lower Ascii chars :A-Z, a-z, 0-9, @#$%^&*( )_+- =~`{}|[] \;':",. /<>? etc.	Words of length 9 chars or less chosen from a set of around 103 chars can fit within 1 Numeric-String.
10	All commonly used lower-ascii, except lowercase a-z.	Words of length 10 or less chars chosen from a set of around 76 chars can fit within 1 Numeric-String. Please note that this implies case-insensitive matching.
11	All commonly used lower-ascii, except lowercase a-z and except some special characters.	Words of length 11 or less chars chosen from a set of around 56 chars can fit within 1 Numeric-String. Matching will be Case-insensitive.
12	Only A-Z, 0-9 and one special character.	Words of length 12 or less chars built from a set of 37 chars can fit within 1 Numeric-String. Case-insensitive.
13	Only A-Z.	Words of length 13 or less built from a set of 26 chars can fit within 1 Numeric-String. Case-insensitive.
> 13	Limited character sets. For example GeneticSequences etc.	Subsequences of Length upto 23 characters can fit in one SN. For example a Gene sequence comprised of GATC etc.

V. Computing String-Numbers of Arbitrary length strings

Strings of length over 13 chars are commonplace in computing and in sentences. Multiple SNs can represent such longer strings.
As discussed earlier, the String-Number for a large string of arbitrary length is a sequence of String-Numbers, which are linked together.
For example, the ASCII string "String Numbers are way Cooler and way Faster", has a string length of 44 characters. Considering a word-size for String-Numbers of 8, this string can

be mapped into the numeric plane via 6 Chained String-Numbers. This is because, [(44 Div 8) + 1] equals 6. A 1 is added to accommodate the last few characters only if (44 Mod 8) > 0. In general, an arbitrary ASCII String of length N can be mapped to the Numeric plane via (N/8 +1) String-Numbers, where
/ is the Div operator, and , (N Mod 8) != 0.

As discussed earlier, if the test dataset is limited to a subset of the ASCII characters, for example, only A-Z, I can improve the word-size from 8 to 13 for Case-Insensitive matches.

In general, an arbitrary UTF-8 String of length N can be mapped to the Numeric plane via (N/4 +1) String-Numbers using a word-size of 4, where
/ is the Div operator, and , (N Mod 4) != 0.

Also, for a string of arbitrary length, an increase in the word-size is directly proportional to the performance gains. Reason being that determining the strings' equality or inequality, possibly requires fewer SN numeric comparisons.

A. Relevant optimizations.

If the sample-space of Strings for a spoken or written language is finite and its cardinality is less than the value of Max(SN), then I can map any possible string from this set into a single unique SN, thus obviating the need to represent long strings with Multiple SNs. However, this work considers the Set of possible strings as Infinite, and therefore, it discusses how to represent large length strings as a chain of SNs.

B. Optimizations: Properties of Arbitrary Strings being compared.

Depending on the dataset, if most Strings have a common prefix, then the chain of String Numbers can be compared for in-equality from the end of the chain and traversing backwards, resulting in performance benefits. However, the SN performance results are impressive even without this optimization.

VI. Finding Substrings: Needle in a Haystack.

SNs offer substantial performance benefits for substring lookups. This is especially true if one of the strings is large. For e.g., If I have to find the following regarding two strings A and B.

1). Find the first occurrence of B in A.
2). Find the last occurrence of B in A.
3). Find ALL occurrences of B in A.

For the sake of simplicity and feasibility, it is assumed that the minimum length of B is twice the SN word-size. Word-Size relates to the maximum number of characters that an SN can hold without overflowing the 64-bit number. In this example, A and B have a word-size of 8. Alternately, the word size can be set to lower size to accommodate the requirement that strlen(B) is twice the word size.
A is like a haystack and B is like a needle. The goal is to find a needle-in-a-haystack.

A == "This is a very very very long sentence that has been broken into Number-Strings. This could might as well be a long paragraph, document or a book or a collection of books."
B == "sentence that has"

First, I break the smaller String B, and create N versions of it, where N is the word-size. For instance, String B can have 8 versions for a word-size of 8. Then I create Numeric Strings for those 8 versions of B. Next, I traverse through the chain of Numeric-String blocks of String A, and try to find a match with any of the 8 versions of String B.
Example:

B1 = "sentence that has" == "sentence" --> " that ha" --> "s"
B2 = "entence that has" == "entence "--> "that has"
B3 = "ntence that has" == "ntence t" --> "hat has"
B4 = "tence that has" == "tence th" --> "at has"
B5 = "ence that has" == "ence tha" --> "t has"
B6 = "nce that has" == "nce that" --> " has"
B7 = "ce that has" == "ce that " --> "has"
B8 = "e that has" == "e that h" --> "as"

String number chain for the larger string, A == A1-->A2-->A3-->A4-->A5-->A6-->A7-->A8-->A9-->A10-->A11-->A12-->A13-->A14-->A15-->…
Where,
A1 == "This is "
A2 == "a very v"
A3 == "ery very"
A4 == " long se"
A5 == "ntence t"
A6 == "hat has "
A7 == "been bro"
A8 == "ken into"
A9 ==" Number-"
A10 =="Strings."
A11 ==" This co"
A12 == "uld migh"
A13 == "t as wel"
A14 =="l be a l"
A15 =="ong para"
And so on…

Hence, the needle B3 matches substrings A5 & A6 in the haystack.

Essentially, I have reduced the problem to exact string-matching involving substrings of substrings, where the matching of substrings is simply comparing the relevant String-Numbers; an arithmetic operation. To conclusively determine if an absolute Match exists, further computation is required to match the prefix and suffix. This required extra computation has a very small performance penalty because I know exactly where and how many chars to look for when matching prefix and the suffix of the string from the possible-match location. After identifying the vicinity of the match, a traditional match for the remaining characters in the prefix and the suffix of the match suffice.

The lookup of the Bi in Ai can be sped up by a binary search. Because the comparisons in SN based Binary search are numeric instead of being character-based, the search is incredibly fast. Performance gains were observed to occur for repeated string operations. This scheme performed better than other search algorithms, including Rabin-Karp **[5]**, KMP **[4]**, and glibc:strnstr after around 25 searches. It certainly is asymptotically faster. The setting up of binary-search and other book-keeping operations dominated the time consumed by SN scheme for lower search counts. However, for search volumes in excess of 50 searches, OR for repeated sub-string lookups the SN-based search performance is an order of magnitude better than the others. This observed warm-up penalty for SNs seemed to occur only for sub-string lookups and was not observed when running tests regarding Binary-Search, Sorting-Algorithms or String-Matching.

VII. SNs and Fuzzy Matching.

Fuzzy String-matching applications don't require exact matches and work even if the strings match with a relatively high probability. The strings may be equal or may not be equal, and, for these software, the fuzziness in matching may suffice.

For such applications, matching the first block of Numeric-String in the chain may be sufficient and performant. For dictionary words, this is true due to the data property that 2 arbitrary strings from the dictionary are extremely unlikely to share a common stem that is larger than 13 characters. Of course, some exceptions exist for large dictionary words with > 13 characters, especially around the derivatives of such words. Although, such word derivatives will be within a bounded distance of the root word and the length of the words will be likely different. Another way to achieve fuzziness could be checking if the two SNs are within a certain numeric-value of each other, implying that the corresponding Strings have a fuzzy match.

VIII. Performance Aspects.

A. *Data Impact*

Performance numbers for String-Number matching was higher than traditional string comparisons when the data being compared shared a prefix stem/characters. This observation is expected due to the fact that traditional string comparisons occur character by character, and if there are more characters to match before determining the result, it takes longer.

B. *Repetitive Comparisons*

It is not uncommon in the lifespan of a process to make comparisons for strings that it has compared in the past. String-Numbers are better suited for such repeated String comparisons and the performance improvements in this case are even more significant as the SN book-keeping re-computation is not required in this case. The prevalent string functions when replaced or supplemented by an equivalent String-Number function seem to compute the results much faster.

C. *Advantages Of String-Numbers.*

By replacing string operations from OS/Platform provided routines to String-Number routines in test programs, significant performance improvements were observed in the areas of Sorting, Searching, Sub-String-Lookups and String-Matching algorithms. These advantages were primarily observed as a reduction in time taken to perform string operations and translated to faster results, sometimes by an order of magnitude. SN based algorithms outperformed other similar algorithms like KMP, Rabin-Karp etc **[3]**.

D. *Memory Overhead.*

Replacing Strings with String-Numbers does add a memory overhead. In the contemporary scheme of things, the memory requirements for a char* or string is typically an array of bytes, whose length is proportionate the string length. An identical string when represented as a String-Number, also requires storing the String-Number portion of the string. This memory overhead includes storing the String-Number; a sequence of 64 bit-numbers, whose length is a fraction of the string length. Also, some string operations like substring-lookups require the String-Number scheme to store other book-keeping items in memory. For computing systems that are not memory-bound, this memory overhead might be acceptable. Moreover, String-Numbers can be applied to only those strings that are high-traffic-string-operation candidates during the process lifecycle, thereby alleviating the memory overhead.

IX. In Wider Outlook.

Prevalent Search-Engines and Databases mostly process ASCII or UTF-8/Unicode Data. For data in other encodings/charsets, the number of distinct characters which define a written language is less than 256 in most cases. Most written languages in the known world have a character set cardinality of 256 or less. Acknowledged that there are a few languages like Chinese, which have over thousands of Characters in prevalent dictionaries. As a further optimization, it is possible to extend the data set for non-ASCII characters via employing the Arithmetic DIV operation on the maximum character codes of characters/symbols with the character-count, thereby bringing their effective code/character value lower. This allows us to Map the relevant multi-byte words/strings to their corresponding String-Numbers and keep the String-Number chain-length size to 1 or a relatively smaller size for such strings.

Table-2 captures the number of characters per language for popular languages of the known world **[1]** & **[2]**. Additionally, there are modifier letters like é, ç etc. that contribute to an increase in the letter count for String-Number purposes in some way, alongside the upper/lower case variations.

Table 2: Number of letters in Alphabet for various languages[1].

Language	Number of letters/characters in the alphabet
English	26
Spanish	29
French	26
Italian	21
Arabic	28
Chinese	Thousands.
Japanese	46
Korean	24
Russian	33
Greek	24
German	26
Hindi	52
Punjabi/Gurmukhi	35
Urdu	36

X. Applications to Pattern-Matching: Genetic Alignment.

A. *Performance Benefits to Alignment.*

I explored SN's applicability to Pattern-Matching problems when aligning a Genetic Sequence. The alignment of Sequence Data to a Reference Genome is the first phase in bioinformatic analysis. The Sequence Alignment phase involves aligning or mapping the reads to a reference genome **[8], [9] & [10]**. This step tells us which precise location in the genome does each base pair in each sequencing read comes from. One way to accomplish this is to find occurrences or approximate matches for short reads or long reads in the Reference Genome Sequence. Often, repeated sub-string comparisons are required to be carried out during this process. For finding the Long and Short reads' substrings within the reference genome, I tried various algorithms and observed performance improvements when SN-based algorithms were used.

The alphabet representing a sequence of DNA and RNA molecules includes 'A', 'C', 'G', 'T', 'U' and 'N'. The four types of bases found in a DNA molecule are adenine (A), cytosine (C), guanine (G), and thymine (T). RNA uses uracil (U) instead of thymine (T). 'N' is used to represent skipped bases on reference when sequencer software cannot make a base-call for this base. This succinct Genome character set allowed for opportunities in optimization during Pattern matching. I used a single SN to represent 23 or more Bases at a time. This allowed SN based algorithms to gain considerable speedups in pattern matching and lookup operations.

I modified KMP and RK search algorithms to support SNs as described in Section-XI. Next, I compared the performance of the modified and the unmodified versions. Results yet again proved that SN based algorithms were faster by an order of magnitude.

XI. Comparisons with Contemporary Sub-string Search Algorithms.

A. *KMP, RK etc.*

I compared popular Genetic Pattern matching algorithms like Knuth-Morris-Pratt, Rabin-Karp etc. with the corresponding versions of SN algorithms. Both RK and KMP algorithms were modified to support SNs. Thereafter, performance tests were run comparing these SN-enhanced-version of algorithms with their unmodified counterparts. **Figure-1** shows the time taken by non-SN-versions to complete sub-string lookups and it compares them with the SN-versions.

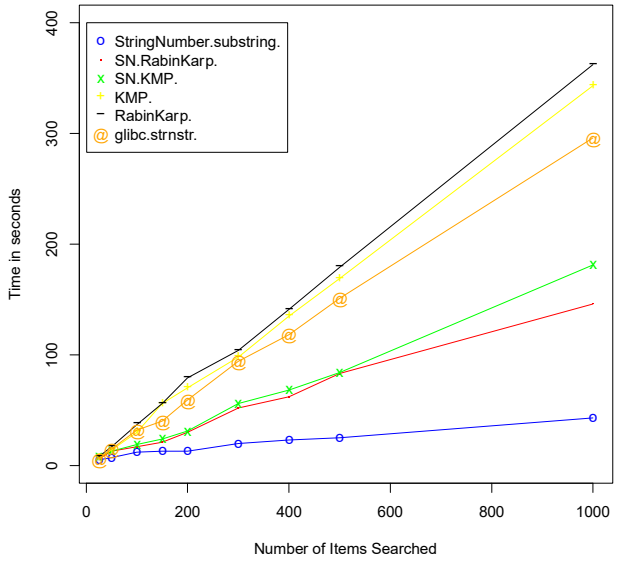


Figure-1: Genome Sequence Lookup in a FASTA file: SN versus Various Algorithms.

SN-Substring routines were compared with contemporary substring search algorithms like RabinKarp, KnuthMorrisPratt and glibc:strnstr Algorithms.

Lookup size for substrings was 400 characters. So needle size was 400. FileSize of Fasta file was 145MB. So, haystack size was 145MB.

The following Algorithms were tested with identical substring search queries. Each iteration's needle was chosen as a random substring from the haystack. Keeping the haystack constant, identical needle values were searched via the following algorithms.

1). Knuth-Morris-Pratt

2). Rabin-Karp

3). SN-RK

4). SN-KMP

5). SN-Pure

6). OS-glibc:strnstr

Table-3 compares the Time and Space complexities of SN-based substring search algorithm with KMP and RK versions.

Table 3: Worst-case Time/Space complexities for KMP, RK and StringNumber substring search algorithms.

Algorithm	Preprocessing	Time Complexity	Space Complexity
KMP	**O(M)**	**O(M+N)**	**O(M)**
RK	**O(N)**	**O(MN)**	**O(1)**
SN	**O(M+N)**	**O(log(M))**	**O(M+N)**

Results in **Figure-1** proved conclusively that SN based algorithms performed faster than all other tested algorithms. The SN-Pure substring searches outperformed all other tested Algorithms. It was followed by SN-RK and SN-KMP.

XII. TEST SETUP

The test-setup for ALL tests included the following:

- Tests were run on an ARM M1 chip based Mac computer.

For verification purposes, some tests were also run on an AMD Ryzen9 and Intel I5 chips based Linux computers. However, surprisingly, when comparing two large sized ascii strings, where the first 24 characters between the those strings were identical, the Linux strcmp outperformed SN routine. I am attributing this to some Linux/x86-64 internals that require further investigation because the same Linux strcmp surprisingly outperformed Linux's own memcmp routine!, which just compares raw bytes. The Linux strcmp routine was invoking its AVX2 based implementation as was revealed by some profiling. I suspect that Linux's memcmp routine too was AVX2 based. Also worth mentioning is that, when comparing ascii strings that differed within the first 24 characters, the SN-based routine comfortably out performed Linux's strcmp routine with results matching what was observed on ARM/MacOS. Additionally, the SN-based routines allow to be modified to their AVX2 form if required. This implementation was accomplished as part of this work too.

The Test Programs relevant to all the test-results shared herewith were implemented in C-Language. As a test, I also created AVX2 and Assembly based implementations for SNs, and both versions performed marginally better than the plain C-implementation of SNs. The relevant source-code and libraries are available upon request **[11].**

For comparison purposes, a test program implemented popular Sorting Algorithms like QuickSort, HeapSort, MergeSort and Binary Search Algorithms etc. The test results are shown in Sections XIII, XIV and XV.

XIII. SORTING RESULTS

Figure-2: Shows the time it took for various Sorting Algorithms like QuickSort, MergeSort and HeapSort to sort identical data sets and compares it with the time taken by their SN versions.

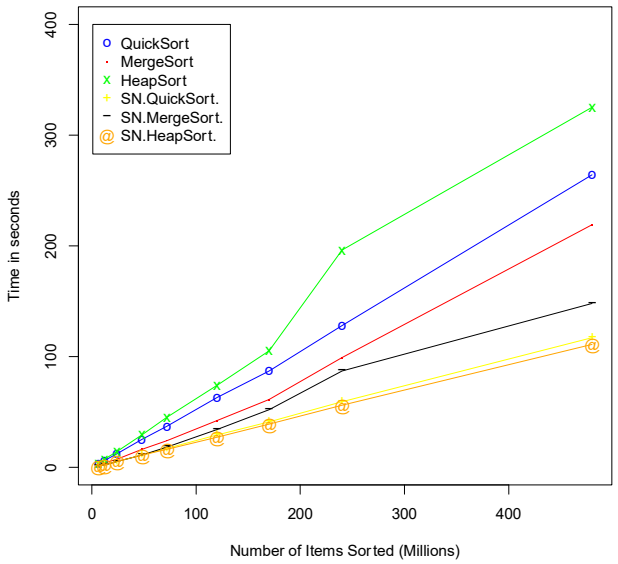


Figure-2: Sorting: SN versions versus regular versions of sorting Algorithms. SN-Sorting routines compared with regular string sorting routines for QuickSort, MergeSort and HeapSort Algorithms. All SN versions of Sorting Algorithms performed much better than equivalent non-SN versions.

XIV. Binary Search Results.

Figure-3 compares the time taken for BinarySearch by SN-version and the Contemporary BinarySearch Algorithm.

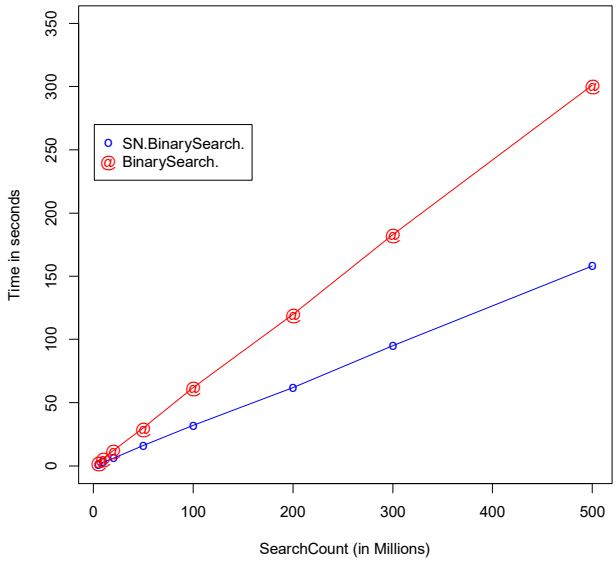


Figure-3: BinarySearch: SN Vs Contemporary BinarySearchAlgorithm. The SN-BinarySearch Algorithm performed up to 2.5Times faster than the Contemporary Binary Search Algorithm.

XV. String Lexicographic Comparisons Results .

Figure-4 shows the time it took to complete String Comparison operations via SN-strcmp and via the OS provided strcmp function: SN Vs OS glibc:strcmp function. Stings of various Sizes were compared with one another via the SN-strcmp and the glibc-strcmp functions and the time taken to complete the string compare operations was recorded.

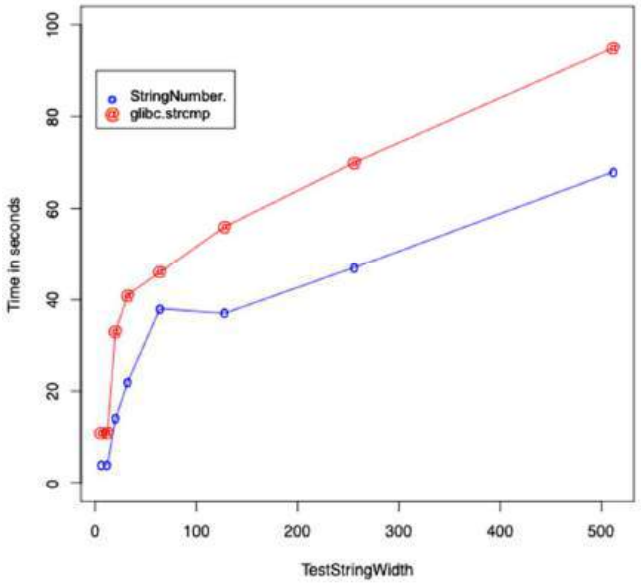


Figure-4: String Comparisons: SN-strcmp versus OS glibc:strcmp function. 5-Billon String Comparisons per test. The X-Axis reflects various String-Sizes. The Y-Axis represents time taken to complete the string matching operations.

XVI. Sub-String Lookup Results.

Table-4 shows the time it took to complete Sub-string Lookup Operations via glibc: strstr function and via SN-based sub-string lookup function. glibc:strstr implementation was the OS/Platform provided unmodified function. The SN-bsased strstr functions completed much faster than the glibc:strsstr functions for identical substring lookups.

XVII. PERSPECTIVE.

SNs offer a performant way to supplement and possibly replace how Strings are stored and computed upon by modern operating systems. They also indicate potential to bring improvements in areas like Data Compression and Data Transmission. I have observed Data Compression properties of SNs based on preliminary tests; an effort better suited for a future publication.

XVIII. CONCLUSION.

String-Numbers introduce an alternate way to represent Strings and provide a much faster way to perform substring lookups and string comparisons. Comparisons are made with the modern Operating Systems' provided substring-lookup-routines from glibc and state of the art contemporary algorithms like KMP and Rabin-Karp. SNs show considerable potential for speeding up algorithms in the areas of Databases, Web-Searches, Gene-Sequencing etc. SNs also show substantial performance improvements when implementing Sorting and Searching algorithms.

DATA Type	String (Haystack)	Sub - string (Needle)	DATA Length in bytes (HayStack, Needle)	TIME-TAKEN-(secs)	
				strstr	SN
UNICODE UTF-8	Cryllic, Gurmukhi and Chinese Characters	Cryllic	(11191, 13)	25.32	0.3
‘ ‘	‘ ‘	Gurmukhi	(11191, 48)	37.35	0.6
‘ ‘	‘ ‘	Chinese	(11191, 33)	19.82	0.2
ASCII-8 bit	8-byte characters	US-ASCII	(13204, 63)	45.19	0.2
‘ ‘	‘ ‘	US-ASCII	(13204, 59)	73.32	2.1
‘ ‘	‘ ‘	8-bit ASCII	(13204, 44)	32.21	1.5
US-ASCII	US-ASCII	US-ASCII	(12830, 72)	41.65	4.7

Table 4: Sub-string Lookups: glibc:strstr versus SN :strstr function. 10Million lookups were performed during each test.

REFERENCES

[1] How Many Letters are there in the Alphabet. https://wordcounter.net/blog/2015/11/24/10950_how-many-letters-alphabet.html

[2] Language Recognition Chart, https://en.wikipedia.org/wiki/Wikipedia:Language_recognition_chart

[3] Introduction To Algorithms: Book by Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest. Prentice-Hall.

[4] KMP, https://en.wikipedia.org/wiki/Knuth–Morris–Pratt_algorithm

[5] RK, https://en.wikipedia.org/wiki/Rabin–Karp_algorithm

[6] Distribution of Word Lengths in Various Languages, https://web.archive.org/web/20230405155600/http://www.ravi.io/language-word-lengths

[7] What is the average length of English words, https://www.quora.com/Whats-the-average-length-of-English-words

[8] https://eriqande.github.io/eca-bioinf-handbook/alignment-of-sequence-data-to-a-reference-genome-and-associated-steps.html

[9] https://www.ncbi.nlm.nih.gov/books/NBK464187/

[10] https://www.ncbi.nlm.nih.gov/books/NBK20261/

[11] Source code for the Test programs available upon request via email. Also **libStringNumbers.a** is available upon request via email vikramsw@mac.com.

CLIP-ViT Detector: Side Adapter with Prompt for Vision Transformer Object Detection

1st Han Yang
Institute of Intelligent Machines, HFIPS
Chinese Academy of Sciences
Hefei, China
University of Science and Technology of China
Hefei, China
matrixyh@mail.ustc.edu.cn

2nd Minxia Xu
Hangzhou Institute for Advanced Study
University of Chinese Academy of Sciences
Hangzhou, China
xuminxia22@mails.ucas.ac.cn

3rd Zhiyong Sun
Institute of Intelligent Machines, HFIPS
Chinese Academy of Sciences
Hefei, China
sunzymnzym@gmail.com

4th Bo Song
Institute of Intelligent Machines, HFIPS
Chinese Academy of Sciences
Hefei, China
songbo@iim.ac.cn

5th Erkang Cheng*
Institute of Intelligent Machines, HFIPS
Chinese Academy of Sciences
Hefei, China
twokang.cheng@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899669

***Abstract*—Object detection represents a fundamental and pivotal task within the domain of computer vision, which has attracted considerable interest in approaches that directly utilize Vision Transformer (ViT) to perform region-level recognition. However, despite the efforts of early pioneers in exploring vanilla ViT detectors, a significant performance disparity remains. Addressing this limitation, in this work, we propose an ViT-based detector to further enhance the ability of plain ViT in object detection by incorporating a frozen vision-language large model. Specifically, to boost the ViT detector with the frozen CLIP model, we construct ViT-based Side Prompt-Adapter Tuning, which align and fine-tune CLIP features without requiring gradients to flow through CLIP to provide additional rich semantic information for the ViT detector. Furthermore, a CLIP-based visual token selection module is proposed to leverage fine-tuned CLIP features to filter out irrelevant background visual tokens, resulting in decreased computational complexity and memory usage of the ViT-based detector. Additionally, we introduce query denoising training and adapt the position embeddings to further enhance training efficiency. Compared to the latest ViT-based detector, experimental results show that our method converges $3\times$ faster and achieves promising performance.**

***Index Terms*—object detection, Vision Transformer, CLIP, prompt, adapter**

I. Introduction

Object detection has been one of the most important tasks in computer vision, with the primary goal of identifying and localizing specific target objects in images. Recently, owing to the powerful representation of deep learning, modern detectors have made remarkable progress in object detection tasks. These detectors can be roughly divided into CNN-based methods [1], [2] and Transformer-based approaches [3], [4].

Inspired by the success of transformers in NLP, Vision Transformer (ViT) [5] demonstrates that directly applying a pure transformer to sequences of image patches can achieve comparable image classification performance. In recent studies, ViT has been extended for object detection tasks by utilizing it as a backbone. For instance, ViT-FRCNN [6] employs a pre-trained ViT as the backbone for Faster R-CNN [1]. Additionally, several works explore the use of ViT with multi-scale feature maps for object detection [7], [8], [9], [10], [11]. In contrast, YOLOS [12] proposes a different approach by using a vanilla pre-trained ViT encoder with minimal adaptation for object detection. However, this method may suffer from slow convergence and high computational complexity, which can be influenced by the pre-training scheme [12].

To enhance detection performance, researchers have been incorporating additional semantic information into pre-trained models by leveraging CLIP [13]. Recent advancements [14], [15], [16], [17] have demonstrated the effectiveness of using the CLIP image encoder as a backbone for addressing open vocabulary object detection challenges. Certain approaches [14], [15] employ a frozen CLIP image encoder as the backbone. Others [16], [17] utilize the frozen CLIP image encoder in conjunction with a knowledge distillation strategy. In addition, DenseCLIP [18] directly fine-tune the CLIP image encoder specifically for object detection tasks. Despite the object detection performance improvement, full fine-tuning of the entire CLIP image encoder increases computational costs and may result in the forgetting of initial weights. Also, DenseCLIP requires an additional object detection decoder to compute the detection results.

This work was supported in part by the National Natural Science Foundation of China under Grant 61973294, in part by the Anhui Provincial Key R&D Program under Grant (2023s07020017, 2022i01020020), in part by the University Synergy Innovation Program of Anhui Province, China, under Grant GXXT-2021-030, and in part by the Anhui Provincial Key Laboratory of Bionic Sensing and Advanced Robot Technology.

*Corresponding author.

To address aforementioned challenges, the prompt technique has been employed by introducing a minimal number of trainable parameters to the frozen model. In Prompt Tuning [19], [20], [21], a few learnable parameters are added alongside input images or middle features. Adapter Tuning [22], [23] inserts a light-weight trainable network into each frozen layer. These techniques allow for fine-tuning specific aspects of the model while keeping the majority of the pre-trained weights frozen. However, these techniques still require gradients to flow through the entire model during training, resulting in increased computational demands and higher GPU memory usage [24]. Instead, Side Adapter Tuning [24], [25], [26] train a light-weight independent network that is fused with the frozen pre-trained model via summation, but there can be misalignment between their feature maps.

In this paper, we present a method that combines a frozen CLIP model with a vanilla ViT detector as a side network. As shown in Fig. 1, the side network is specifically designed to adapt to the frozen CLIP model through the use of a Side Adapter. To address the mismatch between the feature maps produced by CLIP and the Side Adapter, we propose a Side Prompt Alignment module that efficiently utilizes a small number of trainable parameters to align the feature maps. To reduce the computational complexity of ViT-only detector, we propose a light-weight visual token selection module that utilizes the fine-tuned CLIP feature to filter out irrelevant background tokens, significantly reducing computational overhead. In the training stage, we employ a query denoising approach to enhance the training efficiency of the plain ViT detector. Furthermore, we adapt position embeddings for both visual tokens and detection tokens, leading to further improvements in the training efficiency and overall performance of the model.

We evaluate our method on the challenging COCO object detection benchmark. In comparison to YOLOS [12], the newest plain ViT detector, our approach demonstrates a convergence rate that is $3\times$ faster and achieves promising results.

II. Related works

A. Vision Transformer for Object Detection

The Vision Transformer (ViT) [5] is the first to demonstrate that the Transformer architecture can be directly applied to images by treating them as a series of patches. Recent advancements primarily concentrate on extending ViT to object detection by using it as the feature backbone. For example, ViT-FRCNN [6] propose to use a pre-trained vanilla ViT as the backbone for Faster R-CNN. In order to improve single-scale and low-resolution feature map of ViT, several approaches [7], [8], [9], [10], [11] introduce the pyramidal feature hierarchy to ViT design to get multi-scale feature maps for object detection. A Dynamic Token Halting Module [27] dynamically halts tokens at different layers of the ViT backbone to reduce the computational and memory costs of self-attention mechanisms in object detection. Instead, YOLOS [12] explore the transferability of the pre-trained vanilla ViT to object detection, which gets promising results with the fewest possible modifications.

B. CLIP for Object Detection

CLIP [13] is among the most widely utilized large-scale vision-language pre-trained models, which learns high-quality visual representation from a large-scale image-text dataset and shows remarkable capability on visual recognition tasks. Recent studies [14], [15], [18], [16], [17] have explored the application of CLIP image encoder as backbone to enhance the performance of object detectors. Typically, for open vocabulary object detection tasks, CORA [14] and F-VLM [15] utilize the frozen CLIP image encoder as the backbone and decode its output feature maps using trainable downstream modules. ViLD [16] and DetPro [17] freeze CLIP image encoder as a teacher model and train a relatively smaller backbone through the knowledge distillation strategy. This smaller backbone provides features for downstream detection modules. Additionally, DenseCLIP [18] utilizes the CLIP image encoder as a pre-trained backbone and performs full fine-tuning for object detection, which enhances the detection performance. DenseCLIP still relies on an additional object detection decoder for object detection tasks. In our paper, we explore an alternative approach by combining a frozen CLIP image encoder with a plain ViT architecture for object detection, eliminating the need for a separate decoder.

C. Prompt Tuning

In the field of NLP, Prompt Tuning [28] is an important method for efficiently adapting frozen large language models to downstream language tasks. With its proven effectiveness and the increasing application of vision-language large models, Prompt Tuning has also been introduced into the computer vision domain. For instance, various approaches [20], [21], [19], [29], [30] introduce learnable parameters into input images, texts, or the intermediate features of frozen pre-trained models. These learnable parameters, which are initialized randomly and independent of the input, are optimized using loss functions specific to the downstream tasks. Others [23], [31], [32], [33] employ light-weight sub-networks to construct visual prompts based on input images or intermediate feature maps, enriching these prompts with prior knowledge. These methods are referred to as the Inside Prompt methods because gradients propagate through the frozen pre-trained model during training.

D. Adapter Tuning

In the field of NLP, Adapter Tuning [34] is proposed to use frozen pre-trained models for various downstream tasks. It incorporates additional learnable and light-weight sub-networks into a frozen pre-trained model, enabling effective learning for diverse tasks. In the computer vision domain, several approaches [23], [34], [22], [35] have adapted the Adapter Tuning approach within frozen vision-language models. These approaches are referred to as the Inside Adapter. Although the Inside Adapter contains only a few parameters, gradients still propagate through the frozen pre-trained model during the training stage, resulting in significant computational overhead and GPU memory consumption. In contrast, alternative

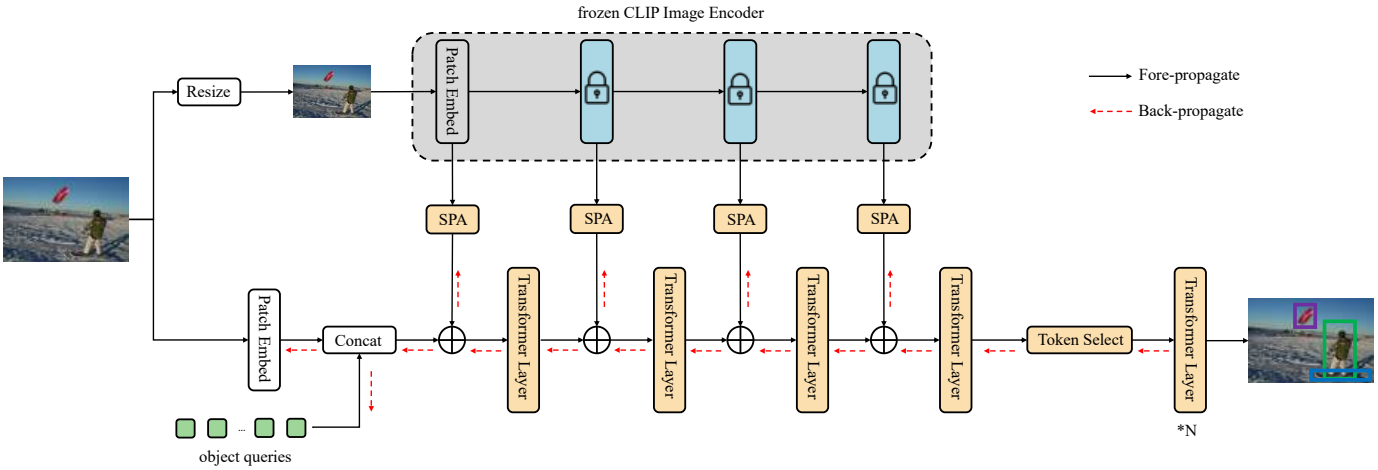


Fig. 1. Overview of our CLIP-ViT detector. In our framework, the frozen CLIP image encoder serves as an additional feature backbone, and the vanilla ViT detector with some object queries serves as a Side Adapter to fine-tune the feature from CLIP and computes detection results without any decoders. The Side Prompt Alignment (SPA) and the Token Select module will be proposed in Sec III-C and Sec III-D, respectively.

approaches [24], [26], [36], [25] take a different way by adapting the frozen pre-trained model through the training of a light-weight conditioned "side" model. This "side" model operates independently of the frozen pre-trained model and is denoted as the Side Adapter. The Side Adapter serves the same role as the Inside Adapter, however, during the training phase, gradients flow solely within the Side Adapter, which greatly reduces computational overhead and GPU memory consumption [24].

III. Methods

A. Overview

The overall architecture of our plain ViT detector with a frozen CLIP image encoder is illustrated in Fig. 1. Inspired by YOLOS [12], we adopt DeiT [37] as the baseline detector. We initialize N object queries as detection tokens, which are randomly initialized and concatenated with all visual tokens along the sequence dimension. In order to combine the plain ViT detector with a frozen CLIP model, the plain ViT detector is utilized as the Side Adapter. A Side Prompt Alignment (SPA) module is used to align the CLIP features with the features of the Side Adapter. In addition, a visual token selection module proposed to utilize the fine-tuned CLIP features to eliminate background tokens, which reduces the computational complexity of the ViT-based Side Adapter. The Side Prompt Alignment module is introduced in Sec III-C. The visual token selection module is listed in Sec III-D.

B. Preliminaries

For a plain ViT [5] with N layers, an input image $\boldsymbol{I} \in \mathbb{R}^{3\times H\times W}$ is divided into fixed-sized patches $\{\boldsymbol{I}_j \in \mathbb{R}^{3\times P\times P} \mid j \in \mathbb{N}, 1 \le j \le \frac{H}{P} \times \frac{W}{P}\}$. H, W are the height and width of the image and P is the size of a image patch. Then each patch is initially embedded into a d-dimensional latent feature space, incorporating positional encoding:

$$\boldsymbol{e}_0^j = \text{PatchEmbed}(\boldsymbol{I}_j) + \text{PE}_j. \tag{1}$$

where $\boldsymbol{e}_0^j \in \mathbb{R}^d, j = 1, 2, \ldots, \frac{H}{P} \times \frac{W}{P}$ represents the j-th image patch embeddings, which we refer to as visual tokens.

We denote the collection of visual tokens as $\boldsymbol{E}_i = \{\boldsymbol{e}_i^j \in \mathbb{R}^d \mid j \in \mathbb{N}, 1 \le j \le \frac{H}{P} \times \frac{W}{P}\}$, which serve as inputs to the $(i+1)$-th vision transformer layer, denoted as L_{i+1}. Along with an additional learnable classification token, denoted as [CLS], the computation of the entire ViT can be formulated as:

$$[\boldsymbol{x}_i, \boldsymbol{E}_i] = L_i([\boldsymbol{x}_{i-1}, \boldsymbol{E}_{i-1}] \tag{2}$$

$$\boldsymbol{y} = \text{MLPHead}(\boldsymbol{x}_N) \tag{3}$$

where $i = 1, 2, \ldots, N$. $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes the embedding of the [CLS] token at the input space of vision transformer layer L_{i+1}. The notation $[\cdot, \cdot]$ indicates concatenation along the sequence dimension. Each vision transformer layer L_i comprises Multi-Head Self-Attention (MSA), Feed-Forward Networks (FFN), Layer Normalization [38], and residual connections [39]. A MLP prediction head with an activation function (e.g., Sigmoid) is employed to project the [CLS]'s embedding from the final layer of the ViT into a predicted probability distribution over the classes.

The vanilla ViT detector proposed by YOLOS [12] replaces the [CLS] for image classification in ViT with some object queries proposed by DETR [3] for object detection, which can be denotes as:

$$[\boldsymbol{Q}_i, \boldsymbol{E}_i] = L_i([\boldsymbol{Q}_{i-1}, \boldsymbol{E}_{i-1}] \tag{4}$$

$$\boldsymbol{C}, \boldsymbol{B} = \text{MLPDetHead}(\boldsymbol{Q}_N) \tag{5}$$

where $\boldsymbol{Q}_i = \{\boldsymbol{q}_i^m \in \mathbb{R}^d \mid m \in \mathbb{N}, 1 \le m \le M\}$ denote object queries at L_{i+1}'s input space. $\boldsymbol{C} = \{\boldsymbol{c}^m \in \mathbb{R}^k \mid m \in \mathbb{N}, 1 \le m \le M\}$ and $\boldsymbol{B} = \{\boldsymbol{b}^m \in \mathbb{R}^4 \mid m \in \mathbb{N}, 1 \le m \le M\}$ are class probability distribution and 2D bounding boxes predicted by each object query $\boldsymbol{q}^m$. k represents the number of categories.

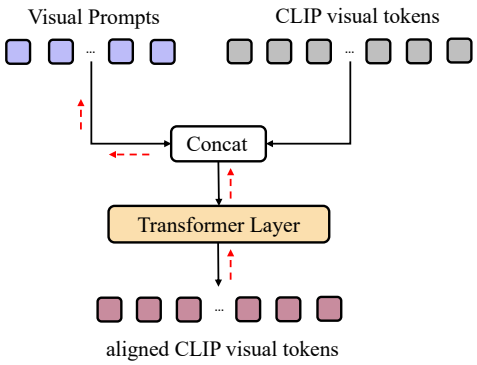


Fig. 2. The Side Prompt Alignment bridges the gap between the feature maps of Side Adapter and the frozen CLIP image encoder with contiguous visual prompts.

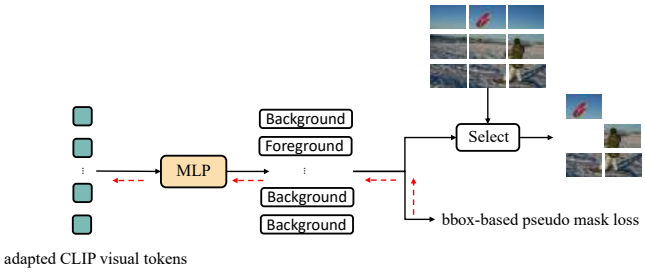


Fig. 3. To decrease the computational complexity, the CLIP-based visual token selection utilizes the fine-tuned CLIP feature to filter out irrelevant background visual tokens.

C. Side Adapter with Side Prompt Alignment

To fine-tune the features of the frozen pre-trained model, Prompt Tuning [19], [20] combines some learnable parameters and the input tokens of each frozen layer along the sequence dimension. Unlike Prompt Tuning, which focuses on fine-tuning at the input level, other approaches perform parameter fine-tuning at the model design level. For example, Adapter Tuning [22], [35] integrates light-weight networks after each frozen layer. These methods allow gradients to propagate through the frozen pre-trained model during training, which is inefficient during the training stage [24]. To address this, Side Adapter [24], [26], [25] constructs a light-weight network that operates independently of the model to combine features via summation. In this paper, we propose Side Prompt Alignment (SPA) to align the features from the frozen pre-trained model and the Side Adapter, effectively bridging the distribution gap between them.

Following SAN [26], we integrate the visual tokens from the first three stages of the frozen CLIP image encoder into the shallow vision transformer layers of the ViT-based Side Adapter. The fused features are subsequently refined by the remaining vision transformer layers. The process of feature fusion on visual tokens can be denoted as:

$$\overline{\boldsymbol{C}}_k = \text{UpSample}(\boldsymbol{C}_k) \tag{6}$$

$$\overline{\boldsymbol{E}}_k = \boldsymbol{E}_k + \overline{\boldsymbol{C}}_k \tag{7}$$

where $k = 1, 2, 3, 4$. $\boldsymbol{C}_k$ and $\overline{\boldsymbol{C}}_k$ is the features of the k-th stage of the ViT-based CLIP without classification embeddings, respectively. $\overline{\boldsymbol{E}}_k$ is the features after integrating CLIP features.

Due to the different resolutions of the input image of the Side Adapter and the frozen CLIP image encoder, there exists a distribution misalignment between both image feature maps. We propose a module to bridge this gap by adapting the CLIP feature with a few learnable visual prompts $\boldsymbol{P} \in \mathbb{R}^{N \times C}$, where N is the number of the visual prompts, and C is the dimension of the CLIP features. Different from the previous visual prompt [19], [21], this module operates independently of the frozen CLIP image encoder, preventing gradients from propagating through CLIP. Therefore, we refer to this module as the Side Prompt Alignment (SPA), which is shown in Fig. 2. With the proposed SPA, (6) is modified as:

$$\overline{\boldsymbol{C}}_k = \text{UpSample}(\text{SPA}([\boldsymbol{P}_k, \boldsymbol{C}_k])) \tag{8}$$

Except for the feature fusion of visual tokens between the CLIP and the shallow vision transformer layers of Side Adapter, the interactions between object queries and visual tokens are similar with YOLOS [12]. As described by (4), in each vision transformer layer, they are treated equally and perform global interactions. Moreover, different from YOLOS, we utilize anchor-based object queries and sinusoidal encoding for the position embeddings of object queries and visual tokens to adapt query denoising training [40], which is introduced in Sec. III-E.

D. CLIP-based Visual Token Selection

One of the major disadvantages of plain ViT for object detection is their relatively high computational and memory costs, even when processing common input image sizes. For instance, in the COCO benchmark [43], images typically have a shorter edge of 800 pixels, which can be quite demanding for the vanilla ViT.

To mitigate the computational complexity associated with the vanilla ViT detector, we adopt a visual token selection strategy [44], [45], [27]. As shown in Fig. 3, we incorporate a token-selecting module before the earliest feasible vision transformer layer. Leveraging the rich semantic information embedded in fine-tuned CLIP feature maps, we can directly identify which pixels belong to the foreground or background, instead of using the cascading architecture proposed by previous methods, which can be denoted as:

$$\boldsymbol{E}_5^{logits} = \text{Sigmoid}(\text{MLP}(\boldsymbol{E}_5)) \tag{9}$$

$$\boldsymbol{E}_5^{mask}(x, y) = \begin{cases} \text{True} & \text{if } \boldsymbol{E}_5^{logits}(x, y) > \tau \\ \text{False} & \text{otherwise} \end{cases} \tag{10}$$

$$\boldsymbol{E}_5^{sparse} = \text{IndexSelect}(\boldsymbol{E}_5^{mask}, \boldsymbol{E}_5) \tag{11}$$

where $\boldsymbol{E}_5^{mask} \in \{\text{True}, \text{False}\}^{\frac{H}{P} \times \frac{W}{P}}$ is a boolean mask, and then we can directly extract the visual tokens belonging to

TABLE I

PERFORMANCE COMPARISON WITH THE NEWEST VANILLA VIT DETECTOR YOLOS [12]. YOLOS-TI (TINY) AND -B (BASE) ARE EQUIVALENT TO DEIT-TI AND -B [12], [37]. AND WE UTILIZE VIT-B/16 AND VIT-L/14@336PX FOR YOLOS-TI AND YOLOS-B, RESPECTIVELY. † DENOTES ADDITIONAL TRANSFORMER-SPECIFIC DISTILLATION TOKEN INTRODUCED BY DEIT [37], WHICH CAN FURTHER IMPROVE DETECTION PERFORMANCE [12]. THE RESULTS OF YOLOS WITH 100 & 50 TRAINING EPOCHS ARE REPORTED IN THE TRAINING LOG OF FULLY CONVERGED YOLOS.

Method	Epoch	AP ↑	AP_{50} ↑	AP_{75} ↑	AP_{small} ↑	AP_{medium} ↑	AP_{large} ↑
YOLOS-Ti [12]	300	28.7	47.3	29.1	**9.7**	29.3	46.1
YOLOS-Ti [12]	100	20.6	37.7	19.5	5.1	19.6	35.8
Ours-Ti	100	**30.2** ↑ 1.5	**48.3** ↑ 1.0	**30.5** ↑ 1.4	7.5	**29.8** ↑ 0.5	**55.9** ↑ 9.8
YOLOS-B† [12]	150	42.0	62.3	**44.6**	**19.6**	45.2	62.7
YOLOS-B† [12]	50	33.0	54.0	33.6	12.1	35.6	54.6
Ours-B	50	42.8	63.5	44.4	17.7	46.4	68.4
Ours-B†	50	**43.0** ↑ 1.0	**63.8** ↑ 1.5	**44.6**	18.1	**46.5** ↑ 1.3	**68.6** ↑ 5.9

TABLE II

COMPARISON WITH CNN-BASED METHOD FASTER R-CNN AND TRANSFORMER-BASED METHOD DETR WITH CNNS BACKBONE, WHICH IS REIMPLEMENTED BY DETR [3] AND CONDITIONAL DETR [41]. * REPRESENTS THE SAME MODELS, BUT TRAINED USING MMDETECTION [42] WITH THE 4X SCHEDULE AND TWO ADDITIONAL EPOCHS, AND NOT INCLUDING A WARM-UP PHASE, ENSURING ALIGNMENT WITH DETR AND OURS.

Method	Epoch	AP ↑	AP_{50} ↑	AP_{75} ↑	AP_{small} ↑	AP_{medium} ↑	AP_{large} ↑
Faster RCNN-R101-FPN [3]	108	44.0	63.9	47.8	27.2	48.1	56.0
DETR-DC5-R101 [3]	500	44.9	64.7	47.7	23.7	49.5	62.3
Faster RCNN-R101-FPN*	50	40.1	40.3	43.8	23.3	44.3	52.5
DETR-DC5-R101 [41]	50	38.6	59.7	40.7	17.2	42.2	57.4
Ours-B†	50	43.0	63.8	44.6	18.1	46.5	68.6

the foreground from $\boldsymbol{E}_5$ with this mask by IndexSelect (e.g., boolean indexing in PyTorch [46]).

In addition to reducing the computational complexity and memory costs of the plain ViT detector, the proposed CLIP-based one-step visual token selection also significantly minimizes the interference from irrelevant background information, which allows object queries and subsequent vision transformer layers to concentrate more effectively on detecting various foreground objects.

E. Query DeNoising Training

In order to stabilize training and accelerate convergence of DETR families [3], [4], Query DeNoising Training (QDT) is first proposed by DN-DETR [47] and subsequently improved by DINO [40]. QDT treats the forward propagation process of anchor-based object queries [48] as a denoising process. Given a slightly coarse 2D anchor around an object to be detected, the model can gradually refine to be more accurate. If the 2D anchors are significantly coarse, the model classifies them as negative samples.

Similar to Conditional-DETR [41], we separate an object query or visual token into content embeddings and position embeddings. In the case of the vanilla ViT, both embeddings are initialized as learnable d-dimensional feature vectors. Without auxiliary losses [3], it becomes challenging for object queries to adequately interact with each vision transformer layer and effectively learn to detect objects within a limited number of training epochs.

To apply QDT to the vanilla ViT detector, we initialize object queries as learnable anchors with random initialization and use sinusoidal encoding to obtain position embeddings for both visual tokens and anchor-based object queries:

$$\boldsymbol{q}_p^i = \text{MLP}(\text{PE}(\boldsymbol{A}_q^i)) \tag{12}$$

$$\boldsymbol{e}_p^i = \text{MLP}(\text{PE}(\boldsymbol{A}_e^i)) \tag{13}$$

where $\boldsymbol{A}_q^i = (x_q^i, y_q^i, w_q^i, h_q^i)$ and $\boldsymbol{A}_e^i = (x_e^i, y_e^i)$ are anchor of i-th object query and position of i-th visual token in the image coordinate system, respectively. PE is sinusoidal encoding without learnable parameters and MLP is shared between object queries and visual tokens. With this encoding method, the position embeddings $\boldsymbol{q}_p^i$ and $\boldsymbol{e}_p^i$ are projected to the same latent feature space. It is easier for the plain ViT detector to capture the positional relationships between object queries and visual tokens, enhancing the suitability for the functionality of QDT.

F. Loss

Our optimization objective aligns with YOLOS [12], which uses Hungarian matching loss in a set prediction manner proposed by DETR [3]. To create foreground and background ground truth for the token selection module, we adopt the approach in DenseCLIP [18]. Specifically, we utilize the bounding boxes and labels in the training set to construct a binary target $\boldsymbol{y}_{mask} \in \{0,1\}^{H \times W}$, considering all categories as foreground. Subsequently, we employ a binary cross-entropy loss [18] and dice loss [49] to supervise the training of the CLIP-based visual token selection method.

IV. EXPERIMENTS

A. Implementation settings

We use DeiT-tiny and DeiT-base [37] as our vanilla ViT detector. The first 9 vision transformer layers of ViT-B/16

TABLE III
ABLATION STUDY OF SIDE PROMPT ALIGNMENT (SPA) ON COCO2017 VAL.

Method	AP ↑	AP_{50} ↑	AP_{75} ↑	AP_{small} ↑	AP_{medium} ↑	AP_{large} ↑
Ours-Ti w/o SPA	27.0	45.2	26.8	5.6	25.9	51.4
Ours-Ti w/ SPA	**27.5** ↑0.5	**45.6** ↑0.4	**27.2** ↑0.4	**6.3** ↑0.7	**26.5** ↑0.6	50.7
Ours-B w/o SPA	42.0	62.8	44.2	16.1	45.4	68.5
Ours-B w/ SPA	**42.8** ↑0.8	**63.5** ↑0.7	**44.4** ↑0.2	**17.7** ↑1.6	**46.4** ↑1.0	68.4

TABLE IV
ABLATION STUDY OF THE IMPORTANCE OF QUERY DENOISING TRAINING (QDT) AND RANDOMLY INITIALIZED LEARNABLE EMBEDDINGS (RILE) V.S. SINUSOIDAL ENCODING WITH ANCHOR-BASED EMBEDDINGS (SEAE). THE TOP SECTION IS VIT-B/16 OF CLIP WITH DEIT-TI AND THE BOTTOM SECTION IS VIT-L/14@336PX WITH DEIT-B.

Method	QDT	RILE	SEAE	AP ↑	AP_{50} ↑	AP_{75} ↑	AP_{small} ↑	AP_{medium} ↑	AP_{large} ↑
Ours-Ti		✓		27.5	45.6	27.2	6.3	26.5	53.7
	✓	✓		29.2	47.4	29.3	**6.6**	28.9	53.8
	✓		✓	**30.2** ↑2.7	**48.3** ↑2.7	**30.5** ↑3.3	6.3	**29.8** ↑3.3	**55.9** ↑2.2
Ours-B		✓		40.3	62.0	41.5	14.8	43.7	65.8
	✓	✓		42.0	63.0	43.9	17.2	46.2	67.1
	✓		✓	**42.8** ↑2.5	**63.5** ↑1.5	**44.4** ↑2.9	**17.7** ↑2.9	**46.4** ↑2.7	**68.4** ↑2.6

TABLE V
ABLATION STUDY OF THE THRESHOLD IN CLIP-BASED VISUAL TOKENS SELECTION ON COCO2017 VAL. THE DROPOUT RATIO REPRESENTS THE PROPORTION OF VISUAL TOKENS IDENTIFIED AS BACKGROUND TOKENS AFTER PROCESSING THROUGH THE CLIP-BASED VISUAL TOKEN SELECTION PROPOSED IN SEC III-D. ALL EXPERIMENTS ARE IN VIT-B/16 OF CLIP WITH DEIT-TI.

τ	Dropout Ratio (%)	AP ↑	AP_{50} ↑	AP_{75} ↑	AP_{small} ↑	AP_{medium} ↑	AP_{large} ↑
0.7	56.9	28.9	47.1	28.8	6.9	28.0	53.6
0.6	55.6	29.1	47.3	29.0	7.1	28.1	53.6
0.5	54.2	30.2	48.3	30.5	7.5	29.8	55.9
0.4	53.4	30.3	48.5	30.5	7.8	30.5	55.2

CLIP image encoder and the first 18 vision transformer layers of ViT-L/14@336px CLIP image encoder are aligned by Side Prompt Alignment and fused with the first 4 vision transformer layers of DeiT-tiny and DeiT-base, respectively. The two variants are denoted as Ours-Ti (Tiny) and Ours-B (Base) in all tables. All trainable layers and modules are trained on the COCO object detection benchmark [43], following YOLOS [12]. Specifically, while the parameters of the vision transformer layers are initialized from the pre-trained DeiT, the remaining trainable parameters are initialized using a truncated normal distribution. The optimization of all trainable parameters is performed using the AdamW optimizer with a learning rate of 2.5×10^{-5} and a batch size of 1 per GPU. We employ a cosine learning rate decay schedule and a weight decay of 1×10^{-4}. For data augmentation, we use the same protocols as those in YOLOS [12] to ensure a fair comparison.

B. Main result

Comparison with YOLOS. As shown in Table I, compared with YOLOS, our method has shown improvements on most evaluation metrics (especially on large objects) while requiring only one-third of the training epochs. By incorporating a transformer-specific distillation token supervised by a CNNs teacher model as used in DeiT [37], our method further enhances detection performance, particularly for small objects. The detection of small objects achieves comparable performance to YOLOS. We believe that leveraging an enhanced version of a region-level training vision-language large model, such as RegionCLIP [50], has the potential to enhance the detection of small objects.

Comparison with Faster R-CNN and DETR. We also conduct a comparison with two popular detectors, Faster R-CNN [1] and DETR [3]. As presented in Table II, our approach demonstrates superior detection performance within a limited number of training epochs, especially for large objects, despite being less competitive when compared to fully converged Faster R-CNN and DETR. However, similar to the experimental findings observed with DETR, the detection of small objects using only single-scale feature maps remains a significant challenge. Therefore, it is crucial to investigate methods for enhancing the local perception capability of the vanilla ViT detector specifically for small objects in our future work.

C. Ablation studies

Effectiveness of Side Prompt Alignment. We evaluate the importance of the alignment between feature maps of the frozen CLIP image encoder and Side Adapter (Table III). In version Ti, overall Average Precision (AP) improves by 0.5 points, AP on small and medium objects improves by 0.7 points and 0.6 points, respectively. In version B, overall AP improves by 0.8 points, with a more significant improvement of 1.6 points and 1.0 points on small and medium objects, respectively. The integration of Side Prompt Alignment boosts the object detection performance. Firstly, it aligns the feature map generated by the frozen CLIP image encoder, which

primarily captures global semantic information, with the feature map utilized by the Side Adapter for local perception. This alignment significantly improves the overall detection performance. Secondly, it extracts and amplifies the relevant semantic information about small and medium-sized instances from the feature map produced by the frozen CLIP image encoder. This enhancement strengthens the local sensing capabilities of the Side Adapter specifically for small and medium-sized instances, further boosting the detection accuracy.

Learnable PE v.s. Sinusoidal PE. In Table IV, when employing ViT-B/16 of CLIP with DeiT-Ti, the utilization of the prediction of object queries as a denoising process, leads to a significant enhancement (❶ v.s. ❷) in Average Precision (AP). We make a similar observation when using ViT-L/14@336px with DeiT-B (❹ v.s. ❺). This conclusion is consistent with that of DINO [40]. In QDT, the vanilla ViT detector approach initializes both position embeddings randomly and allows them to be learnable. In our modified version, we adopt a different approach. Each object query is treated as an anchor and uses sinusoidal encoding to be projected into the latent space. Additionally, the position embeddings of the image are sinusoidally encoded for each pixel's coordinates to ensure consistency between the position embeddings of the object query. These modifications result in the refinement of the position embeddings in the plain ViT detector and the incorporation of learnable anchor-based object queries, leading to notable performance improvements (❷ v.s. ❸ and ❺ v.s. ❻).

Threshold in Token Selection. The threshold employed in CLIP-based visual token selection facilitates the selection of visual tokens by identifying those most likely to belong to the foreground, thereby excluding tokens that lack valid information and mitigating potential interference with the Side Adapter. As demonstrated in Table V, an increase in the threshold results in a reduction of tokens classified as foreground, enhancing computational efficiency. However, this comes at the cost of degrading the spatial information within the feature map, consequently diminishing the Side Adapter's detection performance. This effect becomes more significant as the threshold exceeds 0.5, resulting in a deterioration in Average Precision (AP). For example, when the threshold is set to 0.6, AP decreases by 1.1. Balancing computational efficiency with detection performance, a threshold of 0.5 is selected as the optimal choice, aligning with standard threshold selection practices in binary classification tasks.

V. Conclusion

In this paper, we present CLIP-ViT, which further improve the vanilla ViT detector in object detection with frozen CLIP image encoder. To achieve this, we use a vanilla ViT detector as a Side Adapter to fine-tune CLIP features. To address the distribution discrepancy, we introduce Side Prompt Alignment with learnable visual prompts for alignment. Additionally, we implement a light-weight CLIP-based visual token selection mechanism to filter out extraneous background tokens, reducing computational overhead. To enhance training efficiency, we further introduce the query denoising training strategy and adapt the position embedding for visual tokens and object queries. The approach demonstrates significantly better performance than the latest ViT detector, YOLOS, and achieves comparable results to the popular Faster R-CNN and DETR in terms of training convergence speed and detection performance. Furthermore, CLIP-ViT demonstrates markedly superior performance on large-scale objects compared to these methods, which can be attributed to the enhanced perceptual capabilities of CLIP in recognizing large-scale objects.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[4] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[6] J. Beal, E. Kim, E. Tzeng, D. H. Park, A. Zhai, and D. Kislyuk, "Toward transformer-based object detection," *arXiv preprint arXiv:2012.09958*, 2020.

[7] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11 936–11 945.

[8] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4804–4814.

[9] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[10] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European conference on computer vision*. Springer, 2022, pp. 280–296.

[11] P. Chen, M. Zhang, Y. Shen, K. Sheng, Y. Gao, X. Sun, K. Li, and C. Shen, "Efficient decoder-free object detection with transformers," in *European Conference on Computer Vision*. Springer, 2022, pp. 70–86.

[12] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, and W. Liu, "You only look at one sequence: Rethinking transformer in vision through object detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 183–26 197, 2021.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[14] X. Wu, F. Zhu, R. Zhao, and H. Li, "Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7031–7040.

[15] W. Kuo, Y. Cui, X. Gu, A. Piergiovanni, and A. Angelova, "F-vlm: Open-vocabulary object detection upon frozen vision and language models," *arXiv preprint arXiv:2209.15639*, 2022.

[16] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," *arXiv preprint arXiv:2104.13921*, 2021.

[17] Y. Du, F. Wei, Z. Zhang, M. Shi, Y. Gao, and G. Li, "Learning to prompt for open-vocabulary object detection with vision-language model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 084–14 093.

[18] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, and J. Lu, "Denseclip: Language-guided dense prediction with context-aware prompting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 082–18 091.

[19] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*. Springer, 2022, pp. 709–727.

[20] Y. Wang, Z. Huang, and X. Hong, "S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5682–5695, 2022.

[21] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola, "Exploring visual prompts for adapting large-scale models," *arXiv preprint arXiv:2203.17274*, 2022.

[22] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 664–16 678, 2022.

[23] X. Zhou, D. Liang, W. Xu, X. Zhu, Y. Xu, Z. Zou, and X. Bai, "Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 707–14 717.

[24] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, "Clip-adapter: Better vision-language models with feature adapters," *International Journal of Computer Vision*, vol. 132, no. 2, pp. 581–595, 2024.

[25] J. O. Zhang, A. Sax, A. Zamir, L. Guibas, and J. Malik, "Side-tuning: a baseline for network adaptation via additive side networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 698–714.

[26] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai, "Side adapter network for open-vocabulary semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2945–2954.

[27] M. Ye, G. P. Meyer, Y. Chai, and Q. Liu, "Efficient transformer-based 3d object detection with dynamic token halting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8438–8450.

[28] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[29] Z. Zhou, Y. Lei, B. Zhang, L. Liu, and Y. Liu, "Zegclip: Towards adapting clip for zero-shot semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 175–11 185.

[30] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[31] J. Loedeman, M. C. Stol, T. Han, and Y. M. Asano, "Prompt generation networks for input-based adaptation of frozen vision transformers," *arXiv preprint arXiv:2210.06466*, 2022.

[32] J.-W. Zhang, Y. Sun, Y. Yang, and W. Chen, "Feature-proxy transformer for few-shot segmentation," *Advances in neural information processing systems*, vol. 35, pp. 6575–6588, 2022.

[33] Y. Zhang, K. Zhou, and Z. Liu, "Neural prompt search," *arXiv preprint arXiv:2206.04673*, 2022.

[34] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[35] Y.-L. Sung, J. Cho, and M. Bansal, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5227–5237.

[36] ——, "Lst: Ladder side-tuning for parameter and memory efficient transfer learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 991–13 005, 2022.

[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.

[38] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[40] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[41] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional detr for fast training convergence," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3651–3660.

[42] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[44] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, "Dynamicvit: Efficient vision transformers with dynamic token sparsification," *Advances in neural information processing systems*, vol. 34, pp. 13 937–13 949, 2021.

[45] Q. Tang, B. Zhang, J. Liu, F. Liu, and Y. Liu, "Dynamic token pruning in plain vision transformers for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 777–786.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[47] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 619–13 627.

[48] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "Dab-detr: Dynamic anchor boxes are better queries for detr," *arXiv preprint arXiv:2201.12329*, 2022.

[49] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. Ieee, 2016, pp. 565–571.

[50] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li *et al.*, "Regionclip: Region-based language-image pretraining," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 793–16 803.

Wavelet-Driven Multi-Model Ensemble: A Synthesis Box for Time Series Forecasting

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899633

*Rui Tang
Department of Applied Mathematics
Beijing Normal University-Hong Kong Baptist University United International College
Zhuhai, China
*s230026144@mail.uic.edu.cn

Minglei Lyu
Department of Applied Economics
Beijing Normal University-Hong Kong Baptist University United International College
Zhuhai, China
t330036104@mail.uic.edu.cn

Yuwen Zheng
Department of Finance
Beijing Normal University-Hong Kong Baptist University United International College
Zhuhai, China
s230024309@mail.uic.edu.cn

Abstract—This paper presents a wavelet-driven multi-model ensemble framework for time series forecasting, aimed at addressing the limitations of traditional and machine learning-based methods in handling non-stationary data, periodic components, and stochastic variations. The framework begins with wavelet decomposition, separating the original data into trend, periodic, and stochastic components. The periodic component is analyzed using Fast Fourier Transform (FFT) and predicted via an AutoRegressive Integrated Moving Average model, followed by reconstruction through inverse Fourier transform. For the trend and stochastic components, feature engineering techniques are employed to train an Adaboost model, enabling iterative forecasting with dynamically updated feature sets. Experimental results demonstrate the framework's effectiveness, achieving a SMAPE of 0.8% , thereby verifying the model's robustness in handling diverse temporal dynamics. The results highlight the robustness of the model over different temporal dynamics, and the proposed method will greatly improve the reliability of the model in diverse application scenarios.

Keywords—Wavelet Transform, FFT, Adaboost, Feature Engineer, Iterative, ARIMA, Time Series

I. INTRODUCTION

Time series forecasting is crucial for decision-making in various fields, raising a significant question: how can we effectively automate the selection and application of the most suitable forecasting method [1][2]? Traditional approaches typically assume stationarity in the data, limiting their ability to handle non-linear dynamics, complex seasonal fluctuations, or long-term dependencies. Furthermore, they are sensitive to outliers, often resulting in inaccurate predictions.

Machine learning methods, on the other hand, excel at capturing complex non-linear relationships and diverse patterns, enabling them to identify intricate temporal dependencies and trends. However, when faced with noisy data or insufficient training samples, these methods are prone to overfitting, leading to poor generalization. Both approaches face challenges in accurately predicting long-term trends, periodic changes, and dynamically adapting to new conditions.

This raises the following research question:

Q1: How can we construct a forecasting framework entirely based on historical data to address the shortcomings of standalone machine learning and traditional models in handling non-stationarity and periodic features, thereby offering a more comprehensive forecasting approach?

Most datasets exhibit complex dynamic characteristics comprising three main components: long-term trends, periodic fluctuations, and unpredictable stochastic variations. This gives rise to the following research question:

Q2: How can we more accurately identify and handle these distinct components in time series data to deepen our understanding of their underlying mechanisms and enhance the robustness of forecasting models [3]?

In many forecasting scenarios, developing a model that is independent of external factors yet capable of efficient predictions based on the intrinsic properties of the data is critical. Such a model should adapt to real-time data and market changes while maintaining predictive accuracy. This leads to the next research question:

Q3: How can we design a model that ensures automatic extraction of intrinsic data features, thereby guaranteeing adaptability and minimizing reliance on external factors [4]?

This approach will greatly improve the usefulness and reliability of the model in diverse application scenarios.

Contribution

Our study makes three primary contributions:

1.We introduce wavelet decomposition to precisely identify the components of time series data, including long-term trends, periodic fluctuations, and stochastic variations.

2.We construct a comprehensive forecasting framework combining machine learning and feature engineering with Fourier series and traditional forecasting models.

3.We establish a feature engineering approach based on historical data and incorporate the ARIMA model to extract autocorrelation features for predicting the periodic component.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

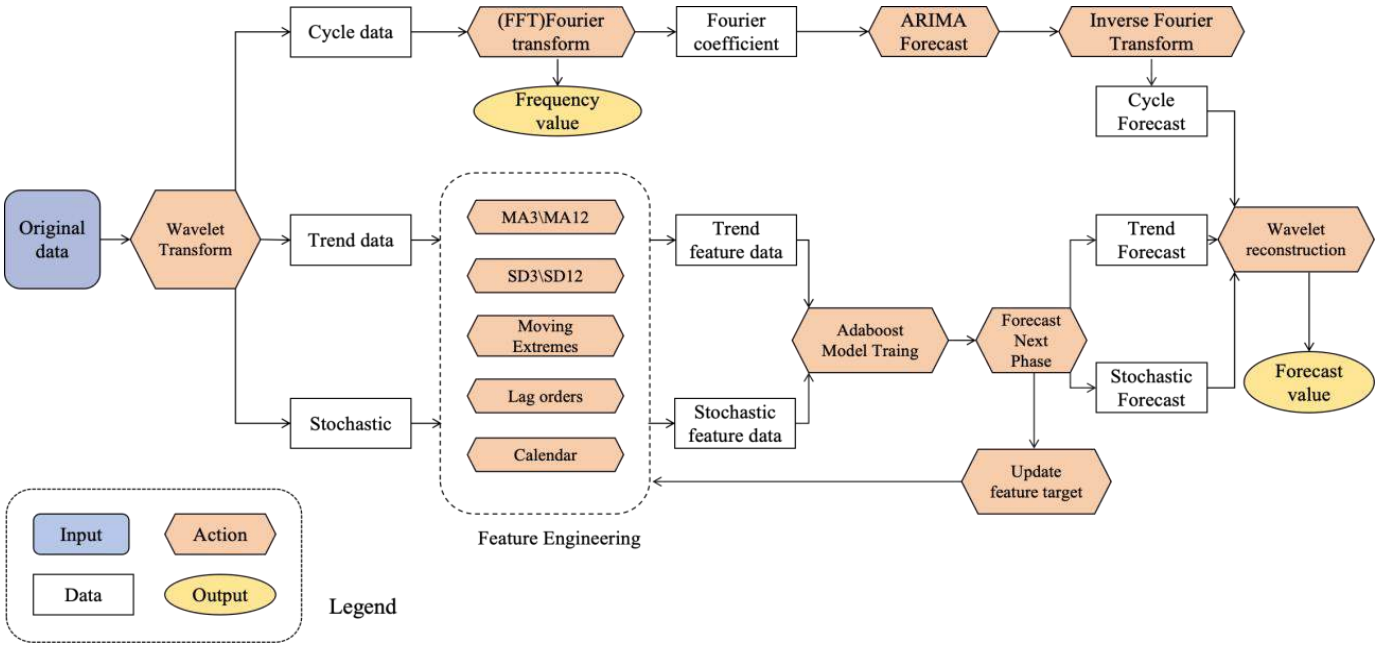


Fig. 1. Multi-Model ensemble workflow

II. MULTI-MODEL ENSEMBLE FRAMEWORK

Figure 1 illustrates the proposed prediction framework based on wavelet decomposition and multi-model integration. First, the original data is decomposed into trend, periodic, and stochastic components using wavelet transform. For the periodic component, its frequency characteristics are extracted using Fast Fourier Transform (FFT), and the Fourier series is forecasted with an AutoRegressive Integrated Moving Average (ARIMA) model. The predicted periodic component is then reconstructed into the time domain using inverse Fourier transform.

For the stochastic and trend components, an Adaboost model is trained using feature engineering. By continuously refreshing the feature vectors, future values are iteratively predicted. Finally, the predicted trend, periodic, and stochastic components are combined to produce the final forecast.

A. Wavelet Transform

Wavelet decomposition is a powerful time-frequency analysis technique that enables the identification and extraction of trend, periodicity, and stochastic components within time series data. By decomposing the data into distinct frequency bands, it provides a detailed representation of its temporal and spectral characteristics.

A time series X_t can be conceptualized as a dynamic system comprising three components: the trend T_t, periodic C_t, and stochastic components S_t. Mathematically, this can be expressed as follows:

$$X_t = T_t + C_t + S_t \quad (1)$$

After applying the wavelet transform, these components can be separated and reconstructed. Specifically, the original time series is decomposed into detailed coefficients (high-frequency components) and scale coefficients (low-frequency components). The reconstructed time series can then be represented as [5]:

$$X_t = g_{N,t} + \ldots + g_{M,t} + g_{M-1,t} + \ldots + g_{1,t} + x_N \quad (2)$$

Among tem:

- x_N is the low frequency component, to the trend term T_t

- $g_{M,t}$ to the $g_{N-1,t}$ is a composition of intermediate frequency, to the periodic item C_t.

- $g_{1,t}$ to the $g_{M-1,t}$ is the high frequency component, corresponding to the random item S_t

B. Periodic Term Prediction

1) Fast Fourier Transform (FFT)

A fast Fourier transform is an algorithm used to convert a time-domain signal into a frequency-domain representation. For a periodic time series $T(t)$, the FFT is expressed as:

$$X(f) = \int_{-\infty}^{\infty} T(t) e^{-j2\pi ft} dt \quad (3)$$

Where $X(f)$ is the complex representation in the frequency domain, f is the frequency, and j is the imaginary number unit.

2) Fourier Coefficient

The Fourier coefficient is calculated by FFT and expressed as[6]:

$$a_n = \frac{2}{T}\int_0^T T(t)\cos\left(\frac{2\pi nt}{T}\right)dt \quad (4)$$

$$b_n = \frac{2}{T}\int_0^T T(t)\sin\left(\frac{2\pi nt}{T}\right)dt \quad (5)$$

Where a_n and b_n are the Fourier coefficients of the cosine and sine components, respectively, and T is the period of the signal.

3) ARIMA model prediction

In the frequency domain, the ARIMA model is used to predict Fourier coefficients. The general form of the ARIMA model is:

$$\hat{y}_t = \mu + \sum_{i=1}^{p}\phi_i \hat{y}_{t-i} + \sum_{j=1}^{q}\theta_j \epsilon_{t-j} \quad (6)$$

p, q, and d are the orders of the autoregressive, moving average, and differencing terms, the $\hat{y}_t$ is predicted, μ is a constant term, $\phi_i$and θ_j is autoregressive and moving average parameters respectively, ϵ_{t-j} is the error term.

For optimal parameter selection of p , q , and d , the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) is utilized. By systematically comparing the AIC or BIC values across different parameter combinations, the configuration that minimizes the information criterion is chosen to balance model accuracy and complexity.

4) Inverse Fourier transform

The predicted Fourier coefficients are converted back to the time domain by inverse Fourier transform to generate the predicted period data:

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{y}_t\, e^{j2\pi ft} df \quad (7)$$

C. Feature Engineering

Let y_t is current time point.

- Calculate the moving average over the past 3 or 12 cycles to provide a smooth trend for the data:

$$MA_3 = \frac{y_{t-2} + y_{t-1} + y_t}{3}$$

$$MA_{12} = \frac{y_{t-11} + y_{t-10} + \cdots + y_{t-1} + y_t}{3} \quad (8)$$

- Calculate the moving standard deviation over the past 3 or 12 periods, which is a measure of how time series data has fluctuated over a certain time window:

$$SD_3 = \frac{\sqrt{\sum_{i=t-2}^{t}(y_i - y_{t-2,t})^2}}{3}$$

$$SD_{12} = \frac{\sqrt{\sum_{i=t-11}^{t}(y_i - y_{t-11,t})^2}}{12} \quad (9)$$

- Calculate the maximum and minimum values of the past 3 or 12 periods, an indicator that measures the extreme values of time series data within a time window:

$$MAX_n = max(y_{t-n+1}, \ldots, y_{t-1}, y_t)$$

$$MIN_n = min(y_{t-n+1}, \ldots, y_{t-1}, y_t) \quad (10)$$

- Lag feature: Using past values to predict future values, based on the self of time series

$$Lag_1 = y_{t-1}\ , Lag_2 = y_{t-2}\ , Lag_3 = y_{t-3} \quad (11)$$

- Weekday feature: This represents the day of the week and helps capture the difference between weekends and weekdays.

D. Adaboost Model

In ensemble learning, multiple decision tree classifiers are combined to build a strong classifier, and these weak classifiers are trained in each iteration round[7]. In each iteration, the new weak learner is forced to pay more attention to the difficult examples by increasing the weight of those examples that were misclassified by the previous weak learner and decreasing the weight of those that were correctly classified, this is Adaboost[8]. The framework is shown in Figure2. Where $f_1, \ldots, f_k$ stands for weak classifier—Decision tree and $f(x)$ stands for combined classifier.

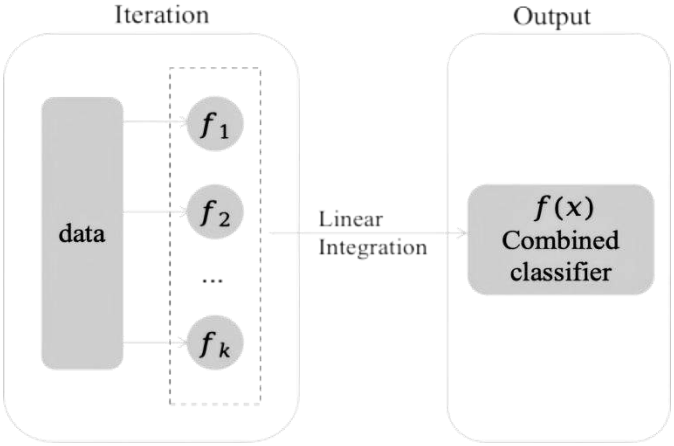


Fig. 2. AdaBoost framework diagram

E. Trend and Random Term Forecasting

Feature engineering is performed separately for the trend and stochastic components to generate feature sets, referred to as feature_trend and feature_stochastic. These components serve as the target variables for supervised learning models. Subsequently, an Adaboost model is independently trained for each component to capture their distinct temporal patterns.

An iterative forecasting approach is adopted to predict the future trends of the time series data. After each prediction, the model's feature set is dynamically updated to incorporate the

most recent data points. This update involves recalculating essential time series features, including moving averages, standard deviations, extremes, and growth rates. Such dynamic feature refinement ensures that the model adapts to the latest data characteristics and accurately reflects the evolving temporal dynamics.

F. Wavelet Reconstruction

After the predicted values of trend, random and periodic items are obtained, wavelet reconstruction technique is used to recombine them to accurately recover the original shape of the time series.

III. Experiments and Results

We select the sales volume of a certain electronic shopping item in the past half year to verify the accuracy of the algorithm framework. Data for the next twenty periods of its prediction[9]. The popular symmetric Mean Absolute Percentage Error (SMAPE) metric is used to evaluate the global prediction performance[10].

$$SMAPE = \frac{1}{N}\sum_{i=P+1}^{P+N} \frac{/y_i - \hat{y}_i/}{(/y_i/+/\hat{y}_i/)/2} \times 100\% \quad (12)$$

The Daubechies wavelet, db4, was chosen as the basis for the analysis because of its ability to provide good time and frequency resolution when processing the data. The approximation coefficients containing the trend and the detail coefficients representing the periodic and random components are obtained. Then, the dynamic soft threshold filtering based on its maximum absolute value multiplied by a threshold (0.7) is applied to these detail coefficients to reduce noise and extract signal features. The periodic, trend and random terms are obtained after signal reconstruction. Figure 3 shows the original data and the data after WT.

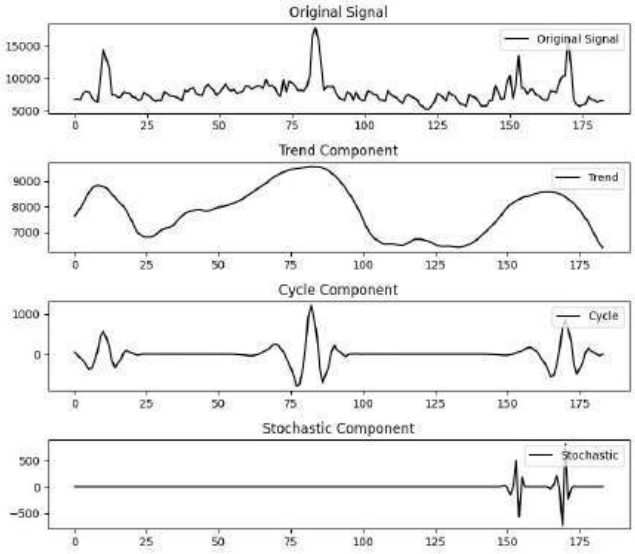


Fig. 3. Original data and the data after WT.

Firstly, the Fast Fourier Transform (FFT) is applied to the original time series data to convert the time domain signal into a frequency domain representation. In the frequency domain, by comparing the BIC values of different parameter combinations, the parameter combination (1,0,0) that minimizes the information criterion is selected to construct the ARIMA model for the real and imaginary parts of each frequency component. The dynamic changes of the frequency components are captured. By fitting these models, each frequency future value is predicted. The real and imaginary parts of the predicted frequency components are combined to form Fourier coefficients in the complex form. Then, IFFT is applied to convert these coefficients back into the time domain to generate the final predicted time series. Figure 4shows the comparison, amplitude spectrum, phase spectrum, real and imaginary part of the original signal and the predicted signal.

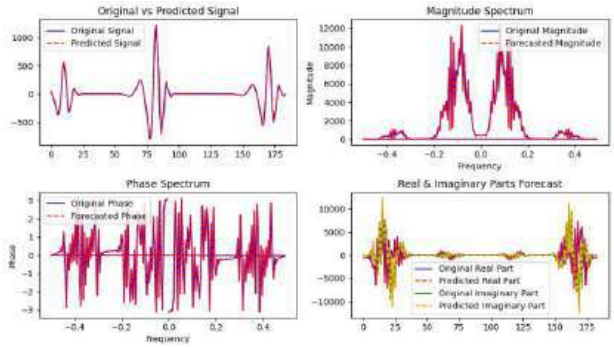


Fig. 4. Comparison of raw and predicted data

Feature engineering and adaboost models were built for trend and random items, respectively.Feature engineering and adaboost models were built for trend and random items, respectively. For AdaBoost, setting random states 42 ensures model repeatability, and 120 weak learners are used to balance model performance and complexity, thereby improving accuracy and reducing the risk of overfitting[11].

Recursive prediction techniques are used to simulate the future trend of time series data[12]. After each forecasting iteration, the time series model refreches its feature set with the newly generated data, which includes recalculating important time series metrics. Figure 5 shows the 20-period predicted values for the three items.

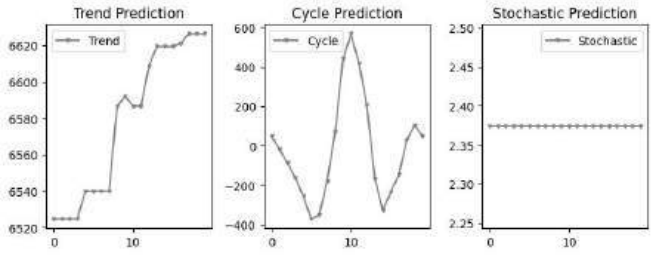


Fig. 5. 20-period predicted values for the three items.

Finally, the predicted periodic term, trend term and random term are reconstructed by wavelet, and the final predicted value is obtained. The SMAPE between the calculated real value and the predicted value is 0.8%, and the goodness of fit is 0.818, indicating that the error of the prediction model is very small, which is considered to be a very good prediction performance.

Figure 6 shows the comparison between the actual and predicted values and the QQ plot. In the QQ plot, most of the blue points are close to the red line, indicating that the residuals of the model are close to normal distribution[13].

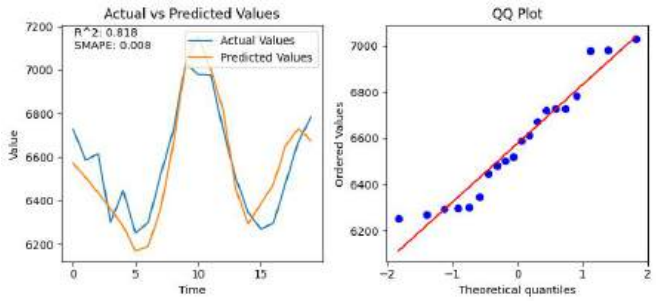


Fig. 6. Actual vs. Prediction values

IV. CONCLUSION

In this study, we propose a wavelet transform based multi-model ensemble framework that aims to challenge the traditional "no free lunch" theory and address the limitations of single machine learning and traditional models when dealing with non-stationary and periodic features[14][15]. Our framework provides a comprehensive prediction approach, thus enhancing the usefulness and reliability of the model in diverse application scenarios. Through wavelet transform decomposition and multi-model integration, our models deeply learn the inherent dynamics of historical data, including trends, periodicity, and randomness, enabling accurate predictions of future time series changes.

The problems proposed in this paper have also been effectively solved:

1. We have successfully built a prediction framework based entirely on historical data, which not only overcomes the shortcomings of single machine learning and traditional models.

2. We accurately identify the different components in the time series data by wavelet decomposition, and deeply analyze and predict these components by feature engineering techniques and ARIMA model. This approach allows us to more accurately identify and deal with long-term trends, periodic fluctuations, and random variations in time series data.

3. The model we designed can automatically extract the intrinsic features of the data, thus ensuring the adaptability of the model. It solves the problem that in the absence of external data support, it can only rely on historical data for prediction. This model based on the intrinsic properties of the data is able to adapt to real-time data and market changes while maintaining the accuracy of the prediction.

Overall, our research not only addresses the issues raised in the cited literature, but also, through our contribution, provides new tools to the field of time series forecasting. Our framework performs well in our experiments, and the results validate the potential of our approach to improve the usefulness and reliability of models in diverse application scenarios.

REFERENCES

[1] R. J. Hyndman and G. Athanasopoulos, Forecasting: principles and practice. Melbourne, Australia: OTexts, 2018.

[2] Shumway, R.H., Stoffer, D.S. (2017). ARIMA Models. In: Time Series Analysis and Its Applications. Springer Texts in Statistics. Springer, Cham. https://doi.org/10.1007/978-3-319-52452-8_3

[3] exandridis, A. K., Panopoulou, E., & Souropanis, I. (2024). Forecasting exchange rate volatility: An amalgamation approach. Journal of International Financial Markets, Institutions & Money, 97, 102067.https://doi.org/10.1016/j.intfin.2024.102067

[4] A. Bauer, M. Züfle, N. Herbst, S. Kounev and V. Curtef, "Telescope: An Automatic Feature Extraction and Transformation Approach for Time Series Forecasting on a Level-Playing Field," 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 2020, pp. 1902-1905, doi: 10.1109/ICDE48307.2020.00199.

[5] Qiong Yu, Ming Qi, Shujuan Wang and Guofu Zhai, "Research on life prediction based on wavelet transform and ARMA model for space relay," 2009 4th IEEE Conference on Industrial Electronics and Applications, Xi'an, China, 2009, pp. 1275-1280, doi: 10.1109/ICIEA.2009.5138407.

[6] Issac, B. , Singh, H. , Kaur, H. , & Raghava, G. P. S. . (2002). Locating probable genes using fourier transform approach. Bioinformatics, 18(1), 196-197.

[7] Wu, X., Lu, X., & Leung, H. (2018). A Video Based Fire Smoke Detection Using Robust AdaBoost. Sensors (Basel, Switzerland), 18(11), 3780.

[8] Li, K., Zhou, G., Zhai, J., Li, F., & Shao, M. (2019). Improved PSO_AdaBoost Ensemble Algorithm for Imbalanced Data. Sensors (Basel, Switzerland), 19(6), 1476.

[9] Peralta Donate, J., Cortez, P., Gutiérrez Sánchez, G., & Sanchis de Miguel, A. (2013). Time series forecasting using a weighted cross-validation evolutionary artificial neural network ensemble. Neurocomputing, 109, 27-32.

[10] https://doi.org/10.1016/j.neucom.2012.02.053

[11] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, 22(4), 679-688. https://doi.org/10.1016/j.ijforecast.2006.03.001

[12] Wang, R. (2012). AdaBoost for feature selection, classification and its relation with SVM, a review. Physics Procedia, 25, 800-807. https://doi.org/10.1016/j.phpro.2012.03.160

[13] R. Jiang, P. Li and K. Zhang, "Quantile-Quantile Plot of Folded-Normal Distribution and its Applications in Reliability and Quality Modeling," 2024 10th International Symposium on System Security, Safety, and Reliability (ISSSR), Xiamen, China, 2024, pp. 44-50, doi: 10.1109/ISSSR61934.2024.00011

[14] (2019).Information Technology - Data Streaming; Researchers at University of Louisville Target Data Streaming (No Free Lunch Theorem for concept drift detection in streaming data classification: A review).Information Technology Newsweekly

[15] Biagio Ciuffo & Vincenzo Punzo.(2014)."No Free Lunch" Theorems Applied to the Calibration of Traffic Simulation Models..IEEE Trans. Intelligent Transportation Systems(2),553-562.

Stacking LSTMs to extract features in two-dimensional data for prediction tasks on travel time and crimes frequency

1st Xiangdong Ran*
School of Artificial Intelligence
Beijing Information Technology College
Beijing, China
ranxiangdong@hotmail.com
*Corresponding author

2rd Kai Niu
School of digital Business
Beijing Information Technology College
Beijing, China
Niuk@bitc.edu.cn

3ndFanxing Deng
School of Artificial Intelligence
Beijing Information Technology College
Beijing, China
Dengfx@bitc.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899699

***Abstract*—Data-based methods have some advantages over other methods for prediction tasks, with domain knowledge is not required. In these methods, Long-short Term Memory has achieved tremendous success in recent years, because its capable of feature extraction of time series. However, there are fewer studies on using Long-short Term Memory to extract the features in spatial dimension. In our work, we attempt to stack Long-short Term Memorys to extract the spatial dimensional features while completing feature extraction of time series for prediction tasks. An multilayer stacking method with multiway inputs and an grid mode stacking method were proposed to extract the features in two dimensions. In the grid mode stacking method, an memory block was proposed to merge the gated cell vectors and each Long-short Term Memory is for one embedding vector separately. Based on two public datasets, we evaluated the proposed methods on travle time prediction and crime prediction. Travel time dataset is provided by Highways England, and crime dataset is provided by the Home Office. Experimental results on the datasets demonstrate that the proposed methods perform well compared to the baseline methods.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

A useful property of Long-short Term Memory (LSTM) is that it can learn how to map the history information among data into a fixed dimensional vector representation. In recent years, we have seen a revival of LSTM [1] with its effectiveness on a wide range of tasks such as traffic predict [2], speech recognition [3], sentiment classification [4], and machine translation [5] [6]. In these works, models were trained and tested on datasets with no domain knowledge is required, and the time series information in datasets were coded and summarized in Memory Cells into a time sequence representation. However, there are no studies on using Long-short Term Memory to extract the features in spatial dimension in above works.

In this paper, the datasets that is fed to the proposed models for prediction consist of one time series data and one data that is spatially related to the time series data. The merger of the time series data and the spatially related data is called two dimensional data. Such as, one time series data is from one link [2], and then two dimensional structure is a merger of the link and the other link that is spatially related to the link. To gain the better accuracy of prediction, we proposed an stacked LSTMs method into a grid structure model to extract features from datasets in two dimensions for prediction task. We proposed an memory block to consider the history memories of multiple child nodes. In terms of the architecture, our model is perhaps the most similar to the multilayer LSTM network [3]. In terms of the memory block, our model is perhaps the most similar to the S-LSTM [7]. In our work, all layers used an same hidden layer function. The main contributions of our work can be summarized as follows.

- We described an multilayer LSTMs network(M-LSTM) on two dimensional data for prediction task. Based on the multilayer LSTMs network, we proposed a grid-structure LSTMs(Grid-LSTM) on two dimensional data.
- We proposed an memory block to extract the features from multiple child nodes in Grid-LSTM. The discriminative regions within the extracted features are effectively highlighted and localized in the different areas in both temporal and spatial dimensions, demonstrating the generalization ability of our model on two-dimensional data.
- We conducted experiments on the Highways England dataset and the Home Office dataset. Experimental results demostrate that the proposed models can work well on two dimensional data and improve the prediction performance compared with the baseline methods.

The rest of this paper is organized as follows. Section II reviews the related studies. Section III describes MM-LSTM and Grid-LSTM methods. Section IV presents the experiments based on the Highways England dataset and the Home Office dataset, analyses the experimental results. Section V provided our conclusion.

II. Related work

In data-based prediction methods, functions that relate the explicative variables with the target variable $f(\dots)$ usually is not obtained from domain knowledge, it instead is determined using statistical, machine learning or deep learning methods. The main advantage of these methods is that they do not require expertise in the field of transportation.

Deep learning methods

RNN suffers from the problem of the vanishing or exploding gradients that may grow or decay exponentially over long sequences [8] [1]. RNN have difficulty performing long-term memorization, for example, an simple copying task that outputs the same input sequence they just read [9]. LSTM is capable of capture long-term dependencies. A distinguishing feature of LSTM that differs from Rnn is the complex structure in its hidden layer [1] called memory block. One shortcoming of RNN is that they are only able to make use of previous context. Such as, whole utterances are transcribed at once in speech recognition, there is no reason not to exploit future context. Bidirectional RNN can do this by processing the data in positive and negative time direction with two separate hidden layers and obtained a better results than other approaches in some regression and classification experiments [10]. In [11], the authors study proposes a method combining a convolutional neural network (CNN) and LSTM for analyzing and compensating spatiotemporal features in residents' travel data. Scatter Wavelet is a form of Deep Learning that derives features from Gabor filters, utilizing an architecture similar to convolutional neural networks [12].

To benefit from depth in spatial dimension, Alex et.al introduced a deep LSTM and assess their potential for speech recognition [3]. In [3] [13], M-LSTM took a fixed size vector as input and built up progressively the higher level representations of acoustic data, and gives good results, such as in speech recognition. To utilize the given structures of data, Xiaodan et.al proposed to extend LSTM to a tree structures, in which a memory cell can reflect the history memories of multiple child nodes in a recursive process [7]. The tree structures provide a principled way of considering long-distance interaction over hierarchies. The memory block in the tree struture LSTM, consisting of an input gate, two forget gates (the same as the number of child nodes) and an output gate. Experiments shows that the proposed method can achieve a better performance than the baseline methods without considering the structures.

Traffic prediction methods

There are many data-based prediction methods for traffic prediction. Linear regression is one of the most typical used methods for traffic prediction. In [14], a linear relation between future travel time and current travel time is discovered. The authors call τ the 'current time' and their aim is to predict future travel time $TT_e(\tau+\delta)$ for a given (nonnegative) 'lag' δ. Two naive predictors for $TT_e(\tau+\delta)$, $T_e^\star(\tau)$ and $\mu_{TT}(\tau+\delta)$, were proposed. $T_e^\star(\tau)$ predicts well for small δ and $\mu_{TT}(\tau+\delta)$ predicts better for large δ. $\mu_{TT}(\tau+\delta)$ is the historical mean travel time. It is important to notice that the computation of $TT_e(\tau+\delta)$ only requires information available at time t. The authors also presented an experimental comparison with other methods, including principal components and nearest neighbors. The authors have gathered data from single link, and have not gathered data from multiple links. The work consider the linear relationship between the future travel time and two naive predictors of the future travel time, and don't consider the non-linear relationship between them.

Nikovski et al., presented an experimental comparison of several machine learning methods for short-term travel time prediction on road segments. The compared methods include linear regression, neural networks, regression trees, k-nearest neighbors and locally weighted regression, which were tested on the same historical data. Despite the expected superiority of non-linear methods over linear regression, the only non-linear method that could consistently outperform linear regression was locally weighted regression [15].

Neural network is also one of the most typical used methods for traffic prediction. Many different types of neural networks have been applied to traffic prediction, such as regular multilayer feed-forward neural networks [16], recurrent neural networks [17] [2] and stacked auto-encoder model [18]. In [18], the authors stacked autoencoders to form a deep neural network that is used to learn traffic flow features.

In [19], LSTM has been suggested to estimate the daily traffic volume of rural roads, becuase LSTM resulted in the highest percentage of estimation accuracy while having no overfitting issue. In [20],

Crime Prediction Methods

Crime prevention has entered a new, more robust phase of research activity, which is more relevant to current policies and practices than ever before [21]. A recent research trend is the development of models for crime prevention. Many of these models are hot spot analysis, there are fewer models focusing on the methods of crime prevention. In [22], Ginger et.al focused on the prediction of crime frequency as a numeric value rather than as a label (hot/cold spot). They explored three models for predicting the frequency of several types crimes and anti-social behavior crimes. Metaheuristic algorithms [23] [24] can either reduce the complexity of optimization problems, effectively reducing the scope of the search space, especially in the case of multi-objective optimization for complex problems.

These prediction methods can make use of historical data, but they fewer focused multi-dimensional data. It remains a challenge on how to effectively utilize multi-dimensional data to improve the performance of prediction model. Therefore, it is motivated that devising a LSTM neural network model to achieve better prediction accuracy by utilizing multi-dimensional data.

III. LSTM-based model for prediction

A. Multilayer LSTMs neural network

The transition function of LSTM has three gate units and one cell: forget gate unit f_t, input gate unit i_t, output gate unit o_t, and memory cell state c_t. Cell c_t is like a conveyor

belt that is the key to the success of LSTM. The gate units adaptively keep or override information in cell c_t. Gate f_t decides what information will throw away from cell c_t, gate i_t decides what new information will store in cell c_t, gate o_t decides how to access cell c_t to output information [1]. LSTM is so more sophisticated and powerful because of the gates and cell allow long-distance correlations in sequences to be better modeled. The transformation function of LSTM that parameterize the input gate, forget gate, output gate and cell state respectively are described in detail as Equ.1-6. Where $W_i,\ W_f,\ W_o,\ W_c \in R^{d\times 2d}$ are weighted matrices, and $b_i,\ b_f,\ b_o,\ b_c \in R^d$ are bias parameters, which will be learned during model training. σ is an sigmoid function, and $\odot$ denotes element-wise multiplication. x_t denotes the input sequence for LSTM. h_t denotes the output vector of LSTM.

$$i_t = \sigma(W_i \cdot [h_{t-1};\ x_t] + b_i) \tag{1}$$
$$f_t = \sigma(W_f \cdot [h_{t-1};\ x_t] + b_f) \tag{2}$$
$$g_t = tanh(W_c \cdot [h_{t-1};\ x_t] + b_c) \tag{3}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{4}$$
$$o_t = \sigma(W_o \cdot [h_{t-1};\ x_t] + b_o) \tag{5}$$
$$h_t = o_t \odot tanh(c_t) \tag{6}$$

M-LSTM (see Fig. 1) is an utilization of deep architecture on vertical direction that can build up progressively higher level representations of input data. When M-LSTM is used to solve prediction problem, a predictor need be linked to the tail of M-LSTM. M-LSTM feeds h_{mn} to the predictor that transforms h_{mn} into a floating-point scalar $\hat{y}$ that is regarded as the prediction value. In [3] [13], M-LSTM only extract some temporal dimensional features from the phoneme recognition data, and fed the features into a predictor that transforms the features into a floating-point scalar. The floating-point scalar is regarded as a prediction value. M-LSTM do not take two dimensional data as the input sequences, the spatial dimensional features that is related to prediction task can't be extracted.

In our work, the proposed method considers two dimensional data for prediction task. The proposed method consists of an M-LSTM and an predictor. We use a single neural network layer $n \times 1$ as the predictor. Every LSTM unit has the same weighted matrices parameters $W_i, W_f, W_o \in R^{d\times 2d}$ and the same bias parameters $b_i, b_f, b_o \in R^d$. These parameters will be learned in an end-to-end training way. Based on M-LSTM, we extended M-LSTM in structure to incorporate multiway inputs from two dimensional data. One input is the spatial dimensional information that are extracted by LSTM unit. The other input is the temporal dimensional information that are extracted by LSTM unit. When LSTM unit receive input sequences, both spatial and temporal dimensional information will be captured simultaneously. The overview of M-LSTM for multiway inputs is illustrated in Fig.1. Where D_i, D_j and D_k have the same data structure. There is a spatial dimensional relationship between D_i, D_j and D_k. There is a temporal dimensional relationship between $D_{i,t-3}$, $D_{i,t-2}$ and $D_{i,t-1}$. D_i, D_j and D_k are concatenated into an multiway input to predict the information of D_i at next lag.

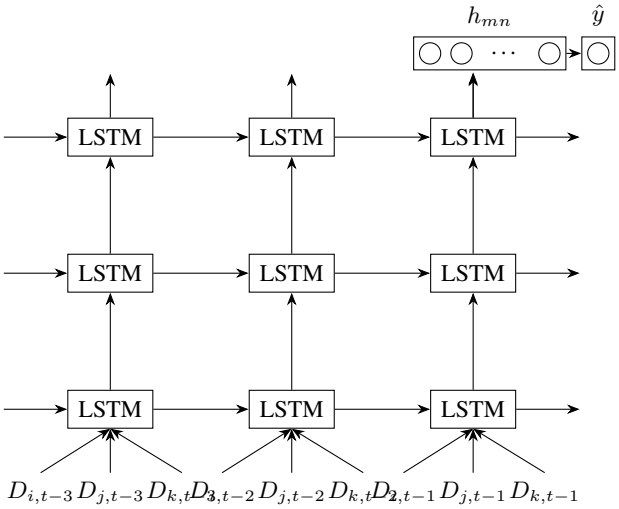


Fig. 1. The architecture of multiway input for M-LSTM. M-LSTM is created by vertical stacking of LSTM units, where the outputs of bottom layer form the inputs of upper layer. In M-LSTM with n layers, the first layer receives the inputs from training set or test set, and the outputs of the $i-th$ layer form the inputs of the $(i+1)-th$ layer. The output of last LSTM unit h_{mn} is the final representation.

B. Grid-structure LSTMs neural network

Based on M-LSTM, we propose a Grid-LSTM for prediction task, in which each LSTM unit receives one embedding vector that is an element in an matrix representation of two dimensional data. The bottom layer of Grid-LSTM takes sequence D_i and the upper layers take other sequences related to sequence D_i in spatial dimension. The overview of the Grid-LSTM is illustrated in Fig.2. Where, D_i, D_j and D_k have the same data structure as those in Fig.1.

To capture the long-distance interplays over grid structure, We propose an LSTM unit that is perhaps the most similar to S-LSTM in [7]. The proposed LSTM unit is described in Fig.3. Through merging the gated cell vectors of the input node, the proposed LSTM unit can reflect multiple direct or indirect input cells.

More specifically, the transition functions of the proposed LSTM unit are described as in Equ.7-13.

$$i_t = \sigma(W_i \cdot [h1_{t-1};\ h2_{t-1};\ x_t] + b_i) \tag{7}$$
$$f1_t = \sigma(W_f \cdot [h1_{t-1};\ h2_{t-1};\ x_t] + b_f) \tag{8}$$
$$f2_t = \sigma(W_f \cdot [h1_{t-1};\ h2_{t-1};\ x_t] + b_f) \tag{9}$$
$$g_t = tanh(W_c \cdot [h1_{t-1};\ h2_{t-1};\ x_t] + b_c) \tag{10}$$
$$c_t = f1_t \odot c1_{t-1} + f2_t \odot c2_{t-1} + i_t \odot g_t \tag{11}$$
$$o_t = \sigma(W_o \cdot [h1_{t-1};\ h2_{t-1};\ x_t] + b_o) \tag{12}$$
$$h_t = o_t \odot tanh(c_t) \tag{13}$$

$$\hat{y} = W_s \cdot h_n + b_s \tag{14}$$

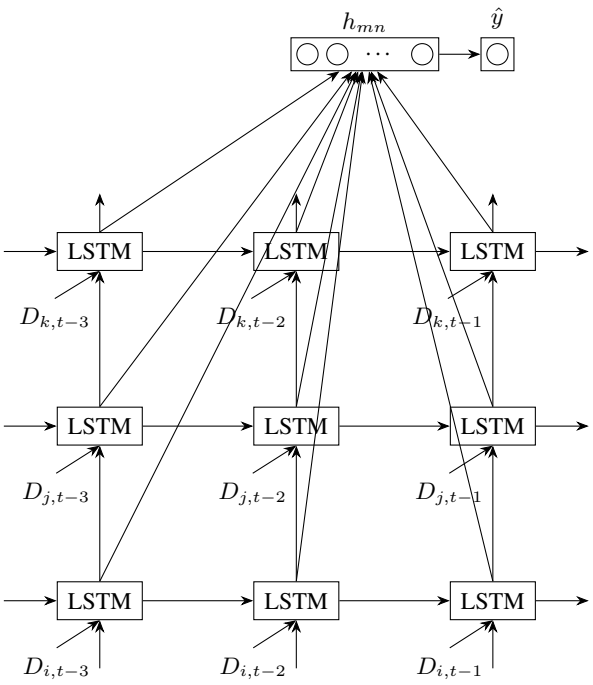


Fig. 2. Architecture of the Grid-LSTM. The data which indexes are in $\{t-3, t-2, t-1, \dots\}$ represent the temporal dimension input sequence, which length is the number of time step. The data which indexes are in $\{i, j, k, \dots\}$ represent the spatial dimensional input sequence, which length is the number of spatial dimensional data.

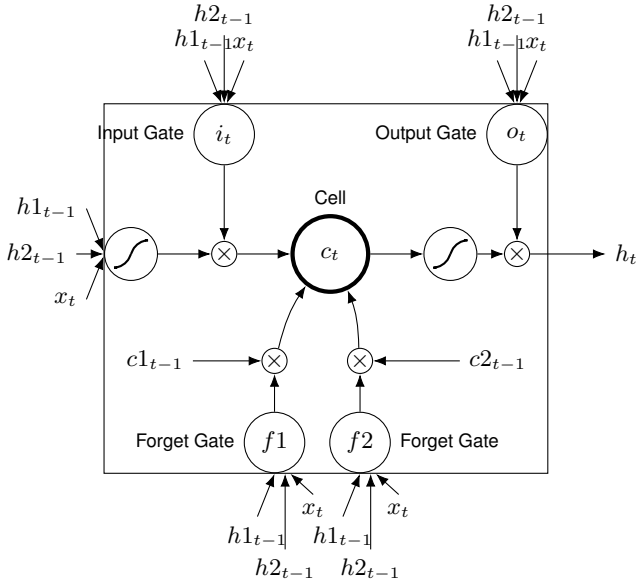


Fig. 3. Architecture of the proposed memory block. Where, the proposed LSTM unit consists of one input gate, one output gate and some forget gates. The number of forget gates is equal to the number of input nodes in LSTM unit. There are two input node in LSTM unit in Fig.2, so we have two forget gates. One gate is the left node. The other is the below node. $W_i,\ W_{f1},\ W_{f2},\ W_o,\ W_c \in R^{d\times 2d}$ are the weighted matrices of the proposed LSTM unit, and $b_i,\ b_{f1},\ b_{f2},\ b_o,\ b_c \in R^d$ are the biases of the proposed LSTM unit. These parameters are shared by Grid-LSTM, and will be learned during end-to-end model training. The left forget gate f_t^1 and the below forget gate f_t^2 can be controlled independently to allow the pass-through of information from input nodes.

Where, σ is an sigmoid function, and $\odot$ denotes the element-wise multiplication. x_t denotes an input sequences that is fed into the proposed LSTM unit. $h1_{t-1}$ denotes an output vector from the node at left. $h2_{t-1}$ denotes an output vector from the node at below. $c1_{t-1}$ denotes the cell state of the node at left. $c2_{t-1}$ is the cell state of the node at below. Predictor $\hat{y}$ is a single $n \times 1$ neural network layer.

C. Models training

The proposed models have been effectively trained end-to-end by the error back-propagation algorithm, which has been widely applied in neural networks training. The training is a process of determining the models parameters to minimize function loss value. The parameter set includes weight parameters, bias parameters, and hyper-parameters. Adjusting the weight and bias parameters of the models relys on a training set. Choosing the hyper parameters relys on a validation set. Evaluating the prediction performance of the models relys on a test set.

In our work, the loss function is the mean squared error of the observed value and the predicted value of the models. The loss function is desecribed in Equ.15 and Equ.16. Where y denotes the observed value , $\hat{y}$ denotes the predicted value, J denotes the mean squared error between y and $\hat{y}$, n denotes the total number of the training set. The mini-batch gradient descent algorithm and AdaGrad algorithm [25] were used for training. AdaGrad has the ability to adaptively regulate the learning rate of neural network. AdaGrad's main weakness is that the accumulating of history gradients continues to grow as the denominator during training [25]. The growth causes the learning rate to shrink and eventually become infinitesimally small, at which point the algorithm is no longer able to acquire additional knowledge.

$$loss = \underset{\Theta}{argmin}\, J(y, \hat{y}) \tag{15}$$

$$J(y, \hat{y}) = \sum^{n} (y - \hat{y})^2 \tag{16}$$

It is the main benefit of AdaGrad that eliminating the need to manually tune the learning rate. Based on the accumulating of history gradients, AdaGrad can perform much larger updates for infrequent parameters compared to frequent parameters. For this reason, it is well suited for dealing with sparse data. Dean et al. used AdaGrad to optimize the parameters of large-scale neural networks at Google, which learned to recognize cats in Youtube videos. Dean et al. found that AdaGrad greatly improved the robustness of stochastic gradient descent method [26]. Moreover, Pennington et al., used AdaGrad to training GloVe word embeddings, where performing larger updates for infrequent word embeddings and smaller updates for frequent word embeddings [27]. The AdaGrad equations are described

as Equ.17-19.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \varepsilon}} \cdot g_{t,i} \tag{17}$$

$$g_{t,i} = \nabla_\theta J(\theta_i) \tag{18}$$

$$G_{t,ii} = \sum^{t} g_{t,i}^2 \tag{19}$$

At time step t, the optimization of parameter $\theta_{t+1,i}$ is performed according to equation (17). Where, most implementations of learning rate η use a default value of 0.01. Based on the gradient value at the previous step, AdaGrad modifies learning rate η. $G_t \in R^{d\times d}$ is a diagonal matrix. Each element on diagonal line $G_{t,ii}$ is the square sum over history gradient value θ_i. ε denotes the smoothing term that avoids division by zero (usually on the order of $1e-8$).

The parameters of M-LSTM are $\{W_f, b_f, W_i, b_i, W_o, b_o, W_c, b_c, W_s, b_s\}$.
The parameters of Grid-LSTM are $\{W1_f, b1_f, W2_f, b2_f, W_i, b_i, W_o, b_o, W_c, b_c, W_s, b_s\}$.
Compared to the parameters of M-LSTM, $\{W_f, b_f\}$ are added into the parameters of Grid-LSTM because Grid-LSTM has one more forget gate than M-LSTM. The proposed models and baseline methods were trained on a 16-G GPU computer with CUDA [28]. CNMeM is enabled with an initial size of 80.0% of the memory. Python and Pytorch are used to implement the proposed models and baseline methods.

IV. Experiment for travel time prediction

We adopt three performance indices to evaluate the prediction accuracies of the proposed models: mean absolute error (MAE), mean absolute percentage errors (MAPE), and root mean square error (RMSE). The performance indices are formulated in Equ. 20-Equ.22.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}| \tag{20}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i - \hat{y}|}{y_i} \tag{21}$$

$$RMSE = [\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y})^2]^{\frac{1}{2}} \tag{22}$$

where n denotes the length of the given test set, y denotes the observed value, and $\hat{y}$ denotes the predicted value that is the output of the model.

A. Dataset

We perform some comparative experiments on the dataset that was provided and managed by Highways England [29]. Highways England operates, maintains and improves England's motorways and major A roads. The raw dataset consists of average journey time, average speed and average of the observed flow attributes value for the link, time period, day type and etc. The selection of the attributes for our experiments are listed in Table I.

TABLE I
A Selection of Attributes from Highways England's Data for Our Experiments.

LinkRef	Date	TimePeriod (0–95)	AverageJT
AL100	2015-03-01 00:00:00	0	138.26
...	...	...	...

LinkRef denotes the unique alphanumeric link id that represents the road from a junction to other junction on the road network. Date denotes the date of travel. TimePeriod denotes the 15-minute intervals in one day that refers to 0-95, where 0 indicates 00:00 to 00:15. AverageJT denotes the average journey time (in seconds) of vehicles entering the link within a given 15-minute time period to pass through the link.

B. Task definition

In our experiments, we select a dataset from Mar 1, 2015 to Mar 31, 2015. The dataset are divided into three parts: dataset from Mar 1 to Mar 28 for training (approximately 90.4%), dataset from Mar 29 to Mar 30 for validation (approximately 6.4%), dataset from Mar 31 for testing (approximately 3.2%).

Input representation: The input sequences are constructed from the dataset as Equ.23-Equ.24.

$$L: \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdots \\ r_{2975} \\ r_{2976} \end{bmatrix} \xrightarrow{f} X: \left[\begin{array}{cccc:c} r_1 & r_2 & \cdots & r_7 & y_1 \\ r_2 & r_3 & \cdots & r_8 & y_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & r_{2975} & y_{2969} \end{array}\right] \tag{23}$$

$$y_i = r_{i+7}.AverageJT \tag{24}$$

Where matrix l denotes the observed data of one link, and matrix X denotes the input sequence set of the models. Matrix l has $n = 96\times 31$ rows because the dataset consists of 31 days and there are 96 timeperiods per day. Suppose that the number of the horizontal layer of the proposed models is d, matrix X has $n-d$ rows. Each row of X denotes one input sequence for the proposed models. $x(r_1, r_2, \ldots, r_7)$ is the embedding vector. r concatenates the Date, TimePeriod and AverageJT attribute values. In our work, y denotes the observed values that are the AverageJT values on next time lag.

In the experiments, we predicted the AverageJT on the link AL282. The two dimensional data contains the observed data from link AL286, AL292, AL2274, and AL282. The road network that contains the link AL282 is described in Fig. 4.

C. Training details

Normalization: In our work, normalization is a very good practice method for preprocessing the input data because the gates of the LSTM unit have sigmoid or tangent activation functions. The input vectors are rescaled to the range of 0 to 1 by the normalization method.

Default setting: Regrading the structure of the proposed models, we need to determine the depth of the vertical

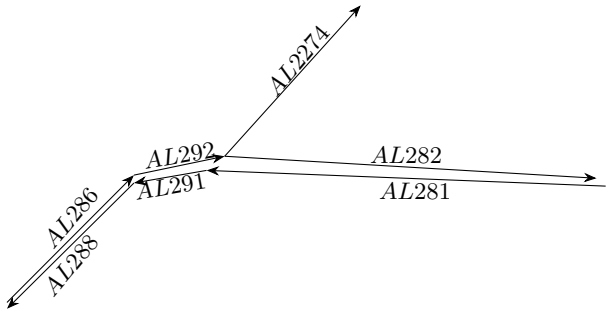


Fig. 4. The road network contains the link AL282.

layers and horizontal layers. In our experiments, the depth of vertical layers equals the number of spatial dimensional data. The depth of horizontal layers equals the length of an input sequence for every horizontal layers that is $r = 7$. The dimension of the input vector x is 3. Therefore, the dimension of the input sequence is $len \times 3$. The dimension of the output vector and the hidden unit is 4.

The hyper parameters of the proposed models were obtained based on the validation set. L1 regularization was performed and L2 regularization was not performed. The optimal learning rate of SGD for W_s parameter is 0.1. The optimal learning rate of AdaGrad for the other parameters is 0.01.

D. Experiments and results

Baseline methods: We compared the proposed models with several baseline methods for travel time prediction on the link AL282. The comparison methods include Elman NN, linear regression, and SAEs.

The linear regression method that is used in our experiments arose of Rice and Zwet [14]. In our experiments, the learn rate of the parameters is set to 0.05, Adagrad is used, and the time lag δ is set to 0.

The Elman NN model that is used in our experiments is based on the work of Elman [30]. In our experiments, The topology of the Elman NN is as follows: the depth of input layers is 7, the depth of hidden layers is 7, the depth of output layers is 7. The recurrent connections of the Elman NN is fixed at 1.0 and is not subject to adjustment that is formulated as $x_c(k) = x(k-1)$. The context units are initially set to 0.5.

Regarding the structure of a SAE network, we need to determine the size of the input layer, the number of hidden layers, and the number of hidden units in each hidden layer. The SAEs model in our experiments is based on the work of Yisheng et. al [18]. Based on the work of Yisheng et. al, the hidden layer size is set to 3. The dimension of the input vector is $mr = 28$ and the number of hidden units is 28 because the spatial dimension of the two dimensional data is $m = 4$ and the depth of horizontal layers is $r = 7$. The dimension of the output vector is 1 because the predicted AverageJT is only for one link.

Results: The comparative experimental results of the proposed models are listed in Table II. The baseline methods are performed 10 times on AL282 data. The proposed model

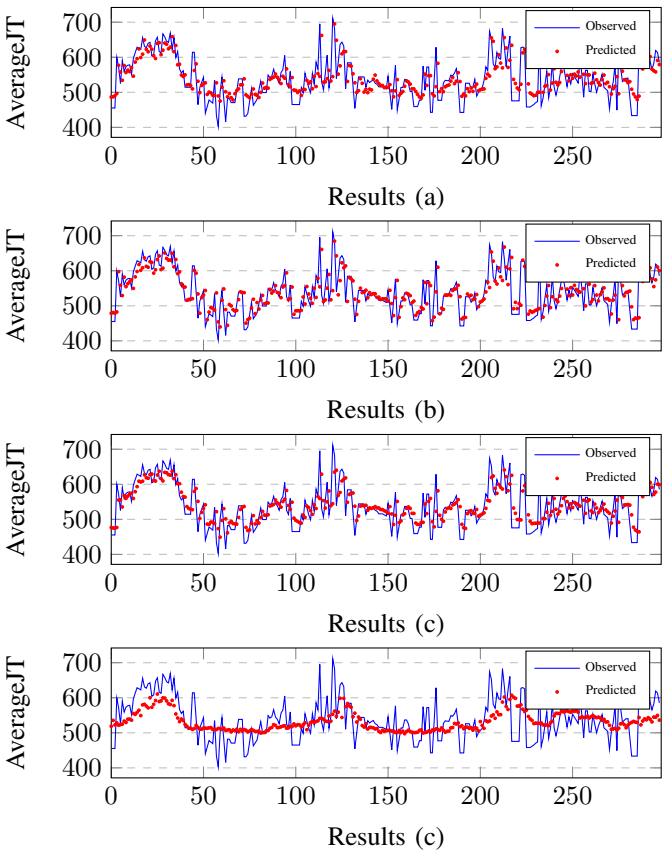


Fig. 5. Three travel patterns of the proposed models and the observed values on link AL282 road. (a) The experimental results of MM-LSTM and the observed values. (b)The experimental results of Grid-LSTM with S-LSTM memory block the observed values. (c)The experimental results of Grid-LSTM with the proposed memory block the observed values.

are performed 10 times on the two dimensional data, which consist of the data from the link AL286, AL292, AL2274, and AL282. We can see from Table II that the proposed models achieve better prediction accuracy than the baseline methods.

The proposed models outperform the baseline methods because it makes good use of the spatial dimensional data and captures the spatial dimensional relationships on two dimensional data. The proposed models with the proposed memory block outperforms all other methods in terms of prediction accuracy.

TABLE II
THE EXPERIMENTAL RESULTS COMPARING THE PROPOSED MODEL WITH THE BASELINE METHODS.

Models	**15-min**		
	RMSE	***MAE***	***MAPE***
Elman NN	55.24	6.65	0.0826
Linear Regression	57.10	6.75	0.0873
SAEs	48.46	5.98	0.0675
MM-LSTM	47.38	5.92	0.0661
Grid-LSTM with S-LSTM memory block	51.59	6.15	0.0713
Grid-LSTM with the proposed memory block	**47.48**	**5.78**	**0.0627**

E. Case study

The predicted values of the proposed models and the observed AverageJT values for comparison are presented in Fig.5. As shown in Fig. 5, the predicted values of the proposed models and the observed AverageJT values exhibit similar travel patterns.

The input data for the proposed models are two dimensional data. The temporal dimensional features from each link and the spatial dimensional features from the link AL286, AL292, AL2274, and AL282 can be extracted by the proposed models. Specifically, four input sequences from the above links are fed to the proposed models at time t, and the input sequence format of each link is $\{x_{t-1}, x_{t-2}, ..., x_{t-d}\}$. Therefore, the proposed models can extract the features in the spatiotemporal dimensions from the two dimensional data.

The results in Table II shows that Grid-LSTM with the proposed memory block outperforms MM-LSTM because Grid-LSTM make good use of the spatiotemporal structure of the two dimensional data. Regarding the grid structure, we compare the S-LSTM memory block and the proposed memory block of the Grid-LSTM. The results in Table 3 show that the proposed memory block outperform the S-LSTM because the gate functions of the proposed memory block do not take the long term output of the cell state into their input, and only take the short term output of the cell state into their input. The gate function of S-LSTM take the long and short term output of cell state into their input.

It is important for the proposed model to perform well on prediction tasks that achieving a high performance on localization as it involves identifying both the spatial and temporal features from two-dimensional data accurately. We show four group images from Grid-LSTM in Fig. 6-Fig.13, each from the features weights for the test datasets and their prediction activation maps below them. Each small square in these figures corresponds to an LSTM in Fig.2. The figures were feched by registering a forward hook on the code of Grid-LSTM. The hook was called every time after function 'forward' has computed an output. Columns correspond to the LSTMs of temporal dimension. Rows correspond to the LSTMs of spatial dimension. We observe that the discriminative regions are often highlighted and localized in the different areas of Fig. 6-Fig.13 in both temporal and spatial dimensions. This demonstrates the generalization ability of our proposed models.

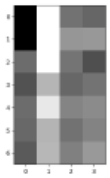


Fig. 6. The feature of the 7nd test sample

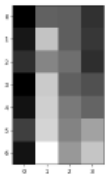


Fig. 7. The feature of the 57nd test sample

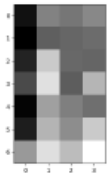


Fig. 8. The feature of the 177nd test sample

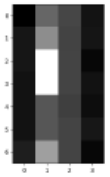


Fig. 9. The feature of the 229nd test sample

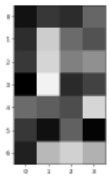


Fig. 10. The weights of the features of 7nd test sample

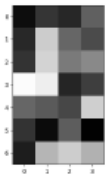


Fig. 11. The weights of the features of 57nd test sample

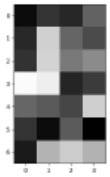


Fig. 12. The weights of the features of 177nd test sample

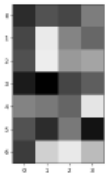


Fig. 13. The weights of the features of 229nd test sample

V. Experiment for crime prediction

We perform experiments on the crime data to validate the proposed models. The crime data is about the crime and policing in England, Wales and Northern Ireland. The crime data are described as Table III.

A. Dataset

The crime data that are used in the experiments is an open data and can be accessed from the web site https://data.police.uk/. The attributes/features of the crime data are described in Table III. In the crime data, the "Reported by" and "Falls within" attributes have the value "City of London Police". The crime data are originally reported by the 43 geographic police forces and go through a rigorous quality control process before being published by the Home Office. The crime data are generated in the past three years and are updated on a monthly basis. The crime data are published in monthly files, such as "2015-01-city-of-london-street.cvs". The crime data that contain street-level crime, outcome, stop and search data were download in clear and simple CSV format to validate the proposed models. The CSV files provide the street-level crime, outcome, stop and search information that are broken down by police force and 2011 lower layer super output area (LSOA). We focused on the crime data that falled within London from September 2014 to August 2017 (36 months).

B. Task definition

In our work, the experiments were performed to predict the crime frequency at next month on one LSOA. In the data

TABLE III
A DESCRIPTION OF THE CRIME DATA'S FEATURES.

Field	Meaning
Crime ID	Id of Crime.
Month	Date of the crime in the format yyyy-mm.
Reported by	The force that provided the data about the crime.
Falls within	At present, also the force that provided the data about the crime. This is currently being looked into and is likely to change in the near future.
Longitude and Latitude	The anonymised coordinates of the crime. See Location Anonymisation for more information.
LSOA code and LSOA name	References to the Lower Layer Super Output Area that the anonymised point falls into, according to the LSOA boundaries provided by the Office for National Statistics.
Crime type	One of the crime types listed in the Police.UK FAQ.
Last outcome category	A reference to whichever of the outcomes associated with the crime occurred most recently. For example, this crime's 'Last outcome category' would be 'Formal action is not in the public interest'.
Context	A field provided for forces to provide additional human-readable data about individual crimes. Currently, for newly added CSVs, this is always empty.

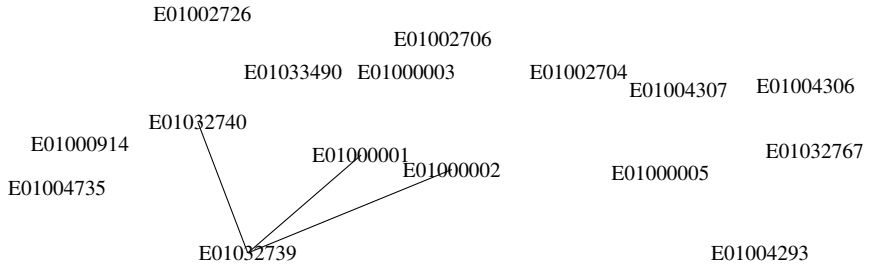


Fig. 14. The map is a spatial distribution description of crime data, where the nodes represent LSOA codes, and the location of the LSOA codes on the map represents the location of the crime data in the city.

TABLE IV
THE SUMMARY OF THE CRIME DATA THAT USED IN OUR EXPERIMENTS.

LSOA code	LSOA name	Instance Number
E01032739	City of London 001F	11888
E01000002	City of London 001B	811
E01000001	City of London 001A	632
E01032740	City of London 001G	2803

understanding step [31], The monthly files were integrateed into one file that was used in our experiments. In the data preparation phase, four features were selected from the integrated file, which are Month, LSOA code, LSOA name, and Frequency as the input for our experiments. We do not selected that Crime ID, Reported by, Longitude, Latitude, Location, Crime type, Last outcome category and Context Consequently fields from the integrated file because that are no irrelevant to the crime frequency based on common sense.

In our experiments, we predicted the crime frequency of LSOA E01032739 at next month. Based on the Fig.14, the crime data of E01000001, E01000002, LSOA E01032739 and E01032740 were selected as the two dimensional data because LSOA E01000001, E01000002 and E01032740 are closer to E01032739 in the LSOA spatio distance. The summary of the two dimensional data is listed in Table IV.

C. Experiments and results

The proposed models and the baseline models are performed ten times on the two dimensional crime data. The training details is the same as section IV-C, except for using stochastic gradient descent instead of mini-batch gradient descent. The indices of MAE, MAPE and RMSE were used on comparing the proposed models and the baseline methods for the crime prediction. These indexes are formulated in Equ. 20-Equ.22.

TABLE V
THE EXPERIMENTAL RESULTS COMPARING THE PROPOSED MODEL WITH THE BASELINE METHODS.

Models	15-min		
	RMSE	*MAE*	*MAPE*
Elman NN	47.54	6.18	0.1347
Linear Regression	68.10	7.51	0.1941
SAEs	51.15	6.58	0.1564
MM-LSTM	**42.78**	**5.69**	**0.1089**
Grid-LSTM with S-LSTM memory block	51.52	6.53	0.1526
Grid-LSTM with the proposed memory block	52.31	6.63	0.1530

The experimental results are listed in Table V. The experimental results shows that MM-LSTM achieves the best performance. MM-LSTM outperforms Grid-LSTM with the proposed memory block and Grid-LSTM with S-LSTM memory block in the indices of MAE, MAPE and RMSE. Therefore, MM-LSTM used the spatial dimensional data and captured the spatial dimensional relationships on two dimensional data, and then achieves the best performance comparing the other methods. The structure of Grid-LSTM is more complex than MM-LSTM, may requiring more data to achieve higher accuracy.

VI. Conclusions and future work

In this paper, we stacked LSTMs to extract features in two-dimensional data for prediction tasks on travel time and crime. The stacking methods of MM-LSTM and Grid-LSTM with memory blocks were proposed. The key strategies of these models is to use two-dimensional data as input and use the stacking methods to extract the spatial-temporal features from two-dimensional data.

To validate the proposed models, we performed our experiments on a traffic data and a crime data. Based on the traffic data, the experimental results show that Grid-LSTM with the proposed memory block achieves higher accuracy compared to all other methods for travel time prediction. The Grid-LSTM effectively extracts features, highlighting and localizing discriminative regions within the extracted features in both temporal and spatial dimensions. Based on the cime data, the experimental results show that MM-LSTM outperforms all other methods for crime prediction. Therefore , the proposed models have a competitive advantage for prediction tasks based on two-dimensional data.

As future work, we will investigate how to improve the proposed memory block to better extract the spatio-temporal features from two-dimensional data. An interesting potential direction is to attempt to apply convolutional neural network to enhance the extract ablity of the proposed memory block, which would perhaps allow us to utilize techniques from some relevant literatures.

Acknowledgment

The authors extend their appreciation to Mr. Congpeng Li for generously providing the computing resources for model training. The work was funded by the Scientific Research Project of Beijing Information Technology College (number JB27132). The code is available at https://github.com/puma-running/StackingLSTMs/.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735.1780, 1997.

[2] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C*, 2015.

[3] Alex Graves, Abdelrahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. 38(2003):6645–6649, 2013.

[4] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. Neural sentiment classification with user and product attention. In *Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659, 2016.

[5] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. 4:3104–3112, 2014.

[6] Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. Linguistically regularized lstm for sentiment classification. In *Meeting of the Association for Computational Linguistics*, pages 1679–1689, 2017.

[7] Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. Long short-term memory over tree structures. In *ICML*, 2015.

[8] Yoshua Bengioy, Patrice Simardy, and Paolo Frasconiz. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions*, 5(2):157–166, 1994.

[9] Wojciech Zaremba and Ilya Sutskever. Learning to execute. In *ICLR*, 2015.

[10] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 2002.

[11] Adi Alhudhaif and Kemal Polat. Spatio-temporal characterisation and compensation method based on cnn and lstm for residential travel data. *PeerJ Computer science*, 10:e2035, 2024.

[12] Improving recognition accuracy for facial expressions using scattering wavelet. 3, Mar. 2024.

[13] Alex Graves, Navdeep Jaitly, and Abdel Rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding*, pages 273–278, 2014.

[14] John Rice and Erik van Zwe. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 2004.

[15] D. Nikovski, N. Nishiuma, Y. Goto, and H. Kumazawa. Univariate short-term prediction of road travel times. *IEEE Intelligence Transportation Systems Conference*, 2005.

[16] J. W. C. van Lint. Online learning solutions for freeway travel time prediction. *ieee transactions on intelligent transportation systems*, 2008.

[17] J.W.C. van Lint, S.P. Hoogendoorn, and H.J. van Zuylen. Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks. *Transportation Research Record Journal of the Transportation Research Board*, January 2002.

[18] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei Yue Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[19] Farshid Afshar Mojtaba Mohammadzadeh, Abdoul-Ahad Choupani. The short-term prediction of daily traffic volume for rural roads using shallow and deep learning networks: Ann and lstm. *The Journal of Supercomputing*, 79:17475–17494, 2023.

[20] Zhe Zheng, Bo Zou, Wenbin Wei, and Wen Tian. A data-light and trajectory-based machine learning approach for the online prediction of flight time of arrival. *Aerospace*, 10(8), 2023.

[21] Brandon C. Welsh and David P. Farrington. Science, politics, and crime prevention: Toward a new crime policy. *Journal of Criminal Justice*, 40(2):128–133, 2012.

[22] Ginger Saltos and Mihaela Cocea. An exploration of crime prediction using data mining on open data. *International Journal of Information Technology and Decision Making*, 2017.

[23] Ata Jahangir Moshayedi, Seyed Taha Mousavi Nasab, Zeashan Hameed Khan, and Amir Sohail Khan. *Meta-heuristic Algorithms as an Optimizer: Prospects and Challenges (Part I)*, pages 131–154. Springer Nature Singapore, Singapore, 2025.

[24] Ata Jahangir Moshayedi, Seyed Taha Mousavi Nasab, Zeashan Hameed Khan, and Amir Sohail Khan. *Meta-heuristic Algorithms as an Optimizer: Prospects and Challenges (Part II)*, pages 155–180. Springer Nature Singapore, Singapore, 2025.

[25] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, (12):2121–2159, 2011.

[26] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc' Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. Large scale distributed deep networks. *In Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[27] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[28] Vincent Vanhoucke, Andrew Senior, and Mark Z. Mao. Improving the speed of neural networks on cpus. In *In Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*, 2011.

[29] Highways England. Highways england network journey time and traffic flow data. *Retrieved October*, 3:2018, 2018.

[30] J. Elman. Finding structure in time. *Cognitive Science 14, pp. 179-211*, 1990.

[31] Ginger Saltos and Mihaela Cocea. An exploration of crime prediction using data mining on open data. *International Journal of Information Technology and Decision Making*, 2017.

Microarchitectural Analysis of Pre-Processing Stage in Machine Learning Workloads

Muge Zhang*, Dae Yeol Lee†, Vasudevan Janarthanan*, Jeeho Ryoo*
* Fairleigh Dickinson University, Vancouver, BC, Canada
m.zhang1@student.fdu.edu, v_janart@fdu.edu, j.ryoo@fdu.edu
† Dolby Laboratories Inc., Sunnyvale, CA, USA
DaeYeol.Lee@dolby.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899564

***Abstract*—As Machine Learning (ML) has become integral for various applications, ML workloads are now important considerations for deployment across diverse use cases, ranging from data centers to edge devices. ML encompasses diverse application fields, including vision, audio, text, and multimodal areas, each involving specific raw data formats that often needs pre-processing to become more interpretable for the models and to ensure a more balanced and standardized data distribution. This stage can also include data augmentation to improve model robustness and performance. Therefore, most ML workloads incorporate a stage, commonly referred to as pre-processing, prior to processing the actual data in complex ML model. As the amount of data size increases at a drastic rate, the pre-processing stage now requires closer attention given its significant computation time. In this paper, we conduct an in-depth micro-architectural analysis of the ML pipeline's pre-processing stage to uncover bottlenecks and utilize a bottom-up approach to deliver valuable insights by identifying code hotspots.**

I. Introduction

ML workloads have been growing its popularity and has been an integral part of daily life beyond scientific community. Recent mobile devices that people use daily now have processors with neural engines. As the amount of data that is processed by the ML workloads has surged, both industry and academia acknowledge that technology scaling will not be able to match the input data growth rate, prompting a search for alternative algorithms.

ML algorithms typically receive raw inputs like visual, text, or audio data, which require pre-processing before entering the ML pipeline. In the past, this pre-processing has not been considered significant since a majority of execution time was spent in the ML pipeline where complex ML algorithms, requiring abundant compute resources, take place. However, as data volumes grow, pre-processing has begun to occupy a higher fraction of the execution time.

In this paper, we identify microarchitectural bottlenecks using hardware performance counters and system tools. We make the following contributions:

- We conducted a detailed and comprehensive microarchitectural exploration of the pre-processing stages in representative ML workloads, showing inefficiencies overlooked at higher levels of abstraction.
- We observed memory consumption and bandwidth patterns during model execution across pre-processing and inference stages.
- We employed a bottom-up approach from microarchitectural metrics to application source code to find optimization opportunities.

II. Background

Common pre-processing tasks include data cleansing, normalization, feature extraction, data augmentation. High-quality pre-processing ensures input data is optimized for the ML pipeline, enhancing model performance. For inference, pre-processing ensures input data is formatted similarly to training data, helping models deliver accurate predictions. The focus of this paper is specifically on the pre-processing for inference stage. This stage is essential for transforming raw data into a suitable format for model inference. The pre-processing stage during training is aimed at enhancing data generalization and model performance, involving more complex operations to manipulate data, while during inference, preprocessing is more streamlined and focus only on necessary steps to make the data into a usable format. While pre-processing can generally refer to both training and inference pre-processing stages, the scope of this paper is limited to the inference stage.

Note that the focus of our work is on the analysis of the pre-processing stage to identify inefficiencies in the **inference** stage for various workloads. Therefore, we do not consider the data read performances of the ground-truth labels, which are primarily used in the training and validation processes to supervise the network weights. Consequently, the dataloader we focus on differs from the training stage dataloader, whose main purpose is to prepare the data for model learning.

III. Related Work

Benchmarks: MLPerf Inference [1] provides a benchmark suite focused on inference workloads that covers a wide range of ML scenarios. It evaluates performance of pre-processing and pipeline together using hardware-based metrics. The XR-Bench is a suite of benchmarks for extended reality platforms that includes features absent in traditional workloads [2]; nonetheless, focusing only on the inference stage.

Domain Specific Applications: EyeCoD [3] is a framework for accelerating eye tracking models. It identifies pre-processing as a performance bottleneck in XR devices. Similarly, CoVA [4] observes that the decoding process in pre-processing of specifically video analytics workloads as a bot-

979-8-3315-2931-4/24/$31.00 © 2024 IEEE

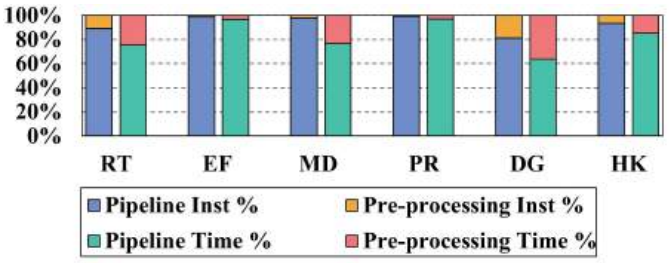


Fig. 1: Instruction count along with execution time spent in pre-processing vs pipeline

tleneck While formerly mentioned work exclusively evaluates domain specific applications, we examine a much diverse set of general ML workloads.

GPU Offloading: To mitigate pre-processing bottlenecks, some approaches offload tasks to the GPU. FusionFlow [5] improves efficiency by offloading pre-processing from the CPU, though its focus remains on training rather than inference. NVIDIA DALI [6] provides GPU acceleration for data loading and pre-processing, supporting both training and inference. While effective, DALI doesn't deeply analyze pre-processing's performance impact on inference. GPU offloading can ease CPU bottlenecks but may cause resource contention between data transformation and model computation, leaving the decision to the user on CPU vs. GPU execution.

IV. Motivation

In this section, we present the significance of pre-processing in various ML workloads. The yellow-blue bars shown in Figure 1 depict the ratio of the computation time spent in the pre-processing stage and the rest of the pipeline. On average, 17.54% of the runtime is spent on pre-processing. In D2GO workload, up to 36.15% of execution time is spent on pre-processing. The sub-optimal performance of pre-processing becomes more prominent when compared with the pink-green bars on the right, also in Figure 1, which show the distribution of instruction counts between the pre-processing and the inference pipeline stage for the same workloads. For D2GO, the pre-processing instruction count accounts for 18.54%, yet these instructions are responsible for 36.15% of the execution time. On average, pre-processing takes only 6.58% of total instructions in the entire inference pipeline, yet it is responsible for 17.54% of the execution time. This significant discrepancy between the execution time and instruction percentages signals potential bottlenecks in the pre-processing stage.

Cycles per instruction (CPI) is a metric that shows how many processes cycles are consumed per instruction on average. The relative CPI is computed by normalizing the pre-processing CPI to the CPI of the entire execution. If this relative CPI is low, it signifies that the pre-processing is executing instructions faster than the entire execution on average. However, in our case, for all models, the CPI in pre-processing is much higher. Figure 2 shows the relative CPI of all our models. The average relative CPI across all our models is 4.35, which is a key observation of this work where

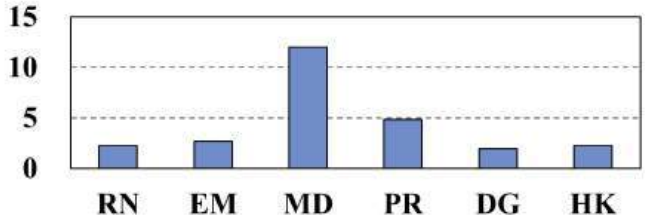


Fig. 2: Relative CPI (Pipeline as baseline)

the pre-processing is not processing instructions as efficiently as other stages in the execution. In case of MiDaS, it is taking 12 times longer to execute a single instruction than the rest of the pipeline. This provides significant performance-enhancing opportunities. With such observations, this paper aims to utilize micro-architectural metrics to investigate sources of the bottlenecks in the pre-processing stage and offer performance tuning guides.

V. Methodology

All experiments were conducted on an i5-11400H Processor with 12 logical cores (2.70–4.50 GHz) and a shared 12 MB last-level cache.

A. Models

We used various models relevant to industrial use cases. Specific tasks performed in pre-processing are described here.

RITnet (RN) [7]: An eye segmentation model dividing eye images into regions like the pupil and iris, crucial for eye-tracking in interactive systems. Pre-processing enhances grayscale eye images using gamma correction and CLAHE.

MiDaS (MD) [8]: A depth estimation model predicting object distances to generate 3D scene representations, used in areas such as robotics. Pre-processing involves resizing, normalization, and data augmentation of RGB images.

D2GO (DG) [9]: An object detection model identifying and localizing objects in images, applied in areas like autonomous driving. Pre-processing applies data augmentation to RGB images, converting them into tensors.

PlaneRCNN (PR) [10]: A plane detection model identifying planar surfaces, fundamental for applications including multimodal language models. Pre-processing generates depth and segmentation maps from RGB images and camera parameters.

Emformer (EF) [11]: A speech recognition model converting speech to text, essential for applications like live translation. Pre-processing transforms audio signals into spectrograms using feature extraction.

Honk (HK) [12]: A keyword detection model recognizing predefined words in audio, used in devices like virtual assistant. Pre-processing employs MFCCs to process audio inputs.

B. Datasets

In order to make inferences using these workloads, different datasets were selected for each benchmark.

OpenEDS [13]: Provides 12,759 annotated 640×400 grayscale images for RITnet.

COCO [14]: Used with D2GO for object detection tasks; we utilized 41,000 RGB test images.

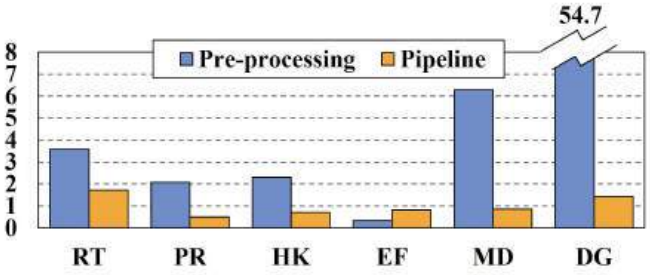


Fig. 3: MPKI for Pre-processing vs Pipeline

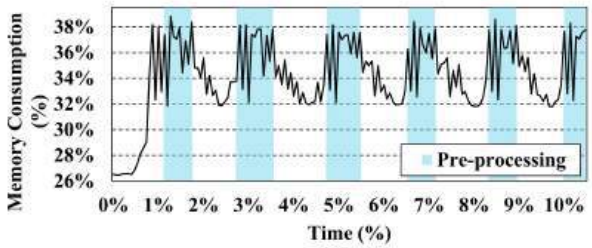


Fig. 4: Zoomed-in memory consumption (RITnet)

KITTI [15]: Employed by MiDaS and PlaneRCNN; 7,518 RGB test images for depth estimation and plane detection.
LibriSpeech [16]: A speech corpus for testing Emformer; we used the clean test set containing 5.4 hours of speech comprised of 3–4-second audio files.
Google Speech Commands [17]: Used with Honk; we utilized 4,890 test examples of single spoken words.

C. Tools

We utilized Intel® VTune™ Profiler in Microarchitecture Exploration mode to collect hardware performance data. Intel® Instrumentation and Tracing Technology APIs with ITT-python profiled pre-processing and pipeline stages separately. Cprofile identified hot functions in pre-processing code, while Windows Performance Monitor (PerfMon) recorded memory consumption during execution.

VI. Evaluation

In this section, we analyze ML pre-processing pipeline stages using micro-architectural metrics and take a bottom-up approach to identify performance improvement opportunities in key code segments, known as *hotspots*.

A. Memory Access Analysis

The Misses Per Kilo Instructions (MPKI) indicate memory intensity, with higher MPKI meaning slower main memory accesses. We use MPKI in LLC to measure number of missed memory requests in LLC, and thus, reach the main memory. Figure 3 compares MPKI of pre-processing and pipeline across six workloads. Higher MPKI means more main memory accesses that are slower than on-chip memory accesses. Lower MPKI improves performance. Except for Emformer, pre-processing consistently has higher MPKI than pipeline. For instance, D2GO's MPKI drops from 54.7 in pre-processing to 1.44 in pipeline, and MiDaS falls from 6.29 to 0.858. This difference suggests pre-processing has more intensive or irregular memory access patterns across almost all workloads.

These models rely on operations such as image resizing and loading, which involve intensive pixel data processing, including integer-to-floating-point conversion and adjacent pixel loading. Although this sequence of operations has high spatial locality, they exceed the on-chip cache capacities, resulting in high LLC MPKI. In pre-processing, intermediate copies of input data, required for scaling and transformation operations, further increases memory footprint because the complexity of the algorithms requires additional buffers and data structures to store computed values. We discuss the algorithmic details of these operations in Section VI-D.

In contrast, Emformer shows a low MPKI, with 0.34 during pre-processing and 0.817 during the pipeline, thanks to PyTorch's optimized tensor operations. They are built to enhance spatial locality through orchestrating memory hierarchies in order to maximize memory efficiency and thus to decrease cache misses. In Emformer, Pytorch reconfigures the tensor access metadata to minimize data copying and reallocation rather than physically rearranging the data. Unlike models that rely on multiple third-party packages like PIL and OpenCV for various data manipulations, Emformer primarily uses a spectrogram transform, which is sequential and regular, further reducing MPKI by optimizing locality [18].

B. Memory Consumption Analysis

The memory access analysis in Section VI-A reveals that one pre-processing bottleneck is related to main memory accesses. Therefore, we analyze memory usage, as memory footprint proves to be an issue in pre-processing. In all workloads except Honk, frequent small fluctuations in memory consumption occur due to the memory-intensive pre-processing, followed by the less demanding inference stage.

During pre-processing, data are read, transformed, and temporarily stored, resulting in peaks in memory consumption. In RITnet, PlaneRCNN, and D2GO, this stage processes batches of images or audio files, with multiple inputs being handled concurrently. Once pre-processed, the data are passed to the model for inference, where memory demand decreases, as this phase is more compute-intensive. This alternating pattern between pre-processing and inference leads to the characteristic memory usage oscillations noted across these workloads, aligning with the findings in Section VI-A. We focus on RITnet's memory behavior in Figure 4. This figure is zoomed-in for the first 10% of the entire execution, which reveals a consistent pattern: six peaks corresponding to six batches being pre-processed, followed by lower memory usage during inference. The pre-processing stages, highlighted in blue, show higher memory consumption due to frequent I/O operations, image decoding, data type conversions, and transformations, while inference stages exhibit reduced memory usage.

C. Bandwidth Analysis

Figure 5 shows bandwidth profiled during inference and pre-processing stages of the workloads. The y-axis denotes the

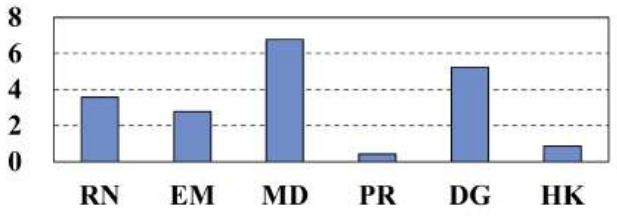


Fig. 5: Relative bandwidth (Pipeline as baseline)

bandwidth ratio between the pre-processing and the inference pipeline stage. Honk shows the highest ratio of 6.78, indicative of significantly greater bandwidth utilization during the pre-processing phase compared to pipeline. MiDaS, RITnet, and PlaneRCNN also show a high ratio of 6.24, 3.59, and 2.78, respectively, suggesting higher bandwidth demand during pre-processing. In contrast, Emformer and DGO have ratios lower than 1, implying that their pre-processing stages require less bandwidth compared to their inference stages.

For D2GO and Emformer that show a ration that is less than 1, this phenomenon can be primarily attributed to a technique called "channel expansion" in the inference stages of these models. This refers to the process where the number of channels in a neutral network increases when passing through certain layers of the network [19]. This technique can be used to improve the capacity of a model to represent more generalized data without increasing the spatial dimensions. It increases the size of the data being processed by doubling the number of channels in the input image tensor, leading to the memory allocation of a much larger tensor. This process not only increases memory usage but also doubles the bandwidth requirements, as more data must be handled during reading and writing operations.

Conversely, for RITnet, Honk, PlaneRCNN, and MiDaS, bandwidth usage is higher during pre-processing due to intensive data manipulation tasks, particularly image resizing, which is explained in Algorithm 1. Lines 1-6 calculate new dimensions for the resized image tensor. Line 7 allocates memory based on the calculated result. If a larger image is read, a larger amount of data exercises memory bandwidth. Line 8 calls the 'resizing_calc' function to perform the resizing operation of reading pixel values and writing transformed pixel values to the destination image. Upscaling or padding contribute each pixel to many pixels in the destination, thereby also increasing data transfers. The need to load, process, and store the resized images leads to increased bandwidth consumption when a large amount of data is transferred between storage and processing units. In addition, operations such as cropping, padding, and flipping, which are prevalent in pre-processing, also contribute to bandwidth.

D. Runtime Performance Analysis

In this section, we present a systematic identification of performance bottlenecks in the code segments using our previously discussed micro-architectural analysis.

In RITnet, the most time-consuming function is related to I/O, specifically built-in method io.open, which is responsible

Algorithm 1: resize Function

```
   Data: Source image src, Destination size dsize,
         Scaling factors fx, fy
   Result: Resized image dst
1  if dsize is not provided then
2  |  dsize.width = src.width * fx;
3  |  dsize.height = src.height * fy;
4  else if fx ≤ 0 or fy ≤ 0 then
5  |  fx = dsize.width / src.width;
6  |  fy = dsize.height / src.height;
7  dst = allocate_memory(dsize.width,
     dsize.height);
8  resizing_calc(src, dst, fx, fy);
9  return dst;
```

for accessing input data. During pre-processing, image data are read from the disk or the camera, which leads to high I/O access latency. In addition, image decoding and manipulation introduces latency. For example, method 'decode' of 'ImagingDecoder' objects involves decoding image files while operations like method 'tobytes' of 'numpy.ndarray' objects, LUT, and method 'apply' of 'cv2.CLAHE' objects operate on images. For RITnet, image loading functions take more execution time than actual image-manipulation operations commonly found in neural networks. This observation is in line with the profiling results presented earlier in Figure 5, in which a much higher pre-processing bandwidth is found.

For MiDaS, the functions `resize` and `transforms.py:205(__call__)` are the two hotspots in the entire workload. They handle image resizing and transformations. Again in Algorithm 1, the OpenCV resize function is shown. The `resize` function accesses the memory quite heavily and frequently, which contributes to the high overhead. Specifically, function `allocate_memory` (Line 7), allocates memory for the resulted image, which is attributed to constant memory consumption as shown in Section VI-B. Furthermore, resizing calculations in `resizing_calc` (Line 8) incur a high number of memory accesses. When processing larger images, the function becomes a even more pronounced hotspot.

The pre-processing pipeline in D2GO involves intensive memory accessing by image reading function (imread), data conversion (method 'tobytes' of 'numpy.ndarray' objects) and resizing, as analyzed above. This aligns with the much higher pre-processing MPKI data we have seen in Figure 3.

In the PlaneRCNN pre-processing pipeline, the most time-consuming functions are image loading function (imread), data type casting functions like method 'astype' of 'numpy.ndarray' objects, and resizing as in RITnet and MiDaS. Other pre-processing functions such as normalization are also observed, which adjusts the pixel values to ensure the format of test images are the same as the training data, in which every pixel value is read, processed, adjusted, and written back to memory. We have observed that the bandwidth for PlaneRCNN during

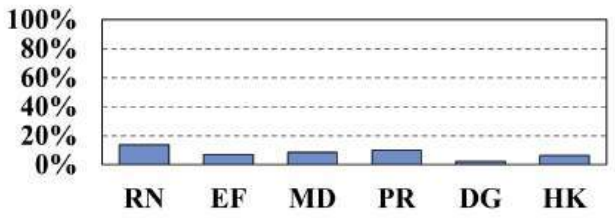


Fig. 6: Percentage of core bound instructions in pre-processing

pre-processing more than doubles the bandwidth during the pipeline, as shown in Figure 5.

In constrast, Emformer's top overhead attributing operation is spectrogram computation `functional.py:57` (`spectrogram`), which is used to convert audio signals into spectral format for subsequent processing. As a transformer model, Emformer involves operations such as multi-head attention, feed-forward networks, and layer normalization in the inference stage. Even though operations such as spectrogram computation cause notable performance overheads in pre-processing, their impact on performance is less than that of the inference operations which involve complex calculations. This agrees with the relative bandwidth discussion in Section VI-C.

Honk interacts with file systems and manipulates data. The most time-consuming operation is related to obtaining the final path name using {built-in method nt._getfinalpathname}. This method handles resolving symbolic links and returning the final path of a file, indicating high file I/O activities. This aligns with our bandwidth analysis in Section VI-C.

E. Core Bound Analysis

We conducted further research to determine if any workload exhibits a bottleneck in the compute functions. We collected the instruction mix in the pre-processing stage, and here we present the percentage of core-bound versus memory-bound instructions. Figure 6 shows that the average fraction of core-bound instructions during pre-processing is approximately 4.35. It should be noted that the core bound instruction is as low as 2.4% in D2GO and as high as 13.8% in RITnet. It is apparent that no workload shows a significant amount of core bound instructions. Since our workloads represent a diverse spectrum in various ML use cases, we can safely assume that pre-processing is predominantly memory bound.

VII. Conclusion

In this paper, we have presented bottleneck analysis in an often overlooked pre-processing stage of ML workloads. Our evaluations using micro-architectural metrics show that much of execution time spent in pre-processing is due to subpar memory performance such as high memory or I/O accesses. In addition, source code analysis found that certain functions are designed with an emphasis on enhancing computation or algorithms. As input data scales, the role of pre-processing in ML workloads will grow in importance. Hence, this work highlights key takeaways for future work in ML where input processing algorithms in pre-processing can be optimized by understanding micro-architectural features.

References

[1] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou *et al.*, "Mlperf inference benchmark," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2020, pp. 446–459.

[2] H. Kwon, K. Nair, J. Seo, J. Yik, D. Mohapatra, D. Zhan, J. Song, P. Capak, P. Zhang, P. Vajda *et al.*, "Xrbench: An extended reality (xr) machine learning benchmark suite for the metaverse," *Proceedings of Machine Learning and Systems*, vol. 5, 2023.

[3] H. You, C. Wan, Y. Zhao, Z. Yu, Y. Fu, J. Yuan, S. Wu, S. Zhang, Y. Zhang, C. Li *et al.*, "Eyecod: eye tracking system acceleration via flatcam-based algorithm & accelerator co-design," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022, pp. 610–622.

[4] J. Hwang, M. Kim, D. Kim, S. Nam, Y. Kim, D. Kim, H. Sharma, and J. Park, "{CoVA}: Exploiting {Compressed-Domain} analysis to accelerate video analytics," in *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, 2022, pp. 707–722.

[5] T. Kim, C. Park, M. Mukimbekov, H. Hong, M. Kim, Z. Jin, C. Kim, J.-Y. Shin, and M. Jeon, "Fusionflow: Accelerating data preprocessing for machine learning with cpu-gpu cooperation," *Proceedings of the VLDB Endowment*, vol. 17, no. 4, pp. 863–876, 2023.

[6] NVIDIA, "Nvidia dali," 2024. [Online]. Available: https://github.com/NVIDIA/DALI

[7] A. K. Chaudhary, R. Kothari, M. Acharya, S. Dangi, N. Nair, R. Bailey, C. Kanan, G. Diaz, and J. B. Pelz, "Ritnet: Real-time semantic segmentation of the eye for gaze tracking," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 3698–3702.

[8] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 3, pp. 1623–1637, 2020.

[9] Meta, "D2Go: A Deployable Framework for Object Detection based on Detectron2," GitHub repository, 2022. [Online]. Available: https://github.com/facebookresearch/d2go

[10] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "Planercnn: 3d plane detection and reconstruction from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4450–4459.

[11] Y. Shi, Y. Wang, C. Wu, C. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," *arXiv preprint arXiv:2010.10759*, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2010.10759

[12] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5484–5488.

[13] S. J. Garbin, Y. Shen, I. Schuetz, R. Cavin, G. Hughes, and S. S. Talathi, "Openeds: Open eye dataset," *arXiv preprint arXiv:1905.03702*, 2019.

[14] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[17] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[18] J. Song, H. Jeong, and J. Jeong, "Performance optimization of object tracking algorithms in opencv on gpus," *Applied Sciences*, vol. 12, no. 15, p. 7801, 2022.

[19] Y. Yang, X. Wang, B. Sun, and Q. Zhao, "Channel expansion convolutional network for image classification," *IEEE Access*, vol. 8, pp. 178 414–178 424, 2020.

BanglaEmbed: Efficient Sentence Embedding Models for a Low-Resource Language Using Cross-Lingual Distillation Techniques

Muhammad Rafsan Kabir
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
muhammad.kabir@northsouth.edu

Md. Mohibur Rahman Nabil
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mohibur.nabil@northsouth.edu

Mohammad Ashrafuzzaman Khan
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mohammad.khan02@northsouth.edu

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899723

***Abstract*—Sentence-level embedding is essential for various tasks that require understanding natural language. Many studies have explored such embeddings for high-resource languages like English. However, low-resource languages like Bengali (a language spoken by almost two hundred and thirty million people) are still under-explored. This work introduces two lightweight sentence transformers for the Bangla language, leveraging a novel cross-lingual knowledge distillation approach. This method distills knowledge from a pre-trained, high-performing English sentence transformer. Proposed models are evaluated across multiple downstream tasks, including paraphrase detection, semantic textual similarity (STS), and Bangla hate speech detection. The new method consistently outperformed existing Bangla sentence transformers. Moreover, the lightweight architecture and shorter inference time make the models highly suitable for deployment in resource-constrained environments, making them valuable for practical NLP applications in low-resource languages.**

***Index Terms*—Sentence Transformer, Knowledge Distillation, Paraphrase Detection, Semantic Textual Similarity**

I. Introduction

In recent years, there have been remarkable advancements in the field of natural language processing (NLP) [1]. One significant development is the emergence of high-quality sentence embedding models [2], which can effectively map sentences into an embedding space based on their meaning and context. However, low-resource languages like Bangla still lag behind high-resource languages like English in terms of the quality of sentence transformers. This disparity exists primarily due to the limited availability of high-quality data in languages like Bangla.

Low-resource languages lack high-quality, large text corpora necessary for developing effective sentence embedding models [3]. To address this gap, our work proposes an approach that utilizes a machine translation dataset [4] instead of a large text corpus to train lightweight sentence transformers for the Bangla language. In this study, we introduce two distinct embedding models [2] for Bangla, trained using an English-Bangla machine translation dataset [5] and two different loss functions—mean squared error (MSE) and multiple negatives ranking loss [6].

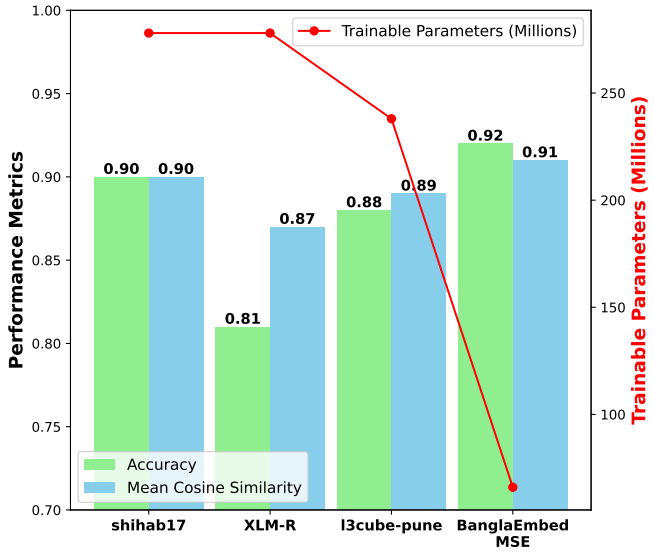


Fig. 1. Performance comparison of our proposed sentence transformer, `BanglaEmbed-MSE`, on the paraphrase detection task, evaluated using accuracy, mean cosine similarity, and the number of trainable parameters.

In our study, we employed a knowledge distillation pipeline [7] that utilizes a pre-trained teacher sentence embedding model alongside a custom student sentence embedding model. Since Bangla lacks a high-performing pre-trained sentence embedding model, we used an English pre-trained sentence transformer as the teacher model. As an English sentence and its translated Bangla counterpart convey the same contextual meaning, the embeddings for both sentences should map to the same embedding space. This approach allows us to supervise the training of lightweight Bangla sentence transformers (student models) using a pre-trained English sentence transformer (teacher model). Our proposed methodology facilitates the training of an embedding model for a low-resource language like Bangla by leveraging a pre-trained English sentence trans-

former and a machine translation dataset, rather than relying on a large text corpus. Specifically, we used the English-Bangla Machine Translation dataset introduced by [5] for training. Figure 1 presents a performance comparison of our introduced sentence embedding model, `BanglaEmbed-MSE`, with other existing Bangla sentence transformers on the paraphrase detection task. The significant contributions of this work are as follows:

- Introduction of two lightweight sentence transformers for a low-resource language, Bangla, utilizing mean squared error (MSE) and multiple negatives ranking loss.
- The novel training approach employs cross-lingual knowledge distillation and leverages a pre-trained model from a high-resource language to overcome the lack of large Bangla corpora, using a smaller translation dataset.
- The proposed models are evaluated against existing Bangla sentence transformers across multiple downstream tasks, demonstrating superior performance with reduced computational requirements.

II. Related Works

Sentence Embeddings. Sentence embedding plays a crucial role in modern language models, capturing semantic meaning that informs the reasoning abilities of these models. Numerous efforts have been made to improve sentence embeddings, particularly in high-resource languages like English [8], [9]. However, significant gaps remain for low-resource languages. To address this, various multilingual approaches [10], [11], have been proposed to extend sentence representation learning to a wider set of languages.

Multilingual Sentence Embeddings. The Multilingual Universal Sentence Encoder (mUSE) [12] employs a dual encoder architecture and was trained using a multi-task learning approach on the SNLI dataset [13]. The model's training included a translation ranking task, where it had to identify the correct translation from a set of candidates. To improve performance, it used hard negative samples—sentences similar to the correct translation but not entirely accurate. Trained on 16 languages, the model showed strong effectiveness in multilingual tasks. For low-resource language translation tasks, Gao et al. [14] employed the pre-trained SimCSE model to generate sentence embeddings, which were combined with representations using an embed-fusion module. This output was fed into the encoder-decoder attention layer of the model's decoder block. During training, only the parameters of the embed-fusion module and transformer model were updated, while the SimCSE parameters were frozen. During inference, the SimCSE and embed-fusion modules acted as plug-ins, ensuring compatibility with various model architectures.

Distillation for Sentence Transformers. Knowledge distillation (KD) has proven to be an effective approach for transferring knowledge from a large, high-capacity model (teacher) to a smaller, more efficient model (student) [7]. In [15], SimTDE was introduced, employing knowledge distillation to train a student model with a shallower token embedding block and fewer transformer layers than the teacher model. Despite its simpler architecture, the student model achieved performance comparable to the teacher while offering twice the inference efficiency. Similarly, Reimers and Gurevych [10] used a KD-based approach, where a pre-trained English sentence transformer served as the teacher model, and a lightweight student model was trained to learn sentence representations for both English and additional languages. This process enabled the student model to capture cross-lingual representations under the guidance of the more robust teacher model.

III. Methodology

A. Problem Formulation

Let D be a machine-translation dataset, where each data point consists of a Bangla sentence BN_i and its corresponding English translation EN_i, i.e., $D = \{BN_i, EN_i\}_{i=1}^{N}$. When an English sentence EN_i is passed through the teacher model T, it generates an embedding $E_T(EN_i)$, which can be considered the *target embedding*, denoted as $\mathbf{y}_i$. Similarly, when the corresponding Bangla sentence BN_i is passed through the student model S, the model produces an embedding $E_S(BN_i)$, which can be considered the *predicted embedding*, denoted as $\hat{\mathbf{y}}_i$. The goal is to optimize the parameters of the student model S such that the distance between the *predicted embedding* $\hat{\mathbf{y}}_i = E_S(BN_i)$ and the *target embedding* $\mathbf{y}_i = E_T(EN_i)$ is minimized.

$$L = \frac{1}{N} \sum_{i=1}^{N} \text{distance}\left(\mathbf{y}_i, \hat{\mathbf{y}}_i\right) \tag{1}$$

Here, *distance* represents a metric that quantifies the difference between the teacher's and student's embeddings. By minimizing this distance, the student model S learns to align its output embeddings $\hat{\mathbf{y}}_i = E_S(BN_i)$ as closely as possible to the teacher's embeddings $\mathbf{y}_i = E_T(EN_i)$.

B. Training Pipeline for `BanglaEmbed`

Low-resource languages like Bangla still face a shortage of high-performing sentence embedding models compared to high-resource languages. In this study, we introduce two distinct Bangla sentence transformers: `BanglaEmbed-MNR` and `BanglaEmbed-MSE`. These sentence transformers are trained using a cross-lingual knowledge distillation approach on an English-Bangla machine translation dataset.

Training Data. For training purposes, we utilized the publicly available BanglaNMT machine translation dataset [5], which consists of parallel English and Bangla statements. The initial size of the corpus was 2.75 million sentence pairs. After pre-processing, the dataset was reduced to approximately 2.66 million sentence pairs, which is sufficient for training a sentence transformer. This dataset is the largest available machine translation resource for the Bangla language. The original and the pre-processed datasets are accessible through

English Sentence	Bangla Sentence
He eats nothing other than fruit.	উনি ফল ছাড়া আর কিছুই খান না।
Internet Gaming Worries Saudi Parents.	সৌদি অভিভাবকরা ইন্টারনেট গেমিং নিয়ে চিন্তিত।
Tunisians remembered the 10th anniversary of the death of their country's first president - Habib Bourguiba.	তিউনিশিয়া বাসী তাদের প্রথম প্রেসিডেন্ট হাবিব বুরগুইবার ১০ম মৃত্যুবার্ষিকী পালন করেছেন।

Fig. 2. Sample EN-BN sentence pairs from the machine translation dataset.

the official GitHub repository[1]. A few sample pairs from the training dataset are shown in Figure 2.

Knowledge Distillation. Instead of following the standard training approach, we adopted a unique strategy for training sentence transformers in a low-resource language by leveraging knowledge from a pre-trained sentence transformer in a high-resource language, as proposed by [10]. To implement this, we utilized a knowledge distillation pipeline [7], creating a teacher-student framework between a high-performing English sentence transformer (teacher) and a custom lightweight Bangla sentence transformer (student).

In this process, the pre-trained English sentence transformer (teacher) was provided with English statements from the machine translation dataset (discussed in Section III-B), while the custom lightweight embedding model (student) was fed the corresponding Bangla statements. Since the English and Bangla embeddings represent the same contextual meaning, they should ideally map to the same embedding space. The student model's output embeddings were supervised using the embeddings generated by the pre-trained teacher model. Consequently, knowledge was distilled from the pre-trained English sentence transformer into the lightweight Bangla sentence transformer, completing the distillation process. This cross-lingual distillation method allows for the training of a Bangla sentence embedding model without requiring an extremely large text corpus. Figure 3 illustrates the training pipeline used in this study.

Loss Functions. We employed two different loss functions during the training phase. Specifically, we utilized Mean Squared Error (MSE) and Multiple Negatives Ranking Loss for training `BanglaEmbed-MSE` and `BanglaEmbed-MNR`, respectively.

1) **Mean Squared Error:** To train the `BanglaEmbed-MSE` sentence transformer, the MSE loss function was used, which calculates the squared difference between the embeddings from the English transformer (teacher) and the Bangla transformer (student), as shown in equation 2. The goal is to minimize this loss by aligning the two embeddings in the embedding space.

$$L_{MSE} = \frac{1}{N}\sum_{n=1}^{N}(E_T^i - E_S^i)^2 \tag{2}$$

Here, E_T and E_S refer to the embeddings generated by the teacher and student models, respectively, and N represents the total number of sentence pairs.

2) **Multiple Negatives Ranking Loss:** This loss function is employed to train `BanglaEmbed-MNR`. The objective of this ranking loss is to minimize the distance between the teacher embedding and the positive sample (correct translation), while maximizing the distance between the teacher embedding and the negative samples (incorrect translations) in the embedding space. During training, one sample from the batch is considered the positive sample, while the remaining samples are treated as negative examples. Equation 3 illustrates the multiple negatives ranking loss.

$$L_{MNR} = -\log \frac{e^{(\cos(E_T, E_S^+))}}{e^{(\cos(E_T, E_S^+))} + \sum_{n=1}^{K} e^{(\cos(E_T, E_S^{-(n)}))}} \tag{3}$$

Here, E_S^+ refers to the positive sample, and $E_S^{-(n)}$ refers to the n-th negative sample, and K represents the total number of negative samples.

C. Model Overview

For our training pipeline, we utilize two models—a teacher and a student. The teacher model is a pre-trained sentence transformer capable of generating high-quality embeddings for English sentences. Specifically, we use the pre-trained **multi-qa-distilbert-cos-v1** from SBERT, which contains 66.4 million parameters. This model has been trained on 215 million diverse question-answer pairs, making it highly effective for generating robust sentence embeddings.

The student models, `BanglaEmbed-MSE` and `BanglaEmbed-MNR`, are custom embedding models based on the **distilbert-base-uncased** architecture [16], a lightweight variant of BERT [17]. Like the teacher model, the student model also has 66.4 million parameters. The student model was adapted for sentence embedding tasks by adding a pooling layer that averages token embeddings, resulting in compact and efficient sentence representations. The lightweight student model, along with its pooling mechanism, makes it ideal for downstream tasks in low-resource languages like Bangla.

IV. Experiments

A. Setup

Dataset. To evaluate the performance of the introduced sentence transformers, we assessed them on three distinct tasks: paraphrase detection, semantic textual similarity (STS), and hate-speech classification. For the paraphrase detection task, we employed the test set from the public *BanglaParaphrase* dataset by [18], which contains 23,300

[1]https://github.com/csebuetnlp/banglanmt

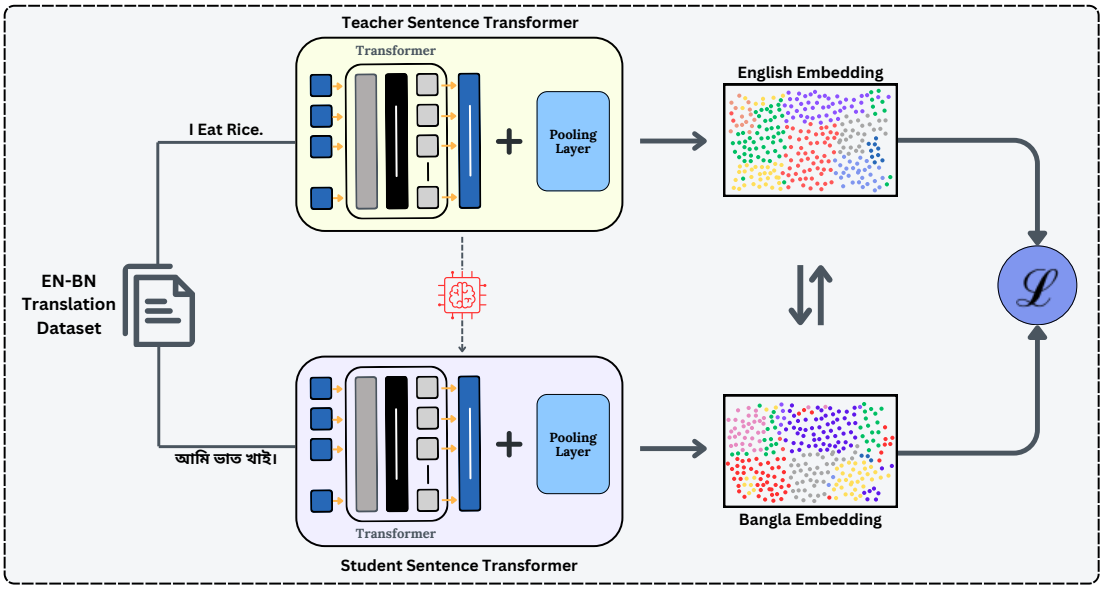


Fig. 3. Proposed cross-lingual knowledge distillation methodology for training the Bangla sentence transformer, leveraging an English-Bangla machine translation dataset. The bidirectional arrow (↓↑) indicates that both English and Bangla embeddings are aligned to map into the same embedding space.

sentence pairs (source and target). For the STS evaluation, we utilized the *SemEval Task 1: Semantic Textual Similarity (STS)* dataset, consisting of 250 pairs of statements with labels ranging from 0 to 5, a widely recognized gold-standard dataset [19]. Given the dataset's high quality, we translated it into Bangla rather than using a lower-quality Bangla STS dataset. The English sentence pairs were translated into Bangla using GPT-4o, followed by manual human validation, while preserving the original labels. Finally, we conducted a qualitative evaluation on the Bangla hate-speech classification task using the *Bengali Hate Speech Dataset* introduced by [20]. This dataset contains 3,420 instances and categorizes hate speech into five distinct types: personal, political, geopolitical, gender abuse, and religious hate speech.

Implementation Details. We introduced two versions of a sentence transformer for Bangla, a low-resource language. These two versions were trained on the BanglaNMT machine translation dataset [5], using mean squared error and multiple negatives ranking loss, respectively. Both sentence transformers were trained for 10 epochs with a batch size of 4, utilizing the *SentenceTransformerTrainer* from Sentence Transformers (SBERT) for efficient model training. The experiments were implemented using the *PyTorch* framework. All training and evaluation were conducted on an NVIDIA RTX 3080 GPU with 12 GB of VRAM. Table I presents the hyperparameters employed during our experiments.

Evaluation. We used three strategies to evaluate the performance of our sentence transformers, alongside other available Bangla sentence transformers. First, we evaluated all models on the Bangla paraphrase detection task by calculating the Mean Cosine Similarity (MCS) between source and target

TABLE I
HYPERPARAMETERS USED DURING OUR EXPERIMENTS.

Hyperparameter	Value
Epochs	10
Batch Size	4
Warmup Ratio	0.1
Learning Rate	5e-5
Optimizer	AdamW

embeddings. Classification accuracy was also measured using a cosine similarity threshold of 0.8; pairs with cosine similarity $\geq$ 0.8 were classified as paraphrases, while those below were classified as non-paraphrases. Second, we assessed the sentence transformers' performance on the Semantic Textual Similarity (STS) task, where the objective was to assign a similarity score (ranging from 0 to 5) to pairs of Bangla sentences. We used two evaluation metrics for this task: Pearson correlation (r) and Spearman correlation (ρ). Lastly, we conducted a qualitative evaluation by generating t-SNE clustering plots [21] for a Bangla hate-speech classification task consisting of five distinct classes. By examining the separation and clustering of the generated t-SNE plots, we compared the quality of the embeddings given by each of the sentence transformers.

B. Results and Analysis

We compare the performance of our proposed Bangla embedding models across three downstream tasks: Paraphrase Detection, Semantic Textual Similarity (STS), and Hate Speech Classification. Table II presents the results for the paraphrase detection and STS tasks, achieved by both the introduced sentence embedding models and existing embeddings for the Bangla language. Additionally, the table includes the

TABLE II
COMPARISON OF EVALUATION RESULTS FOR TWO DOWNSTREAM TASKS: PARAPHRASE DETECTION AND SEMANTIC TEXTUAL SIMILARITY (STS). HERE, *MCS* DENOTES MEAN COSINE SIMILARITY.

Models	**Parameters**	**Inference Time**	**Paraphrase Detection**		**STS**	
			MCS	Accuracy	ρ	r
Bangla Sentence Transformer [2]	278 M	13.4 s	0.90	0.90	0.65	0.65
XLM-R 100L BERT NLI STSB [2]	278 M	13.9 s	0.87	0.81	0.65	0.64
BengaliSBERT-STS [22]	238 M	12.7 s	0.89	0.88	0.72	**0.72**
BanglaEmbed-MNR (Ours)	66 M	12.3 s	0.87	0.82	0.63	0.62
BanglaEmbed-MSE (Ours)	66 M	**12.0 s**	**0.91**	**0.92**	**0.73**	0.70

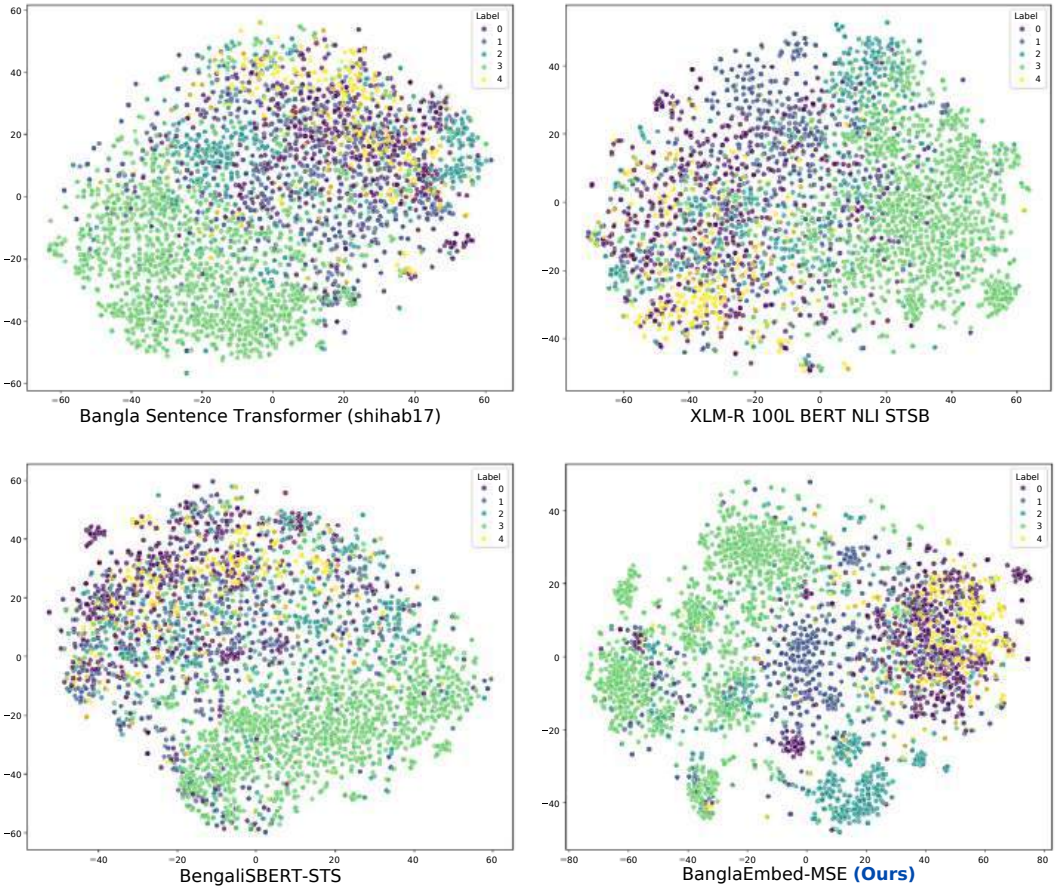


Fig. 4. t-SNE visualizations of four distinct sentence transformers. The `BanglaEmbed-MSE` model shows superior performance in separating clusters, indicating higher quality sentence embeddings compared to the other models.

inference time required to compute embeddings for the source sentences in the BanglaParaphrase dataset [18].

From Table II, we observe the following: ***(1)*** Both of the introduced embedding models, `BanglaEmbed-MNR` and `BanglaEmbed-MSE`, are lightweight, each with approximately 66 million parameters, making them computationally efficient. This reduced parameter count also results in shorter inference times compared to other existing models. ***(2)*** `BanglaEmbed-MSE` achieves the best performance in paraphrase detection, with a mean cosine similarity of 0.91 and an accuracy of 0.92, outperforming all other models. ***(3)*** In the semantic textual similarity (STS) task, `BanglaEmbed-MSE` shows strong performance with a Spearman correlation of 0.73 and Pearson correlation of 0.70, while remaining efficient with fewer parameters. ***(4)*** Although `BanglaEmbed-MNR` does not surpass the highest-performing models, it remains competitive, offering computational efficiency with fewer parameters and a shorter inference time.

[2]https://huggingface.co/shihab17/bangla-sentence-transformer

For the hate speech classification task, we conducted a qualitative evaluation using t-SNE visualizations, as shown in Figure 4. This visualization compares the clustering produced by four different sentence embeddings. As illustrated, `BanglaEmbed-MSE` shows superior clustering and clearer separation of hate speech classes, indicating the generation of higher-quality embeddings compared to the other models.

V. Conclusion

In this paper, we introduced two novel Bangla sentence transformers, `BanglaEmbed-MSE` and `BanglaEmbed-MNR`, which outperform existing models with fewer parameters. By employing cross-lingual knowledge distillation, we successfully trained these models for the Bangla language, making a significant contribution to advancing natural language understanding in this low-resource language. While our models demonstrate superior performance on multiple downstream tasks, they also establish a strong baseline for future work in Bangla NLP. However, there remains room for improvement. Future efforts could focus on exploring more efficient architectures and training on more diverse and larger datasets to further enhance the models' performance.

References

[1] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimedia tools and applications*, vol. 82, no. 3, pp. 3713–3744, 2023.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410

[3] B. P. King, "Practical natural language processing for low-resource languages." Ph.D. dissertation, 2015.

[4] H. Wang, H. Wu, Z. He, L. Huang, and K. W. Church, "Progress in machine translation," *Engineering*, vol. 18, pp. 143–153, 2022.

[5] T. Hasan, A. Bhattacharjee, K. Samin, M. Hasan, M. Basak, M. S. Rahman, and R. Shahriyar, "Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 2612–2623. [Online]. Available: https://aclanthology.org/2020.emnlp-main.207

[6] M. Henderson, R. Al-Rfou, B. Strope, Y.-H. Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil, "Efficient natural language response suggestion for smart reply," *arXiv preprint arXiv:1705.00652*, 2017.

[7] G. Hinton, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[8] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 670–680. [Online]. Available: https://aclanthology.org/D17-1070

[9] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, "Universal sentence encoder for english," in *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, 2018, pp. 169–174.

[10] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 4512–4525. [Online]. Available: https://aclanthology.org/2020.emnlp-main.365

[11] A. Saraswat, K. Abhishek, and S. Kumar, "Text classification using multilingual sentence embeddings," in *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 1*. Springer, 2021, pp. 527–536.

[12] M. Chidambaram, Y. Yang, D. Cer, S. Yuan, Y.-H. Sung, B. Strope, and R. Kurzweil, "Learning cross-lingual sentence representations via a multi-task dual-encoder model," *arXiv preprint arXiv:1810.12836*, 2018.

[13] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 632–642. [Online]. Available: https://aclanthology.org/D15-1075

[14] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6894–6910. [Online]. Available: https://aclanthology.org/2021.emnlp-main.552

[15] J. Xie, C. He, J. Wang, C. Qiu, A. Kebarighotbi, and F. Ghassemi, "Simtde: Simple transformer distillation for sentence embeddings," in *SIGIR 2023*, 2023. [Online]. Available: https://www.amazon.science/publications/simtde-simple-transformer-distillation-for-sentence-embeddings

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[18] A. Akil, N. Sultana, A. Bhattacharjee, and R. Shahriyar, "BanglaParaphrase: A high-quality Bangla paraphrase dataset," in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Y. He, H. Ji, S. Li, Y. Liu, and C.-H. Chang, Eds. Online only: Association for Computational Linguistics, Nov. 2022, pp. 261–272. [Online]. Available: https://aclanthology.org/2022.aacl-short.33

[19] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, S. Bethard, M. Carpuat, M. Apidianaki, S. M. Mohammad, D. Cer, and D. Jurgens, Eds. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14. [Online]. Available: https://aclanthology.org/S17-2001

[20] M. R. Karim, B. R. Chakravarti, J. P. McCrae, and M. Cochez, "Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network," in *7th IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA,2020)*. IEEE, 2020.

[21] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[22] S. Deode, J. Gadre, A. Kajale, A. Joshi, and R. Joshi, "L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert," *arXiv preprint arXiv:2304.11434*, 2023.

An Adaptive Dual-Archive SPEA2 For Solving Multi-Objective Flexible Job Shop Scheduling Problems

Linyuan Hao
Computer School
Beijing Information Science & Technology University
Beijing, China
lyhao@bistu.edu.cn

Xu Liang
Computer School
Beijing Information Science & Technology University
Beijing, China
liangxu@bistu.edu.cn

Zhiyuan Zou*
Computer School
Beijing Information Science & Technology University
Beijing, China
zyzou@bistu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899580

***Abstract*—For high-dimensional multi-objective flexible job shop scheduling problems, balancing solution diversity and convergence is crucial. However, existing studies often focus on either diversity or convergence alone, overlooking the potential of optimizing both synergistically. To address this, we introduced a multi-objective flexible job shop scheduling model with makespan, total machine energy consumption, and total machine load as the optimization objectives. We proposed an Adaptive Dual-Archive Strength Pareto Evolutionary Algorithm (ADA-SPEA2) to improve performance on high-dimensional multi-objective optimization problems. This algorithm proposed primary and auxiliary archives to retain elite solutions and maintain solution diversity, respectively, while dynamically adjusting an adaptive fitness threshold to balance search and convergence. Experimental results confirm the feasibility of this algorithm for multi-objective flexible job shop scheduling optimization, and benchmark tests demonstrate significant advantages in both convergence and diversity.**

***Index Terms*—multi-objective optimization, flexible job shop scheduling problem, multi-objective evolutionary algorithm**

I. Introduction

In the modern industrial era, intelligent manufacturing significantly contributes to the advancement of national economic development [1]. Flexible job shop scheduling problem (FJSP) is a core technology in intelligent manufacturing. By enhancing the flexibility of production processes, it improves machinery efficiency, thereby promoting enterprise profitability. Flexible job shops better suit the current demand for intelligent manufacturing, which is dominated by discrete production modes. They meet diverse, small-batch order requirements and are widely used in fields such as machining, furniture manufacturing, and precision electronic equipment.

Traditional FJSP typically focuses on minimizing makespan as a single optimization objective [2], often solved using metaheuristic algorithms like genetic algorithms [3], tabu search [4], and differential evolution [5]. Sun et al. [6] considered machine load balancing and introduced a variable neighborhood search based on genetic algorithms to minimize makespan. Xie et al. [7] proposed a hybrid genetic-tabu search algorithm that combines the global search capability of genetic algorithms with the local search advantage of tabu search, enhancing the diversity of neighborhood solutions.

Nevertheless, with the growing complexity of production demands, traditional single-objective scheduling optimization solutions can no longer meet the needs of modern enterprises [8] [9]. Driven by an increasingly volatile market environment and customer-specific demands, companies must consider multiple factors in the production process, such as production cost, delivery time, resource utilization, and product quality. Consequently, multi-objective optimization has emerged as a crucial direction in FJSP, helping companies achieve an effective balance among various production goals, thereby realizing efficient, flexible, and refined production management.

The computational process of multi-objective flexible job shop scheduling problem (MOFJSP) is fundamentally an NP-hard problem [10], marked by high complexity and numerous technical challenges. First, multi-objective optimization requires the simultaneous satisfaction of multiple, often conflicting performance metrics, complicating both mathematical modeling and solution search. Second, the dynamic and uncertain nature of flexible job shops necessitates scheduling systems with high adaptability and rapid response capabilities to handle constantly changing production environments. Finally, when addressing large-scale problems, multi-objective optimization algorithms still encounter substantial challenges in enhancing solution diversity and computational efficiency, placing higher demands on the design and optimization of existing algorithms.

Multi-objective evolutionary algorithms are highly versatile for solving optimization problems, unrestricted by specific problem types. Each run produces a set of Pareto optimal solutions, demonstrating good generality and robustness, thus gaining wide application in multi-objective optimization. Typical multi-objective evolutionary algorithms include the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [11], the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [12], and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [13]. Existing multi-objective evolution-

ary algorithms often use mechanisms like crowding distance calculation and reference point decomposition to maintain solution diversity. However, these mechanisms can exhibit issues such as premature convergence and loss of diversity in complex or high-dimensional spaces. To address these challenges, some studies have proposed improvements like dynamically adjusting archive size or adaptively tuning crossover and mutation rates to enhance convergence and diversity. Although these methods improve algorithm performance to some extent, limitations in archive management still exist in high-dimensional scenarios.

SPEA2 maintains an external archive through strength-based fitness calculation to store non-dominated solutions. However, its single-archive strategy lacks sufficient balance between preserving elite solutions and thoroughly exploring the search space, leading to premature convergence and reduced solution diversity. To address this issue, we propose an adaptive dual-archive strategy to dynamically manage the trade-off between exploration and exploitation. This strategy includes a main archive and an auxiliary archive: the main archive stores elite solutions to maintain convergence, while the auxiliary archive preserves population diversity to prevent premature convergence. By incorporating adaptive moderation thresholds, this approach dynamically adjusts the balance between exploration and exploitation during the optimization process, effectively enhancing the algorithm's adaptability and solution quality at different stages. This improvement provides SPEA2 with a novel archive management strategy, significantly extending its existing capabilities and addressing its limitations in high-dimensional scenarios.

II. Problem description and mathematical model

This section begins with A. Problem description, which outlines the core features, constraints, and objectives of the MOFJSP. Following this, B. Mathematical model provides a formalized representation of the problem, defining the key variables, constraints, and optimization objectives to establish a structured foundation for solution development.

A. Problem description

In the flexible job shop scheduling problem, there are $\mathcal{J}$ tasks and $\mathcal{M}$ processing devices, represented by the sets $\{j_1, j_2, \ldots, j_{\mathcal{J}}\}$ and $\{m_1, m_2, \ldots, m_{\mathcal{M}}\}$, respectively. Each task consists of multiple operations $\{\mathcal{O}_{1,j}, \mathcal{O}_{2,j}, \ldots, \mathcal{O}_{\mathcal{K}_j,j}\}$ that must be processed in a specific sequence. These operations may be performed on different devices, each with unique time and energy requirements. Flexible job shop scheduling aims to optimize the assignment of operations to devices to achieve the best overall objective. The constraints of the MOFJSP are as follows:

- At any given time, each machine can handle only one operation.
- Each job can be processed on only one machine at a time.
- Once an operation begins, it cannot be interrupted until completion.
- All jobs have equal priority.
- There are no precedence constraints between different jobs, however, operations within the same job must follow a defined sequence.
- All jobs are available for processing from the initial time point.

The symbols and definitions used in the multi-objective flexible job shop scheduling problem are detailed in Table I.

TABLE I: Definitions of symbols for the MOFJSP.

Symbol	Definition
$\mathcal{J}$	Total number of tasks.
$\mathcal{K}_j$	Total number of steps in task j.
$\mathcal{O}_{k,j}$	The k-th operation in task j.
j	Task index, $j \in [1, \ldots, \mathcal{J}]$.
k	Operation step index within task j, $k \in [1, \ldots, \mathcal{K}_j]$.
m	Device index, $m \in [1, \ldots, \mathcal{M}]$.
$\mathcal{M}$	Total number of available devices.
$\mathcal{P}_{k,j}$	Number of devices for the k-th step of task j.
β_m	Energy consumption rate for device m.
$\phi_{k,j}$	Start time of the k-th step in task j.
$\phi_{k,j,m}$	Start time of the k-th step of task j on device m.
$\psi_{k,j}$	Completion time of the k-th step of task j.
Ψ	Maximum completion time across all steps.
$\mathcal{L}$	Number of optimization objectives.
o	Optimization objective index, $o \in [1, \ldots, \mathcal{L}]$.
Λ	Large constant for constraint handling.
$\tau_{k,j,m}$	Processing time of the k-th step of task j on device m.

B. Mathematical model

This study initially focuses on enterprise efficiency by selecting the minimization of makespan f_1 as the primary objective, aiming to shorten production cycles and enhance response speed. To address the impact of enterprise efficiency on MOFJSP, the minimization of total machine energy consumption f_2 and total machine load f_3 are also considered as secondary objectives, as defined by Equations (1), (2), and (3).

$$f_1 = \min \max_{k,j} \ \psi_{k,j} \tag{1}$$

$$f_2 = \min \sum_{m=1}^{\mathcal{M}} \sum_{j=1}^{\mathcal{J}} \sum_{k=1}^{\mathcal{K}_j} \beta_m \cdot \tau_{k,j,m} \cdot \delta_{k,j,m} \tag{2}$$

$$f_3 = \min \sum_{j=1}^{\mathcal{J}} \sum_{k=1}^{\mathcal{K}_j} \sum_{m=1}^{\mathcal{M}} \tau_{k,j,m} \cdot \delta_{k,j,m} \tag{3}$$

To achieve the stated objective functions, several MOFJSP constraints must be satisfied to ensure a feasible and effective scheduling process. These constraints include:

$$\phi_{k,j} \geq 0, \quad \psi_{k,j} \geq 0 \tag{4}$$

$$\psi_{k,j} \leq \Psi \tag{5}$$

$$\sum_{m=1}^{\mathcal{P}_{k,j}} \delta_{k,j,m} = 1 \tag{6}$$

$$\delta_{k,j,m} = \mathbf{1}\left(\text{operation } \mathcal{O}_{k,j} \text{ is assigned to device } m\right) \tag{7}$$

$$\phi_{k,j} + \tau_{k,j,m} \leq \phi_{l,i} + (1 - \xi_{k,j,l,i,m}) \cdot \Lambda \tag{8}$$

$$\xi_{k,j,l,i,m} = \mathbf{1}\left(\mathcal{O}_{k,j} \text{ precedes } \mathcal{O}_{l,i} \text{ on device } m\right) \tag{9}$$

$$\psi_{k,j} \leq \phi_{k+1,j} + (1 - \xi_{l,i,k+1,j,m}) \cdot \Lambda \tag{10}$$

$$\phi_{k,j} + \delta_{k,j,m} \cdot \tau_{k,j,m} \leq \psi_{k,j} \leq \phi_{k+1,j} \tag{11}$$

$$\sum_{j=1}^{\mathcal{J}} \sum_{k=1}^{\mathcal{K}_j} \xi_{k,j,l,i,m} = \delta_{l,i,m} \tag{12}$$

$$\sum_{i=1}^{\mathcal{J}} \sum_{k=1}^{\mathcal{K}_i} \xi_{k,j,l,i,m} = \delta_{k,j,m} \tag{13}$$

Equation (4) enforces the non-negativity of all time variables. Equation (5) specifies the total completion time constraint, ensuring that each job meets the overall deadline. Equations (6) and (7) assign each operation to a single machine. Equations (8), (9), and (10) ensure that each machine processes only one operation at a time. Equation (11) establishes sequencing constraints to guarantee that operations within each job are executed in the correct order. Finally, Equations (12) and (13) define cyclic operation constraints for machines.

III. Adaptive dual-archive strategy SPEA2

This section introduces the proposed ADA-SPEA2. Figure 1 provides an overview of the adaptive dual-archive SPEA2 structure. Subsection A explains the initialization strategy, highlighting the hybrid approach that integrates random, local, and global selection methods. Subsections B and C discuss the dual-archive structure and the adaptive fitness threshold adjustment mechanism, which jointly balance exploration and convergence. Finally, Subsection D outlines the archive management mechanism, which evaluates and classifies solutions to maintain diversity and enhance convergence.

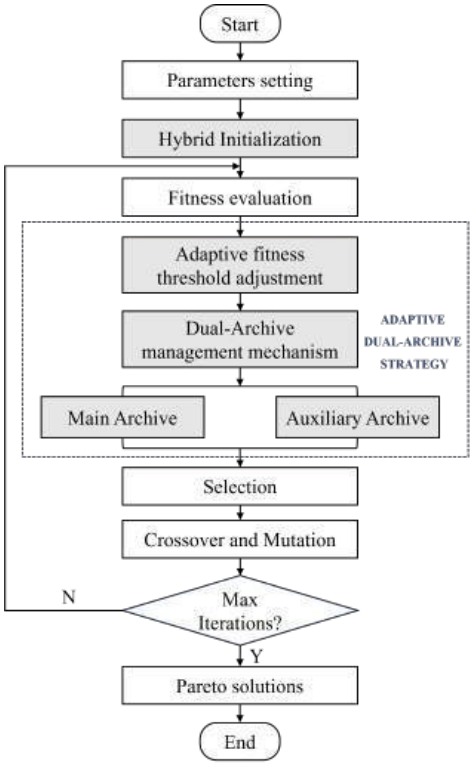


Fig. 1: Adaptive dual-archive SPEA2 structure diagram

A. Initialization strategy

In MOFJSP, the core of scheduling lies in optimizing the decision-making process for operation sequencing and machine selection. This paper employs a serial chromosome encoding method that integrates both operation and machine information. Each chromosome consists of two components: the left side encodes the operation sequence (OS), representing the order of operations, while the right side, termed machine sequence (MS) encoding, has the same length as OS and specifies the machine assigned to each operation, as illustrated in Figure 2. To effectively enhance algorithm performance and maintain solution diversity, this paper proposes a hybrid initialization strategy that combines three methods random selection, local selection, and global selection to optimize the quality of the initial population.

Operation Sequence(OS)							Machine Sequence(MS)						
2	3	1	2	2	1	3	2	1	3	2	1	2	1
$O_{2,1}$	$O_{3,1}$	$O_{1,1}$	$O_{2,2}$	$O_{2,3}$	$O_{1,2}$	$O_{3,2}$	M_3	M_2	M_3	M_2	M_2	M_3	M_1

Fig. 2: Chromosome encoding

B. Structure of the dual-archive strategy

In multi-objective optimization, maintaining both solution convergence and diversity is a critical challenge. To address this, we propose a dual-archive mechanism that dynamically balances exploration and exploitation, thereby enhancing the algorithm's overall performance. This dual-archive structure consists of a main archive and an auxiliary archive, each serving distinct roles that collectively achieve comprehensive coverage of the Pareto front.

The main archive stores elite non-dominated solutions, filtered based on an adaptive fitness threshold to ensure optimal convergence. The archive size is strictly controlled to prevent redundancy and improve computational efficiency. In parallel, the auxiliary archive retains solutions that do not meet the main archive's admission criteria but are crucial for preserving population diversity. This archive covers diverse regions of the solution space, thereby broadening the Pareto front coverage.

C. Adjustment mechanism of the adaptive fitness threshold

To achieve an effective balance between exploration and exploitation, we propose an adaptive fitness threshold strategy that dynamically adjusts with the iteration count, thereby modifying the admission criteria for the main archive. In multi-objective optimization, maintaining a balanced emphasis on exploration and convergence is essential, particularly at different stages of the algorithm. To this end, we employ a non-linear decay function for the adaptive threshold, allowing for an emphasis on exploration in the early stages and facilitating faster convergence in the later stages.

Initially, the fitness threshold is set high, permitting a larger number of solutions to enter the main archive. This encourages extensive exploration, ensures broad population

coverage of the solution space, and prevents premature convergence to local optima, thus establishing a solid foundation for subsequent convergence. As optimization advances, the threshold gradually decreases, making the selection criteria more stringent and guiding the algorithm to focus on retaining elite solutions, thereby accelerating convergence. The dynamic threshold adjustment follows an exponential decay model, as specified below:

$$F_i = F_0 \cdot \exp\left(-\frac{r \cdot i}{G}\right) \tag{14}$$

Here, F_i represents the fitness threshold at generation i, with F_0 as the initial threshold, r as the decay rate, i as the current generation, and G as the total number of iterations. This exponential decay strategy facilitates ample exploration in the early stages and strengthens convergence in the later stages, achieving a dynamic balance between solution quality and diversity throughout the optimization process.

IV. EXPERIMENTS AND ANALYSIS OF RESULT

The experimental operations in this study were implemented in Python using Visual Studio Code on a Windows 11 operating system, with hardware comprising an Intel Core i7-14700HX CPU @ 3.2 GHz and a 64-bit architecture.

A. Experimental settings and performance metrics

The performance of ADA-SPEA2 was validated using the Brandimarte and FT datasets [14] [15], which are standard job shop scheduling benchmarks with configurations ranging from 6 to 20 jobs and 6 to 15 machines, providing a comprehensive test of the algorithm's effectiveness across varying levels of complexity.

The evaluation metrics included inverted generational distance (IGD) [16], where lower values indicate better performance, and hypervolume (HV) [17], where higher values reflect superior results. IGD and HV are widely used in multi-objective optimization to assess solution convergence and distribution, offering a robust evaluation of algorithm performance. Each test was conducted over multiple runs to ensure consistency and statistical reliability.

B. Comparative Analysis and Discussion of Results

To validate the effectiveness of the proposed algorithm, we conducted comparative analyses against traditional mainstream multi-objective optimization algorithms: MOEA/D [14], NSGA-II [18], and SPEA2 [13]. These comparative algorithms, along with the proposed ADA-SPEA2, utilize identical encoding and initialization methods, as well as consistent crossover and mutation operations. Each experiment was independently run 20 times to ensure the reliability of the results.

TABLE II: IGD values for different instances

	IGD			
Instance	NSGA-II	MOEA/D	SPEA2	ADA-SPEA2
Mk01	0.319	0.373	0.297	**0.291**
Mk02	0.173	**0.140**	0.145	0.143
Mk03	0.026	0.035	0.062	**0.018**
Mk04	**0.068**	0.076	0.096	0.070
Mk05	0.162	0.123	0.119	**0.104**
Mk06	0.206	0.067	0.205	**0.067**
Mk07	0.098	0.099	0.108	**0.092**
Mk08	0.040	0.051	0.028	**0.025**
Mk09	0.113	0.128	0.092	**0.084**
Mk10	**0.156**	0.268	0.164	0.162
FT10	0.046	0.028	0.031	**0.021**
FT20	0.204	0.196	**0.182**	0.188

Table II demonstrates ADA-SPEA2's superior performance, achieving 8 optimal and 2 sub-optimal IGD values. These results underscore the strength of its adaptive dual-archive strategy in consistently generating high-quality solutions near the Pareto front, enhancing both convergence and solution diversity across diverse instances. ADA-SPEA2 consistently surpasses other algorithms in effectiveness and efficiency, highlighting its robustness and reliable convergence behavior.

Similarly, Table III further validates ADA-SPEA2's capabilities, with 7 optimal, 3 sub-optimal, and 1 near-optimal HV results, demonstrating its effectiveness in complex multi-objective optimization scenarios. ADA-SPEA2 excelled in challenging instances such as Mk01, Mk03, Mk04, Mk07, Mk08, and Mk09, outperforming the comparison algorithms. Although ADA-SPEA2 fell slightly short of the optimal in a few instances, it maintained strong global search capability and consistent convergence overall. Collectively, these findings affirm ADA-SPEA2's robustness, positioning it as a leading choice for high-performance optimization.

TABLE III: HV values for different instances

	HV			
Instance	NSGA-II	MOEA/D	SPEA2	ADA-SPEA2
Mk01	0.078	0.135	0.119	**0.146**
Mk02	0.105	**0.127**	0.100	0.121
Mk03	0.114	0.077	0.129	**0.135**
Mk04	0.143	0.263	0.167	**0.270**
Mk05	**0.198**	0.194	0.134	0.153
Mk06	0.136	**0.186**	0.161	0.172
Mk07	0.156	0.184	0.119	**0.202**
Mk08	0.102	0.075	0.101	**0.118**
Mk09	0.238	0.215	0.255	**0.294**
Mk10	0.196	**0.232**	0.155	0.209
FT10	0.037	**0.089**	0.006	0.027
FT20	0.194	0.197	0.119	**0.222**

V. CONCLUSIONS

In this paper, we addressed the challenge of balancing diversity and convergence in high-dimensional multi-objective flexible job shop scheduling problems. We introduced a multi-objective scheduling model with makespan, total machine energy consumption, and total machine load as the primary optimization objectives. To enhance algorithm performance,

we proposed ADA-SPEA2, which incorporates a primary archive to retain elite solutions and an auxiliary archive to preserve diversity. This dual-archive structure, along with a dynamically adjusted fitness threshold, effectively balances exploration and exploitation. Experimental results on benchmark datasets demonstrate that ADA-SPEA2 achieves superior convergence and diversity, outperforming other algorithms across various instances. Overall, this study validates ADA-SPEA2 as a robust and efficient approach for high-dimensional multi-objective scheduling, providing a valuable solution for complex operational environments.

ACKNOWLEDGMENT

This research is partially supported by the R&D Program of Beijing Municipal Education Commission (KM202411232003), Young Backbone Teacher Support Plan of Beijing Information Science & Technology University (YBT 202425) and Research Foundation of Beijing Information & Science Technology University (2023XJJ19).

REFERENCES

[1] S. A. Alvi, X. Zhou, S. Durrani, and D. T. Ngo, "Sequencing and scheduling for multi-user machine-type communication," IEEE Transactions on Communications, vol. 68, no. 4, pp. 2459–2473, April 2020.

[2] Y. Wang, L. Fu, Y. Su, Q. Yang, and L. Wu, "Genetic algorithm in flexible work shop scheduling based on multi-objective optimization," Journal of Interdisciplinary Mathematics, vol. 21, no. 5, pp. 1249–1254, 2018.

[3] R. Li, W. Gong, L. Wang, C. Lu, and S. Jiang, "Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time," Swarm and Evolutionary Computation, vol. 74, p. 101139, 2022.

[4] Y. Du and J.-q. Li, "A deep reinforcement learning based algorithm for a distributed precast concrete production scheduling," International Journal of Production Economics, vol. 268, p. 109102, 2024.

[5] X. Wu, X. Liu, and N. Zhao, "An improved differential evolution algorithm for solving a distributed assembly flexible job shop scheduling problem," Memetic Computing, vol. 11, pp. 335–355, 2019.

[6] K. Sun, D. Zheng, H. Song, Z. Cheng, X. Lang, W. Yuan, and J. Wang, "Hybrid genetic algorithm with variable neighborhood search for flexible job shop scheduling problem in a machining system," Expert Systems with Applications, vol. 215, p. 119359, 2023.

[7] J. Xie, X. Li, L. Gao, and L. Gui, "A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems," Journal of Manufacturing Systems, vol. 71, pp. 82–94, 2023.

[8] C. Lu, Y. Huang, L. Meng, L. Gao, B. Zhang, and J. Zhou, "A Pareto-based collaborative multi-objective optimization algorithm for energy-efficient scheduling of distributed permutation flow-shop with limited buffers," Robotics and Computer-Integrated Manufacturing, vol. 74, p. 102277, 2022.

[9] K. Peng, Q.-K. Pan, L. Gao, X. Li, S. Das, and B. Zhang, "A multi-start variable neighbourhood descent algorithm for hybrid flowshop rescheduling," Swarm and Evolutionary Computation, vol. 45, pp. 92–112, 2019.

[10] I. A. Chaudhry and A. A. Khan, "A research survey: review of flexible job shop scheduling techniques," International Transactions in Operational Research, vol. 23, no. 3, pp. 551–591, 2016.

[11] Y. An, X. Chen, K. Gao, L. Zhang, Y. Li, and Z. Zhao, "A hybrid multi-objective evolutionary algorithm for solving an adaptive flexible job-shop rescheduling problem with real-time order acceptance and condition-based preventive maintenance," Expert Systems with Applications, vol. 212, p. 118711, 2023.

[12] Q. Luo, Q. Fan, Q. Deng, X. Guo, G. Gong, and X. Liu, "Solving bi-objective integrated scheduling problem of production, inventory and distribution using a modified NSGA-II," Expert Systems with Applications, vol. 225, p. 120074, 2023.

[13] M. Wan, C. Ye, and D. Peng, "Multi-period dynamic multi-objective emergency material distribution model under uncertain demand," Engineering Applications of Artificial Intelligence, vol. 117, p. 105530, 2023.

[14] Z. Wang, M. He, J. Wu, H. Chen, and Y. Cao, "An improved MOEA/D for low-carbon many-objective flexible job shop scheduling problem," Computers & Industrial Engineering, vol. 188, p. 109926, 2024.

[15] S. Shi and H. Xiong, "Solving the multi-objective job shop scheduling problems with overtime consideration by an enhanced NSGA-II," Computers & Industrial Engineering, p. 110001, 2024.

[16] E. Yuan, L. Wang, S. Cheng, S. Song, W. Fan, and Y. Li, "Solving flexible job shop scheduling problems via deep reinforcement learning," Expert Systems with Applications, vol. 245, p. 123019, 2024.

[17] Z. Xu, Z. Zheng, and X. Gao, "Energy-efficient steelmaking-continuous casting scheduling problem with temperature constraints and its solution using a multi-objective hybrid genetic algorithm with local search," Applied Soft Computing, vol. 95, p. 106554, 2020

[18] C. Song, "Improved NSGA-II for solving multi-objective hybrid flow shop scheduling," Computer Integrated Manufacturing Systems, vol. 28, no. 6, p. 1777, 2022.

Analysis of User Attention Behavior and Its Driving Factors on WeChat Public Platform

Shu Yang
Hubei Institute of Geoscience (Hubei Selenium Industrial Research Institute)
Hubei Key Laboratory of Resources and Eco-Environment Geology
WuHan, China
1848320893@qq.com

Weilu Hu
WSGRI Engineering & Surveying Incorporation Limited Wuhan
WSGRI Smart City(Wuhan) Engineering Technology Co. , Ltd. Wuhan
WuHan, China
825027247@qq.com

Qanguo Kang
Hubei Institute of Geoscience (Hubei Selenium Industrial Research Institute)
Hubei Key Laboratory of Resources and Eco-Environment Geology
WuHan, China
75755400 @qq.com

Li Zhou
Hubei Institute of Geoscience (Hubei Selenium Industrial Research Institute)
Hubei Key Laboratory of Resources and Eco-Environment Geology
WuHan, China
394590260 @qq.com

Jiewei Yi
Hubei Institute of Geoscience (Hubei Selenium Industrial Research Institute)
Hubei Key Laboratory of Resources and Eco-Environment Geology
WuHan, China
23610014 @qq.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899711

***Abstract*—In recent years, with the development of new media, such as WeChat Public Platform and Tiktok have become an important source of news and knowledge for the public. All kinds of official account have sprung up one after another. How to operate them to attract more public attention, so as to better serve the science popularization and knowledge dissemination of public welfare units, is the problem faced at this stage. This paper takes the WeChat Public Platform of Hubei Institute of Geoscience as an example, based on the data of the platform, uses XGBoost algorithm to evaluate the factors that attract the public and analyze the driving factors of their attention behavior from several aspects, such as the number of posts, article types, layout design and interaction. The following conclusions can be drawn: ① The public attention behavior factor evaluation model based on XGBoost algorithm integrates machine learning technology and data mining ideas and achieves good evaluation results; ② According to the ranking of feature importance, the type of article has the greatest impact on public attention behavior, followed by interactivity the evaluation results show that the contents of the WeChat public platform of the Hubei Institute of Geoscience involve popular geological knowledge, the latest research results, project activities and other contents, taking into account the booking consultation and link functions of the Hubei Geological Museum, which can effectively serve the public. The research results provide decision-making basis for the next operation of the public platform.**

Keywords—Public attention behavior, Evaluation of driving factors, XGBoost algorithm

I. INTRODUCTION

With the development of the internet and mobile devices, the public has more and more ways to obtain news and knowledge. The integration of new media has brought new opportunities for knowledge dissemination [1], and WeChat official account has also become an important platform for spreading knowledge and influence [2,3]. As one of the core functions of WeChat, the official account has become an important channel for information dissemination with far-reaching influence. It has strong interactivity. Users can not only obtain information, but also participate in the creation and dissemination of information. The functions of commenting, sharing, liking, etc. make the flow of information more bidirectional and increase user engagement; The information of the official account spreads very fast, and a message can quickly spread to a large number of users in a short time; The platform offers a diverse range of content forms, including text, images, audio, videos, etc., to meet the needs of different users; The influence of official account is not limited to the internal platform, and its content can often cross platform boundaries and spread on other social media through sharing and forwarding; The official account is also used for education and learning. Many educational institutions and experts provide knowledge and educational content through the official account. In short, information dissemination in the new media environment is more flexible, fast, and personalized, and it is also an important medium for cultural digitization. As an important part of the modernization and civilization construction of the Chinese nation, the cultural digitization strategy needs to shoulder new cultural missions in the new era and cultivate new quality productive forces in the cultural field. The cultural digitization strategy requires a deep integration of culture and technology to better unleash cultural productivity. The fundamental path of cultural digitization lies in the deep exploration, organization, innovation, and utilization of cultural resources through modern digital technology, enabling them to radiate new vitality and energy. As a key role, the official account has not only changed the way of production, dissemination and consumption of information, but also profoundly affected the daily life and information acquisition habits of the public.

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

Therefore, how to increase public attention has also become a problem worthy of attention and research. [4] Research has shown that platform data can reflect the underlying reasons for user behavior, and user evaluation is a reference for consumer consumption decisions, as well as an important basis for manufacturers to grasp consumer preferences and optimize production behavior[5]. The operation of WeChat official account also belongs to the category of news communication. The data of news distribution includes not only the user related data that provides algorithm support for accurate push, but also the specific feedback of news release spread through different channels after release, including reading, clicking, commenting, liking, forwarding and staying time. On the one hand, the data from the news distribution process is used to evaluate the effectiveness of news releases and provide reference for the next news topic selection[6]. On the other hand, these data are also important user data that can lay a more solid foundation for more accurate personalized push in the future. Based on the above analysis, this article attempts to make breakthroughs in new methods and technologies to improve the scientific, objective, and rational evaluation methods of factors that attract the public. Kang et al.'s [7]research shows that the XGBoost model has advantages such as shorter training time, smaller memory usage, and the highest prediction accuracy. In recent years, with the progress of data mining and machine learning technology, evaluation factors have gradually changed from case studies to data driven[8]. At present, there are few relevant studies on the evaluation of public attraction factors based on machine learning methods, and there are not many achievements to systematically study them. Based on this, this paper uses the data in the operation of Hubei Institute of Geological as training samples, constructs characteristic factors from the number of papers, article types, layout design and interactivity, and uses XGBoost algorithm to conduct empirical research on the driving factors, with a view to providing reference and reference for the improvement of the service scope and service level of the official account.

II. RESEARCH METHOD

A. *theoretical analysis*

Machine learning methods establish the relationship between feature variables and known sample categories through nonlinear fitting. This method does not require a complete indicator system for prediction, avoiding problems such as inconsistent data dimensions, correlation of feature factors, and weight settings, and improving the reliability of prediction results. And its series of model optimization strategies make its training results stable and less prone to overfitting. The importance of features can be evaluated without excessive data preprocessing [8].

Feature contribution is the evaluation of the importance of each input feature by a machine learning model. Simply put, it can tell us which features the model considers to have played a key role in the final prediction result. In datasets with numerous features, reasonable utilization of feature contribution can help us screen the most critical features, thereby improving the efficiency and performance of the model.

B. *XGBoost algorithm*

XGBoost, also known as eXtreme Gradient Boosting, is a representative boosting algorithm designed and optimized by Chen et al. [9] based on the Gradient Boosting Decision Tree (GBDT). It aims to break through the computational limitations of boosting trees and achieve engineering goals of excellent performance and fast computation. Among the numerous models in machine learning, tree models are widely favored by data scientists due to their ease of interpretation, ability to handle multiple data types, and strong generalization ability. As a member of the tree model family, XGBoost has become one of the first choices in regression and classification tasks due to its efficient parallel computing and good predictive ability[10]. XGBoost reduces the risk of overfitting in the model by randomly sampling features and samples[11]. Efficient parallel computing utilizes hardware resources to accelerate the training process and automatically calculate feature contribution. By splitting the decision tree, we can easily obtain the importance of each feature.

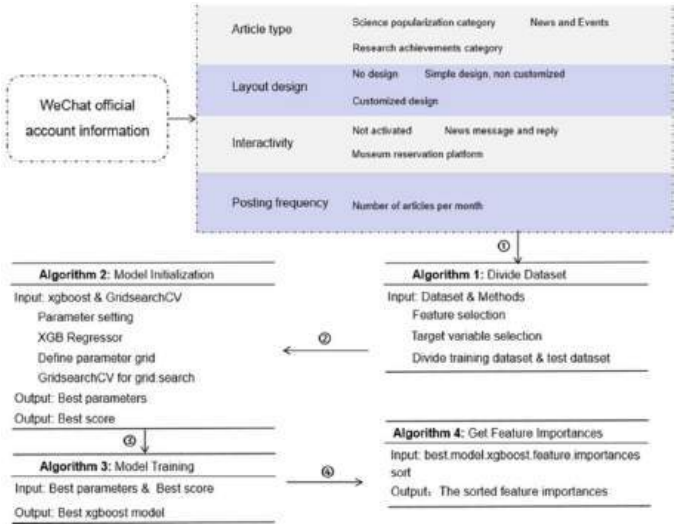


Fig. 1. Model Framework Diagram

Figure 1 shows the technical roadmap of this study. Firstly, relevant data is obtained from the WeChat public platform, and the data is preprocessed based on the factors represented by feature factors, and a feature matrix is constructed Extract the feature values and target variables corresponding to the sample points, and randomly extract 70% of the samples as the training set to import into the Python pre trained model, while the remaining 30% of the data is used as the testing set for validation Optimize hyperparameters such as learning rate, number of trees, maximum depth, etc. using GridSearchCV for XGBoost model initialization and parameter setting By comprehensively utilizing methods such as online search and learning curves to adjust model parameters, and testing the various classification evaluation indicators of the model under 10 fold cross validation, the optimal parameters of the model are found, and the performance of the model is evaluated based on negative mean square error (RMSE) The trained model will automatically provide the importance values of each feature, arranged in descending order, and draw a bar chart of feature contribution

based on the results to visually display the importance ranking of each feature by the model.

The following are the steps of the XGBoost algorithm in this study:

The loss function of XGBoost adopts a second-order Taylor expansion and adds a regularization term to the GBDT objective function. It approximates the objective function using first and second derivatives, simplifying the model while effectively reducing the risk of overfitting. The objective function of XGBoost includes two parts: loss function and regularization term.

$$\mathrm{L}(\emptyset) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{l}(\hat{\mathrm{y}}_{\mathrm{i}}, \mathrm{y}_{\mathrm{i}}) + \sum_{k=1}^{k} \theta(f_k)$$

In the formula: i represents the i-th sample in the dataset; n is the number of training samples; k is the k-th tree among all trees (K); $\mathrm{l}(\hat{y}_{\mathrm{i}}, \mathrm{y}_{\mathrm{i}})$ representing the traditional differentiable convex loss function, measuring the difference between the true label and the predicted label; $\theta(f_k)$ as a regularization term, it helps to smooth the learning of weights and avoid overfitting of the model.

$$\theta(\mathrm{f}) = \gamma^{\mathrm{T}} + \frac{1}{2}\delta\|\mathrm{w}\|^2$$

In the formula: γ and δ are regularization parameters, and w is the leaf node weight vector. When the regularization parameter is set to zero, it is a traditional gradient boosting tree. The parameters in the tree ensemble model in equation (1) contain unknown functions, so traditional optimization methods cannot be used for optimization in Euclidean space. Assuming $\hat{y}_i^{(t)}$ is the prediction of the i-th instance in the t-th iteration, add f_t to minimize the following objectives:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \theta(f_k)$$

By adding the f_t that can improve the model the most to achieve the goal of optimizing the objective function. In general, second-order approximation can quickly optimize the objective, and the objective function can be transformed into:

$$L^{(t)} \cong \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \theta(f_k)$$

In the equation, g_i and h_i are the first-order and second-order gradient statistics of the loss function, respectively. Removing the constant term can obtain the following simplification objective:

$$\tilde{L}^{(t)} \cong \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} g_i f_t^2(x_i)] + \theta(\mathrm{f_k})$$

The XGBoost algorithm uses the greedy algorithm [4] to recursively select the optimal features of the tree structure starting from the root node. Since CART trees are all binary trees, the difference between the objective function after branching and the structure score in the XGBoost algorithm can be generalized using the following formula:

$$\mathrm{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\delta} + \frac{G_R^2}{H_R+\delta} - \frac{(G_L+G_R)^2}{H_L+H_R+\delta}\right] - \gamma$$

In the formula: γ is the penalty term; G_L and H_L calculated from the left node; G_R and H_R calculated from the right node; $G_L + G_R$ and $H_L + H_R$ it is obtained by calculating intermediate nodes.

By conducting 10 fold cross validation, we identified the optimal parameters of the model and evaluated its performance based on the negative mean square error (RMSE). As the focus of this article is on extracting feature contribution, we do not need to pay too much attention to the accuracy of the model. In fact, feature contribution can also be used to help optimize model performance. In practical projects, readers only need to ensure the accuracy of the model, and the results of feature contribution can also be used to further improve the performance of the model.

III. Empirical analysis

A. Research data sources and preprocessing

Hubei Institute of Geoscience is a scientific research unit that integrates geological science research, geological exploration, geological relic investigation, geological park construction, paleontological fossil research, agricultural geological investigation, selenium rich industry research institute, natural resource research, and geological science popularization. The WeChat public platform of Hubei Institute of Geological was established in 2016, mainly for publishing related research results, popular science propaganda, and related news. From its establishment to January 2020, the number of fans was only 140, mostly employees within the organization. Therefore, this study used data from January 1, 2020 to present as research data. Select data from the 1st of each month, including the growth in fan numbers from the 1st of the previous month to the 1st of this month, the number of articles published (divided into science popularization articles, research results articles, and news activity articles), statistics on the layout and design types of published articles (no design articles, simple design articles, customized design articles), interaction volume, and the number of articles published within one month. Select the above data as the characteristic data for this study.

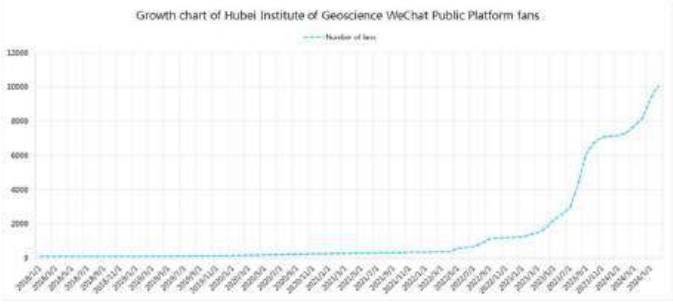


Fig. 2. Change chart of fan count

B. Feature construction

In order to aggregate multi-dimensional spatial data, this article combines the research of scholars[5,9,12] with practical research to carry out corresponding data processing and analysis. Based on the principles of scientific, systematic, representative, and obtainable feature selection, the required feature factors for this study are constructed as shown in Table 1.

TABLE I. EXTRACTION OF DRIVING FACTORS FOR PUBLIC ATTENTION

influence factor	Feature selection	Characteristic definition
Article type	research findings	Research achievements news refers to articles published through the WeChat official account platform, which mainly introduce the latest achievements, progress and findings of scientific research.
	popularization of science	Popular science news refers to the graphic or multimedia content released through the WeChat official account platform to popularize scientific knowledge, disseminate scientific ideas, promote scientific methods, and interpret scientific phenomena.
	News and Events	News activities refer to articles published through the WeChat official account platform, which mainly report the latest news events, social activities, industry trends, enterprise information, etc.
Layout design	No design	Without any design, use text or original images directly
	Simple Design	There is a certain design, but directly using existing templates on the internet
	Customized design	Customized design, incorporating unit logo and concept, using appropriate text and illustration styles for different types of news
interactive quality	No interaction	Not activated for interaction
	Passive interaction	The public can leave comments under the news section, and the management personnel of the public platform will reply regularly
	Active interaction	Open venue reservation platform, administrators can reply to public questions at any time
Posting frequency	Number of articles per month	Number of articles published per month

IV. RESULTS AND ANALYSIS

A. Model reliability assessment

The confusion matrix is a multidimensional measure of binary classification problems and a commonly used method in the field of pattern recognition to present the classification performance of algorithms. It depicts the relationship between the true attributes of sample data and the types of recognition results.

In order to further compare the prediction accuracy, five ten fold cross validation tests were conducted on 1000 quantified valid data points to establish scoring prediction models. The results are shown in Table 2, where the second to sixth columns represent the prediction accuracy of ten fold cross validation, and the seventh column represents the average value. From the average value, it can be seen that the XGBoost algorithm has a prediction accuracy of 80.75%. Overall, it can be concluded that the XGBoost algorithm has a high prediction accuracy.

TABLE II. PREDICTION ACCURACY OF TEN FOLD CROSS VALIDATION

algorithm	1	2	3	4	5	average value
XGBoost	80.02%	80.91%	81.11%	81.17%	80.55%	80.75%

B. *Driving factor analysis*

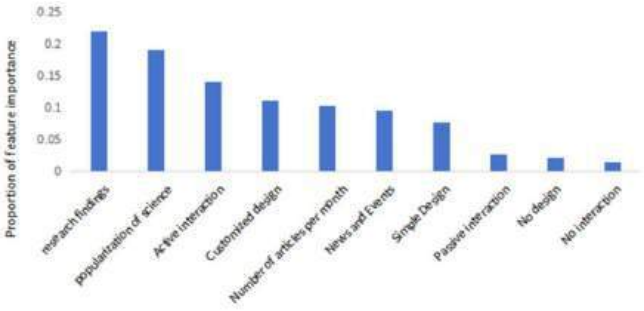


Fig. 3. Feature importance ranking results

By analyzing the ranking results of feature importance returned by the model Figure 2, it is found that article type has a significant impact on user attention. Among them, the contribution rate of research results is the highest, followed by science popularization and news activities, ranking first, second and fourth in their respective characteristic factors. Since our institute is a research institute, the public has certain psychological expectations for our scientific research achievements. The published research achievements are professional, innovative and unique, which meet the public expectations and can attract the most public attention. Moreover, some studies show that the number of articles promoted by the academic journal WeChat official account is significantly higher than the number of papers not popularized on HowNet, indicating that the promotion of the academic journal WeChat official account is positively related to the academic communication effect of the papers [13], and also provides decision-making guidance for the operation of the official account. Science popularization news presents complex scientific research results in a relatively popular way, enhancing public scientific awareness and promoting the transformation and application of scientific research results, as well as increasing public interest and participation in science. Especially in recent years, science popularization learning for primary and secondary school students has become increasingly important, which is also an important factor in attracting public attention to such news. News events refer to the progress of our hospital's related projects, leadership activities, and other related dynamics, indicating the next steps of work. They have a low appeal to the public and mainly attract partners in related businesses. Layout design also has a significant contribution to public attention. From charts and data, it can be seen that the impact of no design, simple design, and customized design on attention shows an upward trend, with customized layout design having the greatest impact. Previous studies have shown that users pay more attention to the overall sensory experience when acquiring knowledge [14]. In the digital age, libraries can activate and showcase the value of physical institutions and spatial locations through sensory design, as well as empower library service capabilities and enhance service effectiveness [15]. Customized design is not only beautiful, but also consistent with the theme of the official account. It can reflect the intention and sincerity of the news editor, which is the factor that can attract the public. The impact index of posting frequency is 11%, with the lowest impact on interactivity. The impact of non interaction and passive interaction is almost the same, while active interaction has a greater impact and ranks third.

According to the research results of this paper, we can draw a official account of the type of research institute, publish more of our research results, play the role of science popularization, and optimize the overall design can improve the number of followers of the official account, and play a better service value.

V. Conclusion and Discussion

This article empirically studies the user attention and driving factors of the WeChat public platform of Hubei Institute of Geoscience using XGBoost algorithm based on machine learning ideas, and achieves good results. The conclusions are as follows:

(1) Based on XGBoost algorithm, the analysis of official account users' attention and its driving factors is based on the idea of data mining and machine learning technology to identify potential patterns or laws from existing data samples, thus avoiding the intervention of human factors to give feature weights, enhancing the objectivity of evaluation results, which is more objective, scientific and reliable than traditional methods.

(2) The ranking results of feature importance show that research achievement news, popular science news, and customized layout design have the highest contribution rate to user attention among article types, indicating that these feature factors have an important positive impact on improving user attention.

This study conducted an empirical evaluation of user attention and driving factors by integrating machine learning and data mining ideas, and achieved good results, but there are also some shortcomings. The summary is as follows:

(1) At the theoretical level, this article is based on relevant theories such as journalism and communication, and innovatively applies machine learning technology to analyze user attention and its driving factors in response to existing problems. However, the components of driving factors still lack a systematic approach, and in the future, more systematic selection and definition of driving factors can be integrated with other disciplines.

(2) The WeChat public platform of Hubei Institute of Geological selected at the data level can only represent the official account of this type of research institute. In future research, user attention and driving factors of more types of news communication platforms can be considered, and similarities and differences can be analyzed to better serve the development of the news communication platform.

By understanding the public's attention preferences, content creators can more accurately target their audience, create content that better meets their needs, and enhance the attractiveness and dissemination of the content. The research results of this paper will be applied to the operation of the official account, adjust the operation strategy according to the research results, optimize the type of news content and interaction methods to improve user satisfaction and better serve the public. In the future, we will also continue to pay attention to the development of the official account for verification analysis and further detailed research.

REFERENCES

[1] Shi Jianlan L H. Research on the Investigation and Optimization of Promoting Red Culture Reading in University Libraries from the Perspective of Converged Media[J]. Library and Information Service, 2024,19(68):15-28.

[2] Yifan Z H C. News Narrative Innovation and Data Core Reconstruction of Data journalism WeChat Public Account ——A Comparative Study of Content Analysis Based on "Data Blog" WeChat Public Account[J]. Journalism & Communication Review, 2019,6(72):55-67.

[3] LIU Qiong L G L Z. Chinese Culture Database: Origin Gradual Sequence and Convergence[J]. Information Science, 2023,41(7):23-31.

[4] Shu Yang E. Research on user collaboration patterns of VGI based on K-means: In 2023 6th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2023)[C], 2023.

[5] Guijun Y, Xue X, Fuqiang Z. Predicting User Ratings with XGBoost Algorithm[J]. Data Analysis and Knowledge Discovery, 2019,3(1):118-126.

[6] Yujie F Huangqian Z. The Hidden Power: Data Power in News Production and Distribution[J]. JOURNAL OF SOUTHWEST MINZU UNIVERSITY (Humanities and Social Science), 2024,45(03):132-141.

[7] Jun-Feng K, Lie-Xing H, Chun-Yan Z, et al. Hourly PM2.5 prediction and its comparative analysis under multi-machine learning model[J]. China Environmental Science, 2020,40(5):1895-1905.

[8] TAN Cui H Q Y B. Application of Random Forest Algorithm in Regional Ecotourism Suitability Assessment[J]. Journal of Geo-Information Science, 2024,26(2):318-331.

[9] CHEN T G C. XGB: a scalable tree boosting system: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C], 2016.

[10] Huang Q T C Z X. Research on the Evaluation Method of Ecotourism Suitability in Subtropical Regions Based on XGBoost Algorithm[J]. Journal of Geo-Information Science, 2024,26(2):303-317.

[11] H F J. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001,5(29):1189-1232.

[12] Yuxin Y E, Jiawen L, Wanxin Z, et al. Ontology-guided knowledge graph construction for mineral prediction[J]. Earth Science Frontiers, 2024,31(4):16-25.

[13] Li Lei W X H K. Research on the Influence of Academic Journal WeChat Official Account Promotion Mode on Paper Dissemination Effect[J]. Library and Information Service, 2024,16(68):50-61.

[14] Y W. Embodiedness, materiality and interactivity: three dimensions of the affordance practice for audio reading[J]. Editorial friend, 2022(3):13-20.

[15] Wang Zheng L Y Z Y. Sensory Design Theory Applied in Library Service Design: Case Study on Library STEAM Education Activities[J]. Library and Information Service, 2024,20(68):37-49.

Predictive Modeling of In-Hospital Mortality in ICU Heart Failure Patients Using Machine Learning Techniques

Firas Jolha
Innopolis University
Innopolis, Russia
f.jolha@innopolis.university

Najlaa Jolha
najlaajolha@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899560

***Abstract*—Heart Failure (HF) is a complex clinical disease that poses significant risks to public health. According to the European Society of Cardiology, the number of people diagnosed with heart failure has been rising dramatically, with approximately 26 million people affected worldwide in 2024. This trend is expected to increase by more than 50% by 2030. Given the seriousness of this disease, early prediction of the risk of death in heart failure patients is crucial for developing individualized and effective treatment plans. While many previous studies have primarily focused on building mortality prediction systems with high predictive accuracy, increasing the sensitivity of such a system has not been considered. This study aims to fill that gap by developing a predictive model that achieves high accuracy while also focusing on sensitivity in detecting the risk of in-hospital death for heart failure patients in the Intensive Care Units (ICUs). The model was trained on the MIMIC III dataset of 1177 patients using various Machine Learning (ML) algorithms, with CatBoost being the most effective. This model demonstrated encouraging results, with an accuracy of 90% and a sensitivity of 81%. The importance of this sensitivity lies in its ability to determine if a patient requires further intervention before it is too late and to ensure that more high-risk patients are correctly diagnosed; by doing so, healthcare providers can implement timely interventions that could prevent fatalities and improve clinical outcomes, including mortality rates and quality of life for HF patients.**

***Index Terms*—heart failure, machine learning, hospitalization.**

I. Introduction

In-hospital mortality rates among patients in ICUs are alarmingly high, ranging from 6.7% to 44.0% worldwide [9]. Traditional ICU scoring systems for predicting mortality, such as the Acute Physiology and Chronic Health Evaluation (APACHE), the Simplified Acute Physiology Score (SAPS), and the Sequential Organ Failure Assessment (SOFA), often rely on the physician's history-taking or statistical models [3]. These tools allow comparison of quality of care across different ICUs [4]. However, when applied to larger populations, score-based models have relatively poor diagnostic performance [8]. The in-hospital mortality rate for patients who received treatment in an ICU was reported to be 10.6% [1]. Given the significantly higher in-hospital mortality, heart failure patients admitted to the ICU could greatly benefit from accurate prognosis tools that allow for receiving intensive care with closer follow-up. Identifying those patients at the highest risk of poor outcomes following hospital discharge can improve outcomes for ICU-admitted HF patients [14]. However, many existing prediction algorithms focus exclusively on accuracy, sometimes ignoring sensitivity indicating the inability to correctly identify high-risk patients who require urgent care. This is where ML comes into play. These algorithms have the potential to analyze large datasets effectively, and automatically reconstruct relationships between input variables and response values that traditional methods, such as logistic regression, might miss in identifying critical predictors [7]. Our contribution to this paper is to address the current shortcomings in the literature, providing a descriptive analysis of the MIMIC III data obtained from ICU patients and a comparative evaluation of ML models in predicting heart failures of ICU patients. By prioritizing sensitivity along with accuracy, we hope to improve outcomes for this vulnerable group of patients.

The rest of the paper is structured as: Section II presents the recent works in ML-based heart failure detection; Section III includes the description of the used dataset and the pipeline of the ML model; Section IV shows the results of predictive modeling and discusses the findings of this research; and Section V concludes the study and reflects on our own work.

II. Related works

This section discusses the studies that applied ML approaches to predict mortality in heart failure among ICU patients.

Predictive modeling using ML has become a powerful tool due to the versatility and flexibility of modeling techniques to understand the root causes of diseases and expand clinical knowledge; therefore, gathering and analyzing large volumes of critical care data is essential. Researchers have recently focused on developing predictive models based specifically on ML algorithms. For example, Li et al. [16] managed to create a viable ML model to predict the mortality risk for HF patients

in ICUs using the eICU Collaborative Research Database. The prediction performance of the extreme gradient boosting (XGBoost) model was the best (accuracy = 82.4%, sensitivity = 59.5%) in comparison with other machine learning models in terms of accuracy, though its sensitivity remains low. Similarly, Choi et al. [15] analyzed data from all adult patients admitted to the ICUs of two university hospitals, comparing predictive models of ICU inpatients with conventional scoring systems using the area under the receiver operating characteristic curve (AUROC). Their findings revealed that ML models achieved the highest AUROC (97.7%–95.5%) in these hospitals, but did not report the sensitivity of these models. Another study by Li et al. [14] developed the first in-hospital mortality prediction system for ICU-admitted HF patients. This system is designed to routinely monitor patients and automatically alert ICU staff to the patient's condition changes using data extracted from the MIMIC-III database. Here, XGBoost was also selected as the final model due to its conciseness, achieving an accuracy of 76% and a sensitivity of 53%. A recent study by Yang et al. [17] evaluated the effectiveness of a conditional medical generative adversarial network (c-med GAN) for predicting mortality in ICU patients, addressing issues of limited clinical data. The c-med GAN was compared to SAPS II, support vector machine (SVM), and multilayer perceptron (MLP), using data from the MIMIC-III database. The results showed that the c-med GAN outperformed all other models (AUCROC = 91%, sensitivity = 44%), even when trained on a smaller dataset. This suggests that the c-med GAN could enhance mortality predictions in ICU settings, indicating potential for further clinical research. Despite the advances in predictive modeling, we find that ML models usually suffer from low sensitivity. Furthermore, data on predictive models for heart failure patients in the ICU are publicly limited due to privacy issues, making collecting such data challenging and requiring much effort.

TABLE I
OVERVIEW OF PREDICTIVE MODELS AND THEIR PERFORMANCE IN RELATED WORKS

Study	Model	Performance
Li et al. [16]	XGBoost	accuracy = 82.4%, recall = 59.5%
Choi et al. [15]	XGBoost	AUROC = 97.7%
Li et al. [14]	XGBoost	accuracy = 76%, recall = 53%
Yang et al. [17]	XGBoost	AUROC = 91%, recall = 44%

The related studies did not pay much attention to the recall or sensitivity of trained ML models as shown in Table I, while our study aims to develop and validate a highly accurate and potentially sensitive predictive model for in-hospital mortality in patients with HF admitted to the ICU using data from the MIMIC-III database.

III. RESEARCH METHODOLOGY

In this section, we will present the data used in the study and how we performed the predictive ML modeling.

A. Data source

The dataset used in this study belongs to the MIMIC-III database (version 1.4, 2016)[1], which is a publicly available critical care database containing de-identified data on 46,520 patients. Adult patients diagnosed with heart failure were included, and characteristics were extracted within the first 24 hours of all admitted patients, and laboratory variables were measured during unit stay. The following data was extracted: demographic characteristics, vital signs, urine output (first 24 hours), comorbidities, and laboratory variables, and this data is freely available for researchers worldwide.

The extracted data contained 1177 records and 51 columns representing 50 features, and the attribute to be predicted (label) is the "outcome" field, that indicates the death or survival of the patient. It is composed of only two values, where the value 0 represents that the HF patient is valid and the value 1 represents the non-survived patient.

TABLE II
FEATURES WITH MOST MISSING VALUES

Data feature	Frequency of missing values
PCO2	294
PH	292
Basophils	259
Lactic acid	229
BMI	215
Creatine kinase	165
Lymphocyte	145
Neutrophils	144
Urine output	36
INR	20
PT	20
temperature	19
glucose	18
Diastolic blood pressure	16
Systolic blood pressure	16
SP O2	13
Respiratory rate	13
heart rate	13
Blood calcium	1

B. Data preprocessing

We encountered two main issues with the selected dataset. The dataset contains *too many null values* in certain columns and some rows, as it is obvious in Table II. The 'Data feature' column exhibited in Table II represents the extracted MIMIC III dataset feature columns. The data is also highly imbalanced (see Figure 1), and as the target class has an uneven distribution of observations. We have applied different preprocessing techniques to overcome these problems.

One of the approaches we used to remove the missing values was to remove the top columns with missing values. These values negatively affect the performance and accuracy of any machine learning algorithm. Four features that contain most null values are removed. Some features were dropped, including "group" and "ID". They are considered as insignificant features in our prediction problem. Another method for

[1]https://physionet.org/content/mimiciii/1.4/

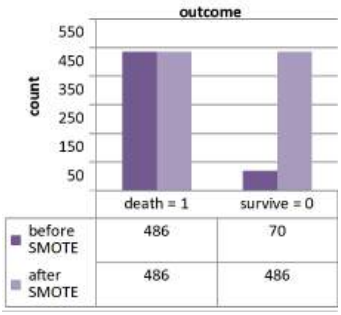


Fig. 1. Distribution of HF cases in train set before and after oversampling

fixing the missing values in rows, was to drop the rows which do not contain values for all significant columns. Eventually, we got a data set consisting of 44 features and 696 records. The numerical input data X has been standardized using the standard scaler method from `sklearn` package [2] to change the size of the distribution of numerical features so that the mean of the observed values is 0 and the standard deviation is 1.

The data preprocessing and modeling pipeline is shown in Figure 2. After specifying the set of inputs X and output Y, the distribution of the target value of the data distribution representing people who could not survive (1) versus those who survived (0) was examined. From Figure 1, we note that the number of values in category (0) equals 608, which represents 87.36%, while the number of values in category (1) equals 88, and this represents 12.64% of all data. This ratio is very variable. The dataset is divided into two subsets, which are 70% train set and 30% test set. This problem is called the training set imbalance problem, with 486 survived people and 70 died. By using the SMOTE method to achieve some balance by oversampling the minority class to create many data points that are in the majority class. As a result, the modified data set was 972.

A correlation analysis is performed to identify pairs of highly correlated columns, specifically with a correlation exceeds the threshold of 0.5. When such a correlation was found, the corresponding column was included in a set of related columns. Subsequently, this set of columns, comprising Lymphocyte, Neutrophils, INR, Chloride, MCV, PT, Creatinine, Blood Sodium, Anion Gap, Urea Nitrogen, Hematocrit, Bicarbonate, MCH, MCHC, and RBC—was excluded from the actual dataset, (see Figure 3). For further details on these features, readers are encouraged to refer to the original paper [5].

C. Predictive modeling and model evaluation

Mortality prediction is considered a classification problem (binary classification) since it has only two outcomes, whether the patient will die or survive. In this case, the classifier requires training data to understand the relationship between the input variables and the outcome. We selected four machine learning algorithms for the classification problem. Our objective was to train the model that can be used to build the

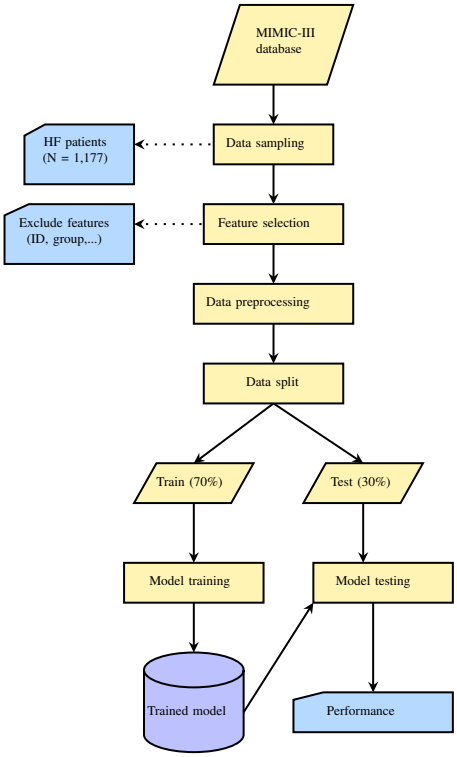


Fig. 2. ML modeling pipeline

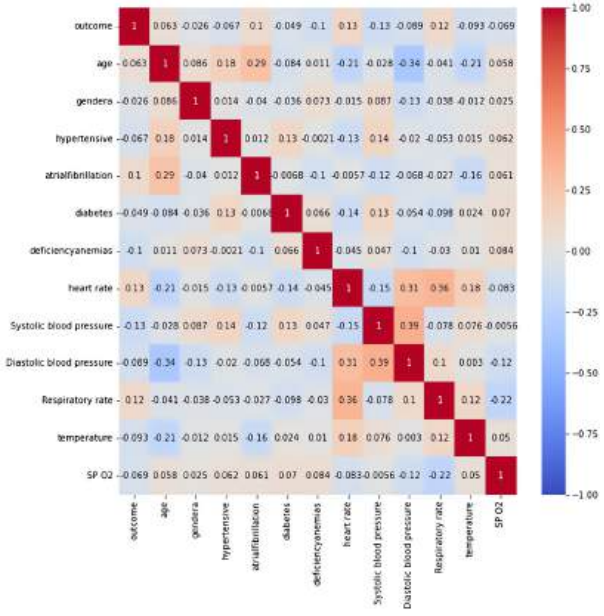


Fig. 3. Distribution of HF cases in train set before and after oversampling

best possible prediction system in terms of both accuracy and sensitivity. Note that we do not aim here to study the trade-off between accuracy and sensitive measures in ML. In the following paragraphs, we briefly describe the ML algorithms used in this study:

- *Logistic Regression Algorithm:* In logistic regression, the target variable is usually binary. This means that only data classified as 1 or 0 are included. The goal of the logistic regression algorithm is to find the best fit that is diagnostically meaningful to describe the relationship between the target and predictor variables [11]. The logistic regression algorithm is based on the linear regression model given in equation (1) below:

$$y = h_\theta(x) = \theta^T x \tag{1}$$

  Where $h_\theta(x)$ is the hypothesis on which the algorithm is based. θ is the regression coefficient, while x is a sample of the data. The output of the algorithm must be in the form of a probability value, so the values must be confined within $(0, 1)$, so we will use the *sigmoid* function, whose mathematical formula is given as:

$$h_\theta(x) = g(\theta^T) = \frac{1}{1 + e^{-\theta^T x}} \tag{2}$$

- *Support Vector Machine Algorithm:* The SVM model can be referred to as a "non-probabilistic classifier"; it constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier. SVM supervises the machine learning algorithm used for both classifications and regression problems [12]. Support vectors are the set of closest data points on the hyperplane, and they are points of great importance in the process of data classification because through them the best hyperplane is determined in the data separation process, and therefore removing them or changing their location requires another hyperplane again. The margin is the distance between the hyperplane and the nearest point of the dataset, so the greater the distance, the greater the probability of correctly classifying new data. The Linear SVM Classifier is a linear support vector machine classifier to find the ideal values of the straight line equation that separates this data as best as possible, i.e. by the largest margin distance between any point of the support vector and the hyperplane. Whereas the non-linear SVM Classifier is used to separate data that is not linearly separable by a line. The mathematical representation of the hyperplane in the SVM algorithm is $w^T x = 0$, where w and x are vectors and their product is the dot product, knowing that the vector w is the vector of the weights being trained and x is a sample of the training data. Once we have the values of the weights that represent the hyperplane, we can then use that plane to make predictions, defining the hypothesis function h as follows:

$$h_\theta(x) = \begin{cases} +1; 0 \leq w^T.x \\ -1; 0 > w^T.x \end{cases} \tag{3}$$

- *eXtreme Gradient Boosting Algorithm:* XGBoost is a supervised learning algorithm that implements a process called boosting to yield accurate models. It is at its core a decision tree boosting algorithm. Boosting refers to the ensemble learning technique of building many models sequentially, with each new model attempting to correct for the deficiencies in the previous model. In tree boosting, each new model that is added to the ensemble is a decision tree [6]. Instead of adjusting the weights of data points, gradient boosting focuses on the difference between the prediction and the ground truth.
- CatBoost Algorithm is also a kind of boosting family algorithm, which is known to perform better than XGBoost and Light GBM in terms of algorithm accuracy. Where innovative algorithms that automatically process categorical features into numerical features and then use those built-in features to take advantage of the connections between features, greatly enrich the feature dimension. Then, the rank-boosting method is used to avoid the noise points in the training set, and the deviation in the gradient estimation, and then solve the forecast offset problem [13].

As mentioned earlier, the study aimed to build a predictive model with high performance in accuracy and sensitivity as well, but the dataset that we obtained had many problems, so the preprocessing was essential to prepare the cleaned data and find the best algorithm that would be effective in classifying patients. Each ML model uses algorithms; each algorithm has a set of hyperparameters that were tuned to get the best accuracy and recall using the Grid Search technique. A set of variables was identified to be controlled to get the best performance of the algorithm, and these variables included the use of the oversampling technique to solve the problem of the imbalanced dataset by a Boolean value of TRUE/FALSE, deleting the attributes that are related to each other with a value that exceeds the threshold defined using feature correlation, and in addition to the threshold value, the number of columns with the largest number of nulls to be deleted. The variable k was defined as the number of deleted columns containing the largest null values (see Figure 4), the correlation threshold between the attributes, and whether to use the SMOTE method, as it cannot be applied to the test data until there is no data leak known as data leakage. To properly test the model's performance, the stratify option was used while splitting the data into a training data set and a test data set. This option considers the proportion of samples from each class during the stratification process [10]. There are various types of algorithms that can be applied to datasets. The following measures are calculated for performance analysis:

$$Recall(Sensitivity) = \frac{t_p}{t_p + f_p} \tag{4}$$

$$Accuracy = \frac{t_p + t_n}{n} \tag{5}$$

Here, n is the total number of instances, t_p is the number of true positives, t_n is the number of true negatives, and f_p is the number of false positives.

IV. Results and Discussion

Here, we present the modeling results and discuss the findings of our work.

The results presented in Tables III to VI illustrate the performance of various machine learning algorithms —including Logistic Regression (LR), Support Vector Machine (SVM), XGBoost, and CatBoost— in predicting mortality in the ICU for heart failure patients. The evaluation of these models under different configurations of column selection, SMOTE application, and threshold values provides insights into their effectiveness in this high-stakes environment.

Logistic Regression (Table III) achieved a maximum accuracy of 86% with a recall of 45% when no columns were dropped and SMOTE was not applied. While this model demonstrates reasonable accuracy, the recall figures indicate that it may struggle to identify all patients in immediate need of intervention, which can decrease the chance of catching potential health issues early on. The introduction of SMOTE and adjustments to threshold values led to a peak recall of 83% with a threshold of 0.3 after dropping 4 columns. This suggests that while SMOTE can improve sensitivity in detecting the condition of the patient, it may also introduce complexity that affects overall accuracy. Given the importance of identifying HF patients in the ICU, these results highlight a need for careful consideration when choosing models. While Support Vector Machine (Table IV) results show a maximum accuracy of 87.5% and a recall of 66% with 4 columns dropped and SMOTE applied. SVM's performance indicates its robustness in handling imbalanced datasets, which is critical in the context of heart failure patients where events may be infrequent yet crucial for timely interventions. The fluctuations in recall with different threshold settings suggest that while SVM can effectively manage decision boundaries, careful tuning is essential to ensure high sensitivity. XGBoost (Table V) emerged as the leading algorithm, achieving an accuracy of 91% and a recall of 77% when dropping 4 columns and using a threshold of 0.7. This model's capability to maintain high accuracy while enhancing recall signifies its potential in clinical settings, where understanding patient status is vital for decision-making. XGBoost's ensemble approach allows it to model complex interactions within the data effectively, making it particularly well-suited for the intricate patterns associated with heart failure.

The shadowed lines in Tables III to VI indicate the optimal model settings for each algorithm, which include the application of SMOTE, the chosen threshold value, and the number of columns dropped to enhance feature selection. These settings yielded the highest accuracy and recall, demonstrating the effectiveness of tailored preprocessing and hyperparameter tuning in improving model performance for critical decision-making in the ICU setting.

The Table VII summarizes the accuracy and recall performance metrics for the best models of Logistic Regression, Support Vector Machine, XGBoost, and CatBoost. Notably, XGBoost achieved the highest accuracy at 91% but the sensitivity was not high at 77%. On the other hand, CatBoost (see Tables V and VI) demonstrated competitive performance, achieving an accuracy of 90% and a recall of 81% with similar configurations as in XGBoost, highlighting the effectiveness of CatBoost in balancing these critical metrics for timely decision-making in the ICU setting. Its high recall indicates a strong ability to identify true positives, which is essential to

TABLE III
LR model performance in different preprocessing settings

Dropped columns (K)	SMOTE oversampling	Threshold	Accuracy after tuning	Recall after tuning
0	no	1	86%	45%
0	yes	1	78.29%	60%
1	yes	1	78%	55%
4	yes	1	81.34%	61.5%
5	yes	1	77.6%	61%
8	yes	1	72%	59.6%
4	yes	0.7	82.77%	57.7%
4	yes	0.3	**82%**	**83%**
1	yes	0.3	68%	75%

TABLE IV
SVM model performance in different preprocessing settings

Dropped columns (K)	SMOTE oversampling	Threshold	Accuracy after tuning	Recall after tuning
0	no	1	86%	55%
0	yes	1	**84%**	**68%**
1	yes	1	80%	50%
4	yes	1	**87.5%**	**66%**
5	yes	1	80%	65%
8	yes	1	81%	61%
4	yes	0.7	86.12%	62%
4	yes	0.3	70%	63%
1	yes	0.3	75%	67%

TABLE V
XGBoost model performance in different preprocessing settings

Dropped columns (K)	SMOTE oversampling	Threshold	Accuracy after tuning	Recall after tuning
0	no	1	89%	67%
0	yes	1	85%	67%
1	yes	1	86%	67%
4	yes	1	89.8%	69%
5	yes	1	83%	68%
8	yes	1	82%	64%
4	yes	0.7	**91%**	**77%**
4	yes	0.3	82%	63%
1	yes	0.3	77%	58%

TABLE VI
CatBoost model performance in different preprocessing settings

Dropped columns (K)	SMOTE oversampling	Threshold	Accuracy after tuning	Recall after tuning
0	no	1	88%	66%
0	yes	1	88%	73%
1	yes	1	83%	72%
4	yes	1	**90%**	**81%**
5	yes	1	83%	66%
8	yes	1	84%	66%
4	yes	0.7	80%	69%
4	yes	0.3	76%	58%
1	yes	0.3	75%	54%

TABLE VII
PERFORMANCE COMPARISON OF FOUR MACHINE LEARNING ALGORITHMS IN PREDICTING IN-HOSPITAL MORTALITY IN HEART FAILURE PATIENTS IN THE ICU

ML Algorithm	Accuracy	Recall
LR	82%	83%
SVM	84%	68%
XGBoost	91%	77%
CatBoost	**90%**	**81%**

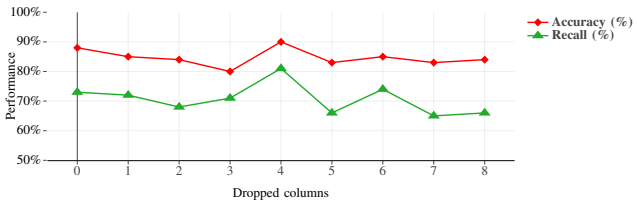


Fig. 4. Impact of Column Selection on CatBoost Model's performance

the ICU for ensuring that healthcare providers do not overlook patients who may be deteriorating. The model's robustness against overfitting and its effective handling of categorical features enhance its applicability in a clinical context, where data can often be noisy and complex.

The comparison between our study and the findings of Li et al. [14] (see Table VIII) highlights significant advancements achieved through our methodological approach. Utilizing the extracted data with all features from MIMIC III database, our study's application of CatBoost resulted in a marked improvement in accuracy (90% compared to 76%) and recall (81% versus 53%). The rigorous preprocessing and hyperparameter tuning employed in our study likely contributed to this enhanced performance, emphasizing the importance of tailored approaches in machine learning for critical care applications. The improved predictive capabilities can be essential for effective decision-making in the ICU, potentially leading to better patient outcomes.

TABLE VIII
PERFORMANCE COMPARISON OF OUR STUDY AGAINST LI ET AL. [14] (2021), SHOWING IMPROVED RESULTS

	Our study	Li et al. [14]
Database	MIMIC III	MIMIC III
Algorithm	CatBoost	XGBoost
Data preprocessing	Yes	No
Hyperparameter tuning	Yes	No
Accuracy	90%	76%
Recall	81%	53%

Despite the encouraging results, the study had several limitations. Beginning with the single dataset reliance while building the predictive model, although the MIMIC-III database is the largest in terms of HF-diagnosed patient data, may not reflect the diverse demographics and clinical situations found in various ICU settings. This raises concerns about the generalizability of our model's performance to other populations, as it was difficult to reach models with the best metrics. Additionally, our study lacks external validation of the CatBoost model across other HF mortality rate datasets, which is important for assessing the robustness of our predictive model. Furthermore, the feature selection process has inherent limitations; while it aims to enhance model performance, it may overlook important clinical variables that could play a big role in accurately predicting patient outcomes. Our analysis also highlights a sensitivity versus accuracy trade-off, as models optimized for high sensitivity may sacrifice some accuracy, for example in Table IV, potentially affecting clinical decision-making.

V. CONCLUSION AND OWN REFLECTIONS

In this study, we evaluated multiple ML algorithms to find the best algorithm for predicting mortality in heart failure patients. Our results underscore the importance of feature selection, hyperparameter tuning through Grid Search, and class balancing in optimizing model performance. The LR model achieved an accuracy of 86% without SMOTE, with a recall of 45%. While using SMOTE improved recall to 83% with a threshold of 0.3, the accuracy declined to 82%. This highlights the important trade-off between accuracy and recall in sensitive medical applications, where misclassification can have serious consequences. The application of SMOTE generally improved recall across models but often came at the cost of accuracy. This reinforces the need for a careful balance between these metrics, particularly in high-stakes environments like the ICU. The SVM model initially performed with lower metrics—86% accuracy and 55% recall without SMOTE. While applying SMOTE improved recall, it did not reach the effectiveness of the other models. However, the XGBoost model demonstrated strong performance, achieving an accuracy of 88% without SMOTE, which then slightly increased to 91% with SMOTE. However, its recall improved to 77% with a threshold of 0.7, showcasing the positive impact of class balancing too. This model proved particularly effective for our application, making it a strong candidate as it was used in the majority of related works for identifying high-risk patients who are more likely than other patients to experience mortality. The CatBoost model achieved an accuracy of 90% and a recall of 81% with optimal configurations. This model's strong performance highlights its effectiveness in handling categorical data and its ability to maintain high sensitivity, which is crucial for detecting true positives in a clinical setting. Overall, the findings highlight that while accuracy is important, maximizing recall is crucial in critical care scenarios. Although the CatBoost model with SMOTE emerged as the best choice for our specific application, achieving a balance between high recall and competitive accuracy, the interpretability of these models is vital for stakeholders in healthcare to understand how the model arrived at a specific conclusion like the survival and death of the patient. This study emphasizes the necessity of thorough preprocessing, strategic feature selection, and appropriate model tuning to optimize predictive performance

in healthcare settings, particularly for critical decision-making in the ICU for heart failure patients. In the future, we plan to examine neural networks and deep learning models to improve mortality rate prediction accuracy and sensitivity. Future studies should look at combining real-time data with additional variables to better monitor the risky states of HF patients before any deterioration. eXplainable Artificial Intelligence (XAI) techniques can provide new insights to understand the hidden links between features and outcomes, for instance, Shapely values for feature significance.

REFERENCES

[1] Kirkwood F Adams Jr et al. "Characteristics and outcomes of patients hospitalized for heart failure in the United States: rationale, design, and preliminary observations from the first 100,000 cases in the Acute Decompensated Heart Failure National Registry (ADHERE)". en. In: *Am Heart J* 149.2 (Feb. 2005), pp. 209–216.

[2] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[3] Romain Pirracchio et al. "Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): a population-based study". In: *The Lancet Respiratory Medicine* 3.1 (Jan. 2015), pp. 42–52. ISSN: 2213-2600. DOI: 10.1016/S2213-2600(14)70239-5. URL: https://doi.org/10.1016/S2213-2600(14)70239-5.

[4] Abdelbaset Saleh et al. "Comparison of the mortality prediction of different ICU scoring systems (APACHE II and III, SAPS II, and SOFA) in a single-center ICU subpopulation with acute respiratory distress syndrome". In: *Egyptian Journal of Chest Diseases and Tuberculosis* 64.4 (2015), pp. 843–848. ISSN: 0422-7638. DOI: https://doi.org/10.1016/j.ejcdt.2015.05.012. URL: https://www.sciencedirect.com/science/article/pii/S042276381530025X.

[5] Alistair E.W. Johnson et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific Data* 3.1 (May 2016), p. 160035. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35. URL: https://doi.org/10.1038/sdata.2016.35.

[6] Rory Mitchell and Eibe Frank. "Accelerating the XGBoost algorithm using GPU computing". In: *PeerJ Computer Science* 3 (2017), e127.

[7] Lina Zhou et al. "Machine learning on big data: Opportunities and challenges". In: *Neurocomputing* 237 (2017), pp. 350–361. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.01.026. URL: https://www.sciencedirect.com/science/article/pii/S0925231217300577.

[8] Raheleh Davoodi and Mohammad Hassan Moradi. "Mortality prediction in intensive care units (ICUs) using a deep rule-based fuzzy classifier". In: *Journal of Biomedical Informatics* 79 (2018), pp. 48–59. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2018.02.008. URL: https://www.sciencedirect.com/science/article/pii/S1532046418300273.

[9] Wojciech Weigl et al. "ICU mortality and variables associated with ICU survival in Poland: A nationwide database study". In: *European Journal of Anaesthesiology — EJA* 35.12 (2018). ISSN: 0265-0215. URL: https://journals.lww.com/ejanaesthesiology/fulltext/2018/12000/icu_mortality_and_variables_associated_with_icu.8.aspx.

[10] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. "Tunability: Importance of Hyperparameters of Machine Learning Algorithms". In: *Journal of Machine Learning Research* 20.53 (2019), pp. 1–32. URL: http://jmlr.org/papers/v20/18-444.html.

[11] Changsheng Zhu, Christian Uwa Idemudia, and Wenfang Feng. "Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques". In: *Informatics in Medicine Unlocked* 17 (2019), p. 100179. ISSN: 2352-9148. DOI: https://doi.org/10.1016/j.imu.2019.100179. URL: https://www.sciencedirect.com/science/article/pii/S2352914819300139.

[12] Andrey Zahariev et al. "Debt management evaluation through Support Vector Machines: on the example of Italy and Greece". In: *Entrepreneurship and Sustainability Issues* 7 (Mar. 2020), pp. 2382–2393. DOI: 10.9770/jesi.2020.7.3(61).

[13] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. "A comparative analysis of gradient boosting algorithms". In: *Artificial Intelligence Review* 54.3 (Mar. 2021), pp. 1937–1967. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09896-5. URL: https://doi.org/10.1007/s10462-020-09896-5.

[14] Fuhai Li et al. "Prediction model of in-hospital mortality in intensive care unit patients with heart failure: machine learning-based, retrospective analysis of the MIMIC-III database". en. In: *BMJ Open* 11.7 (July 2021), e044779.

[15] Min Hyuk Choi et al. "Mortality prediction of patients in intensive care units using machine learning algorithms based on electronic health records". en. In: *Sci Rep* 12.1 (May 2022), p. 7180.

[16] Jili Li et al. "Predicting Mortality in Intensive Care Unit Patients With Heart Failure Using an Interpretable Machine Learning Model: Retrospective Cohort Study". en. In: *J Med Internet Res* 24.8 (Aug. 2022), e38082.

[17] Wei Yang et al. "Mortality prediction among ICU inpatients based on MIMIC-III database results from the conditional medical generative adversarial network". In: *Heliyon* 9.2 (2023), e13200. ISSN: 2405-8440. DOI: https://doi.org/10.1016/j.heliyon.2023.e13200. URL: https://www.sciencedirect.com/science/article/pii/S2405844023004073.

WAVE HEIGHT INVERSION ALGORITHM BASED ON MULTIMODAL DATA FUSION AND ATTENTION MECHANISM

Ma Ruidi [1,2]
QingDao, China
maruidi@ncs.mnr.gov.cn

Hu Wei [1,2]
QingDao, China

Zhao Chuanting * [1,2]
QingDao, China

Wanglan [1,2]
QingDao, Chin

Jiangfan [1,2]
QingDao, China

Yubo [1,2]
QingDao, China

Tian Haoqiang [1,2]
QingDao, China

Song Yanchen [1,2]
QingDao, Chin

Geyong [1,2]
QingDao, China

Ren Dianjun [1,2]
QingDao, China

1 Shandong Provincial Key Laboratory of Marine Ecological Environment and Disaster Prevention and Reduction
2 North China Sea Marine Forecasting and Hazard Mitigation Center of MNR

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899453

***Abstract*—A wave height inversion algorithm that integrates multimodal data and attention mechanism is proposed by combining shore based monitoring videos and CNN, which can effectively utilize monitoring videos to automatically detect wave height and update wave monitoring technology. This algorithm designs a parallel depth separable convolution module with image denoising function, reducing the algorithm's dependence on high-quality data samples; Propose a multimodal data fusion module based on self-attention mechanism to compensate for the problem that single video image information cannot effectively handle the complex changes of ocean waves. Aiming at the problems of uneven quality of existing datasets and lack of introduction of meteorological data, build the "XiaoMai Island Wave Level Annotation Dataset". The results show it achieves a Top-1 of 91.3% on Maidao dataset, which is 5.1% higher. This indicates that the algorithm can effectively detect wave levels and improve algorithm performance.**

Keywords—computer vision, Multimodal data, Wave height inversion, Self-attention mechanism

I. INTRODUCTION

Waves are an ocean's common physical phenomenon, containing immense power[1]. According to the China Marine Disaster Bulletin, there were 5 wave disasters in China in 2022, causing direct economic losses of 24.1177 million RMB and 9 deaths or missing persons[2]. People are paying increasing attention to accurate monitoring and timely forecasting of nearshore waves, and have put forward higher and higher requirements [3-5].

Hao et al. proposed a Wave CNNS algorithm model [6] to address the rich texture features and periodic variations of ocean wave images; Gao Libin et al. proposed a deep learning model based on LSTM, which is mainly used for wave forecasting in the Taiwan Strait and surrounding waters [7]; Song Wei et al. proposed a wave height automatic detection method for nearshore wave videos based on multi-layer local perception neural networks [8]; Android proposes a wave high-level classification network based on 3D convolutional neural network to address the complex and variable nearshore wave conditions, effectively achieving accurate prediction of wave parameters in the waters near Dalian [9]; Gao Yafei et al. proposed a method for inverting the effective wave height of ocean waves based on multi-layer perceptrons and SAR parameters [10]; Yuan Chaowen et al. proposed a study on SAR image wave parameter inversion based on quasi linear approximation method to address the problem of easily amplifying non wave wave texture information in the low wavenumber domain during the inversion process [11]. The above algorithms have the following problems: (1) SAR image based algorithms are difficult to obtain SAR images and have high costs; (2) The algorithm based on surveillance video runs slowly and fails to effectively solve the problem of low resolution of surveillance video affecting algorithm performance; (3) The formation of ocean waves is associated with multiple meteorological factors, and most of the above algorithms have not introduced meteorological data or studied their correlation.

In response to the above issues, this article proposes a wave height inversion model based on convolutional neural networks, which uses shore based monitoring videos for wave element measurement. A parallel depth separable convolution module is proposed to reduce the algorithm's dependence on high-resolution data samples and solve the problem of low resolution in monitoring videos [12-17]. The algorithm introduces meteorological data and combines it with a multimodal segmentation attention feature fusion module to capture complementary information between multimodal data, improving the algorithm's ability to handle complex changes in ocean waves. This paper proposes the "Xiaomai Island Wave Image Wave Level Annotation Dataset" to address the issues of long shooting time, uneven quality, and lack of introduction of meteorological data in the existing dataset [24-27] .

Fundings: North China Sea Bureau Marine Science and Technology Project(202405), and Shandong Provincial Key Laboratory of Marine Ecological Environment and Disaster Prevention and Reduction(202309)

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

II. OVERALL

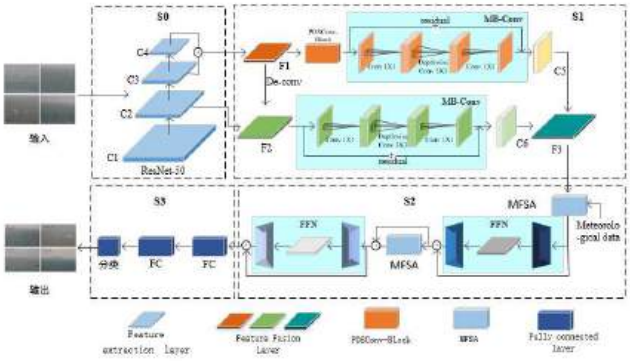


Fig. 1. Network architecture diagram

The entire network adopts a multi-level layout structure, as shown in Figure 1, consisting of four stages: S0 feature extraction, S1 feature enhancement, S2 multimodal data fusion, and S3 classification. The first stage S0 is used to extract feature information, generate feature maps, and in the process of generating C4 from C3, a mixed dilation convolution is used instead of a pooling layer to increase the receptive field and obtain richer feature information; S1 consists of MB-Conv and PDSC. PDSC is a parallel deep separable convolution module that can compress the receptive field, allowing features to have a global receptive field. It also models high-order statistical data to achieve image denoising and reduce the algorithm's dependence on high-resolution data samples; In the S2 stage, the multimodal data fusion module MF attention and the multi-layer feedforward neural network FFN are alternately used. MF attention utilizes self attention mechanism to introduce meteorological factor data into the algorithm, and FFN enhances feature expression to obtain the final feature map for S3 classification task.

A. Parallel Depth Separable Convolution Module

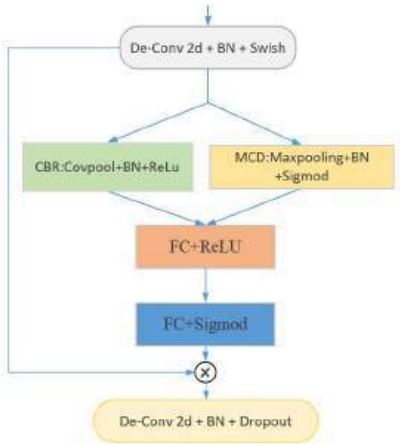


Fig. 2. Parallel deep separable convolution module(PDSC)

When the wave level is relatively small, the lines of the waves are not clear, especially in poor weather conditions, and the quality of surveillance videos may be poor. These factors can make it difficult for algorithms to extract useful feature information. This article designs a simple parallel deep separable convolution module(PDSC) that combines spatial attention mechanism with channel attention mechanism using a parallel structure. It can capture global second-order statistical information along the channel dimension and enhance the network's nonlinear modeling ability; It can also enable the model to adaptively learn attention weights for different regions, allowing it to focus more on important image areas and ignore unimportant areas. This module has the function of image denoising, reducing the algorithm's dependence on high-quality data samples, and can be easily inserted into existing network architectures to further improve its performance with minimal overhead.

Figure 2 shows the structure diagram of PDSC, which adopts two steps of Squeeze Excitation. Firstly, deep convolution operation is performed to adjust the size of the feature map. Then perform Squeeze operation, including MCD and CBR parts. MCD compresses the feature map along the channel dimension, while CBR models second-order statistical information along the channel dimension of the input tensor. By calculating the correlation between channels, the covariance matrix is obtained, resulting in the final attention enhanced feature map. This enhanced feature will be used as input for subsequent network layers to suppress noise and irrelevant information while retaining key information.

B. Multimodal Data Fusion Module

In complex marine environments, a single sensor information cannot effectively handle changes in the scene. This article proposes a multimodal segmentation attention feature fusion module based on segmentation attention mechanism, which introduces meteorological data into the algorithm and can effectively fuse different modal information, thereby improving the performance of the model.

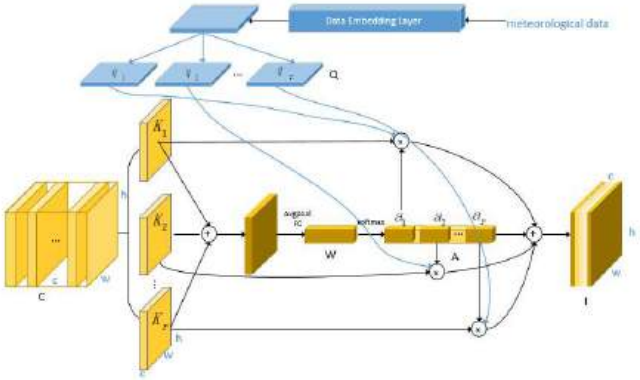


Fig. 3. Multimodal Data Fusion Module

Data Embedding Layer: Firstly, a fully connected layer is used to convert the original input $\chi \in \mathbb{R}^{T\times N\times C}$ into a high-dimensional representation $\chi_{data} \in \mathbb{R}^{T\times N\times d}$. Due to the similar sub processes between similar phases of meteorological data in different periods, this paper uses fast Fourier functions to extract the inherent periodicity of this sequence and obtain time period embeddings $\chi_t \in \mathbb{R}^{T\times d}$. The output of the final

data embedding layer is the result of adding the embedding vectors mentioned above: $\chi_{emb} = \chi_{data} + \chi_t$.

As shown in Figure 3, C is a feature block for ocean wave images and Q is a feature block for meteorological data, both consisting of r feature maps, where h, w, and c represent the height, width, and number of channels of the feature blocks, respectively. Feature block C is first segmented along the channel dimension to generate r sets of individual feature maps(K_1, K_2, …, K_r), where the number of channels is equal to c. Then, the r sets of features are added element by element and subjected to two-dimensional adaptive average pooling (Avgpool) and fully connected (FC) operations in sequence to generate a weighted vector W, which represents the importance between different channels of the cascaded feature map. Next, perform the softmax operation to obtain A, which is composed of r sets of channel attention weights a_1, a_2, …, a_r form. Finally, the feature I is obtained by multiplying K_r with the corresponding vectors a_r and q_r , then adding the elements. It is a feature mapping based on hierarchical perception.

III. EXPERIMENT

A. Dataste

The data used in this article were provided by the North Sea Forecasting Center of the State Oceanic Administration. Based on the observation and image data of the waters around Xiaomai Island, a "Xiaomai Island Wave Image Wave Level Annotation Dataset" is proposed. The dataset consists of three parts: image data for annotating wave levels, meteorological factor data (average wind speed, wind direction corresponding to average wind speed, air pressure, relative humidity), and Xiaomai Island wave level classification standards(TableI).

TABLE I. WAVE LEVEL

Wave level	*Hs*	*Wave level*	*Hs*
0	$0.1 \le H_s < 0.3$	4	$0.7 \le H_s < 1$
1	$0.3 \le H_s < 0.4$	5	$1 \le H_s < 1.25$
2	$0.4 \le H_s < 0.5$	6	$1.25 \le H_s < 1.8$
3	$0.5 \le H_s < 0.7$	7	$1.8 \le H_s \le 2.5$

B. Ablation experiment

TABLE II. EXPERIMENTAL RESULTS ON THE WAVE LEVEL ANNOTATION DATASET OF XIAOMAI ISLAND WAVE IMAGES

PDSC	*MFSA*	*Top-1(%)*
/	/	86.2
✓	/	88.5
/	✓	89.2
✓	✓	91.3

In TableII, " ✓" indicates the addition of this module, and "/" indicates the absence of this module. The first line shows that the performance of the algorithm before improvement was 86.2%, and with the addition of only PDSC algorithm Top-1, it was 88.5%, an improvement of 2.3%; Only incorporating the MFSA algorithm Top-1 resulted in an improvement of 89.2%, representing a 3% increase; Adding Top-1 algorithm to both modules can achieve 91.3%, an increase of 5.1%. This indicates that our proposed module can leverage its advantages to improve the shortcomings of the algorithm and greatly enhance its performance.

C. Parallel depth separable convolution module

Firstly, experiments were conducted on the arrangement of ResNet-50 and MB-Conv, as shown in Figures 4-a, 4-b, 4-c, and 4-d. According to Table III, the parallel structure showed a better performance of 86.2%, and the accuracy increased by 0.6% with the increase of modules. However, the computational complexity also increased accordingly. As algorithm speed is also an important measure of algorithm performance, Figure 4-c structure was chosen.

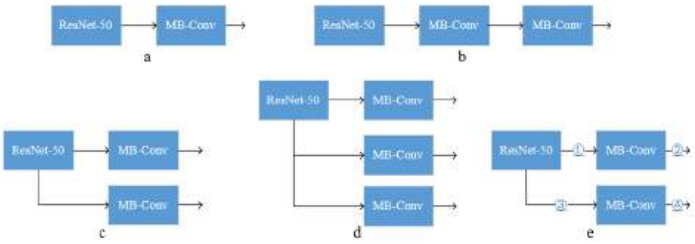


Fig. 4. Five different insertion methods for different positions

In order to find the most suitable placement position for the module plug and play, multiple placement positions have been designed, as shown in Figure 4-e. Where ① * represents placing two PDSC blocks at position ①, and ①+② represents placing one block each at positions ① and ②. According to Table III, in the experiment where one piece is placed, the structure with the same ①/③ has the best effect; The superposition of modules can improve the accuracy of the algorithm, but the improvement is not high and the increase in modules will inevitably lead to an increase in computational complexity. Taking into account both algorithm accuracy and speed, when selecting positions ①/③, the Top-1 accuracy is 88.5%, which is 2.3% higher than the previous 4-c. This indicates that adding this module after early convolution operations to extract image features is more conducive to high-order statistical modeling of the overall image, which can significantly improve algorithm performance.

TABLE III. EXPERIMENTAL RESULTS FROM DIFFERENT LOCATIONS

Location	*Top-1(%)*	*Location*	*Top-1(%)*
ResNet-50	82.3	②/④	86.8
9-a	83.6	①*	88.7
9-c	86.2	①+②	86.7
9-b	85.9	①+③	89.0
9-d	86.5	①+④/③+②	88.9
①/③	88.5	④+②	85.3

D. Multi modal cross attention feature fusion module

The formation and variation of ocean waves are influenced by multiple meteorological factors such as wind, temperature, and pressure, and accurate and effective inversion of wave height cannot be achieved solely through video images. This article proposes a multimodal data fusion module based on segmentation attention mechanism, which uses a data

embedding layer to process meteorological data and extract periodic features; Using segmentation attention to process multimodal features, the experimental results are shown in Table IV. After adding MFSA, the algorithm achieved a Top-1 accuracy of 89.2%, which is 3% higher than before improvement. This is because waves are formed under the drive of sea surface winds. There is a correlation between ocean waves and meteorological factors, and temperature and humidity also affect sea surface visibility, which in turn affects the video imaging effect of waves. In addition, the experimental results demonstrate that the feature fusion module proposed in this paper can effectively fuse and complement multimodal features.

TABLE IV. EXPERIMENTAL RESULTS OF MULTIMODAL SEGMENTATION ATTENTION FEATURE FUSION

MFSA		Top-1(%)
Data Embedding	**Multimodal data fusion**	
/	/	86.2
✓	/	87.1
/	✓	87.5
✓	✓	89.2

E. Analysis of results on public datasets

TABLE V. EXPERIMENTAL RESULTS ON IMAGENET-1K DATASET

Models	Top-1（%）	Years
Internlmage-DCNv3-G (M3l Pre-training)[18]	90.1	2022
DaViT-G[19]	90.4	2022
CoCa[20]	90.6	2022
RevCol-H[21]	90.0	2023
ONE-PEACE[22]	89.8	2023
CoAtNet[23]	88.56	2021
Ours	91.3	/

To further evaluate the superior performance of the MFSAM algorithm, experiments were conducted on the ImageNet-1K dataset and compared with advanced detection methods, achieving the best performance. The experimental results are shown in Table V. Under similar experimental conditions, the algorithm model proposed in this paper achieved a Top-1 accuracy of 91.3%, which is 2.74% higher than before the improvement. And it is also superior to some ViT variant algorithms, indicating that this algorithm has good detection ability, higher Top-1 accuracy and robustness.

F. Model testing results in different scenarios

To verify the inversion performance of the algorithm in complex weather scenarios, it was tested on a randomly divided test set, and the results are shown in Figure 5. As shown in the figure, in the real environment of waves, there are situations such as foggy weather, dark light, small samples, and ship influence. The algorithm can still accurately classify in different scenarios and achieve high accuracy, indicating that the robustness and generalization of our method are good.

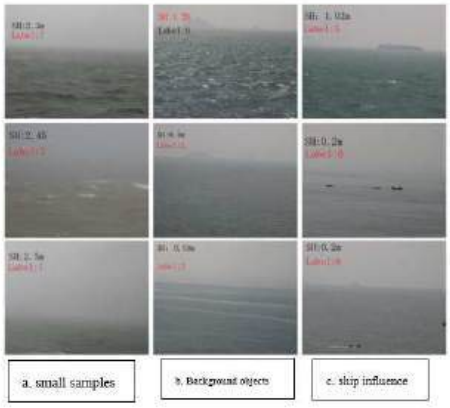


Fig. 5. Model testing results in different scenarios

IV. CONCLUSION

This article proposes a wave height inversion algorithm based on multimodal data fusion and segmentation attention mechanism. This algorithm extracts the feature information of waves in nearshore videos through convolutional neural networks and automatically detects the wave height, effectively compensating for the incompleteness of manually designed features; To address the issue of low resolution in surveillance videos, this algorithm combines channel attention mechanism with depthwise separable convolution and proposes a parallel depthwise separable convolution module, which can effectively improve the algorithm's processing capability for low resolution videos; In response to the problem that a single video image information cannot effectively handle the complex changes of ocean waves, this paper introduces meteorological factor data into the algorithm and proposes a multimodal segmentation attention feature fusion module combined with segmentation attention mechanism. This module can effectively combine information from different modalities, thereby improving the algorithm's ability to handle the complex changes of ocean waves; In response to the problems of long shooting time, uneven quality, and lack of meteorological data in the existing dataset, this paper proposes the "Wheat Island Wave Image Wave Level Annotation Dataset", which includes image data annotated with wave levels, meteorological factor data, and wheat island wave level classification standards, which can effectively support algorithm training and evaluation. The experimental results show that the Top-1 algorithm proposed in this paper can reach 91.3%, which is 5.1% higher than the original algorithm. These results indicate that our proposed module can leverage its advantages, improve its shortcomings, and significantly enhance algorithm performance.

ACKNOWLEDGMENT

Thanks to the North China Sea Marine Forecasting and Hazard Mitigation Center of MNR, the North China Sea Bureau Marine Science and Technology Project(202405), and Shandong Provincial Key Laboratory of Marine Ecological Environment and Disaster Prevention and Reduction(202309) for their support.

REFERENCES

[1] Yu Haitao, Tang Zeyan, Wei Yongliang, etc Advanced Synthetic Aperture Radar Wave Pattern Algorithm for Spaceborne Applications and Verification of Its Inverted Data Accuracy [J] Marine Science, 2022, 46 (9): 1-11 DOI: 10.11759/hykx20210712001.

[2] 2022 China Marine Disaster Bulletin (excerpt) [N] China Natural Resources News, April 14, 2023 (005) DO I:10.28291/n.cnki.ngtzy.2023.001129.

[3] Wang Zhiyong, Wang Weili, Hu Wei, etc Research on the Characteristics of Wave Elements in the Coastal Waters of Qingdao [J] Journal of Marine Technology, 2021, 40 (02): 61-68.

[4] Du Zhaojun, Wang Yanping, Mao Xinyan Characteristics analysis of storm surges and wave disasters in Shandong Ocean Ranch over the past 30 years based on ADCIRC-SWAN coupling model [J] Marine Science, 2023, 47 (6): 1-11 DOI: 10.11759/hykx20200919001.

[5] Gao Song, Zhong Shan, Li Yaru, etc Research on Comprehensive Risk Assessment of Marine Natural Disasters in Shandong Province [J] Marine Science, 2018, 42 (9): 55-63 DOI: 10.11759/hykx20180131001.

[6] Zheng Zongsheng, Hao Jianbo, Huang Dongmei, etc Deep learning based video monitoring of nearshore wave levels [J] Marine Environmental Science, 2017,36 (06): 934-940. DOI: 10.13634/j.cnki.mes. 2017.06.22.

[7] Gao Libin Research on Wave Forecasting in the Taiwan Strait and Surrounding Waters Based on Deep Learning [D] Fujian Agriculture and Forestry University, 2019.

[8] Song Wei, Zhou Xu, Bi Fan, etc Automatic detection of wave height in nearshore wave videos [J] Chinese Journal of Image and Graphics, 2020,25 (03): 507-519.

[9] Android Research on wave height inversion algorithm based on 3D convolutional network [D] Dalian University of Technology, 2021.DOI: 10.26991/d.cnki.gdlulu.2021.000224.

[10] Gao Yafei, Wang Yunhua, Zhang Yanmin, etc Effective wave height inversion method based on multi-layer perceptron and SAR parameters [J] Journal of Ocean University of China (Natural Science Edition), 2024, 54 (02): 121-133. DOI: 10.16441/j.cnki. hdxb. 20220489.

[11] Yuan Chaowen, Zhang Yanmin, Jiang Wenzheng, etc Research on SAR image wave parameter inversion based on quasi linear approximation method [J] Journal of Ocean University of China (Natural Science Edition), 2024, 54 (01): 144-155. DOI: 10.16441/j.cnki. hdxb. 20220442.

[12] GB/T 42176-2022, Wave Grade [S].

[13] Wu Shuping, Wang Juanjuan, Xing Chuang, etc Analysis of catastrophic waves in China's coastal waters in 2022 and prediction for 2023 [J] Ocean forecast, 2023, 40 (04): 1-9.

[14] Li Wenbo, Li Rui, Wang Bin, etc Evaluation of Forecasting Levels of Different Numerical Wave Forecasting Products in Bohai Sea and Yellow Sea [J] Ocean forecast, 2023, 40 (04): 10-21.

[15] Xu Teng, Cai Jingze, Li Rui, etc Wave level annotated images near the coast of Qingdao [J] Ocean Forecast, 2021,38 (05): 76-80.

[16] Gao Song, Xu Jiangling, Liu Guiyan, etc Machine Learning based Integrated Forecasting Method for Coastal Sea Fog in Qingdao City [J] Marine Science, 2021, 45 (03): 33-42

[17] Qin Yifan, Luo Feng, Zhang Jie, etc Research on Deep Learning Models for Predicting Significant Wave Height [J] Ocean Bulletin, 2024, 43 (03): 382-390.

[18] Wang W, Dai J, Chen Z, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions[C]// Conference on Computer Vision and Pattern Recognition. 2023: 14408-14419.

[19] Ding M, Xiao B, Codella N, et al. Davit: Dual attention vision transformers[C]//European conference on computer vision. Cham: Springer Nature Switzerland, 2022: 74-92.

[20] Yu J, Wang Z, Vasudevan V, et al. Coca: Contrastive captioners are image-text foundation models[J]. arXiv preprint arXiv:2205.01917, 2022.

[21] Cai Y, Zhou Y, Han Q, et al. Reversible column networks[J]. arXiv preprint arXiv:2212.11696, 2022.

[22] Wang P, Wang S, Lin J, et al. One-peace: Exploring one general representation model toward unlimited modalities[J]. arXiv preprint arXiv:2305.11172, 2023.

[23] Dai Z, Liu H, Le Q V, et al. Coatnet: Marrying convolution and attention for all data sizes[J]. Advances in neural information processing systems, 2021, 34: 3965-3977.

[24] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]// Conference on Computer Vision and Pattern Recognition. 2018: 4510-4520.

[25] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions[J]. arXiv preprint arXiv:1511.07122, 2015.

[26] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[27] Wilson E B, Hilferty M M. The distribution of chi-square[J]. Proceedings of the National Academy of Sciences, 1931, 17(12): 684-688.

Towards a Conversational Invoice Issuance LLM-based Agent

1st Runze Nie
Dept. AI of Research Institute
Aisino Co.,Ltd.
Beijing, China
nierunze@aisino.com

2nd Hao Wu
Dept. of Artificial Intelligence
Aisino Co.,Ltd.
Beijing, China
wuhao01@aisino.com

3rd Lan Ma
Dept. of Artificial Intelligence
Aisino Co.,Ltd.
Beijing, China
malan@aisino.com

4th Zhenyu Liu
Dept. of Artificial Intelligence
Aisino Co.,Ltd.
Beijing, China
liuzhenyu@aisino.com

5th Zhigang Wang
Dept. of Artificial Intelligence
Aisino Co.,Ltd.
Beijing, China
wangzhigang@aisino.com

6th Ping Zhang
Dept. of Artificial Intelligence
Aisino Co.,Ltd.
Beijing, China
zhangping@aisino.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899737

***Abstract*—The traditional invoice issuance process within tax administration is labor-intensive and prone to errors, necessitating a shift towards digitalization. Despite the advent of digital invoicing systems that streamline invoice generation and automate rule-based audits, integration with existing financial accounting systems remains a challenge. Particularly in the hospitality and bookkeeping sectors, the adoption of these systems is hindered by the lack of standardized software, high costs, and the absence of technical expertise among small and micro enterprises.**

The integration of digital invoicing systems with diverse financial software presents significant barriers to uniform adaptation. Furthermore, the complexity of tax regulations and the dynamic nature of tax categories require advanced understanding beyond the capabilities of standard Large Language Models (LLMs). The need for a specialized system that can comprehend finance and tax contexts, securely handle sensitive information, and adapt to user interactions is paramount.

This paper introduces an autonomous agent based on a finance and tax-specific Large Language Model (LLM) designed to address the aforementioned challenges. The system includes a Specialized Training Framework to enhance domain comprehension, a Hierarchical Memory Architecture for dynamic user interaction, and a Tax Domain Security Module to ensure compliance with tax regulations. The proposed agent aims to improve the efficiency and accuracy of the invoice issuance process, providing a robust solution for tax administration in the digital era.

***Index Terms*—Large Language Models, Autonomous Agents, E-Invoice System, Natural Language Processing**

I. Introduction

In the realm of tax administration, the issuance of invoices serves as a critical component in the documentation and tracking of financial transactions. The process of invoice issuance is a pivotal yet traditionally labor-intensive aspect. The industry has long been conducting intelligent research related to invoice recognition and entry [1]. The conventional approach to invoice generation has been heavily dependent on manual intervention, which involves a multitude of human resources dedicated to data entry and verification. This method, while ensuring a certain level of control, is inherently slow and susceptible to human error, thereby compromising the efficiency and accuracy of the tax system. The reliance on manual processes not only prolongs the invoice issuance timeline but also increases the likelihood of discrepancies, which can lead to subsequent complications in tax audits and financial reconciliations. This inefficiency underscores the necessity for a more streamlined and automated approach to invoice issuance within the tax system.

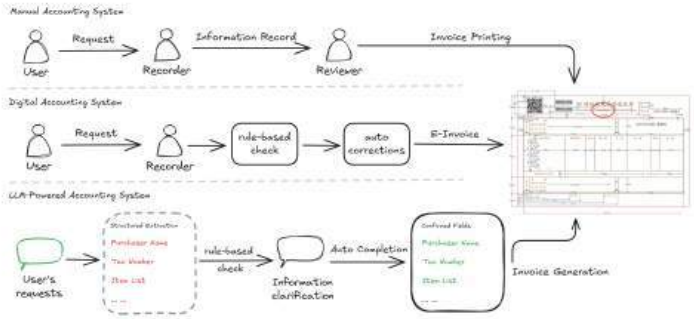


Fig. 1. The Concept of Our Conversational Invoice Issuance System

With the proliferation of digital invoicing systems, the process of invoice issuance has been significantly streamlined, encompassing the automation of invoice generation based on financial records and the implementation of automated rule-based audits [2]. However, the realization of this automated process is contingent upon the integration of these digital invoicing systems with a company's financial accounting systems. The challenge lies in the fact that financial systems from various software companies are not easily standardized for adaptation, presenting a barrier to uniform integration. This disparity highlights the need for more accessible and adaptable invoicing solutions that can cater to the diverse needs of different business scales and operational capacities.

In the hospitality sector, particularly in catering, and the bookkeeping industry, the adoption rate of digital invoicing systems remains relatively low. A multitude of products from different suppliers prevents enterprises from connecting data through a centralized system [3]. High procurement and R&D costs, lack of digital intelligent skills training, and data security concerns often result in the use of only those products that are widely adopted and relatively easy to operate. For small and micro enterprises, or individual business owners, the inability to bear the costs associated with advanced ERP, CRM, OA, and HR systems, and even the reluctance to invest in basic restaurant SaaS systems, contributes to the low coverage of digital invoicing systems. The challenge is further exacerbated by the fact that these businesses may not have the technical expertise or the financial resources to integrate such systems with their existing financial accounting infrastructure.

Agent technology has been a focal point of research in both academia and industry for an extended period. With the rapid progression of artificial intelligence, Large Language Models (LLMs) have emerged as a significant force. Characterized by their extensive parameter scale, profound linguistic knowledge, and superior generative abilities, LLMs have shown remarkable performance across a multitude of domains [4]. These models are capable of sophisticated reasoning, which positions them as pivotal components in the development of autonomous agents [5]. Such agents are designed to emulate human-like decision-making capabilities. Owing to their pre-training on extensive textual datasets, LLM-based agents have a comprehensive and deep internal knowledge base. Remarkably, even in the absence of domain-specific training data, these agents display high adaptability. They harness their profound understanding and processing of a wide array of information to function effectively [6].

Additionally, agents based on Large Language Models (LLMs) offer natural language interfaces, facilitating flexible and intuitive interactions with users, thereby enhancing the system's interpretability. [7] In the context of electronic tax systems, the multitude of tax categories, intricate calculations, and the dynamic complexity of tax regulations pose significant challenges for pre-trained models to comprehend the pertinent business contexts fully. Moreover, the stringent requirements of tax risk management imply that standard LLMs cannot be directly applied to electronic tax scenarios without undergoing further customization.

To tackle these challenges, this paper introduces an autonomous agent leveraging a finance and tax-specific Large Language Model (LLM), as depicted in Figure 1. The proposed system encompasses several key components:

(1) Specialized Training Framework: This framework is engineered to bolster the model's comprehension of finance and tax-related business contexts and to refine its precision in discerning domain-specific jargon.

(2) Hierarchical Memory Architecture: The agent's memory is architected to comprise both long-term storage and the capacity to assimilate external data, optimized for dynamic user interactions.

(3) Tax Domain Security Module: Incorporating a command intervention mechanism, this module is designed to detect and mitigate the presence of sensitive terms during interactions. It also intelligently directs users towards adherence to pertinent tax statutes and regulations.

II. Preliminary

Large pre-trained models are typically based on deep learning architectures, such as Transformers. By being pre-trained on large-scale datasets, they can learn a wide range of linguistic features. The pre-training process involves self-supervised learning on massive amounts of unlabeled data, which allows the model to capture complex patterns and semantic relationships, thereby providing a strong initial state for downstream tasks.

Large language models, often grounded in deep learning architectures as decoder-only Transformers, are capable of learning a broad spectrum of linguistic features. This capability arises from their pre-training on extensive datasets, which employs self-supervised learning techniques on vast quantities of unlabeled data. This process enables the models to discern complex patterns and semantic relationships, thereby establishing a robust foundation for subsequent downstream tasks. Through pre-training on large-scale textual data, these models accumulate substantial language expertise. Subsequently, LLMs are fine-tuned to align with human values and adapting to specific tasks, including but not limited to text classification, question-answering systems, and machine translation.

Autonomous agents based on LLMs demonstrate a strong ability to perform diverse tasks by fully leveraging the reasoning and judgment capabilities of LLMs. A substantial amount of research has explored and constructed numerous highly promising agent architectures. The core strategy involves integrating crucial auxiliary mechanisms into large-scale language models, such as memory systems and planning abilities, with the aim of providing the core control unit, the LLM, with human-like working and decision-making capabilities, thereby enabling it to efficiently execute complex tasks.

III. Methodology

The autonomous agent architecture, as depicted in Figure 2, encompasses several critical components. Initially, a pre-processing safety verification module classifies user queries to filter out inputs containing negative or harmful information. Subsequently, the LLM, in conjunction with a memory component, extracts relevant information from the input query. A post-processing verification module then audits the output content. If key fields are missing, the module provides feedback to the user; otherwise, the automated invoicing process is executed. This section begins by introducing the methodology for constructing the data used in the Supervised Fine-Tuning (SFT) training of the autonomous agent. It proceeds to describe the design of the agent's memory component and concludes by outlining the design method for the safety verification module, which is specifically tailored to the tax domain.

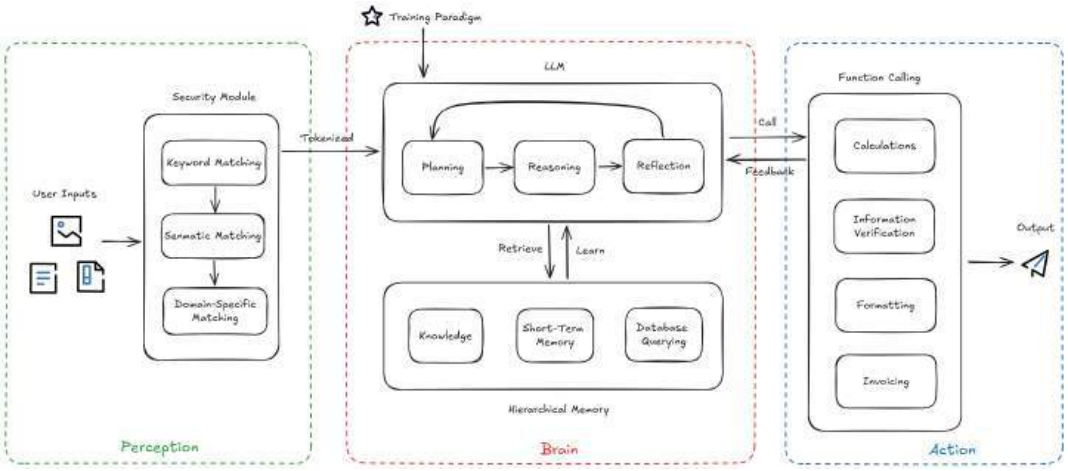


Fig. 2. The Overall Architecture of Our Proposed System

A. ***Perception:*** *Security Enhancement in Taxation*

In response to the stringent security demands within the tax sector, an integrated safety verification module has been implemented. This module is designed to ensure that the content generated by the agent adheres to legal statutes, regulatory guidelines, and ethical norms. It is crucial for safeguarding the legitimate interests of users and for maintaining social order and public security. The module employs a dual-auditing approach consisting of keyword matching and a text classification model, with the following operational protocols:

1. **Keyword Matching**: This method leverages a curated lexicon of sensitive terms and rule sets to screen the dialogue text produced by the agent. The process involves scanning for the presence of sensitive keywords or content that is deemed prohibited.

2. **Text Classification Model**: Utilizing a RoBERTa-based text classification model, this method assesses the dialogue text for the presence of prohibited content. The model quantifies the likelihood that the text contains banned material and compares this probability against a predefined threshold to ascertain whether the content is to be flagged as prohibited.

3. **Real-time Online Review**: This process involves a collaborative effort between keyword matching and the text classification model to perform real-time audits of both incoming user queries and outgoing content streams. In the event that sensitive content is identified, the system is designed to immediately interrupt the content stream and default to a pre-approved, non-sensitive language response.

The RoBERTa model, an enhancement of Google's BERT architecture, has been selected for its robust optimization capabilities. By employing advanced preprocessing techniques and leveraging substantial computational resources, this study harnesses an open-source Chinese RoBERTa model to perform sensitive text classification tasks with enhanced accuracy.

B. ***Brain:*** *Supervised Fine-Tuning for Agents*

Supervised fine-tuning enhances the specialized capabilities of pre-trained models by further training LLMs on datasets composed of (instruction, output) pairs or multi-turn conversation datasets involving the roles of system, user, and assistant. This process enables large models to follow human instructions to generate the desired output. In this context, the instruction represents a command issued by a human to the model, and the output represents the expected result that follows the instruction. This process helps bridge the gap between the LLMs' next-word prediction objective and the goal of making them adhere to human instructions. Instruction fine-tuning is a process that involves further training large language models (LLMs) on datasets composed of (instruction, output) pairs, thereby enhancing the capabilities and controllability of LLMs. The uniqueness of instruction fine-tuning lies in the structure of its dataset, which is composed of paired human instructions and expected outputs. This structure allows instruction fine-tuning to focus on enabling the model to understand and follow human instructions. Template 1 describes the primary instruction template used in this system.

Algorithm 1 Example of Instruction Template with ChatML

<|im_start|> system
Your are a helpful tax assistant. You follow user's instructions strictly. Do not hallucinate. **<|im_end|>**
<|im_start|>user
You help users with invoicing tasks. You identify key information from the user's request to fill in the template.
User's request:
xxx
Your reply:
<invoice>{"key": "",...}</invoice>
<satisfaction></satisfaction>
<clarify></clarify> **<|im_end|>**
<|im_start|>assistant
<invoice>{"key": "value",...}</invoice>
<satisfaction>{"lack_key":"key1"}</satisfaction>
<clarify> Please provide information such as key1 </clarify> **<|im_end|>**

The ChatML based conversational format has extended to encompass generative AI tasks. Each message should begin with the <|im_start|> token followed by the role (user or assistant) and end with the <|im_end|> token. To prevent hallucinations while interfering with other modules of the agent system, we introduce special tokens like <*invoice*>, <*satisfaction*> and <*clarify*> to expand the vocabulary. Special tokens delineate different types of information to optimize the gradient update during the training process. LLM learns to perform key information extraction in a sequential manner, to await calibration results, and to prompt the user for additional information when necessary.

To ensure the agent's accurate generation of invoicing information, a memory component structure has been designed that integrates long-term memory, short-term memory, and external information sources. The long-term memory is tasked with recording the user's background information, including tax numbers, payees, and other pre-filled fields. The short-term memory acts as the contextual memory for the ongoing invoicing process, which is facilitated by the conversation prompt structure designed in conjunction with the LLM. This setup enables users to flexibly complete or adjust invoicing information through multi-turn interactive dialogues, thereby significantly enhancing the accuracy of invoicing operations and optimizing the user experience. External information, sourced from a local database and external interfaces, is utilized to fill in details such as the purchaser's name, tax number, address, business card information, and product classification codes.

C. ***Action:*** *Multiple Function Calling*

In our intelligent agent system, we have engineered an integrated suite of tools that encompasses mathematical computations, information verification, and data formatting. This comprehensive toolkit is designed to automate various functionalities, thereby enhancing the reliability and usability of the entire intelligent agent system [8]. By leveraging the synergistic automation of these diverse tools, we ensure that the system can maintain operational continuity and deliver consistent performance across a spectrum of tasks. The mathematical computation module provides precise numerical analysis, the information verification component guarantees data integrity, and the data formatting tool ensures that information is presented in a standardized and accessible manner. Collectively, these tools not only bolster the system's robustness but also facilitate seamless interaction with users and other systems, making our intelligent agent system a dependable solution for complex problem-solving and data management.

IV. Experiments

The experiment was conducted using the Qwen1.5-32B-Chat model [9], a pre-trained language model developed and released by Alibaba Cloud. This model boasts 32 billion parameters and was trained on a corpus comprising 65.79 GB of text data. A computational platform with NVIDIA A800-PCIe GPUs were employed for both training and testing phases.

A. Datasets

TABLE I
Statistics of Training Dataset

Dataset	# Sents	# Tokens
Domain	107,483	29.8M
General	500,654	119M
Security	51,897	10.2M

This study, in collaboration with tax experts and by adhering to pertinent legal frameworks, has curated a tax domain corpus. This corpus has been meticulously assembled by collecting a diverse array of actual invoice data to bolster the model's comprehension of tax-related intricacies.

Furthermore, to enhance model safety, a dedicated safety data corpus has been meticulously crafted. This corpus encompasses a spectrum of texts that include negative information, harmful content, politically sensitive material, violent themes, and offensive remarks, as well as malicious content and instances of harassment. The inclusion of such texts aims to ensure stringent compliance throughout the model's training and application phases, thereby significantly reducing the likelihood of regulatory infractions.

The data undergoes a standardized processing workflow that encompasses text extraction, quality filtering, and data de-duplications. Empirical research has indicated that optimal training outcomes are achieved when the specialized data to general data ratio is 1:5 on existing pre-trained models [10]. To attain the best possible training performance, the adopted data ratio is 2:10:1, with the total data volume amounting to 837.7 MB as detailed in Table I.

B. Experiments

From the tax domain corpus, we extract 1380 data samples from actual invoicing business scenarios and compare the extraction performance of the open-source model, our proposed method, our method with an added memory module, and two commercial models. We aim not only to enhance the model's ability to extract invoicing information but also to assess the model's generalization capabilities. Therefore, our evaluation of the models includes both general ability assessment and invoicing information extraction ability assessment.

The statistics of our testing dataset are shown in Table II. Completeness of the extraction results is assessed using an automated evaluation method, while accuracy is evaluated using a combination of automated evaluation and assessment by tax business experts. The final scoring results are presented on a scale of 100 points. To empirically validate the efficacy of our intelligent agent system, we conducted a series of controlled experiments designed to compare the system's performance against human benchmarks. To ensure the quality and reliability of our evaluation, we assembled a team of nearly twenty seasoned professionals with extensive experience in the field of accounting.

TABLE II
STATISTICS OF TESTING DATASET

Dataset	# Samples
Invoicing	1380
Security	487

TABLE III
EXPERIMENTS RESULTS

Model	Score	Time(s)
Manual Accounting	**90.28**	374
Qwen1.5-32B	73.45	4.46
Our model	86.67	3.91
+ memory	**89.96**	6.43
Commercial Model1	79.45	14.55
Commercial Model2	81.26	17.34

To verify the security of our agent system, we combined industry knowledge to construct a security verification dataset. The safety module's output is tested for accuracy, recall, and F1 score to evaluate its safety interception rate and false positive rate.

C. Results

During the evaluation phase, we enlisted the expertise of domain specialists to conduct manual assessments. As show in Table III, our proposed method achieved the highest scores in terms of accuracy, demonstrating its efficacy in handling industry-specific applications. Moreover, it exhibited a significantly faster response time compared to the commercial models against which it was benchmarked. These findings underscore the superiority of our approach in both precision and efficiency.

V. ANALYSIS

The integrated security module, augmented with advanced thesaurus logic, exhibited robust performance on the test set labeled as *testresult-sec*. As shown in Table IV, it achieved a negative recall rate of 0.87 and a positive recall rate of 0.99, culminating in an overall recall rate of 0.93 and an accuracy rate of 0.91. These metrics underscore the module's effectiveness in accurately identifying and categorizing content.

Similarly, when evaluated on the *devdata* test set, the enhanced security module demonstrated a negative recall rate of 0.86, a positive recall rate of 0.99, an overall recall rate of 0.925, and an impressive accuracy rate of 0.97. The classification accuracy for both datasets surpassed 90%, indicating the model's proficiency in detecting financial and tax risk-related texts.

TABLE IV
SECURITY

	Negative recall	recall	accuracy
testresult-sec	0.87	0.93	0.91
devdata	0.86	0.925	0.97

VI. CONCLUSION

To tackle the challenge of low user interaction efficiency in electronic tax systems, this paper introduces an autonomous agent framework leveraging large language models. The framework incorporates a specialized fine-tuning training paradigm designed to bolster the agent's comprehension of tax-specific knowledge. Furthermore, a hierarchical memory architecture is implemented to streamline the invoicing process, enhancing both efficiency and user experience.In addition, a dedicated security module has been developed to mitigate security concerns within the tax domain. Comparative experiments conducted demonstrate the framework's ability to comprehend complex tax regulations and user requirements, thereby enabling the provision of tailored tax services.

Looking ahead, the optimization design that integrates multi-agent collaboration holds the promise of delivering a more intelligent experience within our system [11]. This study fills the gap in de-manualizing invoice issuance in the tax field. In the future, with the introduction of multi-agent collaboration and multimodal technology, we will explore real-time interactive intelligent assistants. Our future design will leverage the strengths of multi-agent system to enhance the intelligence and efficiency.

REFERENCES

[1] B. Klein, S. Agne, and A. Dengel, "Results of a study on invoice-reading systems in germany," in *International workshop on document analysis systems*. Springer, 2004, pp. 451–462.

[2] Y. Sun, X. Mao, S. Hong, W. Xu, and G. Gui, "Template matching-based method for intelligent invoice information identification," *IEEE access*, vol. 7, pp. 28 392–28 401, 2019.

[3] J. J. Nay, D. Karamardian, S. B. Lawsky, W. Tao, M. Bhat, R. Jain, A. T. Lee, J. H. Choi, and J. Kasai, "Large language models as tax attorneys: A case study in legal capabilities emergence," *arXiv preprint arXiv:2306.07075*, 2023.

[4] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[5] H. Yang, B. Zhang, N. Wang, C. Guo, X. Zhang, L. Lin, J. Wang, T. Zhou, M. Guan, R. Zhang *et al.*, "Finrobot: An open-source ai agent platform for financial applications using large language models," *arXiv preprint arXiv:2405.14767*, 2024.

[6] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International conference on machine learning*. PMLR, 2022, pp. 9118–9147.

[7] F. Xing, "Designing heterogeneous llm agents for financial sentiment analysis," *ACM Transactions on Management Information Systems*, 2024.

[8] N. Li, C. Gao, M. Li, Y. Li, and Q. Liao, "Econagent: large language model-empowered agents for simulating macroeconomic activities," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 15 523–15 536.

[9] Q. Team, "Introducing qwen1.5," February 2024. [Online]. Available: https://qwenlm.github.io/blog/qwen1.5/

[10] C. Wen, X. Sun, S. Zhao, X. Fang, L. Chen, and W. Zou, "Chathome: development and evaluation of a domain-specific language model for home renovation," *arXiv preprint arXiv:2307.15290*, 2023.

[11] Y. Yu, Z. Yao, H. Li, Z. Deng, Y. Cao, Z. Chen, J. W. Suchow, R. Liu, Z. Cui, D. Zhang *et al.*, "Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making," *arXiv preprint arXiv:2407.06567*, 2024.

Sequential Recommendation via Temporal Data Augmentation and Fourier Convolution

1st Junming Luo
School of Computer Science
South China Normal University
Guangzhou, China
luojunming@m.scnu.edu.cn

2nd Tinghua Zhang
Software and Systems Research Department
China Electronics Product Reliability
and Environmental Testing Research Institute
Guangzhou, China
zhangtinghua@ceprei.com

3rd Ke Jin
School of Computer Science
South China Normal University
Guangzhou, China
2023023198@m.scnu.edu.cn

4th Weihao Yu
Network technology research department
Research Institute of China Telecom Corporation Ltd.
Guangzhou, China
https://orcid.org/0000-0003-0727-4744

5th Jin Huang*
School of Computer Science
South China Normal University
Guangzhou, China
huangjin@m.scnu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899499

***Abstract*—Sequence recommendation aims to predict user preferences by modeling users' dynamic historical behaviors. In recent years, advancements in sequence deep learning models such as Transformer have significantly enhanced sequence recommendation. However, most existing sequence recommendation approaches primarily focus on the sequential information of user behaviors while neglecting the critical temporal information. Additionally, noise in interaction records has consistently impeded the performance of sequence recommendations. To address these two shortcomings, we propose a model named TAFC4Rec. First, to better exploit the relationship between temporal information and user interest changes, we improve three classical stochastic data augmentation methods by incorporating time interval calculations. Furthermore, following data augmentation, to enhance the encoder's noise filtering capability for better handling augmented data, we design a convolutional structure based on the Fourier transform to denoise the data and capture sequence features. Finally, we conducted experiments on four real-world datasets, and the experimental results demonstrate the effectiveness of our model.**

***Index Terms*—Sequential Recommendation, Time Information, Data Noise, Data Augmentation, Fourier Transform**

I. Introduction

Sequential recommendation systems (SRSs) [3], [20], [25] aim to predict the next interaction item for users by modeling the sequential dependencies of user-item interactions in a sequence, which has been extensively utilized in contemporary online systems, such as news, shopping, video, etc.

To capture relationships among items in sequences, the existing sequential recommendation methods mainly focus on the order of user interaction items, such as typical solutions based on RNNs [7], [17] and CNNs [4], more advanced neural network architectures (including memory networks [13] and self-attention mechanisms [16]). More recently, transformer [14], [24] structures have shown excellent performance in sequential recommendation tasks so that many subsequent studies used data augmentation [21], [28], [30] combined with it to further optimize recommendation results.

Although there are some initial efforts on these methods, two issues have not yet received adequate attention. Firtst, most of the above works do not fully utilize time information to consider the correlation between items, which is not conducive to capturing the evolution of user preferences. For example, in Figure 1, two users have the same sparse interaction sequence, but the time intervals between their items are different. Based on the logic of real life and the analysis of this time information, it can be concluded that at the time of making recommendations, we should recommend electronic products and food to user A but food to user B. If the time interval is not considered, the recommendation lists for the two users will tend to be the same and lack personalization.

Second, data noise issues [1], [27], [31] also require more attention. Random or unintentional user interaction records, externally injected malicious data, and data augmentation methods changing item attributes introduce noise problems, thereby reducing model performance. Inspired by [22], [32], the periodic characteristics displayed in the frequency domain differ between noisy and normal data. By analyzing the frequency components, the model can better filter the noise. In addition, performing multiplication on frequency components is equivalent to performing convolution in the frequency domain, which is beneficial for learning the correlation between effective data after denoising. Consequently, frequency serves as a valuable tool for aiding in the process of data denoising.

In response to the above issues, we propose a new model called TAFC4Rec, in which we perform the following work: (1) Utilize temporal information to optimize three data augmentation methods: Insert, Mask, and Substitute. The Insert method enriches the sequence by adding new items, Mask captures high-order relationships between items, and Substitute further explores core user preferences. Specifically, we

* Corresponding author.

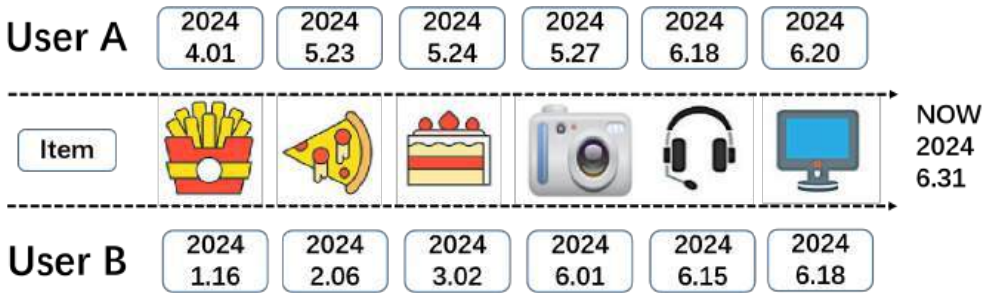


Fig. 1. Time factors in user interaction sequences

determine the positions and items for augmentation through temporal calculations and select new items based on item similarity calculations. (2) Ensure the similarity of newly generated sequences through contrastive learning. This helps to retain users' genuine interests. (3) A Fourier convolutional structure is proposed as a new encoder. We utilize the Fourier transform to convert time-domain data into frequency-domain data and then convolve the frequency-domain components using a learnable filter, filtering the data noise in the process. Afterward, the frequency domain data are converted back to the time domain. Finally, features are further captured through a fully connected neural network. (4) Perform joint optimization training on both contrastive learning tasks and recommendation tasks. Ultimately, we selected four datasets and three distinct types of sequence recommendation models for comparative experiments, confirming the feasibility and effectiveness of our newly proposed model. Our contributions are summarized as follows:

- We propose a data augmentation method that leverages temporal information by analyzing item time intervals in user interaction sequences. This method enhances the model's ability to capture implicit relationships between items and mitigates data sparsity issues.
- We designed a Fourier convolution structure as an encoder. By performing convolution operations in the frequency domain and combining it with a fully connected layer, this encoding structure can not only learn the sequence characteristics of the data but also achieve the effect of data denoising.
- We conducted extensive experiments on four datasets to analyze the feasibility and effectiveness of the proposed new model in detail.

II. Related Work

A. Sequential recommendation

Existing sequential recommendation methods mainly focus on how to effectively model sequences of interacting items. An initial approach involved treating the sequence of items as the Markov chain [9], where the prediction of the next item is closely tied to the most recent interacting items. However, a notable limitation is the inability to learn dependencies over relatively extensive time steps. Consequently, a more advanced solution based on recurrent neural networks (RNNs) [7], [11] has emerged that leverages neural memory operations to capture both long-term and short-term preferences. Despite the strengths of RNNs, their inefficiency in parallelization becomes apparent, leading to performance bottlenecks when processing substantial amounts of sequential data.

Capitalizing on the success [26] of self-attention models in natural language processing (NLP) tasks, a series of sequential recommendation (SR) models based on transformer have been introduced. SASRec [14] employs a transformer layer to discern the significance of items in a sequence, characterizing intricate item transformation correlations. Subsequently, drawing inspiration from the BERT model, BERT4Rec [24] proposed a bidirectional transformer layer. Other works have extended the transformer to integrate complex signals into sequences, confirming the effectiveness of Transformers in addressing sequential recommendation challenges. Due to the superiority of the Transformer structure in tasks, some studies have begun to use the Transformer structure as the main encoder and combine it with data augmentation techniques for sequence recommendation tasks.

B. Discrete Fourier Transform

The discrete Fourier transform (DFT) is a fundamental mathematical tool widely employed in the realm of digital signal processing. Its core function lies in converting discrete sequences into continuous frequency domain signals through the Fourier transform, thereby facilitating the analysis of signaling components at different frequencies within the frequency domain. The Fast Fourier Transform (FFT) [22] is extensively utilized in digital signal processing for efficiently filtering out noisy signals. This efficacy arises from its ability to transform input signals into the frequency domain, where periodic features are more readily distinguishable.

Building upon this concept, researchers have integrated the Fourier transform into recommendation systems to enhance model performance. For instance, the Filter [32] converts sequence data from the time domain to the frequency domain, and it designs a globally learnable filter based on the convolution theorem to analyze the frequency of different hidden dimensions in the item embedding matrix, effectively filtering the data and capturing sequential features. Diverging from Filter's global analysis approach, SLIME4RecJ [6] leverages the Fourier transform to introduce a dynamic frequency selection module. SLIME4RecJ evenly partitions all frequency

components based on the number of layers. Through this fine-grained division, the model can meticulously analyze data at each frequency, thereby enabling adaptive filtering of specific frequency components and effectively alleviating the overfitting phenomenon caused by noise.

III. METHOD

As shown in Figure 2, we first introduce three data augmentation method based on temporal information improvement and ensure the correlation between new samples through contrastive learning. Then, we introduce a more robust Fourier convolution structure to capture sequence dependencies in the enhanced data. Finally, we present the overall training algorithm.

A. Data augmentation

Due to the strong randomness of traditional augmentation operations, some operations may even exaggerate the data sparse problem in the sequence such as cropping and reordering. At the same time, randomly modifying items in the sequence may destroy the correlation of items in the sequence, and this type of method can't fully consider time interval information between items, therefore, from the perspective of alleviating data sparsity, we introduce three data augmentation operations which are applied together to each sequence that combine time information as shown in Figure 3.

Insert In practice, most user interaction records are sparse and the model cannot obtain user preferences from sparse data. Therefore, We select k target positions in the original sequence to insert new items by calculating the time interval. These new noninteractive items need to maintain a certain similarity with the context. By adding new items, we increase the number of items within a certain time period. Our insert operation can be formulated as below and v_{x_i} representing the newly inserted item.

$$s_u^I = I(s_u) = [v_1, v_2, \ldots, v_{x_i}, \ldots, v_n, \ldots] \tag{1}$$

Mask We removed items from specific positions by calculating the time intervals to reduce their standard deviation. Generally, the masking operation encourages items near the masked item to be closer, thereby capturing higher order relationships between distant items in the original sequence. $v_i^{'}$ representing the unmasked item.

$$s_u^M = M(s_u) = [v_1^{'}, v_2^{'}, \ldots, v_n^{'}] \tag{2}$$

Substitute Substitution is the practice of recommending alternative items to users that expand the chances of discovering their actual interests. Accordingly, replacing items in a sequence with highly related items can help preserve the semantic integrity of the original sequence. We replace the original item with a new item at a short time interval, ensuring the new item maintains a certain similarity to the original one. $\overline{v}_{x_i}$ representing the newly replaced item.

$$s_u^S = S(s_u) = [v_1, v_2, \ldots, \overline{v}_{x_i}, \ldots, v_{|s_u|}] \tag{3}$$

1) Time interval calculation: The augmentation method of randomness ignores the correlation information implied by the time interval between items. For example, the smaller the time interval, the stronger the correlation between items. Randomly disturbing the sequence can easily lead to drift in user preferences. According to research by [5], sequences with more uniform time intervals perform better in the model. Consequently, we use the reduction of the variance of the time intervals in the sequence as a guiding principle to select specific positions for modifying the sequence.

2) Item dependencies: In order to ensure maximum consistency with the original sequence semantics when modifying sequence content, our three augmentations operations need to be analyzed in conjunction with contextual semantics. We evaluate the similarity between items through two concepts: internal and external. First, internal refers to assessing the correlation between items based on the model. Since item representations are learned within the same encoder, we directly infer item correlations by measuring the similarity between these item representations. In this work, we adopt dot product as the similarity measure. Given items i and j represented by e_i and e_j , the model-based relevance score $INS(i, j)$ is defined as:

$$INS(i, j) = e_i * e_j \tag{4}$$

Second, external refers to inferring the correlation between items from the situations in which the items are interacted with. Formally, when the number of users shared between two items is higher, the correlation between the two is higher, so the calculation method of this correlation is defined as:

$$EXS(i, j) = \frac{1}{\sqrt{N(i) * N(j)}} \sum_{u \in N(i) \cap N(j)} \frac{1}{\log 1 + N(u)} \tag{5}$$

Where u is a user, N(i) and N(j) are the number of users who have interacted with i and j. From the perspective of model training, the correlation cannot be fully explored through item representation in the early stage, and according to the long tail effect, it is difficult to find related items by analyzing the calculation method of the number of shared users for some low-popular items.

Therefore, we combine the two calculations of correlation to deal with different situations, and take the maximum value of the two as the final score of item correlation.

$$\text{Score}(i, j) = \max(INS(i, j), EXS(i, j)) \tag{6}$$

B. Embedding Layer

After the data augmentation process mentioned above, we embed the obtained new sequence. Because users have different interaction records, in order to facilitate model training, we convert the training sequence into a fixed-length sequence, where n is the settable maximum length parameter. When the length is less than n, we will add padding items to the left side of the sequence until the length is n. If greater than n, only the n items that the user has interacted with most recently will be retained. We create an item embedding matrix $\mathbf{M}_I \in \mathbb{R}^{n \times d}$, where d is the latent dimension, and retrieve the

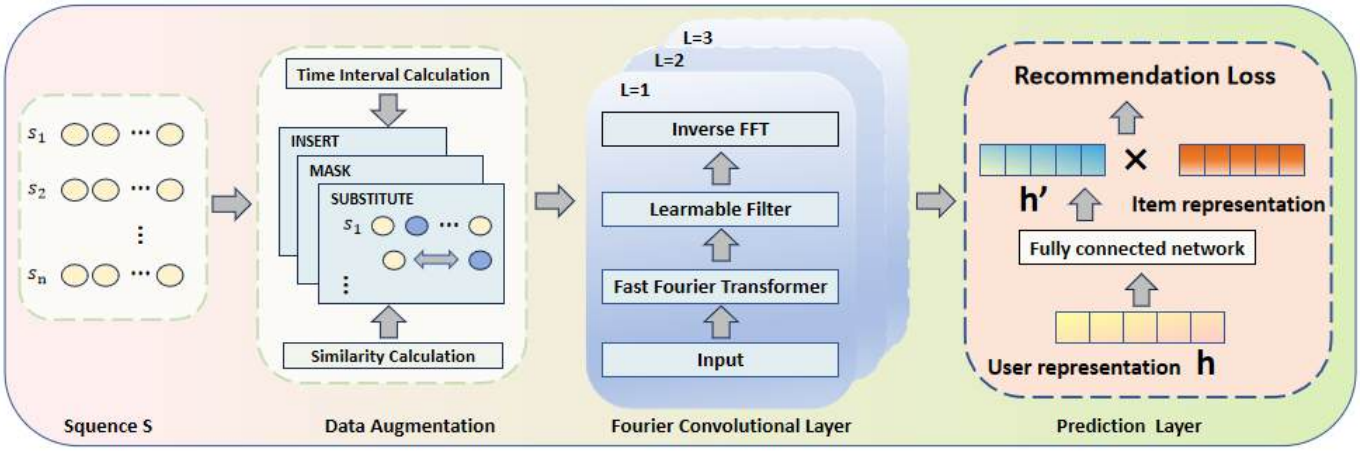


Fig. 2. Overview of TAFC4Rec

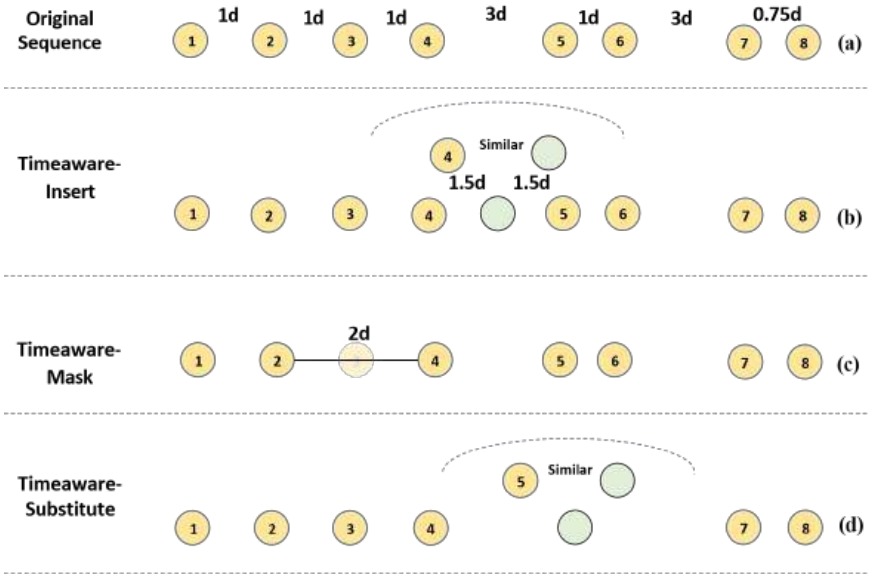


Fig. 3. Data Augmentation

input embedding matrix $\mathbf{E} \in \mathbb{R}^{n \times d}$. The constant zero vector 0 is used as the embedding of the filler terms.

Moreover, in order to learn the position information in the subsequent encoder, we inject a learnable position embedding $\mathbf{P} \in \mathbb{R}^{n \times d}$ into the input embedding. Randomly initialized item and position embedding matrices may cause instability in the training process, so we perform dropout operation and layer normalization operation to alleviate such problems.

C. Fourier convolution layer

Building upon the embedding layer, we construct the item encoder through the stacking of convolution blocks. Each learnable convolution block typically comprises two sublayers: a Fourier filter and a pointwise feed-forward network. We first convert the input in the time domain, that is, the output information of the embedding layer, into a spectrum in the frequency domain. Specifically, given the input item representation matrix $\mathrm{F}^l \in \mathbb{R}^{n \times d}$ of the l-th layer ($\mathrm{F}^0 = \mathbf{E}_I$), we first perform FFT along the item dimension to convert F^l to the frequency domain:

$$\mathrm{X}^l = \mathcal{F}\left(\mathrm{F}^l\right) \in \mathbb{C}^{n \times d} \tag{7}$$

According to the convolution theorem, multiplication in the frequency domain is equivalent to circular convolution in the time domain. We perform element multiplication of the sequence converted to the frequency domain through a learnable filter $\mathbf{W}$, which is equivalent to performing circular convolution on features of different dimensions of x.

$$\tilde{\mathbf{X}}^l = \mathbf{W} \odot \mathbf{X}^l \tag{8}$$

Where $\odot$ is elementwise multiplication. The filter $\mathbf{W}$ can be optimized by SGD to adaptively represent any filter in the frequency domain. Finally, we employ an inverse FFT to

transform the modulation spectrum back into the time domain and update the sequence representation:

$$\mathbf{F}^l \leftarrow \mathcal{F}^{-1}(\mathbf{X}^l) \in \mathbb{R}^{n \times d} \tag{9}$$

Where $\mathcal{F}^{-1}(\cdot)$ represents the inverse 1D FFT, which converts a complex tensor into a real tensor. Through FFT and inverse FFT operations, the noise in the recorded data can be effectively reduced, thereby increasing item embedding. Then we further capture the nonlinear characteristics of the sequence through a fully connected neural network.In the feedforward network, we combine the MLP and ReLU activation functions to further capture nonlinear features. The calculation is defined as follows:

$$\text{FFN}(\tilde{F}^l) = (ReLU(\tilde{F}^l W_1 + b_1))W_2 + b_2 \tag{10}$$

Where W_1, b_1, W_2, b_2 are trainable parameters.

D. Contrastive learning

To optimize the encoder by maximizing the similarity of two sample pairs generated by a sequence, we adopt the NT-Xent loss [2] function to optimize as follows:

$$\mathcal{L}_{ssl}(\tilde{h}_{2u-1}, \tilde{h}_{2u}) = -\log \frac{\exp(\text{sim}(\tilde{h}_{2u-1}, \tilde{h}_{2u}))}{\sum_{m=1}^{2N} \mathbb{I}_{m \neq 2u-1} \exp(\text{sim}(\tilde{h}_{2u-1}, \tilde{h}_m))} \tag{11}$$

Each pair $(\tilde{s}_{2u-1}, \tilde{s}_{2u})$ is regarded as a pair of positive samples obtained after a sequence is enhanced, and the other 2(N-1) augmented views are considered as negative samples of the pair. We use our encoder to characterize each augmented view and represent it as $(\tilde{h}_{2u-1}, \tilde{h}_{2u})$, where sim$(\cdot)$ is a dot product that measures the similarity between two enhanced views, and $\mathbb{I}_{m \neq 2u-1}$ is the indicator function.

E. Joint training

In the recommendation task, we compute the user's preference score for item i at step (t + 1) in the context of the user's history:

$$\mathcal{L}_{\text{rec}} = -\sum_{u \in \mathcal{U}} \sum_{t=1}^{n} \log \sigma \Big(P(i_{t+1}|i_{1:t}) - P(i_{t+1}^{-}|i_{1:t}) \Big) \tag{12}$$

u represents a user, and $\mathcal{U}$ represents the set of users. Since both next-item prediction and comparison SSL model item relationships are in sequence, in order to improve sequential recommendation performance through comparison SSL goals, we leverage a multitask strategy to jointly optimize them as follows and λ is a configurable number:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{ssl}} \tag{13}$$

IV. Experiment

In this section, we conduct experiments on four public datasets and answer the following research questions (RQs):

- RQ1: How does TAFC4Rec perform in sequential recommendation compared to existing method?
- RQ2: Are data augmentation methods that consider time interval information effective?
- RQ3: Can TAFC4Rec maintain robust performance in the presence of noisy interactions?
- RQ4: How does TAFC4Rec perform on sparse datasets? Is joint training effective?

A. Datasets

We conducted experiments using publicly available datasets from various domains collected from real platforms. The Beauty and Sports datasets were obtained from the Amazon review dataset, with the Beauty dataset being relatively sparse. The Yelp dataset was used for commercial recommendations, while the Home dataset focused on furniture-related data. The data were preprocessed following [30], where items with ratings and reviews were designated as positive examples, while the rest were considered negative examples. Additionally, following [12], we analyzed only users who had purchased at least 5 items, and each item in the dataset had interactions with at least 5 users.

B. Comparison Methods and Evaluation Metrics

To evaluate the effectiveness of the new model, we compared it with a total of ten recommended models of three different types as follows.

- BPR [23] Bayesian Personalized Ranking (BPR) is an algorithm that improves recommendation accuracy by ranking items based on implicit feedback, optimizing the ranking of items through a pairwise comparison of user interactions.
- LightGCN [10] involves streamlining GCN architectures and enhancing their efficiency to improve recommendation accuracy and scalability.
- GRU4Rec [11] use RNNs to model user behavior within a session, capturing the sequential nature of interactions to provide accurate next-item recommendations.
- STAMP [18] enhances session-based recommendation systems by prioritizing short-term attention and memory. It focuses on recent user interactions within a session to improve the accuracy of item recommendations.
- SASRec [14] leverages self-attention mechanisms to model user behavior sequences, capturing long-term dependencies and improving recommendation accuracy by focusing on relevant past interactions.
- BERT4Rec [24] utilizes Transformer-based bidirectional encoders to capture sequential patterns in user behavior for improved recommendation accuracy.
- TiSASRec [15] is a technique that enhances recommendation systems by incorporating attention mechanisms that are sensitive to the time intervals between user interactions. This approach aims to improve the modeling of temporal dynamics in user behavior sequences, leading to more accurate recommendations based on the timing of interactions.
- LightSANs [8] proposes a method to improve next-item recommendation systems by using low-rank decomposition in self-attention networks. This approach aims to reduce computational complexity while maintaining or

improving recommendation accuracy, making the model more efficient and effective for practical applications.

- CL4SRec [28] uses pairs of positive and negative interactions to train models, improving their ability to predict which items users will interact with next in recommendation systems.
- CoseRec [19] is a novel framework for sequential recommendation that applies contrastive Self-Supervised Learning (SSL) with specialized augmentation techniques to enhance model performance by leveraging item correlations and addressing challenges like data sparsity and noise.

We use two indicators, Hit@k and NDCG@k, to measure the experimental effect. Hit@k calculates how many of the top k recommended items are relevant to the user's preferences or needs. NDCG@k evaluates the ranking quality of the top k recommendations by considering both relevance and position. (k = 10, 20)

C. Implementation Details

We use the code provided by Ticoserc [5] to perform data augmentation. At the same time, we use the idea provided by the FMLP-REC [32] to design the convolutional layer structure. The training batch is set to 128. We use ADAM optimizers with a learning rate of 0.001, β_1 = 0.9, β_2 = 0.999, and a batch size of 256. The size of the embedded dimension was set to 128.

We implement our method in PyTorch. For common super reassessment, we set the number of self attention blocks and attention heads to 2 according to [29], and the embedded dimension is 64, and the maximum sequence length is 50. We adjust α, β, K, E and λ in [0.1, 0.9], [0.1, 0.9], {4, 12, 20}, {0, 20, 40, 80, 100, 100}, within the range and {0.1, 0.2, 0.3, 0.4, 0.5} range. If the performance of 40 epochs has not improved, we adopt a method of stopping the verification set and report the results of the test set.

D. Performance comparison(RQ1)

Overall, as shown in Table I, the average performance of the nonsequential model is the worst. Compared with BPR, LightGCN pays more attention to neighborhood information and therefore can relatively capture the correlation of adjacent items, thereby achieving relatively superior performance. This underscores the importance of mining item sequence information for prediction. In the experimental setting, sequence models are divided into two types. Both methods utilize the Transformer structure to set up the encoder, and the main difference lies in the use of data augmentation methods. In the first method without utilizing data augmentation, LightSANs achieve the best performance by effectively addressing the redundancy of Transformer structure parameters, which illustrates the need for improvement in handling Transformer parameter redundancy. The performance of the second model with the data augmentation method is generally better than that of the model without the data augmentation method. These results also validate the effectiveness of using contrastive learning and data augmentation. Compared with CL4SRec which uses random enhancement for SSL tasks, CoseRec performs better. The main reason is that CoseRec improves the random enhancement method by utilizing item correlation.

Our model chooses to improve the data by taking into account the temporal information of the data augmentation method, and uses a more robust convolutional structure to capture the data features. Through observation, our method consistently outperforms other methods on all datasets, especially demonstrating more significant improvements on the sparsest dataset Beauty. This further emphasizes the superiority of our method in handling sparse datasets.

E. The effectiveness of utilizing time information(RQ2)

To demonstrate the effectiveness of incorporating time information in the data augmentation method proposed in this paper, we removed the operation of analyzing time intervals in the three augmentation methods and changed the augmentation mode of the three methods to randomness, while keeping the other parts of the model unchanged. This modified variant model, referred to as Variant, was compared with our TAFC4Rec model across four datasets. As illustrated in the Figure 4, TAFC4Rec consistently outperformed Variant across different datasets. Notably, the performance gap was more pronounced in the Beauty dataset, where interaction records were relatively sparse. This improvement is attributed to the fact that data augmentation by analyzing time interval information can better capture changes in user interests, whereas random operations are more likely to distort the original semantics of the sequence. Therefore, incorporating time interval information in the augmentation process significantly enhances model performance.

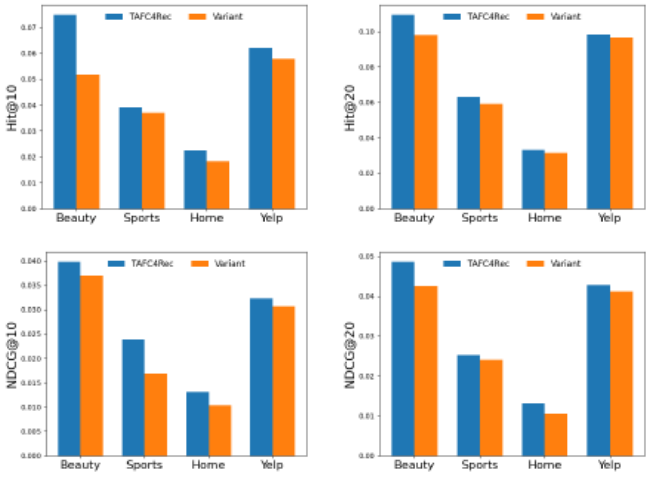


Fig. 4. Temporal Information's Impact on Data Augmentation Methods

F. Robustness of TAFC4Rec(RQ3)

To test the robustness of TAFC4Rec, we designed two sets of comparative experiments using the Beauty and Sports datasets. We randomly added a certain proportion (10%, 20%,

TABLE I
PERFORMANCE COMPARISONS OF DIFFERENT METHODS. THE BEST PERFORMANCE AND THE SECOND BEST PERFORMANCE METHODS ARE DENOTED IN BOLD AND UNDERLINED FONT RESPECTIVELY.

Dataset	Metric	Non-sequential Models		Sequential Models without Data Augmentation						With Data Augmentation		
		BPR	LightGCN	STAMP	GRU4Rec	BERT4Rec	TiSASRec	SASRec	LightSANs	CL4SRec	CoSeRec	TAFC4Rec
Beauty	Hit@10	0.0358	0.0490	0.0458	0.0483	0.0476	0.0502	0.0512	0.0517	0.0559	0.0625	**0.0750**
	Hit@20	0.0576	0.0753	0.0693	0.0717	0.0775	0.0795	0.0767	0.0807	0.0835	0.0985	**0.1097**
	NDCG@10	0.0179	0.0242	0.0233	0.0249	0.0213	0.0244	0.0255	0.0240	0.0227	0.0331	**0.0399**
	NDCG@20	0.0226	0.0308	0.0292	0.0305	0.0288	0.0322	0.0369	0.0313	0.0316	0.0417	**0.0487**
Sports	Hit@10	0.0271	0.0328	0.0277	0.0297	0.0282	0.0247	0.0351	0.0339	0.0346	0.0360	**0.0391**
	Hit@20	0.0410	0.0528	0.0424	0.0507	0.0488	0.0419	0.0565	0.0609	0.0587	0.0579	**0.0632**
	NDCG@10	0.0101	0.0150	0.0145	0.0176	0.0149	0.0148	0.0171	0.0189	0.0205	0.0213	**0.0238**
	NDCG@20	0.0144	0.0201	0.0172	0.0179	0.0176	0.0165	0.0224	0.0207	0.0225	0.0238	**0.0252**
Home	Hit@10	0.0054	0.0086	0.0168	0.0133	0.0155	0.0119	0.0181	0.0113	0.0177	0.0212	**0.0225**
	Hit@20	0.0094	0.0115	0.0223	0.0207	0.0204	0.0193	0.0257	0.0170	0.0249	0.0304	**0.0331**
	NDCG@10	0.0025	0.0042	0.0109	0.0064	0.0079	0.0059	0.0112	0.0063	0.0081	0.0120	**0.0131**
	NDCG@20	0.0043	0.0065	0.0133	0.0090	0.0103	0.0085	0.0131	0.0085	0.0118	0.0137	**0.0154**
Yelp	Hit@10	0.0447	0.0522	0.0402	0.0456	0.0476	0.0569	0.0572	0.0604	0.0554	0.0593	**0.0619**
	Hit@20	0.0683	0.0760	0.0668	0.0848	0.0786	0.0921	0.0895	0.0933	0.0874	0.0945	**0.0983**
	NDCG@10	0.0302	0.0328	0.0214	0.0239	0.0279	0.0309	0.0297	0.0320	0.0302	0.0311	**0.0323**
	NDCG@20	0.0337	0.0373	0.0259	0.0315	0.0332	0.0399	0.0375	0.0406	0.0371	0.0415	**0.0428**

30%, 40%, and 50%) of negative user-item interactions to each test sequence in these datasets.

In the first set of comparative experiments, we compared the proposed TAFC4Rec model with the CoSeRec and CL4SRec models, both of which performed well in the baseline experiments. Figures 5 display the results of this comparison. TAFC4Rec consistently outperforms CoSeRec and CL4SRec across all noise ratios. The performance decline of the models varies between datasets. For instance, in the Beauty dataset, TAFC4Rec's performance drops by approximately 42 % at 50% noise, whereas it drops by 45% in the Sports dataset. Notably, CL4SRec exhibits the largest performance decrease among the three models. This suggests that, in addition to randomness, more specific information is required for constructing views when using data augmentation methods. It's important to note that both CoSeRec and CL4SRec utilize the Transformer architecture.

To directly analyze the robustness of the proposed Fourier convolution layer, we conducted a second set of comparative experiments. Specifically, we replaced the encoding structure of TAFC4Rec with the mainstream Transformer architecture, creating a new model termed T-model. We then compared the T-model with TAFC4Rec. Figures 6 illustrate that TAFC4Rec performs better under noisy conditions, with both models showing relatively close declines. This indicates that the augmentation method incorporating time information aids in robustness, and our Fourier convolution structure handles noise better than the Transformer structure.

G. Performance in Sparse Data and Effectiveness of Joint Training(RQ4)

Data sparsity is a common issue in recommendation systems. In order to test the performance of TAFC4Rec on this issue, we only used partial training data (25%, 50%, 75%,

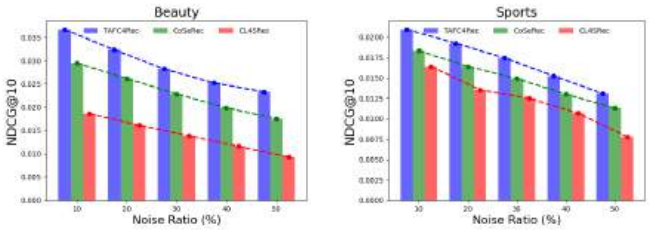


Fig. 5. Comparison of TAFC4Rec, CoSeRec and CL4SRec

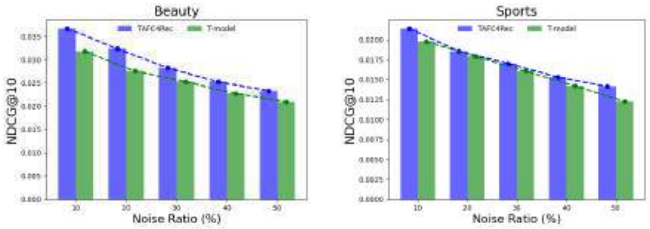


Fig. 6. Comparison of TAFC4Rec and T-model

and 100%) to train the model to simulate data sparsity, while keeping the test data unchanged. We conducted experiments on the proposed method on Beauty and Sports datasets and compared it with two models, CoSeRec and CL4SRec. The results are shown in Table II. We observed that when using less training data, performance significantly decreases, but TAFC4Rec consistently outperforms the other two models and the performance degradation is slower. Furthermore, in the Sports dataset, when all three had only 50% training data, CoSeRec decreased by 45 CL4SRec decreased by 53%, and TAFC4Rec decreased by 37%. These observations indicate that information augmentation in CoSeRec also plays a good role in alleviating data sparsity issues, but the approach of

TABLE II
THE PERFORMANCE ON NOISY DATASET

	Beauty				Sports		
	TAFC4Rec	**CoSeRec**	**CL4SRec**		**TAFC4Rec**	**CoSeRec**	**CL4SRec**
25%	0.0294	0.0226	0.0103	25%	0.0119	0.0101	0.0065
50%	0.0420	0.0342	0.0167	50%	0.0243	0.0200	0.0160
75%	0.0638	0.0490	0.0355	75%	0.0329	0.0000	0.0240
100%	0.0750	0.0625	0.0559	100%	0.0391	0.0360	0.0346

TABLE III
COMPARISON OF TWO STRATEGIES

Home		Yelp	
Strategy	**Hit@10 / NDCG@10**	**Strategy**	**Hit@10 / NDCG@10**
Multi-task	0.0225 / 0.0131	Multi-task	0.0619 / 0.0323
Two-stage	0.0194 / 0.0117	Two-stage	0.0601 / 0.0284

combining time information in TAFC4Rec may have a better effect on data sparsity issues. We also observed that the impact of data sparsity on different datasets varies. Under 50% of training data, TAFC4Rec's performance in beauty decreased by 44%, while in sports it decreased by 37%.

In addition, we conducted experiments to compare the performance of our multi task joint training strategy and the alternative two-stage training strategy. Table III shows the results of both on the Home and Yelp datasets. We observed that compared to the multi task strategy, two-stage training performed worse, indicating that TAFC4Rec can promote each other in joint training by comparing learning objectives and recommendation objectives, while two-stage training reduces the effect of supervised information.

V. CONCLUSIONS

In our work, we investigated two issues in the field of sequence recommendation: how to better utilize time information and alleviate data noise. We proposed a novel learning framework TAFC4Rec, which optimizes three classic information enhancement methods using time information, namely Insert, Mask, and Substitute, mainly through time interval calculation and item similarity calculation. In addition, we also designed a Fourier convolution structure that utilizes the frequency domain to process noisy data and capture sequence relationships, enhancing the robustness of the model. We conducted extensive experiments on four benchmark datasets and demonstrated the effectiveness and robustness of TAFC4Rec. In addition, we also investigated the importance of different modules in TAFC4Rec.

REFERENCES

[1] Chen, H., Lin, Y., Pan, M., Wang, L., Yeh, C.C.M., Li, X., Zheng, Y., Wang, F., Yang, H.: Denoising self-attentive sequential recommendation. In: Proceedings of the 16th ACM Conference on Recommender Systems. pp. 92–101 (2022)

[2] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)

[3] Chen, T., Yin, H., Nguyen, Q.V.H., Peng, W.C., Li, X., Zhou, X.: Sequence-aware factorization machines for temporal predictive analytics. In: 2020 IEEE 36th international conference on data engineering (ICDE). pp. 1405–1416. IEEE (2020)

[4] Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., Zha, H.: Sequential recommendation with user memory networks. In: Proceedings of the eleventh ACM international conference on web search and data mining. pp. 108–116 (2018)

[5] Dang, Y., Yang, E., Guo, G., Jiang, L., Wang, X., Xu, X., Sun, Q., Liu, H.: Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 4225–4232 (2023)

[6] Du, X., Yuan, H., Zhao, P., Fang, J., Liu, G., Liu, Y., Sheng, V.S., Zhou, X.: Contrastive enhanced slide filter mixer for sequential recommendation. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). pp. 2673–2685. IEEE (2023)

[7] Duan, J., Zhang, P.F., Qiu, R., Huang, Z.: Long short-term enhanced memory for sequential recommendation. World Wide Web **26**(2), 561–583 (2023)

[8] Fan, X., Liu, Z., Lian, J., Zhao, W.X., Xie, X., Wen, J.R.: Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 1733–1737 (2021)

[9] He, R., McAuley, J.: Fusing similarity models with markov chains for sparse sequential recommendation. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 191–200. IEEE (2016)

[10] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 639–648 (2020)

[11] Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)

[12] Hidasi, B., Quadrana, M., Karatzoglou, A., Tikk, D.: Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM conference on recommender systems. pp. 241–248 (2016)

[13] Huang, J., Zhao, W.X., Dou, H., Wen, J.R., Chang, E.Y.: Improving sequential recommendation with knowledge-enhanced memory networks. In: The 41st international ACM SIGIR conference on research & development in information retrieval. pp. 505–514 (2018)

[14] Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE international conference on data mining (ICDM). pp. 197–206. IEEE (2018)

[15] Li, J., Wang, Y., McAuley, J.: Time interval aware self-attention for sequential recommendation. In: Proceedings of the 13th international conference on web search and data mining. pp. 322–330 (2020)

[16] Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on

Conference on Information and Knowledge Management. pp. 1419–1428 (2017)
[17] Liu, Q., Wu, S., Wang, D., Li, Z., Wang, L.: Context-aware sequential recommendation. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). pp. 1053–1058. IEEE (2016)
[18] Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1831–1839 (2018)
[19] Liu, Z., Chen, Y., Li, J., Yu, P.S., McAuley, J., Xiong, C.: Contrastive self-supervised sequential recommendation with robust augmentation. arXiv preprint arXiv:2108.06479 (2021)
[20] Lv, Z., Zhang, W., Chen, Z., Zhang, S., Kuang, K.: Intelligent model update strategy for sequential recommendation. In: Proceedings of the ACM on Web Conference 2024. pp. 3117–3128 (2024)
[21] Qiu, R., Huang, Z., Yin, H., Wang, Z.: Contrastive learning for representation degeneration problem in sequential recommendation. In: Proceedings of the fifteenth ACM international conference on web search and data mining. pp. 813–823 (2022)
[22] Rajaby, E., Sayedi, S.M.: A structured review of sparse fast fourier transform algorithms. Digital Signal Processing **123**, 103403 (2022)
[23] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
[24] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 1441–1450 (2019)
[25] Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the eleventh ACM international conference on web search and data mining. pp. 565–573 (2018)
[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
[27] Wang, W., Feng, F., He, X., Nie, L., Chua, T.S.: Denoising implicit feedback for recommendation. In: Proceedings of the 14th ACM international conference on web search and data mining. pp. 373–381 (2021)
[28] Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., Ding, B., Cui, B.: Contrastive learning for sequential recommendation. In: 2022 IEEE 38th international conference on data engineering (ICDE). pp. 1259–1273. IEEE (2022)
[29] Xu, X., Fei, S., Zhaoyang, L., Jinyang, G., Bolin, D., Bin, C.: Contrastive pre-training for sequential recommendation. arXiv preprint arXiv:2010.14395 (2020)
[30] Zhou, K., Wang, H., Zhao, W.X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., Wen, J.R.: S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp. 1893–1902 (2020)
[31] Zhou, K., Wang, H., Zhao, W.X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., Wen, J.R.: S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp. 1893–1902 (2020)
[32] Zhou, K., Yu, H., Zhao, W.X., Wen, J.R.: Filter-enhanced mlp is all you need for sequential recommendation. In: Proceedings of the ACM web conference 2022. pp. 2388–2399 (2022)

FMIP: Feature Map Importance Pruning for Efficient CNN Optimization

Ali Muhammad Shaikh
University of Science and Technology of China
Hefei, China
alims@mail.ustc.edu.cn

Yun-bo Zhao*
University of Science and Technology of China
Hefei, China
ybzhao@ustc.edu.cn

Yu Kang
University of Science and Technology of China
Hefei, China
kangduyu@ustc.edu.cn

Aakash Kumar
Zhongshan Institute of Changchun University of Science and Technology, China
Zhonhshan, China
aakash@cust.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899531

***Abstract*— Deep convolutional neural networks (CNNs) have reformed numerous fields, comprising computer vision and natural language processing. Nevertheless, CNNs' computational complexity frequently raises obstacles to utilization on resource-constrained devices. Different optimization strategies have been proposed in response to this issue, comprising quantization, knowledge distillation, and pruning. This paper familiarizes a novel pruning method, Feature Map Importance Pruning (FMIP), designed to optimize the performance of deep CNNs while reducing computational costs. The FMIP computes the importance of feature maps within convolutional layers according to the area of their activation values, facilitating a systematic approach to pruning decisions. By detecting and eliminating the redundant feature maps, FMIP can considerably cut the number of parameters and FLOPs in a CNN model without conceding accuracy. This is chiefly advantageous for using CNNs on devices with limited memory and computational power.**

We evaluated FMIP on various CNN architectures, including VGG16 and ResNet50, using datasets such as CIFAR-10 and ImageNet. Our experiments demonstrated significant model compression while preserving high accuracy. Specifically, FMIP achieved the following results: VGG16 with CIFAR-10 demonstrated an 81.9% reduction in parameters and a 48.6% reduction in FLOPs, with an accuracy of 93.33%. ResNet50 with ImageNet achieved an 83.11% reduction in parameters and a 71.55% reduction in FLOPs, with an accuracy of 92.03%.

Keywords*— *Deep CNNs, Feature Map Importance, Model compression, Filter Pruning, Optimization

I. Introduction

To enhance the efficiency of Convolutional Neural Networks (CNNs), researchers have utilized advanced techniques across various machine learning domains, such as object detection [1], [2], fault detection in UAVs [3], and image recognition [4], [5]. However, achieving outstanding results also poses significant challenges. These challenges include complex architectures that are suboptimal regarding memory usage and require substantial computational resources, particularly for embedded and mobile devices. Additionally, these architectures incur considerable costs during inference. Consequently, model compression has garnered significant attention from researchers as a means to address the size issue of CNN architectures. Even so, CNNs are inherently computationally intensive, leading to a substantial increase in floating-point operations (FLOPS) [6] due to the extensive trainable parameters and convolution operations involved.

In recent years, researchers have proposed various model compression and acceleration methods [6], [7]. Based on pruning granularity, network pruning can be categorized into unstructured pruning [8] and structured pruning [9]. A common form of unstructured pruning is weight pruning, which reduces network parameters by eliminating insignificant weights in the filters. Unstructured pruning requires specialized software or hardware for model acceleration, whereas structured pruning does not face this issue. Consequently, structured pruning has garnered more attention in recent years. Structured pruning primarily involves filter pruning.

The key to filter pruning lies in selecting which filters to remove. Our proposed pruning strategy FMIP (Feature Map Importance Pruning) is based on the feature map area, targeting layers and ranking feature maps according to "area" in terms of the magnitude of activations. This approach provides a clear, interpretable metric to evaluate the importance of each filter in the network. The underlying assumption is that feature maps with smaller activations (areas) contribute less to the overall task and can be safely removed, thereby reducing the model's complexity. Nevertheless, we rank the feature maps by their scores, which reflect how "active" or "informative" (see Fig.1). This approach is computationally efficient because calculating the area involves only basic summations, making it practical even for large models. Moreover, the combination of iterative pruning and fine-tuning enables the network to gradually adapt to the removal of less significant filters, thereby minimizing the performance loss that often occurs with more aggressive pruning methods.

II. Literature Review

The primary objective of filter pruning is to assess the significance of filters and eliminate those considered

unimportant [10], [11]. Consequently, re-training is necessary after each pruning step to compensate for the reduction in accuracy. In reference [12], the importance of filters was evaluated using a portion of the training data based on the output feature map. In [13], pruning was performed using a greedy technique that determined filter importance by evaluating the model's accuracy post-pruning. Additionally, in [14], feature maps, or pruning activations, were employed to create faster CNNs. This method involves removing filters from specific input locations while often retaining them in other locations, resulting in minimal overall filter compression. This approach may also be viewed as excluding filters at specific input locations, but these filters are often retained at other locations, leading to limited filter compression.

In [15] employs high-rank feature map selection for filter pruning. HRank ranks feature maps based on their information content using matrix rank, ensuring that the most significant feature maps are preserved during pruning. Another method named SCSP (Spectral Clustering Filter Pruning) was introduced in [16], which employs a self-adaptive method to prune filters, concentrating on the spectral characteristics of filters. Our proposed strategy Feature Map Importance Pruning (FMIP) differs from these methods. Firstly, it is more geometrically motivated by spatial coverage, offering a more straightforward metric that can be easily interpreted in terms of input space activation. Secondly, it is based on the area under feature map activations, concentrating on measuring the spatial activation area rather than rank, providing it with a unique advantage in understanding the feature map's contribution through the coverage area. Furthermore, this method is less computationally intensive than spectral clustering or HRank techniques, making it simpler to implement and potentially quicker to execute without extensive fine-tuning.

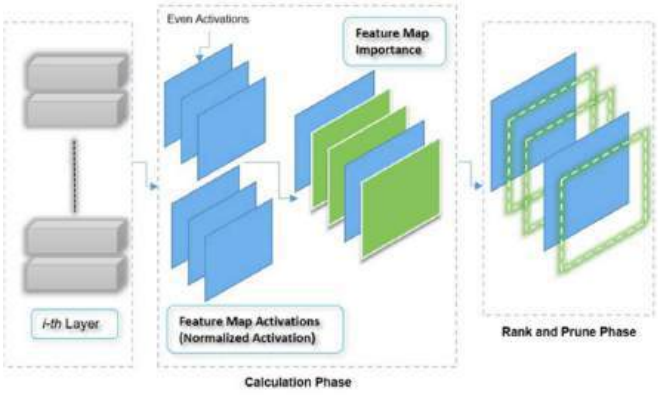


Fig. 1 Feature Map Importance Pruning (FMIP) Process

III. Pruning Filters via Rank and Importance

To optimize (CNNs) through weighing and pruning feature maps characterized by their importance scores, which can be conceptually associated with their active "area" of contribution across each layer. we set off with the activation values computation from the convolutional layers of the CNN model.

A. Ranking and Importance Score Calculation

For a specific convolutional layer n, let X_n signify the input to the layer, and $X_n \in \mathbb{R}^{C_{\text{in}} \times H \times W}$ (with C_{in} is denoted as the number of input channels, H implies the height, and W is the width of the input). The convolution operation provides feature maps L_n as output, where:

$$\mathrm{L_n} = \mathrm{W_n} * \mathrm{X_n} + \mathrm{b_n} \tag{1}$$

Now, W_n are denoted as convolutional filters of layer n alongside dimensions $C_{out} \times C_{in} \times k_h \times k_w$ and b_n denotes the bias. The feature map activations $L_n \in \mathbb{R}^{C_{\text{out}} \times H' \times W'}$ are then acquired after implementing the convolutional filters, where C_{out} is the number of output channels, and H', W' are the height and width of the output feature maps. The importance of feature maps is calculated by determining the spatial area covered by the activations in each feature map. The area is decided through summing the absolute activations to indicate its contribution to the forward pass. The area A_i of feature map i can be calculated as follows:

$$\mathrm{A_i} = \sum_{\mathrm{h=1}}^{\mathrm{H'}} \sum_{\mathrm{w=1}}^{\mathrm{W'}} \left|\mathrm{A_{i,h,w}}\right| \tag{2}$$

Where $A_{i,h,w}$ means the value at position (h, w) in the i-th feature map of a certain layer. $\sum_{h=1}^{H'} \sum_{w=1}^{W'}$ stands for the raw area score, determining the contribution of feature map i. We apply Min-Max Normalization to the areas A_i before ranking the feature maps for pruning. To scale the values between 0 and 1, we ensure that the importance scores are uniformly scaled throughout all feature maps. The normalized importance score I_i is calculated as:

$$\mathrm{I_i} = \frac{\mathrm{A_i} - \min(\mathrm{A})}{\max(\mathrm{A}) - \min(\mathrm{A})} \tag{3}$$

Where $\min(A)$ and $\max(\mathrm{A})$ signify the minimum and maximum area values, respectively throughout all feature maps. After normalization, the feature maps are ranked by their normalized importance values. For example, Lower importance scores imply that the equivalent feature map offers less to the general activations and therefore can be intended for pruning. Let the ranking be denoted via an ordered set R such that:

$$\mathrm{R} = \mathrm{sort}(\mathrm{I_1}, \mathrm{I_2}, \ldots, \mathrm{I_n}) \tag{4}$$

where n denoted as the total number of feature maps, the sort function organizes the feature maps via their normalized importance scores.

B. Pruning Criterion: Feature Map Area

We set a threshold τ is used to prune a definite percentage of feature maps. For instance, if $\tau = 0.3$, the bottom 30% of feature maps according to their importance scores are pruned.

$$\text{Prune } \mathrm{F_i} \text{ if } \mathrm{I_i} < \tau \tag{5}$$

where F_i signifies the i-th feature map. The feature maps that contribute the least to the network's activations are eliminated, successfully dropping the number of filters. where τ is the threshold value defined by the pruning rate r. Pruning a feature map F_i requires the deletion of the connected filter W_i. In the updated network, the pruned filters are efficiently neglected. We then fine-tune the model to regulate the deficit of the pruned feature maps. The final importance score, which merges both the Min-Max Normalization and area-based metric, can be defined as:

$$\mathrm{I_i} = \frac{\sum_{h=1}^{H'} \sum_{w=1}^{W'} \left|\mathrm{A_{i,h,w}}\right| - \min(\mathrm{A})}{\max(\mathrm{A}) - \min(\mathrm{A})} \quad (6)$$

Equation (6) is used for ranking and pruning the feature maps in FMIP.

IV. Experiments

A. Ranking and Importance Score Calculation

We utilized the LeNet model on the MNIST dataset as a prototype. To further demonstrate the efficiency of our proposed approach in shrinking model size, we conducted experiments on mainstream CNN architectures like VGG16 using the CIFAR10 and CIFAR100 datasets. We evaluated the model's performance post-pruning by monitoring accuracy, number of parameters, and required Floating Point Operations (FLOPs). The effectiveness of FMIP was assessed by comparing the performance before and after the pruning process, ensuring that the model maintained its predictive capability while achieving reduced complexity. Through this systematic and iterative approach, FMIP effectively ranked feature map areas and conducted a tailored pruning procedure that enhanced CNN efficiency while preserving the essential performance characteristics.

B. Hyperparameters Settings

All the experiments were conducted using the PyTorch framework on an NVIDIA Tesla GPU, with the execution environment Google Colab Pro. For the VGG16 model on CIFAR-10 and ResNet50 on ImagNet, the settings incorporate a layer-wise learning rate adjustment to fine-tune the learning dynamics for each layer based on its position in the network. In this setup, the learning rate for the initial convolutional layers is set to a lower value, typically 0.001, as these layers are responsible for learning the fundamental visual features. For the middle layers, the learning rate is kept slightly higher at around 0.005. The deeper layers, particularly the fully connected layers, require more aggressive learning to adapt to the classification task. Hence, the learning rate for these layers is set at the base value of 0.01.

Unlike the LeNet-5 model on MNIST, a lower learning rate of 0.001 as the task is simpler because of grayscale input images. We used Stochastic Gradient Descent (SGD) with a momentum value of 0.9 to provide stable updates during training for every model. The training is set for 100 epochs with a batch size of 128 for both ResNet50 and VGG16, but for LeNet-5 batch size is set to 64. To avert the overfitting issue, a weight decay (L2 regularization) of 1e-4 for LeNet-5 and 5e-4 is applied for the ResNet50 and VGG16 models, respectively. The model also applies data augmentation such as random cropping and horizontal flipping to boost simplification.

V. Results

In our prototype study, we applied the Feature Map Importance Pruning (FMIP) method to the LeNet network using the MNIST dataset. This approach led to a significant reduction of 77% in network parameters and a 76.5% saving in FLOPs. Impressively, despite this substantial pruning, the model maintained an accuracy of 98.86%, which is only marginally below the baseline accuracy of 99.21%.

Furthermore, we assign a pruning rate of 0.3 to the VGG16 network and perform iterative pruning. Our proposed FMIP (Feature Map Importance Pruning) approach demonstrates superior performance compared to existing filter pruning methods. Table 1 shows the results for the VGG network, we adopted the 16-layer model (comprising 13 convolutional and 3 fully connected layers) to work with the CIFAR-10 dataset. Despite pruning 81.9% of the parameters and saving 48.6% of FLOPs, we maintained an accuracy of 93.33%, only slightly above our baseline accuracy. These optimizations significantly enhance the VGG model's capability, a popular backbone for object detection and semantic segmentation, to be efficiently deployed on mobile devices.

TABLE I. TABLE 1VGG16 PRUNING PERFORMANCE ON CIFAR10

	VGG-16 on CIFAR-10 datasets (Prune results)			
Approach	Baseline (%)	FLOPs Saved (%)	Pruned (%)	Accuracy (%)
[17]	93.73	52.4	89.7	93.82
[18]	93.25	39.1	73.3	93.18
[19]	-	41.6	73.8	93.02
FMIP	**93.30**	**48.6**	**81.9**	**93.33**
[20]	-	42.5	82.2	90.73

For the ResNet50 network using the ImageNet dataset. This approach resulted in the pruning of 83.11% of the network parameters, concurrently achieving a 71.55% reduction in FLOPs. Despite these significant modifications, the model retained a commendable accuracy of 92.03%, marginally below the original baseline accuracy of 92.40%. These substantial optimizations highlight the efficacy of FMIP in enhancing resource efficiency, thereby rendering ResNet50 more suitable for deployment in environments with limited computational resources (see Table 2).

TABLE II. TABLE 2 RESNET50 PRUNING PERFORMANCE ON IMAGENET

	ResNet50 on ImageNet datasets (Prune results)			
Approach	Baseline (%)	FLOPs Saved (%)	Pruned (%)	Accuracy (%)
[19]	-	68.95	72.94	91.91
[21]	-	45.96	-	90.71
FMIP	92.40	71.55	83.11	92.03
[20]	-	56.96	83.14	90.94

VI. Conclusion

In conclusion, our proposed study exhibits the effectiveness of Feature Map Importance Pruning (FMIP) as a method for optimizing (CNNs). By systematically analyzing and

eliminating redundant feature maps according to their contributions to the model's output, FMIP considerably shrinks computational costs without conceding performance. Our experiments on LeNet, VGG16, and ResNet50 networks utilizing MNIST, CIFAR10, and ImageNet datasets (see Fig.2a,b) reveal that FMIP can effectively streamline model architecture while preserving vital features. Explicitly, we perceived a reduction in computational cost with only an insignificant reduction in accuracy on average across the evaluated models.

These results emphasize the capability of FMIP to make CNNs further practical for utilization in resource-constrained environments, such as mobile devices and embedded systems. Future research could examine the application of FMIP to other CNN architectures and datasets, as well as investigate potential combinations with other optimization methods to further boost efficiency and performance.

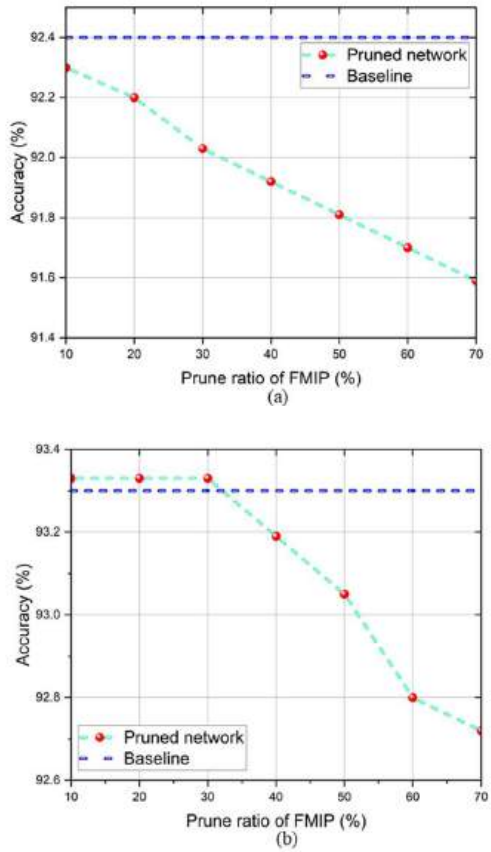


Fig. 2 Pruned ResNet50 on ImageNet (a) and VGG16 on CIFAR10 (b) datasets with variable prune ratio

ACKNOWLEDGMENT

I would like to express my heartfelt thanks to my professors and friends for their invaluable support and insights during the preparation of this paper. I also appreciate the resources and fa cilities provided by the University of Science and Technology of China, which greatly contributed to this research

REFERENCES

[1] L. Zhang, Z. Sheng, Y. Li, Q. Sun, Y. Zhao, and D. Feng, "Image object detection and semantic segmentation based on convolutional neural network," Neural Comput. Appl., vol. 32, no. 7, pp. 1949–1958, 2020, doi: 10.1007/s00521-019-04491-4.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[3] A. Kumar, S. Wang, A. M. Shaikh, H. Bilal, B. Lu, and S. Song, "Building on prior lightweight CNN model combined with LSTM-AM framework to guide fault detection in fixed-wing UAVs," Int. J. Mach. Learn. Cybern., vol. 15, no. 9, pp. 4175–4191, 2024, doi: 10.1007/s13042-024-02141-3.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[5] W. Fang, F. Zhang, V. S. Sheng, and Y. Ding, "A method for improving CNN-based image recognition using DCGAN," Comput. Mater. Contin., vol. 57, no. 1, pp. 167–178, 2018, doi: 10.32604/cmc.2018.02356.

[6] A. Kumar, B. Yin, A. K. Bhatia, A. K. Bhatia, and A. Rohra, "Structure Level Pruning of Efficient Convolutional Neural Networks with Sparse Group LASSO," vol. 12, no. 5, 2022, doi: 10.18178/ijmlc.2022.12.5.1111.

[7] A. M. Shaikh, Y. B. Zhao, A. Kumar, M. Ali, and Y. Kang, "Efficient Bayesian CNN Model Compression using Bayes by Backprop and L1-Norm Regularization," Neural Process. Lett., vol. 56, no. 2, pp. 1–19, 2024, doi: 10.1007/s11063-024-11593-1.

[8] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," pp. 1–14, 2015, doi: abs/1510.00149/1510.00149.

[9] A. Kumar, A. M. Shaikh, Y. Li, H. Bilal, and B. Yin, "Pruning filters with L1-norm and capped L1-norm for CNN compression," Appl. Intell., 2021, doi: 10.1007/s10489-020-01894-y.

[10] Y. He, X. Zhang, and J. Sun, "Channel Pruning for Accelerating Very Deep Neural Networks," Proc. IEEE Int. Conf. Comput. Vis., vol. 2017-Octob, pp. 1398–1406, 2017, doi: 10.1109/ICCV.2017.155.

[11] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures," 2016.

[12] M. Jaderberg and T. Green, "Population Based Training of Neural Networks".

[13] R. Abbasi-asl and B. Yu, "Structural Compression of Convolutional Neural Networks".

[14] C. Fernando et al., "PathNet : Evolution Channels Gradient Descent in Super Neural Networks".

[15] M. Lin et al., "HRank: Filter Pruning using High-Rank Feature Map".

[16] H. Zhuo, X. Qian, Y. Fu, H. Yang, and X. Xue, "SCSP: Spectral Clustering Filter Pruning with Soft Self-adaption Manners," Jun. 2018.

[17] A. Kumar, B. Yin, A. M. Shaikh, M. Ali, and W. Wei, "CorrNet: pearson correlation based pruning for efficient convolutional neural networks," Int. J. Mach. Learn. Cybern., vol. 13, no. 12, pp. 3773–3783, 2022, doi: 10.1007/s13042-022-01624-5.

[18] C. Zhao, B. Ni, J. Zhang, Q. Zhao, W. Zhang, and Q. Tian, "Variational convolutional neural network pruning," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019. doi: 10.1109/CVPR.2019.00289.

[19] Z. Huang and N. Wang, "Data-Driven Sparse Structure Selection for Deep Neural Networks," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018. doi: 10.1007/978-3-030-01270-0_19.

[20] S. Lin et al., "Towards optimal structured CNN pruning via generative adversarial learning," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019. doi: 10.1109/CVPR.2019.00290.

[21] S. Lin, R. Ji, Y. Li, Y. Wu, F. Huang, and B. Zhang, “Accelerating convolutional networks via global & dynamic filter pruning,” in IJCAI International Joint Conference on Artificial Intelligence, 2018. doi: 10.24963/ijcai.2018/336.

A GPGPU-Based Algorithm Acceleration System for ECG Signal Processing

Tianyu Huang*
Department of Electrical and Computer Engineering, Joint Laboratory of Zhuhai UM Science and Technology Research Institute - Lingyange Semiconductor Incorporated University of Macau
Macao, China
mc25002@connect.um.edu.mo

Zhijiong Wang
Department of Electrical and Computer Engineering, Joint Laboratory of Zhuhai UM Science and Technology Research Institute - Lingyange Semiconductor Incorporated University of Macau, Lingyange Semiconductor Incorporated
Macao, China
ckwong9456@foxmail.com

Guannan Hu
Joint Laboratory of Zhuhai UM Science and Technology Research Institute - Lingyange Semiconductor Incorporated Lingyange Semiconductor Incorporated
Zhuhai, China
Robert.hu@lyg-semi.com

Sio Hang Pun
Institute of Microelectronics, Department of Electrical and Computer Engineering, Joint Laboratory of Zhuhai UM Science and Technology Research Institute - Lingyange Semiconductor Incorporated University of Macau
Macao, China
lodgepun@um.edu.mo

Hung Chun Li
Joint Laboratory of Zhuhai UM Science and Technology Research Institute - Lingyange Semiconductor Incorporated Lingyange Semiconductor Incorporated
Zhuhai, China
albert.li@lyg-semi.com

Mang I Vai
Department of Electrical and Computer Engineering, Joint Laboratory of Zhuhai UM Science and Technology Research Institute - Lingyange Semiconductor Incorporated University of Macau
Macao, China
fstmiv@um.edu.mo

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899701

***Abstract*—Electrocardiogram (ECG) is an important non-invasive technique for diagnosing cardiovascular diseases (CVD). After acquiring the patients' raw ECG signal data, signal processing is essential for the diagnosis. Conventional ECG signal processing is usually performed serially, which takes a long time to process signals with large amounts of data. In this study, we have proposed a GPGPU-based algorithm acceleration system to improve the efficiency of ECG signal processing. This system utilizes OpenCL for parallel algorithm development on GPGPU. GPGPU executes certain steps in the QRS complex detection algorithm via parallel computing, thereby accelerating the algorithmic process. Experimental results showed that the system achieved a speedup of 1.279 compared to serial computing when processing 39 ECG signal sequences in parallel. In addition, our system has been equipped with a progressive transmission mechanism and a function that intelligently determines whether an algorithm can be accelerated. These features make this system practical not only in biomedical signal processing but also in engineering and research fields with large amounts of data.**

Keywords—GPU, Parallel Computing, ECG, Biomedical Signal Processing

I. INTRODUCTION

Cardiovascular disease (CVD) is a prevalent and potentially fatal condition, and early detection of CVD has consistently been a crucial subject in the field of medicine [1]. The electrocardiogram (ECG) is a method of recording the electrical activity of the heart over a specific duration. It provides an objective assessment of the heart's condition and serves as a crucial tool for medical research and diagnosing CVD [2]. Due to the progress in modern medicine and the increasing number of aging people, there has been a significant surge in the requirements for ECG testing in medical centers. Previously, cardiologists often conducted manual ECG analysis, but there is currently a growing trend towards computer-assisted analysis of ECG data. Typical computer-assisted systems for ECG analysis consist of signal preprocessing, heartbeat segmentation, feature extraction, and heartbeat classification [3].

A typical ECG waveform includes P waves, QRS complex, and T waves [4], among which the QRS complex has the highest amplitude and the most distinct shape. Accurate localization of the QRS complex is essential for heartbeat segmentation, so the detection and localization of the QRS complex are crucial in ECG signal processing [5]. Before segmenting the ECG signals into separate heartbeats, the raw signal usually contains a substantial volume of data. Preprocessing and waveform detection of the data before segmentation might be time-consuming. Therefore, efficiently and accurately processing and

*Corresponding author.

This work was funded by the National Key Research and Development Program of China (No.2021ZD0201300) and funded by The University of Macau (File no. MYRG2020-00098-FST, MYRG2022-00111-IME) and funded by The Science and Technology Development Fund, Macau SAR (File no. SKL-AMSV(UM)-2023-2025).

The authors would also like to acknowledge the financial support of the ZUMRI-Lingyange Semiconductor Joint Lab (CP-031-2022) and the Lingyange Semiconductor Incorporated, Zhuhai (CP-017-2022) and the Blue Ocean Smart System (Nanjing) Limited (CP-003-2023).

analyzing a large amount of data has become an important problem that needs to be solved.

We have recently completed a research and development project for a high-performance computing (HPC) acceleration system. This system includes a computing server equipped with a general-purpose graphics processing unit (GPGPU). The system utilizes a large number of processing elements in the GPGPU for parallel operations, resulting in reduced computation time for various signal processing algorithms. Based on this system, we developed a GPGPU parallel computing program to accelerate the processing of a large amount of ECG signals. Our signal processing algorithm acceleration experiments on this system showed encouraging results.

This acceleration system employs a hardware design based on an interchangeable GPGPU. To accommodate this design, the development of GPGPU parallel computing programs utilizes OpenCL [6], a heterogeneous computing programming framework compatible with multiple platforms. In addition, the system can intelligently detect parts of the code that can be run in parallel and generate the corresponding OpenCL code. The system is also equipped with a progressive transmission system that minimizes the data transfer time between the user's PC and the computing server.

These features make this acceleration system applicable to a variety of scenarios, including the processing of large amounts of data, the execution of complex algorithms, and the extension of computational durations, in addition to biomedical signal processing.

II. Related Work

Over the past four decades, major advances in ECG algorithms have allowed us to identify useful information in the signal more accurately. The Pan-Tompkins algorithm is the most famous of the algorithms used for QRS complex detection [7]. This algorithm efficiently eliminates noise and baseline drift from the signal while emphasizing the R waves in the signal. Li et al. introduced an approach that utilizes wavelet transform to identify feature points in ECG [8]. The algorithm efficiently identifies QRS complex in waveforms, even when there is baseline drift and significant noise, by utilizing the unique multi-scale wavelet transform.

In recent years, the need to collect biomedical signals has risen. This has resulted in larger amounts of data that need to be processed, leading to longer processing times caused by higher sampling frequencies and longer sample durations. Since the number of computing units in GPU is significantly bigger than that of CPU, researchers have proposed methods to use GPU to improve the efficiency of signal processing [9]. One of the primary conditions for obtaining performance acceleration on GPU is that the program can represent sufficient parallelism.

Stone et al. introduced a technique to improve the speed of MRI reconstruction algorithms using GPU [10]. The algorithm was executed on NVIDIA Quadro GPU for 3D image reconstruction, and it took 23 times longer to run on the CPU compared to the GPU. Dorgham et al. proposed a CUDA-based approach that utilizes GPU acceleration to generate digitally reconstructed radiographs (DRRs) for 2D/3D registration [11]. The experiment results demonstrated that GPU operations enhance the speed of DRR production and maintain registration precision at a clinically acceptable level. Both of their studies showed that GPUs hold promise for algorithm acceleration in biomedical engineering.

Utilizing CUDA to develop GPU programs is a convenient and user-friendly process [12]. However, the closed-source feature of CUDA restricts its usage to only NVIDIA devices. OpenCL is an open-source programming framework for heterogeneous computing that allows for portable programming across many platforms and has better multi-platform compatibility compared to CUDA. OpenCL provides a unified API that supports parallel computing on different brands of CPUs, GPUs, and a variety of other computing devices to improve computing efficiency. Stone et al. introduced OpenCL as a parallel programming standard for heterogeneous computing systems [13].

Fang et al. conducted a complete performance comparison between CUDA and OpenCL [14]. The experiments demonstrated that OpenCL can perform comparably to CUDA, and its cross-platform capability does not significantly affect the performance. This makes OpenCL an acceptable open-source alternative to CUDA. Sanida et al. suggested utilizing the OpenCL framework to accelerate image processing algorithms by implementing Sobel edge detection filters [15]. Experiments conducted with two convolution kernels at different picture resolutions demonstrate that OpenCL achieves a speedup of over 10 times for both kernel sizes. This illustrated the outstanding performance of OpenCL in image signal processing. Fan et al. developed and applied an OpenCL parallel computing algorithm for ECG analysis on a smartphone [16]. The algorithm exhibited a 5.22-fold increase in processing speed for long-term ECG data compared to serial computing. The result indicated that the program, designed using OpenCL, also performs well on mobile devices.

To the best of our knowledge, there are no studies of biomedical signal processing using acceleration systems based on the GPGPU-equipped HPC server. Our acceleration system enables processing signals with various durations and sampling frequencies. The system's architecture was designed in the form of a client and a server. The objective is to enable this system to be used in more scenarios than just biomedical signal processing.

III. Method

A. Architecture

The acceleration system contains the user's personal computer and an HPC server equipped with a GPGPU (Fig. 1). The user's PC establishes a connection to the server via network. When the user gives commands on their PC using the software we developed, the data will be transmitted across the network to the server's GPGPU for the specified processing. Upon completion of the processing, the results will be transmitted back to the user's PC. Since the system operates according to the user's commands, the system is not limited to processing specific ECG signals. It is capable of processing signals of any length and sampling frequency if it does not exceed the server storage limit.

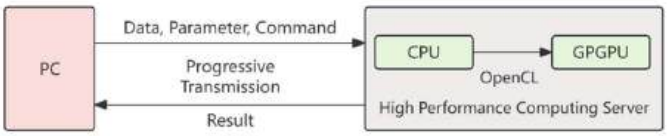


Fig. 1. System architecture

B. Intelligent Judgment Acceleration

Parallel computing can accelerate some algorithms but not all of them. Thus, our HPC server has an intelligent judgment mechanism that can determine whether an algorithm can be accelerated on GPGPU. The judgment mechanism understands code behavior by analyzing code dependencies. It employs this analysis to figure out which code should be performed serially and which in parallel. The intelligent judgment mechanism can then generate OpenCL kernel code based on code that can be executed in parallel.

C. Progress Transmission

In fixed network conditions, the time required for data transmission is positively correlated with the size of the data volume. Consequently, transmission times increase with an increase in data volume. Our acceleration system employs a progressive transmission mechanism to reduce the time required for data transmission. Data is compressed before it is sent to the server. The multi-modal large model is utilized to choose the most appropriate data compression strategy, depending on the type of input raw data. The temporal signals and images are processed using the resampling method. For the ECG signals, the system employs a neural network-based method for compression. The signals are then decompressed after they are received by the HPC server.

D. OpenCL

We use the programming framework named OpenCL to develop parallel computing programs to call GPGPU for computing operations. OpenCL supports CPUs, GPUs, FPGAs, etc., from NVIDIA, AMD, Intel, and other device vendors, unlike CUDA, which only supports NVIDIA devices. When computing devices of the server are replaced, only a few modifications to the OpenCL code are needed to allow different devices to perform the same computational tasks.

E. Modification to the ECG Signal Processing Algorithm

Once the raw ECG signal data is transmitted to the HPC server, the GPGPU will execute the pre-set algorithm program to process these data. In this work, the system was employed to accelerate a QRS complex detection algorithm proposed by Pan and Tompkins. To utilize the benefits of GPGPU in parallel computing, the steps of the algorithm where there is a read-after-read dependency — namely, band-pass filtering, difference, squaring, and moving-window integration, have been redesigned. This enables these steps to be implemented using parallel methods. The redesigned algorithm's flow chart is shown in Fig. 2.

Fig. 2. Algorithm flow chart

The steps of waveform localization, threshold adjustment, and T-wave identification in this algorithm are inter-output data dependent and thus unsuitable for parallel computation. They can only be performed through the serial approach. Consequently, GPGPU does not accelerate these steps in our work.

F. Band-pass Filtering

Fast Fourier Transform (FFT)-based frequency domain filtering methods are more efficient than time domain convolution in parallel digital signal processing because the GPGPU can process the entire sequence in parallel. The acceleration system employs the clFFT library in OpenCL to call GPGPU to perform FFT operations. According to convolution theory, filtering a signal can be equivalent to convolving the signal with the unit sample response of the filter in the time domain. For discrete digital signals, the product of two sequences in the frequency domain after FFT is equivalent to the circular convolution in the time domain. To eliminate the aliasing effect and achieve linear filtering, it is necessary to pad these time domain sequences with a certain number of zeros. Then the product of the two sequences in the frequency domain after padding the zeros is equivalent to a linear convolution in the time domain, as shown in equation (1).

$$y(n) = x(n) * h(n) \overset{FFT}{\longleftrightarrow} Y(k) = X(k)H(k) \tag{1}$$

Where x and y are the time domain sequences of the input and output signal, h is the unit sample response of the filter. X and Y are the frequency domain sequences of the input and output signal, H is the frequency response sequence of the filter.

The frequency response sequence of the filter can be generated in advance using Python. In this work, the filter is an FIR filter with 5-15 Hz passband frequencies and 100^{th} order. The multiplication of two sequences is determined only by the current data points because it requires the pointwise multiplication of corresponding elements. This computing is considered to be an embarrassingly parallel algorithm [17]. To execute this algorithm on the GPGPU using OpenCL, the approach is to create an OpenCL work-item for every data point operation.

G. Difference and Square

To increase the speed of the difference operation, we chose backward difference, which is less computationally intensive. In OpenCL programming, a difference operation on an entire sequence in parallel can be broken down into subtracting the current data point from the previous data point in each work-item, as shown in equation (2). Similarly, the parallel squaring is that each work-item performs the squaring operation on one data point separately, as shown in equation (3).

$$y(n) = x(n) - x(n - 1) \tag{2}$$

$$y(n) = x^2(n) \tag{3}$$

H. Moving-window Integration

The squared signal needs to perform a moving window integration process with a window width of 150 ms. Each data

point is assigned a work-item, which obtains the value of the current point and the values of all points within 150 ms before the current point. The average of these points is then output as the result of the integration, as shown in equations (4) and (5). Moving-window integration is the final step in acceleration, and the integrated signal data is sent back to the user's PC for further processing.

$$y(n) = (1 / N) [x(n - (N - 1)) + x(n - (N - 2)) + \cdots + x(n)] \quad (4)$$

$$N = w \times fs \quad (5)$$

Where w is the width of the window and fs is the sampling frequency of the input signal.

I. *Parallel Processing of Multiple Signals*

In this acceleration system, the GPGPU can process multiple signal sequences simultaneously. The multiple ECG signal sequences are concatenated end-to-end and then transmitted to the GPGPU. For FFT and IFFT operations, the batch processing function in clFFT can be employed to perform transforms on multiple sequences in parallel. For the remaining steps, parallel processing of multiple sequences can be achieved by performing the operations independently for each data point in this composite sequence. More precisely, M×N work-items are required when M signals of length N are needed to process.

IV. Experiment

This section presents experiments to demonstrate the acceleration system's performance in accelerating ECG signal processing algorithms. An experimental group and a control group were established to assess the performance of the system. The former was processed using the GPGPU parallel computing program, while the latter was processed using the CPU serial computing program. After pre-testing, the results of these two operations on the HPC server are identical so the acceleration effect can be evaluated by comparing their execution times.

A. *Data Set*

To evaluate the system's performance, we selected ECG data from the MIT-BIH database for our experiments [18]. This database has 48 ECG signal data for 30 minutes. Each piece of data was sampled at 360 Hz and contained two leads. In the experiments conducted, the MLII lead was selected for processing, while in instances where this lead was absent, the V5 lead was utilized. The same dataset was utilized for both the experimental group and the control group.

B. *Equipment Specifications*

The hardware specifications for the experiment are as follows: The operating system installed in the HPC server is Ubuntu 22.04, and the server has 32GB RAM. The server's CPU is an Intel Core i7-11700. The GPGPU is BF400 from Blue Ocean Smart with 128GB of global memory.

C. *Experimental Setup*

After pre-testing, this system can perform parallel FFT and IFFT operations on up to 39 ECG signals in MIT-BIH simultaneously, so the input data for the experiment is from 100 to 220, a total of 39 signal data. These data were transmitted from a PC to the HPC server. The server processed these data in GPGPU and CPU, using parallel and serial methods respectively, according to the flow in Fig. 2. For each operation, the processing started with one piece of signal data and gradually increased the amount of data, repeating the same processing flow ten times for each additional piece of signal data. Then, the average of the ten execution times was calculated and recorded. The execution time for each operation and the execution time of the entire process in the server from reading the raw data to writing the results were recorded respectively. The processing was performed sequentially for serial operations, so the time spent on each step in a serial operation is the sum of the time spent on each piece of data in that step. Furthermore, this experiment only investigated the acceleration effect of GPGPU, so the data transmission time was not recorded.

D. *Result*

The raw and processed waveforms of ECG signal 100 are shown in Fig. 3.

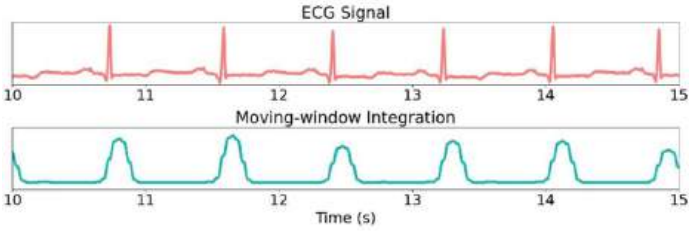


Fig. 3. The raw and processed waveforms of ECG signal 100

The QRS complex waveform in the raw signal corresponds to the rising edge in the processed waveform (Fig. 3). The execution time of each operation step and the entire process are shown in Fig. 4 and 5.

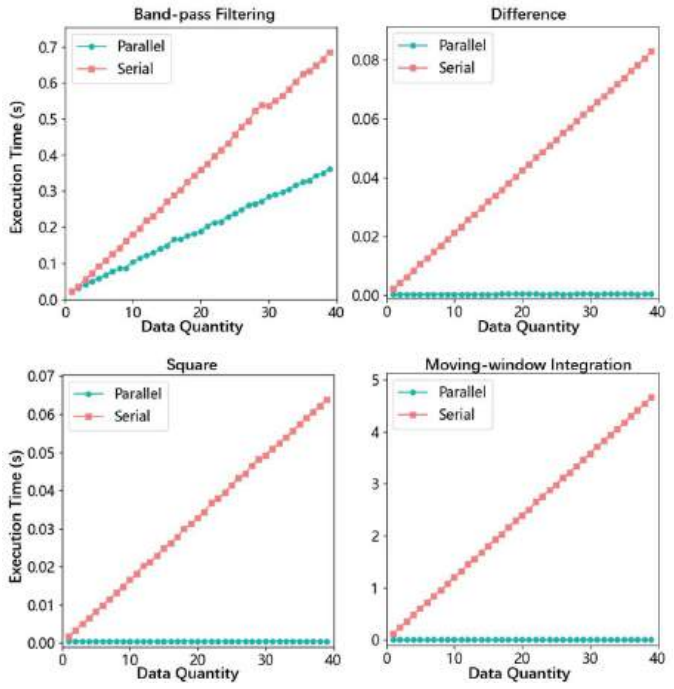


Fig. 4. Parallel and serial execution time of band-pass filtering, difference, square, and moving-window integration vs quantity of ECG data

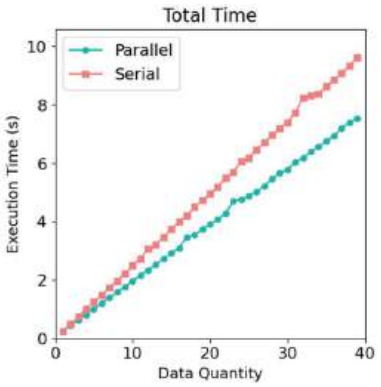


Fig. 5. Parallel and serial execution time of the entire process vs quantity of ECG data

For difference, square, and moving window integration, the execution time for serial operations is positively correlated with the amount of data (Fig. 4). In contrast, the execution time for parallel operations barely increases with the amount of data. This is because all these three operations are the typical embarrassingly parallel algorithm. For this algorithm, each data point is computed simultaneously. When the amount of data increases, the OpenCL program will assign more work-items to be involved in the computation. Therefore, with sufficient computing resources, an increase in the amount of data hardly affects the execution time. For band-pass filtering operations, whether serial or parallel, the execution time increases with the amount of data (Fig. 4). Since FFT and IFFT within band-pass filtering are not embarrassingly parallel algorithms, the amount of data will have an impact on the execution time. However, as the amount of data increases, the gap between the execution times of serial and parallel operations becomes progressively larger. Parallel band-pass filtering can also demonstrate great acceleration when a considerable quantity of data is present.

In terms of total execution time, the acceleration effect of parallel computing becomes more noticeable as the quantity of data increases (Fig. 5). However, the time consumption for parallel operations still increases as the amount of data increases. On one hand, this is due to the effect of the serial part of FFT and IFFT mentioned above. On the other hand, it is because the total time consumed includes not only the execution time of the algorithm but also the time to read and write data. Our program does not employ parallel operations to accelerate the reading and writing of data. Thus, as the quantity of data increases, the time required for reading and writing also increases.

The speedup is also an important benchmark for evaluating the acceleration effect of parallel computing [19], which is defined as (6).

$$S = T_s / T_p \tag{6}$$

Where S is the speedup of a system. *Ts* and *Tp* are the execution times for serial and parallel operations, respectively.

The execution performance between processing one piece of data and 39 pieces of data using serial and parallel operations is presented in Table I. Moving-window integration has the highest speedup, and band-pass filtering has the lowest speedup whether one or multiple pieces of data are processed. When processing 39 ECG signals, the speedup of the whole procedure reaches 1.279 and the speedup of moving-window integration can even reach 10,000. Overall, it can be concluded from Fig. 4, Fig. 5, and Table I that the speedup is positively correlated with the amount of data.

TABLE I. TIME TO PROCESS 1 AND 39 ECG SIGNAL DATA

1 ECG Signal	Band-pass Filtering	Difference	Square	Moving-window Integration	Total Time
Serial Time (s)	0.020300	0.002247	0.001814	0.121036	0.253511
Parallel Time (s)	0.022918	0.000386	0.000367	0.000401	0.250431
Speedup	0.886	5.817	4.950	301.911	1.012
39 ECG Signal	**Band-pass Filtering**	**Difference**	**Square**	**Moving-window Integration**	**Total Time**
Serial Time (s)	0.684212	0.082885	0.638427	4.659227	9.610391
Parallel Time (s)	0.361306	0.000480	0.000471	0.000464	7.513657
Speedup	1.894	172.785	1356.335	10050.100	1.279

E. Discussion

The experimental results demonstrate that the acceleration of difference, square, and moving-window integration is highly effective for our system. However, the speedup of band-pass filtering is relatively not high. The code analysis reveals that the speedup of band-pass filtering is relatively lower due to the limited acceleration of FFT and IFFT by GPGPU, in contrast to the other processes. Even with the strategy of performing FFT on multiple data simultaneously, the time consumption still increases with the amount of data. We believe that the reason for this is that FFT and IFFT are not embarrassingly parallel algorithms. clFFT library implementations still have serial parts of the computation, and thus, the acceleration of the band-pass filtering is limited by these parts.

In addition, due to constraints in hardware storage capacity, such as server memory and global memory of GPGPU, we could not simultaneously handle a larger amount of data in the experiment. Therefore, our future research will focus on optimizing the methods of storing data on the server and discovering the best storage methods to enhance data processing efficiency.

V. CONCLUSION

The utilization of GPGPU in our acceleration system for parallel signal processing has been proven to significantly improve the execution efficiency of the algorithms compared to serial processing on the CPU. Experimental results regarding ECG signal processing showed that the speedup of our system increases proportionally with the amount of data. The operations of difference, squaring, and moving-window integration exhibit a notable acceleration. In our future research, we will explore techniques to maximize the utilization of memory space while optimizing data read/write speeds. Our goal is to utilize the immense computational capabilities of GPGPU fully.

ACKNOWLEDGMENT

Declaration statement: During the preparation of this work, the author(s) used ChatGPT https://chatgpt.com to generate materials for background research and independent study, create materials that have been modified, copy-edit, and/or proofread the writing. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

REFERENCES

[1] C. W. Tsao et al., "Heart disease and stroke statistics-2023 update: a report from the American Heart Association," Circulation, vol. 147, no. 8, pp. 93-621, Feb. 2023.

[2] V. Jahmunah et al., "Computer-aided diagnosis of congestive heart failure using ECG signals - a review," Physica Med., vol. 62, pp. 95-104, Jun. 2019.

[3] W. Lu, H. Hou, and J. Chu, "Feature fusion for imbalanced ECG data analysis," Biomed. Signal Process. Control, vol. 41, pp. 152-160, Mar. 2018.

[4] X. Liu, H. Wang, Z. Li, and L. Qin, "Deep learning in ECG diagnosis: a review," Knowl.-Based Syst., vol. 227, pp. 107187, Sep. 2021.

[5] E. J. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, "ECG-based heartbeat classification for arrhythmia detection: a survey," Comput. Methods Programs Biomed., vol. 127, pp. 144-164, Apr. 2016.

[6] A. Munshi, "The OpenCL specification," in 2009 IEEE Hot Chips 21 Symp. (HCS), Aug. 2009, pp. 1-314.

[7] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," IEEE Trans. Biomed. Eng., vol. BME-32, no. 3, pp. 230-236, Mar. 1985.

[8] C. Li, C. Zheng, and C. Tai, "Detection of ECG characteristic points using wavelet transforms," IEEE Trans. Biomed. Eng., vol. 42, no. 1, pp. 21-28, Jan. 1995.

[9] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU computing," Proc. IEEE, vol. 96, no. 5, pp. 879-899, May 2008.

[10] S. S. Stone, J. P. Haldar, S. C. Tsao, W. M. Hwu, B. P. Sutton, and Z. P. Liang, "Accelerating advanced MRI reconstructions on GPUs," J. Parallel Distrib. Comput., vol. 68, no. 10, pp. 1307-1318, Oct. 2008.

[11] O. M. Dorgham, S. D. Laycock, and M. H. Fisher, "GPU accelerated generation of digitally reconstructed radiographs for 2-D/3-D image registration," IEEE Trans. Biomed. Eng., vol. 59, no. 9, pp. 2594-2603, Sep. 2012.

[12] M. Garland et al., "Parallel computing experiences with CUDA," IEEE Micro, vol. 28, no. 4, pp. 13-27, Jul. 2008.

[13] J. E. Stone, D. Gohara, and G. Shi, "OpenCL: a parallel programming standard for heterogeneous computing systems," Comput. Sci. Eng., vol. 12, no. 3, pp. 66-73, May 2010.

[14] J. Fang, A. L. Varbanescu, and H. Sips, "A comprehensive performance comparison of CUDA and OpenCL," presented at the 2011 Int. Conf. Parallel Process. (ICPP), Sep. 13-16 2011, pp. 216-225.

[15] T. Sanida, A. Sideris, and M. Dasygenis, "A heterogeneous implementation of the sobel edge detection filter using OpenCL," in 2020 9th Int. Conf. Modern Circuits Syst. Technol. (MOCAST), Sep. 7-9 2020, pp. 1-4.

[16] X. Fan, Q. Yao, Y. Cai, and Y. Li, "An efficient automatic electrocardiogram analysis method using smartphones," in 2018 IEEE Int. Conf. Consum. Electron. (ICCE), Jan. 12-14 2018, pp. 1-5.

[17] C. Gong, J. Liu, J. Qin, Q. Hu, and Z. Gong, "Efficient embarrassingly parallel on graphics processor unit," in 2010 2nd Int. Conf. Educ. Technol. Comput. (ICETC), Jun. 22-24 2010, vol. 4, pp. 400-404.

[18] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," IEEE Eng. Med. Biol. Mag., vol. 20, no. 3, pp. 45-50, May 2001.

[19] D. L. Eager, J. Zahorjan, and E. D. Lazowska, "Speedup versus efficiency in parallel systems," IEEE Trans. Comput., vol. 38, no. 3, pp. 408-423, Mar. 1989.

Enhancing Epidemic Prediction Using Simulated Annealing for Parameter Optimization in Infection Network Inference

1st Teun Hoven
Tilburg University
Tilburg, The Netherlands
t.d.hoven@tilburguniversity.edu

2nd Alberto Garcia-Robledo
Conahcyt-CentroGeo
Queretaro, Mexico
agarcia@centrogeo.edu.mx

3rd Mahboobeh Zangiabady
University of Twente
Enschede, The Netherlands
m.zangiabady@utwente.nl

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899726

***Abstract*—Understanding and predicting outbreaks of epidemics has become a major focus since COVID-19. Researchers have explored various methods, from basic curve fitting to complex machine learning techniques, to predict how the virus spreads. One promising method is the Network Inference-based Prediction Algorithm (NIPA), which uses the SIR-model and the least absolute shrinkage and selection operator to estimate how the infections spread over different regions. However, fine-tuning the regularization parameter of NIPA can be complicated because of the time-consuming process and sub-optimal result of k-fold Cross-Validation (CV). To overcome this, we suggest using Simulated Annealing (SA) to optimize NIPA's regularization parameter and find an optimal value for the curing probability. Our study aims to combine SA with NIPA to make the process of choosing the optimal value for the parameters more effective. The results of the research show that the accuracy is improved and therefore indicate that SA is an acceptable alternative to CV, regardless of the computation time being higher.**

***Index Terms*—Hyper-parameter optimization, Regularization parameter optimization, Least Absolute Shrinkage and Selection Operator (LASSO), Network Inference-based Prediction Algorithm (NIPA), Simulated Annealing (SA)**

I. Introduction

Since COVID-19 spread around the world, many research has been conducted in predicting the spread of the pandemic. Predicting how the spread of a virus will evolve is difficult, as it is comparable to weather forecasts and subject to fundamental limits [1]. Many researchers have developed methods to try to predict the spread of COVID-19. This research ranges from simple approaches, such as fitting the number of infections to a sigmoid curve, to using statistical approaches, network-based approaches, machine learning algorithms, and parameter estimations in compartmental models such as the SIR-model [2]. In 2022, Achterberg et al. compared the precision of different network-based techniques to predict cases of COVID-19. The prediction algorithms that were compared in the paper were long short-term memory, Gompertz function, Hill function, logistic function and Network Inference-based Prediction Algorithm (NIPA). In the end, the NIPA proved to be the algorithm with the best accuracy. [2].

At the core of the NIPA algorithm, is the resolution of a Least Absolute Shrinkage and Selection Operator (LASSO) problem to find a network of infection probabilities between pairs of geographical regions. A key parameter in this formulation is the regularization parameter, which determines the network's "sparsity". To find an optimal value for the regularization parameter, k-fold cross-validation or just Cross-Validation (CV) is often used. CV is a method to split the dataset into P amount of sets and for each set, divide it into a train and test set (often 80% and 20% respectively). The model is then trained on the train set and the evaluated on the test set. Although CV is the standard, the resulting value can result in unstable predictions and is computationally expensive to run [3].

In this paper, we propose using the Simulated Annealing (SA) [4] algorithm to find an optimal value for the curing probability and the regularization parameter without using the CV technique which has been used in the implementations of NIPA [2], [5]. LASSO in combination with SA does not have as much research as other regularization parameter optimization algorithms. However, in the research that has been done, the combination with SA has shown promising results against standard LASSO [6], [7]. For this reason, we are researching whether SA can be implemented for the NIPA as well.

We aim to contribute to the scientific community in three ways:

1) We propose a framework to use SA with the NIPA and therefore, LASSO.
2) We define an algorithm to more accurately choose optimal values for the curing probability and the regularization parameter.
3) We show how this framework can produce more accurate prediction of COVID-19 cases with the NIPA.

The structure of the research paper is as follows: In Section II we will discuss previous and related research on the use of SA for hyper-parameter optimization and in combination with LASSO. Section III will first describe the algorithms

and definitions used in the research and secondly, we will describe the approach taken to get the results. The results of the research, will be presented in Section IV. In Section V, we discuss the limitations of the research and what would be interesting to take into consideration for further research. At last, in Section VI the conclusion of the research is given.

II. Related Work

In this section we go over previous and related work of other researchers.

Finding optimal values for hyper-parameters by using SA has been researched for some time. In 2007, Lin et al. [8] proposed a SA method to determine parameters in a support vector machine. This method aims to search for the optimal value of the parameters to maximize the accuracy rate. This approach resulted in a higher classification accuracy rate than performing grid search when finding values for the parameters.

A proposed method called Evolutionary Simulating Annealing LASSO Logistic Regression (ESALOR) was proposed by Tutun et al. in 2016 [6]. This method uses a hybrid meta-heuristic optimization approach, where they prevent over-fitting of the coefficients of logistic regression by using regularization (LASSO) and to optimize these coefficients they use the evolutionary strategy of SA. It shows promising results, being more accurate in classifying readmission of diabetic patients than the other methods (support vector machines, artificial neural networks, naive Bayes algorithm and logistic regression). The researchers did not compare their method with other LASSO techniques, meaning that there is no proof if ESALOR is as accurate as LASSO without SA.

Another paper from 2016, from Zhang et al. [7] proposed a method in which they describe the dynamic shrinking of LASSO resembling an annealing process. Due to this relation, their method integrates a similar type of regularization optimization by using adaptive weights and this results in their solution path being different than the traditional LASSO solution path.

An interesting research about hyper-parameter optimization came from Bertrand et al. in 2020 [9]. It discusses how difficult setting the regularization parameter of LASSO-type estimators is. The paper then dives into the differentiation of the LASSO which allows the researchers to select the hyper-parameter through standard gradient-descent. This method can also scale to a high number of hyper-parameters.

With regards to approaching SA, Guilmeau et al. [10] reviewed different types of SA algorithms and also proposed a new technique which combines two previous SA methods, Fast Simulated Annealing (FSA) [11] and Sequential Monte Carlo Simulated Annealing (SMC-SA) [12]. This combination should allow better state space exploration from FSA while remaining the meaningful exchange between particles from the SMC-SA.

III. Definitions and Methods

In this section, the definitions, algorithms and the approach of the research will be explained and discussed.

A. Algorithms & Definitions

1) NIPA algorithm overview: The well-performing NIPA algorithm uses the Susceptible, Infected and Recovered (SIR) model to predict the spread of COVID-19. The NIPA first infers a matrix that is used as the infection probability between regions when individuals come in contact. After the infection probability matrix is inferred, the NIPA will iterate over each time step and calculates the fraction of susceptible, infected and recovered individuals against the population.

Specifically, the task of NIPA is to estimate each infection probability β_{ij} of the infected people of region j to susceptible people in region i. To infer the network of infections between regions, NIPA uses LASSO. For inferring the network, we have to solve the LASSO for each row i to find the vector of weights, in NIPA's case the infection probabilities matrix B, that minimizes the quadratic error of the linear system:

$$\min_{\beta_i} \left\| y_i - X_i\beta_i \right\|_2^2 \text{ subject to } \sum_i |\beta_i| \leq c, \tag{1}$$

where y_i are the responses, X_i are the predicted variables, β_i are the LASSO estimates, which the sum of these estimates are subjected to a constrained c [13]. This equation can be written in the orthonomal design as:

$$\min_{\beta_i} \left\| y_i - X_i\beta_i \right\|_2^2 + \rho \sum_i |\beta_i|, \tag{2}$$

where ρ is the regularization parameter and for every c of the constraint of Equation 1, there exists a corresponding value for ρ. Equation 2 can be divided into two parts, the Ordinary Least Square (OLS) estimate $\left\| y_i - X_i\beta_i \right\|_2^2$ and the penalty function $\rho \sum_i |\beta_i|$, for $\rho \geq 0$.

Due to this ℓ_1-norm penalty function, LASSO is able to shrink the OLS estimators (β_i) to zero. For this reason, LASSO can be regarded as a variable selection method with ρ as the shrinkage factor or the regularization parameter [14].

2) LASSO formulation in NIPA: As previosily stated, the NIPA is based on the SIR-model, where every individual is in one of three groups: *susceptible* ($\mathcal{S}$), *infected* ($\mathcal{I}$) or *recovered* ($\mathcal{R}$). The first group consists of individuals who have not yet been infected and will go from susceptible to infected when they come in contact with infectious individuals from the second group [5]. The third group consists of people that can not infect others, therefore it is often called "removed".

The network that NIPA is estimating is the matrix B of infection probabilities from the SIR viral state observations $v_i(1), ..., v_i(n)$.

For every city i at time k, we obtain the SIR *viral state* by $v_i(k) = (\mathcal{S}_i(k), \mathcal{I}_i(k), \mathcal{R}_i(k))^T$ [5], where $\mathcal{S}_i$ corresponds to the fraction of susceptible people in region i, $\mathcal{I}_i$ to the fraction of infectious and $\mathcal{R}_i$ to the fraction of recovered individuals.

For every city i at any discrete time k we denote the 3×1 viral state by:

$$v_i(k) = \begin{pmatrix} \mathcal{S}_i(k) \\ \mathcal{I}_i(k) \\ \mathcal{R}_i(k) \end{pmatrix} \tag{3}$$

In the equation above, it holds that $\mathcal{S}_i(k)+\mathcal{I}_i(k)+\mathcal{R}_i(k) = 1$ due to the components of the equation corresponding to the fraction of susceptible, infected and recovered people, respectively. To predict the amount of infected individuals, the *viral state* is updated each time step k according to:

$$\mathcal{I}_i(k+1) = (1-\delta_i)\mathcal{I}_i(k) + (1-\mathcal{I}_i(k)-\mathcal{R}_i(k))\sum_{j=1}^{N}\beta_{ij}\mathcal{I}_j(k) \tag{4}$$

$$\mathcal{R}_i(k+1) = \mathcal{R}_i(k) + \delta_i\mathcal{I}_i(k) \tag{5}$$

$$\mathcal{S}_i(k+1) = 1 - \mathcal{I}_i(k) - \mathcal{R}_i(k) \tag{6}$$

In Equation 4, β_{ij} is an element of matrix B and denotes the infection probability between regions i and j. The δ_i in Equations 4 and 5 denotes the curing probability of region i. These two probabilities are unknown and are based on the viral state $v_i(1), ...v_i(n)$, the estimations $\hat{\delta_i}$ and $\hat{\beta_{ij}}$ can be found by using CV and LASSO [5]. The infection probability β_{ij} specifies the probability of getting infected when infected individuals in region j come in contact with susceptible individuals in region i, where the infection probability matrix between regions is given by an $N \times N$ matrix:

$$B = \begin{pmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{N1} & \beta_{N2} & \dots & \beta_{NN} \end{pmatrix}, \tag{7}$$

where each element represents a probability $0 \leq \beta_{ij} \leq 1$. This matrix is estimated by the LASSO [13]. The network inference approach is suitable for the compartmental epidemic models like the SIR-model. In the equations of the SIR-model (4, 5, 6), it appears that β_{ij} is linear, but $\mathcal{S}_i$, $\mathcal{I}_i$ and $\mathcal{R}_i$ do not. From these equation, the infection probabilities β_{ij} satisfy

$$V_i = F_i \begin{pmatrix} \beta_{i1} \\ \vdots \\ \beta_{iN} \end{pmatrix} \tag{8}$$

for all cities $i = 1, ..., N$. The matrix V_i $(n-1 \times 1)$ and F_i $(n-1 \times N)$ are given by

$$V_i = \begin{pmatrix} \mathcal{I}_i(2) - (1-\delta_i)\mathcal{I}_i(1) \\ \vdots \\ \mathcal{I}_i(n) - (1-\delta_i)\mathcal{I}_{n-1}(1) \end{pmatrix} \tag{9}$$

and

$$F_i = \begin{pmatrix} \mathcal{S}_i(1)\mathcal{I}_1(1) & \dots & \mathcal{S}_i(1)\mathcal{I}_N(1) \\ \vdots & \ddots & \vdots \\ \mathcal{S}_i(n-1)\mathcal{I}_1(n-1) & \dots & \mathcal{S}_i(n-1)\mathcal{I}_N(n-1) \end{pmatrix} \tag{10}$$

To infer the network based on the equations above, we use the LASSO (Equation 2)

$$\min_{\beta_{i1},...,\beta_{iN}} \left\| V_i - F_i \begin{pmatrix} \beta_{i1} \\ \vdots \\ \beta_{iN} \end{pmatrix} \right\|_2^2 + \rho_i \sum_{j=1, j\neq i}^{N} \beta_{ij}, \tag{11}$$
$$\text{s.t. } 0 \leq \beta_{ij} \leq 1, j = 1, ..., N$$

where for each region i there exists an optimal regularization parameter ρ_i. With the LASSO, we can estimate the infection probabilities by using regression. The LASSO works by minimising the OLS with an ℓ_1-norm constraint given by the $\rho_i \sum_{j=1, j\neq i}^{N} \beta_{ij}$ part of Equation 11. The regularization parameter ρ_i determines how many values in the resulting $N \times N$ matrix go to zero.

3) Cross-Validation: In previous research of the NIPA, the regularization parameter and curing probability were selected by CV [2], [5]. The researchers have probably chosen this technique because CV is a common technique for deciding values of unknown parameters [14]. The procedure of CV starts by defining a set of possible choices for the regularization parameter. For NIPA, a set Θ_i with 100 logarithmically equidistant values from a range based of V_i and F_i [2], [5]. The second step is dividing the data into P equal-sized parts (non-overlapping) with approximate s individuals in each part. Each part is then taken as the validation data denoted by x_v and $y_v (v \in \{1, ..., P\})$, the remaining parts $P-1$ we denote as training data x_t and y_t. Per value in Θ_i, we train the model on the training data and then try to predict y_v as $\hat{y_v}(\rho_i)$ with the validation data. The predictive ability is then evaluated and the optimal regularization parameter $\rho_{i,opt}$, which minimizes a certain metric of the prediction accuracy, is chosen to be the regularization parameter [14].

4) Simulated Annealing: The SA algorithm is inspired by the physical process of metallurgy and uses terminology coming from the fields of physics. Atoms in metal experience large disorded movements when the metal is heated. When the metal is cooled down steadily, the movement weakens and the atoms stabilize around a certain position where the energy is minimal. This process of slow cooling the metal is called *annealing* [15]. To understand this intuitively, and therefore also the algorithm: at high temperature, the search (atoms) go through large random movements, resulting in exploration of a wide range of possible configurations, even with high energy. Due to these large random movements, high-energy configurations can be reached even though it might not be the preferred position. When the temperature lessens, the amount of movements also reduces, meaning that the search will prefer low-energy configurations and finally "freezes" into a low-energy minimum which can be a global minimum, although this is not guaranteed. Kirkpatrick et al. came up with the idea in 1983 to use this physics process to search for the global optimum and since then, SA, also called Classic Simulated Annealing (CSA), has been used in many optimization problems [16].

The CSA algorithm is surprisingly simple to implement and functions well with multiple parameters. The pseudocode of the CSA algorithm can be found in Algorithm 1. The algorithm utilizes an objective function of an optimization problem. For each iteration, the algorithm chooses a new location based on the visiting distribution, assumed to be Gaussian (local search) and computes the energy E at that location. If the energy is lower compared to the previous location, the move is always accepted. When the move results in a higher energy, a probability will be calculated that determines if the move will be accepted. This probability is given as follows:

$$P = 1 - e^{\frac{\Delta E}{T_k}} \tag{12}$$

where P is the acceptance probability, ΔE is the difference of energy between the new and the old location and T_k is the temperature at step k [17]. When T is high, it is more likely that a move to a location with a higher energy is accepted. Through the iterations the temperature T will decrease and eventually reach zero. The formula for the decreasing temperature can be linearly, although there has been much research in CSA with non-linear cooling schedules [17]. SA is a widely used global optimization method and much research has been done on SA which led to many versions of the algorithm.

Algorithm 1 Simulated Annealing

Initialization of x_0
for $k = 1, ...$ **do**
  Generate a candidate $y_k \sim G(x_k, dy)$
  Compute the acceptance probability $p_k = e^{\frac{\Delta E}{T_k}}$
  Set $x_{k+1} = \begin{cases} y_k \text{ with probability } p_k \\ x_k \text{ with probability } 1 - p_k \end{cases}$
end

The first modified SA was four years after its introduction in 1983 [11]. The proposal of Szu uses a Cauchy-Lorentz distribution (*semi*-local search) as the visiting distribution that moves the location often in local search space but can occasionally jump to a point further away. With this addition, Szu and Hartly showed that the cooling schedule could be much faster and therefore this algorithm was called Fast Simulated Annealing (FSA). To decrease the temperature at a faster rate, FSA generalizes the accept-reject rule (see Equation 12) to any *acceptance function* q [10]:

$$P_k = q(p_k), \text{ where } pk := (\frac{\Delta E}{T_k}). \tag{13}$$

With this generalization, SA is allowed to decrease more slowly and the convergence will happen with a faster cooling schedule.

In 1996, Tsallis and Strariolo published their research on generalizing each part of the SA algorithm [18]. With the equations in their research, it was possible to obtain the CSA and FSA as well as a faster convergence with certain values for the Generalized Simulated Annealing (GSA) [18]. The GSA uses two artificial temperatures instead of the one temperature in CSA and FSA, where the shape of the distorted Cauchy-Lorentz distribution is controlled by q_v and the acceptance probability is controlled by q_a. The visiting distribution is as follows:

$$g_{q_v}(\Delta x(t)) \propto \frac{(T_{q_v}(t))^{-D/(3-q_v)}}{[1 + (q_v - 1)(\Delta x(t))^2/(T_{q_v}(t))^{2/(3-q_v)}]^{1/(q_v-1)+(D-1)/2}}, \tag{14}$$

where D is the dimension of the variable space and T_{q_v} is the visiting temperature. The parameter q_v also controls the rate of cooling:

$$T_{q_v}(t) = T_{q_v}(1)\frac{2^{q_v-1} - 1}{(1+t)^{q_v-1} - 1}, \tag{15}$$

where q_a controls the acceptance probability which is a generalized Metropolis algorithm [19]:

$$P_{q_a} = \min\{1, (1 - (1 - q_a)\beta\Delta E)^{1/1-q_a}\}, \tag{16}$$

where $\beta = \frac{1}{KT_{q_a}}$. To get CSA and FSA from this generalized form, we set $q_v = 1$ and $q_a = 1$ to get CSA and to get FSA, we set $q_v = 2$ and $q_a = 1$.

5) Evaluation: To evaluate the performance of the prediction ability of NIPA with CV and NIPA with SA, we must define the evaluation metric. Measurement of performance for forecasting is not an easy task; researchers and forecasters have not yet agreed on a measurement technique that should be preferred [20]. Each method of evaluation has its own strengths and weaknesses, making the measurement technique different for each problem. Much research has been conducted on the accuracy of the prediction, as it is the most important criterion to select a prediction technique. Most forecast methods use point forecast which is why researchers are focused on identifying measures to show the accuracy of point forecasts [21].

Popular metrics for forecast evaluation is the mean square error (MSE), the mean absolute percentage error (MAPE) and the symmetric mean absolute percentage error (sMAPE). Where MSE penalizes large errors but is sensitive to outliers and can not handle multiple time series, MAPE handles these drawbacks but creates another. If the true values of the forecast go to zero, we get a large number and it is possible for MAPE to become undefined [20]. To counteract this, the sMAPE can be used. sMAPE is also commonly used by forecast researchers to quantify the predictions [2], [21]. After some discussion and consideration, we have chosen to evaluate our results according to the mean square error which is defined as:

$$e_{\text{MSE}}(t) = \frac{1}{N}\sum_{i=1}^{N}(Y_i(k) - \hat{Y}_i)^2, \tag{17}$$

where Y_i is the true data and $\hat{Y}_i$ is the predicted data.

B. Approach

1) LASSO: To understand how to optimize the NIPA [5], we first have to understand NIPA by examining the working of the algorithm. The NIPA finds its base functionality in

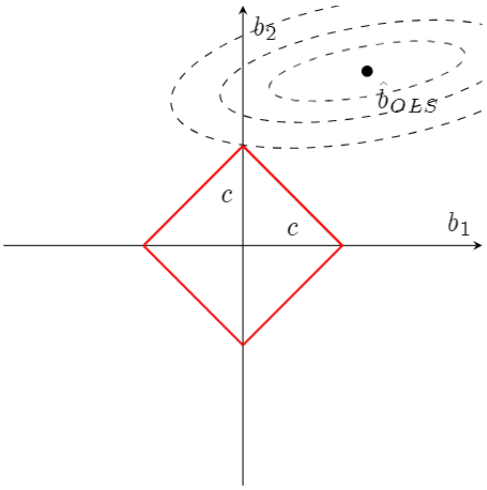


Fig. 1. The LASSO ℓ_1 constraint (red) in a two-dimensional parameter space (b_1 and b_2), where $\hat{b}_{OLS}$ is the (unconstrained) ordinary least square estimate and the contours show the estimates of b with equal variance in terms of squared error loss. c corresponds to the constraint in Equation 1.

the LASSO [13] (Equation 2). With this algorithm, we are minimising the sum of squared elements which is subjected to an ℓ_1-norm penalty. By using this constraint in the equation, it is possible for LASSO to shrink coefficients to zero, creating a sparse matrix of the coefficients and because of this, it can also be regarded as a variable selection method [14].

LASSO can achieve the shrinkage to zero because of the penalty function $\rho \sum_i |\beta_i|$ which results in a constraint with sharp edges (see Figure 1). When ρ is large, all the coefficients will go to zero and when ρ goes to zero, the constraint will be non-existent and the coefficients will equal the OLS estimation.

2) NIPA: How LASSO is integrated into NIPA has been shortly introduced in Section III-A2. Where for each region i there is a specific regularization parameter ρ_i. As mentioned in the previous section, when this parameter is too high, it will result in all infection probabilities β_{ij} for regions i and j to be zero and when ρ_i is too small, the resulting matrix will not be a sparse representation of the infection probabilities between the regions. To confirm the correct regularization parameter is chosen without overfitting the model, Prasse et al. established a range where $10^{-4}(2||F_i^T V_i||_\infty) \leq \rho_i \leq 2||F_i^T V_i||_\infty$ [5]. Within this range, it is possible for all coefficients to go to zero and at the same time preventing the infection probabilities to be above zero for every β_{ij}.

3) Simulated Annealing: Instead of CV, which is often used for determining the regularization parameter for LASSO [14], we propose using SA, introduced in Section III-A4, to find the optimal value for the curing probability δ_i and the regularization parameter ρ_i for every region i. The first part of implementing SA is choosing whether to implement it ourselves or use a third-party library. We have chosen for the latter, as a third-party has optimized the algorithm and our own implementation could be faulty. The implementation of the SA algorithm has been done through the method dual_annealing of the SciPy library [22]. Dual annealing uses the generalization of CSA and FSA and combines it with a local search on each accepted location. As mentioned in Section III-A4, this implementation of SA uses a distorted Cauchy-Lorentz visiting distribution (see Equation 14), where parameter q_v also controls the temperature schedule as seen in Equation 15. For the acceptance probability, it utilizes Equation 16. A crucial step in implementing the SA algorithm is selecting correct values for the parameters q_v, q_a, the initial temperature T_0 and the starting point x_0.

For the parameter q_v, that controls the visiting distribution, it is important that it is not too low ($q_v \leq 1$), because the visiting distribution will be confined to a local search space. If $1 < q_v \leq 2$ (FSA), we get a semi-local search where the search is more efficient than CSA but getting trapped at a local region can still occur. To search the space homogeneously, we set $q_v = 2.62$ which creates the Tsallis-Stariolo form of the Cauchy-Lorentz distribution. With this form of the Cauchy-Lorentz, the search has the possibility of long jumps even at low temperatures. Due to this characteristic, it has a high probability of finding the global minimum instead of being trapped at a local minimum [19].

The value of the parameter that controls the acceptance probability, q_a, is less important as this value determines the initial condition for the temperature. We have chosen for $q_a = -5$ as this is also proposed by Tsallis and Stariolo [18]. With this value we can determine a suitable value for the initial temperature. The following steps have been taken to calculate T_0 [15], [23]:

1) Perform 100 steps of the algorithm.
2) Compute the average difference in energy ($\hat{\Delta E}$).
3) Choosing an initial acceptance probability.
4) Calculate the initial temperature T_0.

Where the initial acceptance probability has been chosen to be 0.8 as to allow the algorithm to make greater jumps at the start. If the initial acceptance probability is lower, the algorithm will not be able to make long jumps and when the probability is higher, it will accept too many bad moves [24]. We have chosen this initial probability because the starting point of the algorithm will be the lower bound of the range for the regularization parameter given in Section III-B2. This starting point has been picked due to the large probability that the optimal value is in close proximity and that the algorithm does not quickly go to the search space where all values of the infection probability matrix go to zero.

Finally, we calculate the initial temperature by inserting the values mentioned above into equation 16. We assume that $K = 1$, preventing another optimization problem. The algorithm is then run over the NIPA model, the result is a local minimum, with a high probability of being the global minimum.

IV. Results

To get the results, we have implemented the NIPA algorithm with CV and the NIPA with SA with the parameters mentioned in Section III-B3.

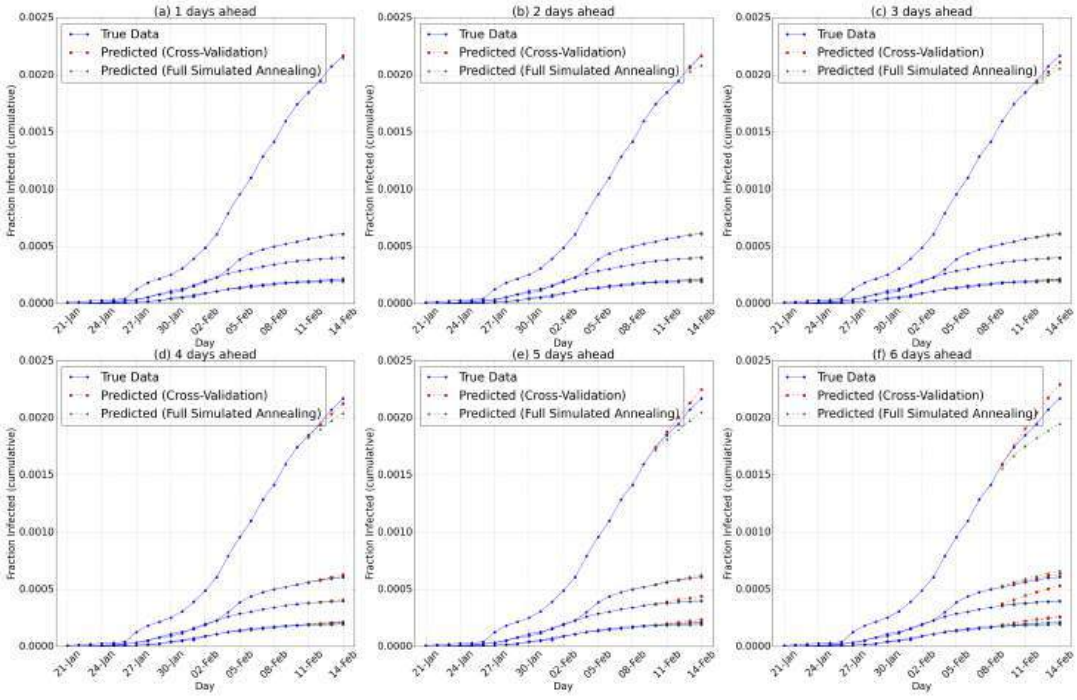


Fig. 2. The prediction of COVID-19 pandemic in Hubei by NIPA with CV and NIPA with SA. Here $N = 16$, but only five are shown. The amount of days predicted is in the title of the graph where $1 \leq m \leq 6$. Each point is the cumulative COVID-19 cases in Hubei.

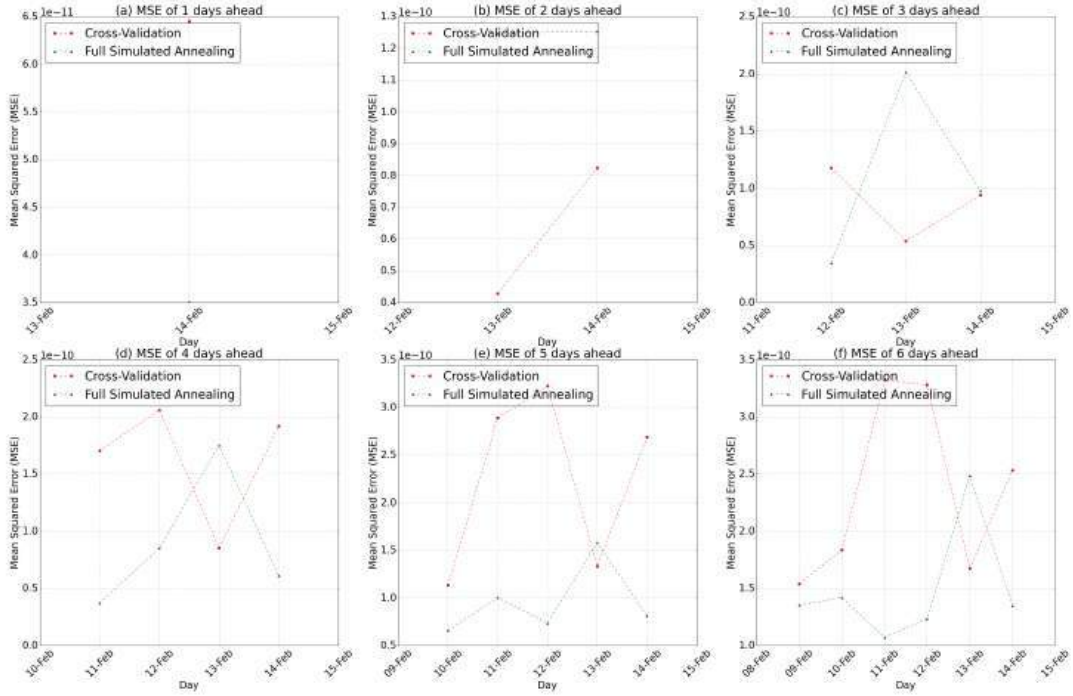


Fig. 3. The evaluation metric of the predictive ability of NIPA with CV and NIPA with SA. The amount of predicted days which is evaluated can be found in the title of the graph where $1 \leq m \leq 6$. Each point corresponds to the MSE between the predicted amount and the true amount of cases (lower is better).

The results will look at the difference in evaluation in the Chinese province of Hubei. The first case was on January 21 and the data lasts until February 14, since the local government changed their diagnosing policy causing an erratic increase in the number of reported cases on February 15 [2].

For the evaluation of both methods, we removed data points for a fixed amount of days m. The NIPA model with CV and with SA are then iterated over these m days. How the disease truly evolves (cumulative) can be seen in Figure 2 by the *blue* line. In the same figure, the predictive ability of NIPA with CV and NIPA with SA (*red* and *green* respectively) can be seen against each other. In Figure 2, it can be seen that both methods have approximately the same predictive results. To clearly see what the differences are between the two methods, we look at the evaluation metrics in Figure 3. These graphs show that for most predictions there is a slight improvement when NIPA is combined with the SA algorithm, except when the NIPA needs to predict two days ahead (Figure 3b). This difference can be explained by the stochastic approach of the LASSO and SA algorithm, where each run has different outcomes.

Besides the predictive ability of both methods, we also want to look at the computational difference. We run the algorithm multiple times with both optimization techniques. It took NIPA with CV, *1 minute and 7 seconds* to infer the infection probability matrix of Hubei, China. Using NIPA with SA, it took on average *4 minutes and 24 seconds* to infer the matrix. The difference in computation time can be attributed to the fact that CV can be executed in parallel while SA has to wait for each iteration, to execute the next. This has a big impact on the performance. The many iterations SA needs to find the optimal value can be another reason why the computation time for SA is higher. Where CV only tries an x amount of values within the range, SA will search throughout the range.

V. Discussion

Optimizing LASSO is possible through multiple approaches, we have chosen the SA algorithm for this research but it is possible to use Bayesian optimization as well. This optimization technique is, like SA, appropriate because of the way the LASSO estimates can be interpreted as a posterior mode. Because of this relation, a Bayesian LASSO has been proposed by Park and Casella [3]. In this approach, we could use the marginal maximum likelihood or hyperpriors for choosing the regularization parameter ρ_i. We could use a Gaussian process (GP) to very closely approximate the optimal value for the Bayesian LASSO which is called BO-GP [25]. To be able to implement this approach, we would have to rewrite LASSO, NIPA and then implement the BO-GP algorithm and would be ideal for further research on this topic.

The SA algorithm has parameters that can be tuned for better and more efficient results. During this research we have tuned these parameters according to research and some trial-and-error. With more time, it would be possible to experiment more with different values and combinations of these values. During the research, the training of the model was hindered as it sometimes tried to only use the curing probability to predict the amount of infected individuals and this resulted in trying out some penalties or constraints (e.g., doubling the score when all infection probabilities of the regions are 0). When more time is put into researching and defining a constraint for this problem, the accuracy would most likely improve.

In this research, we have chosen to implement SA only for the original NIPA. In previous research, by Achterberg et al., the formulation of NIPA was extended to include knowledge of the underlying contact network that was split into two algorithms, one where the prior knowledge is static (*NIPA static prior*) and one where it is dynamic (*NIPA dynamic prior*) [2]. It would be interesting to see the results from more countries as well as seeing the SA algorithm be implemented for the NIPA static prior and the NIPA dynamic prior that were introduced by Achterberg et al. in their research.

VI. Conclusion

In this research, we propose to use the NIPA prediction method with SA. Based on previous research, one region has been chosen to compare the results between NIPA with CV and NIPA with SA: Hubei, China. The results for this region provided valuable information about the use of NIPA with SA. First, the incorporation of SA in the NIPA. As we can see in the results, we can safely assume that the implemented SA works accordingly. The values of the parameters chosen in Section III-B3 are efficient enough to search for an optimal value for the regularization parameter and the curing probability even though the computation time is slower compared to CV.

The results that are shown in Section IV, show that the NIPA with CV and the NIPA with SA show close predictive ability. Looking closer at the predictions, the NIPA with SA has shown better accuracy in predicting the cases of COVID-19. As can be seen in the evaluation metrics in Figure 3, where we see that the NIPA with SA has better evaluations for each of the results except for predicting two days ahead.

We have shown that the SA algorithm improves the overall prediction ability against the CV technique which is used to choose a value for the regularization parameter for the NIPA. Although it has a better prediction accuracy, the computation time for the NIPA with SA is slower compared to the NIPA with CV. To determine which needs to be used is a trade-off between accuracy and computation time.

References

[1] K. R. Moran, G. Fairchild, N. Generous, K. Hickmann, D. Osthus, R. Priedhorsky, J. Hyman, and S. Y. Del Valle, "Epidemic Forecasting is Messier Than Weather Forecasting: The Role of Human Behavior and Internet Data Streams in Epidemic Forecast," *Journal of Infectious Diseases*, vol. 214, no. suppl 4, pp. S404–S408, Dec. 2016.

[2] M. A. Achterberg, B. Prasse, L. Ma, S. Trajanovski, M. Kitsak, and P. Van Mieghem, "Comparing the accuracy of several network-based COVID-19 prediction algorithms," *International Journal of Forecasting*, vol. 38, no. 2, pp. 489–504, Apr. 2022.

[3] T. Park and G. Casella, "The Bayesian Lasso," *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, Jun. 2008.

[4] D. Bertsimas and J. Tsitsiklis, "Simulated Annealing," *Statistical Science*, vol. 8, no. 1, Feb. 1993.

[5] B. Prasse, M. A. Achterberg, L. Ma, and P. Van Mieghem, "Network-inference-based prediction of the COVID-19 epidemic outbreak in the Chinese province Hubei," *Applied Network Science*, vol. 5, no. 1, p. 35, Dec. 2020.

[6] S. Tutun, S. Khanmohammadi, L. He, and C.-A. Chou, "A Meta-heuristic LASSO Model for Diabetic Readmission Prediction," Anaheim, California, May 2016.

[7] K. Zhang, S. Zhe, C. Cheng, Z. Wei, Z. Chen, H. Chen, G. Jiang, Y. Qi, and J. Ye, "Annealed Sparsity via Adaptive and Dynamic Shrinking," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, Aug. 2016, pp. 1325–1334.

[8] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng, "Parameter determination of support vector machine and feature selection using simulated annealing approach," *Applied Soft Computing*, vol. 8, no. 4, pp. 1505–1512, Sep. 2008.

[9] Q. Bertrand, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon, "Implicit differentiation of Lasso-type models for hyperparameter optimization," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, Jul. 2020, pp. 810–821.

[10] T. Guilmeau, E. Chouzenoux, and V. Elvira, "Simulated Annealing: a Review and a New Scheme," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. Rio de Janeiro, Brazil: IEEE, Jul. 2021, pp. 101–105.

[11] H. Szu and R. Hartley, "Fast simulated annealing," *Physics Letters A*, vol. 122, no. 3-4, pp. 157–162, Jun. 1987.

[12] E. Zhou and X. Chen, "Sequential Monte Carlo simulated annealing," *Journal of Global Optimization*, vol. 55, no. 1, pp. 101–124, Jan. 2013.

[13] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[14] Z. Li and M. J. Sillanpää, "Overview of LASSO-related penalized regression methods for quantitative trait mapping and genomic selection," *Theoretical and Applied Genetics*, vol. 125, no. 3, pp. 419–435, Aug. 2012.

[15] B. Chopard and M. Tomassini, "Simulated Annealing," in *An Introduction to Metaheuristics for Optimization*. Cham: Springer International Publishing, 2018, pp. 59–79, series Title: Natural Computing Series.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[17] A. Blum, C. Dan, and S. Seddighin, "Learning Complexity of Simulated Annealing," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Banerjee and K. Fukumizu, Eds., vol. 130. PMLR, Apr. 2021, pp. 1540–1548.

[18] C. Tsallis and D. A. Stariolo, "Generalized simulated annealing," *Physica A: Statistical Mechanics and its Applications*, vol. 233, no. 1-2, pp. 395–406, Nov. 1996.

[19] Y. Xiang and X. G. Gong, "Efficiency of generalized simulated annealing," *Physical Review E*, vol. 62, no. 3, pp. 4473–4476, Sep. 2000.

[20] D. Koutsandreas, E. Spiliotis, F. Petropoulos, and V. Assimakopoulos, "On the selection of forecasting accuracy measures," *Journal of the Operational Research Society*, vol. 73, no. 5, pp. 937–954, May 2022. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/01605682.2021.1892464

[21] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

[22] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[23] F. Busetti, "Simulated annealing overview," May 2001.

[24] S. Ledesma, G. Avia, and R. Sanchez, "Practical Considerations for Simulated Annealing Implementation," in *Simulated Annealing*, C. Ming, Ed. InTech, Sep. 2008.

[25] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.

Optimizing Unmanned Aerial Vehicle Paths with a Spider Wasp Algorithm

1st Zhuo Liu
China Ship Development and Design Center,
Wuhan, China
183288561@qq.com

2nd Mian Zheng*(corresponding author)
China Ship Development and Design Center,
Wuhan, China
zyclzmm@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899696

***Abstract*—Unmanned Aerial Vehicle (UAV) technology, due to its rapid environmental perception capabilities, swift data collection, and transmission, has significantly fueled extensive research in specialized scenarios. However, when executing reconnaissance missions, the UAV face risks if they traverse suboptimal flight paths, underscoring the criticality and complexity of optimal path planning. To address this challenge, this paper introduces an innovative path planning methodology dubbed Spider Wasp Optimization (SWO), a meta-heuristic algorithm deeply rooted in the natural behaviors exhibited by spider wasps during hunting, nesting, and mating rituals. By simulating these intricate behaviors, the SWO algorithm optimizes the UAV flight path points. To validate its efficacy, the proposed SWO algorithm is rigorously compared against two established and widely recognized metaheuristic approaches: Rapidly-exploring Random Tree (RRT) and Astar algorithms. Experimental outcomes conclusively demonstrate that SWO surpasses both algorithms in terms of minimizing the UAV flight path length and search time, thereby showcasing its superiority and potential for enhancing UAV performance in complex reconnaissance missions.**

Keywords—path planning, unmanned aerial vehicle, spider wasp optimization, rescue efficiency

I. Introduction

As mountainous areas become more popular for hiking and outdoor activities, there has been a rise in accidents, leading to increased mountain rescue missions [1][2]. Consequently, the escalating number of mountain rescue missions resulting from these accidents poses additional challenges for rescuers, particularly in navigating rugged terrain and confronting adverse weather conditions [3]. Locating individuals in need within the first 60 minutes is crucial for effective rescue [4]. Unmanned Aerial Vehicle (UAV), equipped with advanced technology, have become vital tools in these efforts, significantly enhancing rescue efficiency and speed [5][6][7][8][9][10]. However, path planning for UAVs presents a significant challenge, requiring the identification of the optimal route while considering various factors and constraints[11][12].

The primary challenge in path planning for UAV lies in identifying the optimal route while balancing various factors and constraints. Historically, UAV path planning has incorporated a variety of algorithms, such as the Artificial Potential Field (APF) [13], the Visibility Graph Algorithm (VGA) [14], the Particle Swarm Optimization (PSO) [15], and the Artificial Neural Network (ANN) [16], each with its unique strengths and limitations. APF approach models UAV movement as a virtual force field, guiding the UAV away from hazards and toward the target in real-time. However, it faces challenges such as local minimum points and target inaccessibility [17][18][19]. In the VGA method, the UAV is treated as a point, connecting it to the target and obstacles to form a visibility graph. This ensures the shortest path but lacks flexibility and requires remodeling upon changes [20][21]. In the PSO algorithm, the optimal solution is discovered by simulating the foraging behavior of birds to discover the optimal solution. It is simple, robust, and converges quickly but can fall into local optimal solutions[22][23][24][25]. ANN algorithm offers adaptability in complex environments and has parallel processing capabilities. However, its generalization ability is poor [26].

In this paper, we employ Spider Wasp Optimization (SWO) [27], simulating the spider wasp hunting, nesting, and mating behaviors, to implement fast path planning for the UAV. The algorithm has strong global search ability and local search ability, which can effectively prevent the algorithm from falling into local optimum. Compared with other algorithms, the SWO algorithm needs more parameters, and its adaptability will be better for complex environments.

II. Problem Modeling

The core of path planning [28] is the design of the model. The path and the smooth cost represent the energy consumption of UAV flights. The high cost and the threat cost are to determine whether the path is feasible. And the inspiration cost is introduced to effectively choose a direction closer to the endpoint. The path planning problem is considered as a cost function minimization problem [23], and five types of costs are proposed as follows.

A. Cost Function

To ensure the UAV's swift arrival at the rescue sites, identifying the optimal path is essential. Considering x_{all}, y_{all}, and z_{all} as sequences representing the x, y, and z coordinates of points, respectively, where i indicates the index in these sequences. For each adjacent point pair (i, i+1), their difference vectors can be calculated by:

$$diff_i = \begin{bmatrix} x_{all}(i+1) - x_{all}(i) \\ y_{all}(i+1) - y_{all}(i) \\ z_{all}(i+1) - z_{all}(i) \end{bmatrix} \quad (1)$$

The Euclidean distance is represented as:

$$\| diff_i \| = \sqrt{\begin{matrix}(x_{all}(i+1)-x_{all}(i))^2+(y_{all}(i+1)- \\ y_{all}(i))^2+(z_{all}(i+1)-z_{all}(i))^2\end{matrix}} \quad (2)$$

Finally, the path cost J_1 is obtained by:

$$J_1 = \sum_{i=1}^{n-1} \| diff_i \| \quad (3)$$

1) INSPIRATION COST

To minimize the path cost, an inspiration cost is introduced. It can ensure that the selected path points are nearest to the endpoint and directs the algorithm's search toward the endpoint. The inspiration cost is represented as:

$$J_2 = \sum_{i=1}^{n} \sqrt{(x_i - x_f)^2 + (y_i - y_f)^2 + (z_i - z_f)^2} \quad (4)$$

Where (x_f, y_f, z_f) represents the endpoint.

2) HIGH COST

UAV restricts their flight altitude to a specific range during rescue missions, and this altitude limitation strategy helps UAV optimize their performance [29]. The altitude cost of each segment is represented as:

$$H_{ij} = \begin{cases} | h_{ij} - \dfrac{(h_{max}+h_{min})}{2} | & \text{if} \quad h_{min} \le h_{ij} \le h_{max} \\ \infty & otherwise \end{cases} \quad (5)$$

Where h_{ij} is the relative altitude to the ground, h_{min} is the minimum flight altitude, and h_{max} is the maximum flight altitude. The total high cost is represented as:

$$J_3 = \sum_{i=1}^{n} H_{ij} \quad (6)$$

3) THREAT COST

During the flight, threats on the path must be avoided to ensure the feasibility of the path [30]. The threat cost is also infinite if the connecting line between two path points is inside a mountain. The threat cost is represented as:

$$J_4 = \begin{cases} \infty & \text{if } i \in obstacle(x_i, y_i) \text{ and} \\ & z_i <= Mouheight \\ 0 & otherwise \end{cases} \quad (7)$$

Where *obstacle* is the matrix used to label the obstacles in mountain region, *Mouheight* is the peak of the mountain.

4) SMOOTH COST

The smooth cost takes into account the steering angle and climbing angle of the UAV during flight, which are also crucial for generating feasible paths [31]. For i=1, 2, 3,..., N-2, we first calculate that the projection of the i-th flight path on the horizontal plane is. Then the angle of climbing θ_i^{climb} is represented as:

$$\theta_i^{climb} = \arctan\left(\frac{z_{i+1} - z_i}{\| \vec{v}_1 \|}\right) \quad (8)$$

Similarly, we can calculate the projection of the (i+1)th flight path on the horizontal plane as $\vec{v}_2 = (x_{i+2} - x_{i+1}, y_{i+2} - y_{i+1})$. Then, the angle of climbing θ_{i+1}^{climb} is represented as:

$$\theta_{i+1}^{climb} = \arctan\left(\frac{z_{i+2} - z_{i+1}}{\| \vec{v}_2 \|}\right) \quad (9)$$

We will calculate the angle between $\vec{v}_1$ and $\vec{v}_2$, and the steering angle θ_{turn} is represented as:

$$\theta_{turn} = \arctan\left(\frac{\| \vec{v}_1 \times \vec{v}_2 \|}{\vec{v}_1 \cdot \vec{v}_2}\right) \quad (10)$$

Finally, check if the angle exceeds the set maximum steering angle and maximum climbing angle. If it is exceeded, the excess is accrued to cost J_5. The smooth cost is represented as:

$$J_5 = \sum_{i=1}^{N-2} \begin{pmatrix} \max(0, | \theta_{turn} | - \theta_{max}^{turn}) \\ + \max(0, | \theta_{i+1}^{climb} - \theta_i^{climb} | - \theta_{max}^{climb}) \end{pmatrix} \quad (11)$$

The maximum allowable climbing and steering angles is θ_{max}^{climb} and θ_{max}^{turn}.

B. Total Cost

When calculating the total cost, we introduce the weighting factor Ω_i [32]. The optimization strategy can be flexibly changed by adjusting the weighting factor, different cost factors can have different degrees of influence on the task, and the weighting factor can adjust the relative importance of each cost factor in the total cost. The total cost is represented as:

$$TotalCost = \Omega_1 \bullet J_1 + \Omega_2 \bullet J_2 + \Omega_3 \bullet J_3 + \Omega_4 \bullet J_4 + \Omega_5 \bullet J_5 \quad (12)$$

III. INTRODUCTION OF SWO AND THE PATH PLANNING METHOD

A. Basic Introduction of SWO

The basic idea of the algorithm is to randomly initialize a set of search agents (spider wasps) within the search space, and the characteristics of the search agents are represented by position and fitness (Cost Function). The position represents the candidate solution to the optimization problem and the fitness represents the quality of the solution. In this algorithm, each search agent updates its position based on the above three behavioral mechanisms and continuously searches for a better solution. This process involves a dynamic trade-off between individual experience and group cooperation. The process of the update about search agent position is as follows.

1) INITIALIZATION

During initialization, the following parameters are defined: the number of search agents (N), the minimum populations (N_{min}), the upper and lower bounds of the search space($\vec{H}$, $\vec{L}$), the trade-off rate (TR), the crossover rate (CR), the maximum number of iterations (t_{max}), and the position of the initial randomly generated solutions in the space.

2) HUNTING BEHAVIOR

Hunting behavior can be divided into two stages, namely the search stage and the follow-escape stage.

Aiming to cover a wide range of search space, the search agent can perform more movements. There are two ways of searching for directions at this stage of the process. The first

method is to search with a random step size. The second method is to search with random directions.

The first search method is represented as:

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^t + \mu_1 \bullet \left(\overrightarrow{SW}_a^t - \overrightarrow{SW}_b^t \right) \tag{13}$$

$$\mu_1 = | rn | \bullet r_1 \tag{14}$$

The second search method is represented as:

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_c^t + \mu_2 \bullet \left(\vec{L} + \vec{r}_2 \bullet (\vec{H} - \vec{L}) \right) \tag{15}$$

$$\mu_2 = B \bullet \cos(2\pi l) \tag{16}$$

$$B = \frac{1}{1 + e^l} \tag{17}$$

where $\overrightarrow{SW}$ is the current solution of the female search agents, *t* denotes the generation index, *i* indicates the population index [27]. *a*, *b*, and *c* are randomly selected indices from the population, r_1 and r_2 are random numbers within the range of [0,1], | *rn* | is a random number generated by a normal distribution, and *l* is a random number generated between -2 and 1.

The two search methods are random and depend on the size of r_1 and r_2. When $r_1 < r_2$, the Equation (13) is chosen to search.

In the follow-escape stage, search agents will move based on known information, or position information from other agents, to perform a more precise search, which optimize and improve the discovered solution. There are two ways of searching directions at this stage of the process. The first method is to search with a random step size. The second method is to search with random directions.

The first search method is represented as:

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^t + C \bullet \left| 2 \bullet \vec{r}_3 \bullet \overrightarrow{SW}_a^t - \overrightarrow{SW}_i^t \right| \tag{18}$$

$$C = \left(2 - 2 \times \left(\frac{t}{t_{max}} \right) \right) \times r_4 \tag{19}$$

The second search method is represented as:

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^t \bullet \overrightarrow{vc} \tag{20}$$

$$k = 1 - \left(\frac{t}{t_{max}} \right) \tag{21}$$

Where $\overrightarrow{vc}$ is a vector generated between -*k* and *k* based on a normal distribution, $\vec{r}_3$ is a random vector, r_3, and r_4 are random numbers within the range of [0,1]. When *C* is less than 0.5, the search agent uses Equation (20) for searching, and Equation (18) is selected when $r_3 < r_4$.

3) NESTING BEHAVIOR

This behavior is primarily to conduct targeted searches within regions where excellent solutions have been discovered, and the aim is to precisely adjust their positions. There are two ways of searching for directions at this stage of the process. The first search method is to move the search agent directly towards the optimal solution region, while the second search method is to move towards the optimal solution region with an additional step size, which can help prevent nesting at the same position.

The first search method is represented as:

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}^* + \cos(2\pi l) \bullet \left(\overrightarrow{SW}^* - \overrightarrow{SW}_i^t \right) \tag{22}$$

The second search method is represented as:

$$\begin{aligned} \overrightarrow{SW}_i^{t+1} &= \overrightarrow{SW}_a^t + r_3 \bullet | \gamma | \bullet \left(\overrightarrow{SW}_a^t - \overrightarrow{SW}_i^t \right) \\ &+ (1 - r_3) \bullet \vec{U} \bullet \left(\overrightarrow{SW}_b^t - \overrightarrow{SW}_c^t \right) \end{aligned} \tag{23}$$

$$\vec{U} = \begin{cases} 1 & \vec{r}_4 > \vec{r}_5 \\ 0 & otherwise \end{cases} \tag{24}$$

where γ is a number generated according to the levy flight, $\vec{r}_4$ and $\vec{r}_5$ are random vectors within the range of [0,1], $\overrightarrow{SW}^*$ is the optimal position up to now.

4) MATING BEHAVIOR

This behavior is parallel to the above two behaviors, and the execution depends on the magnitude of the *TR*. In the later stage of iteration, the solution of the path has been generated. But after the mating behavior, the solution of the previous generation will be retained if the generated offspring is selected as female. The process is represented as:

$$SW_i^{t+1} = Crossover(SW_i^t, SW_m^t, CR) \tag{25}$$

The solution of the previous generation will be replaced with new generated offspring if the generated offspring is selected as male. The process is represented as:

$$\overrightarrow{SW}_m^{t+1} = \overrightarrow{SW}_i^t + e^l \bullet | \beta | \bullet \vec{v}_1 + (1 - e^l) \bullet | \beta_1 | \bullet \vec{v}_2 \tag{26}$$

$$\vec{v}_1 = \begin{cases} \vec{x}_a - \vec{x}_i & f(\vec{x}_a) < f(\vec{x}_i) \\ \vec{x}_i - \vec{x}_a & otherwise \end{cases} \tag{27}$$

$$\vec{v}_2 = \begin{cases} \vec{x}_b - \vec{x}_c & f(\vec{x}_b) < f(\vec{x}_c) \\ \vec{x}_c - \vec{x}_b & otherwise \end{cases} \tag{28}$$

where *Crossover* [33] is the uniform crossover operator, β and β_1 are two random numbers generated according to a normal distribution, *e* is an exponential constant, SW_m represents a male solution, $\vec{v}_1$ and $\vec{v}_2$ and are the differences between the target vectors.

5) POPULATION REDUCTION AND MEMORY SAVING

When a female spider wasp lays eggs, it means that the role of this search agent in the optimization process is almost completed. During the iteration period, some spider wasps are stopped to speed up the convergence of the algorithm. The population is updated as follows:

$$N = N_{min} + (N - N_{min}) \times k \tag{29}$$

B. The Path Planning Based on SWO

Based on the above location, the search agent updates the strategy. Meanwhile, the selected path points are evaluated by the size of the cost function. Finally, the optimal path point is output to form the optimal path.

The mechanism of the three behaviors can be seen in Fig. 1. When *rand*<*TR*, the algorithm prefers to use hunting behavior

in the early iterations to find potential solutions. In the middle and late iterations, it shifts to nesting behavior to refine the current solution, ensuring that known excellent solutions are preserved in the search process. Conversely, when *rand* > *TR*, the algorithm prefers the mating behavior, which facilitates the exchange of information between search agents and helps the whole group to make better use of existing information to generate new solutions and avoid falling into local optima. The interplay of these behaviors forms the basic workings of SWO, enabling it to find better solutions in the search space.The flowchart of the algorithm is represented as:

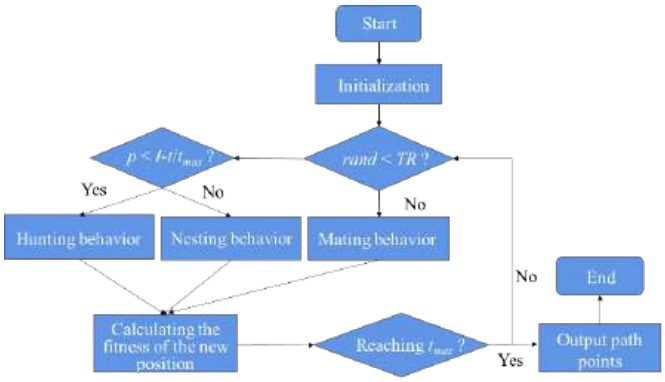


Fig. 1. The flowchart of the algorithm.

where *p* and *rand* represent the random numbers within the range of [0,1].

IV. Simulation Experiment

A. Experimental Environment

We use the *peaks* function created by Matlab software to simulate the experimental environment. The experimental space is 1000 m ×1000 m ×500 m, and the lowest and highest heights of the mountain are 200 m and 500 m, respectively. The mountain represents the obstacles in the experiment. The map in the simulation is shown in Fig. 2.

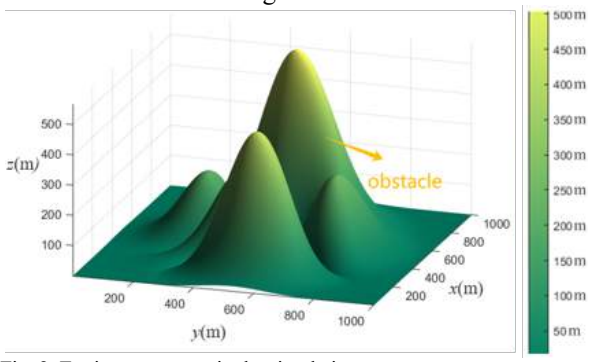


Fig. 2. Environment map in the simulation.

B. Parameter-Optimization

In the third section, we introduced the SWO algorithm, whose *TR* and *CR* values can affect the experimental results. In order to improve the performance, we conducted parameter tuning experiments on these two parameters. In the experiment, we set N=100, N_{min}=50, t_{max}=35.

The *TotalCost* dependence of *TR* and *CR* values is shown in Fig. 3. The *TotalCost* refers to the distance of the path, flight altitude, flight angle, and threat cost during the UAV flight process.

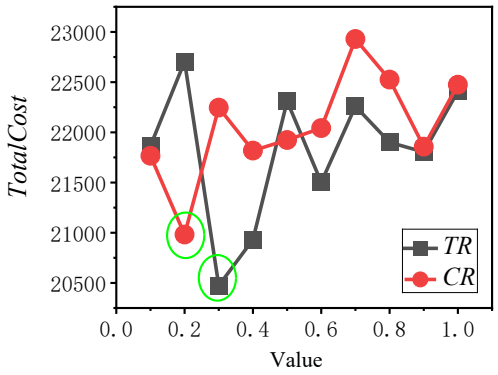


Fig. 3. TotalCost dependence of TR and CR.

The above experimental results indicate that *TR*=0.3 and *CR*=0.2 have the lowest cost, and the following experiments will use the values with the best performance of *TR* and *CR*.

C. Comparative Experiment

To verify the capability of the SWO, RRT [34] and Astar [35] algorithms are used to compare with it. Then, the evaluation experiments of three-dimensional path planning are carried out. The corresponding parameters in the simulation are shown in TABLE I. n is the number of path points of SWO, T_{max} denotes the maximum number of iterations of Astar, *step* denotes the moving step size of RRT, and *peak threshold* is 10, which means the height of the mountain threat.

TABLE I. Experimental Parameters

Algorithm	Parameters
SWO	Ω_1=10, Ω_2=10, Ω_3=1, Ω_4=5, Ω_5=1, n=3,
Astar	T_{max}=1000
RRT	*Step*=5

The preferred path of the three algorithms in the plan section and the vertical section respectively are shown in Fig. 4 and Fig. 5, when the starting point is (150,150,100) and the endpoint is (800,850,80) in the experiment.

As can be seen from Fig. 4 and Fig. 5, the curves generated using SWO are smoother than those generated by the other two algorithms. At the beginning of the SWO algorithm iteration, the evolutionary curve has a partial overlapping crossover phase with Astar. However, after several iterations, SWO can find values with smaller fitness, which indicates that SWO improves the exploration ability at the beginning of the iteration. With the increase in the number of iterations, it utilizes the hunting behavior to search over a wide range of space and quickly locate regions of excellent solutions. When the search agent approaches the endpoint, it utilizes the nesting behavior to fine-tune the region of the current solution. Through the mating behavior to generate the new solutions, the path solution can prevent the algorithm from falling into a local optimum. SWO focuses on the

development capability at the mating stage, which can further improve the quality of the optimal solution and effectively improve the local search capability. Compared with the other two algorithms, SWO does not require searching for nearby nodes for planning the path, so the search time is shorter. To acquire a shorter search path and faster search time, we selected three path points for experiments to ensure path optimality. During the whole search process, the search agent updates its position through three behavioral mechanisms, hunting, nesting, and mating behaviors. Subsequently, the cost function is applied to calculate the cost for each search agent. Then, the positions of the search agents are compared based on the magnitude of their cost functions to determine their quality. Meanwhile, inspiration cost is introduced to effectively choose a direction closer to the endpoint. In the current scenario, the initial total number N of spider wasps is 100, the minimal number N_{min} of spider wasps is 50, and the maximal number of iterations is set to be 35. Reducing the fitness for the current position of spider wasps, when search agents n=3, the path will be the optimal path with the smallest cost functions. In short, SWO shows greater effectiveness in remote scenarios. Farther from the starting point, it exhibits faster search speed and shorter path length.

The results of three algorithms running on five pairs of vertices are shown in TABLE II. The SWO algorithm achieves good results in both path length and search time, which reflects the balanced performance of the algorithm.

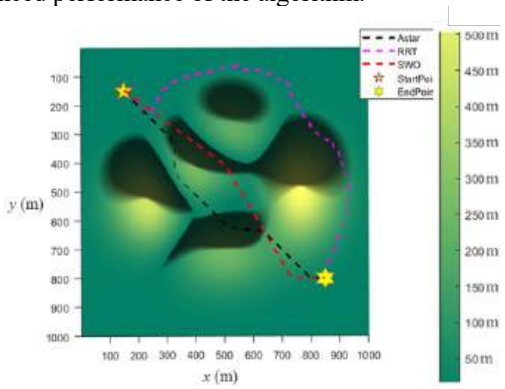


Fig. 4. The 2D view of the three paths generated by Astar, RRT, and SWO respectively. (StartPoint:(150,150,100), EndPoint:(800,850,80))

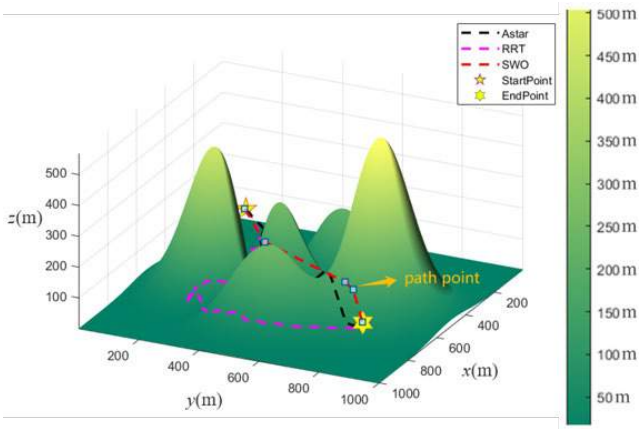


Fig. 5. The 3D path planning diagram of Astar, RRT and SWO. (StartPoint:(150,150,100), EndPoint:(800,850,80))

TABLE II. EXPERIMENTAL RESULTS

StartPoint	EndPoint	Algorithm	Path Length (m)	Search Time (s)
(150,150,100)	(800,850,80)	SWO	**1149.25**	**0.29**
		RRT	1780.04	0.43
		Astar	1175.69	0.72
(150,150,100)	(955,710,80)	SWO	1203.24	0.25
		RRT	1549.08	0.46
		Astar	1264.20	0.87
(235,860,50)	(885,818,60)	SWO	**779.60**	**0.23**
		RRT	809.70	0.23
		Astar	845.82	0.41
(175,245,50)	(720,860,100)	SWO	**1000.18**	**0.21**
		RRT	1384.92	0.27
		Astar	1038.84	0.57
(110,105,10)	(560,568,160)	SWO	**674.38**	0.20
		RRT	748.60	**0.14**

V. CONCLUSIONS

In this paper, we present a path planning algorithm leveraging the innovative Spider Wasp Optimization (SWO) algorithm, with a primary focus on optimizing the design of the cost function for enhanced performance. The cost function meticulously integrates various factors, including the path cost, the inspiration cost, the high cost, the threat cost, and the smooth cost of UAV. By modeling the terrain as an obstacle field, we apply SWO to solve the fitness function, enabling the identification of the optimal path. SWO unique approach optimizes the UAV path points directly within the search space, bypassing the need for sequential neighbor node exploration, thus achieving the shortest search time compared to traditional algorithms. Our experimental findings underscore SWO formidable search capabilities, demonstrating a remarkable balance between global and local search optimization. Additionally, we emphasize the critical role of parameter tuning in the path planning process, as precise settings can significantly accelerate convergence and enhance algorithmic accuracy. In addition, the path planning process involves a large number of parameters, and proper parameter settings may lead to fast convergence or high accuracy of the algorithm. Finally, SWO is a promising algorithm with complementary optimization techniques to tackle complex path planning challenges. To harness its full capabilities, ongoing research should focus on integrating SWO with advanced machine learning techniques, such as reinforcement learning or deep neural networks, to enable the algorithm to adaptively learn from past experiences and dynamically adjust its search strategies in real-time. Additionally, exploring hybrid approaches by combining SWO with other proven optimization algorithms, like genetic algorithms or particle swarm optimization, could further enhance its performance in solving complex and dynamic path planning problems. Ultimately, the continuous development and

refinement of SWO will pave the way for its widespread adoption in diverse real-world applications, revolutionizing the field of autonomous navigation and path planning.

REFERENCES

[1] M. Apollo, "The true accessibility of mountaineering: The case of the High Himalaya," *Journal of Outdoor Recreation and Tourism*, vol. 17, pp. 29–43, Mar. 2017.

[2] B. Soulé, B. Lefèvre, and E. Boutroy, "The dangerousness of mountain recreation: A quantitative overview of fatal and non-fatal accidents in France," *European Journal of Sport Science*, vol. 17, no. 7, pp. 931-939, 2017.

[3] O. K. Toshnazarovich, "Peculiarities of providing medical care in mountain areas," *Multidisciplinary Journal of Science and Technology*, vol. 3, no. 5, pp. 325-331, 2023.

[4] C. Van Tilburg et al., "Wilderness Medical Society Practice Guidelines for Prevention and Management of Avalanche and Nonavalanche Snow Burial Accidents," *Wilderness & Environmental Medicine*, vol. 28, no. 1, pp. 23–42, Mar. 2017.

[5] M. Jurecka and T. Niedzielski, "A procedure for delineating a search region in the UAV-based SAR activities," *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 53–72, Jan. 2017.

[6] P. Andraši, T. Radišić, M. Muštra, and J. Ivošević, "Night-time Detection of UAVs using Thermal Infrared Camera," *Transportation Research Procedia*, vol. 28, pp. 183–190, 2017.

[7] Duansen Shangguan, Yuhui Liu, Liping Chen, Chang Su, Jing Liu; Modeling and experiment of femtosecond laser processing of micro-holes arrays in quartz. J. Appl. Phys. 28 June 2024; 135 (24): 243102.

[8] Z. Mohammad, A. J. Moshayedi, Y. Zhong, A. Khan, A. Kolahdooz and M. E. Andani. "Indoor UAV Object Detection Algorithms On Three Processors: Implementation Test And Comparison." 2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE), 2023, pp. 812-819.

[9] A. J. Moshayedi, A. S. Roy, L. Liao, A. S. Khan, A. Kolahdooz and A. Eftekhari, "Design and Development of FOODIEBOT Robot: From Simulation to Design," IEEE Access, vol. 12, pp. 36148-36172, 2024.

[10] A. J. Moshayedi, KM S. Reza, A. S. Khan and A. Nawaz. "Integrating Virtual Reality and Robotic Operation System (ROS) for AGV Navigation." EAI Endorsed Transactions on AI and Robotics (2023).

[11] Y. Karaca et al., "The potential use of unmanned aircraft systems (drones) in mountain search and rescue operations," *The American Journal of Emergency Medicine*, vol. 36, no. 4, pp. 583–588, Apr. 2018.

[12] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Computer Communications*, vol. 149, pp. 270–299, Jan. 2020.

[13] J. Dai, Y. Wang, C. Wang, J. Ying, and J. Zhai, "Research on Hierarchical Potential Field Method of Path Planning for UAVs," i*n 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, May 2018, pp. 529–535.

[14] L. Blasi, E. D'Amato, M. Mattei, and I. Notaro, "UAV Path Planning in 3-D Constrained Environments Based on Layered Essential Visibility Graphs," IEEE Transactions on Aerospace and Electronic Systems, vol. 59, no. 3, pp. 2359–2375, Jun. 2023.

[15] J. L. Foo, J. Knutzon, J. Oliver, and E. Winer, "Three-Dimensional Path Planning of Unmanned Aerial Vehicles Using Particle Swarm Optimization," *in 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 0 vols., in Multidisciplinary Analysis Optimization Conferences., American Institute of Aeronautics and Astronautics*, 2006.

[16] S. A. Gautam and N. Verma, "Path planning for unmanned aerial vehicle based on genetic algorithm & artificial neural network in 3D," in *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)*, Sep. 2014, pp. 1–5.

[17] J. Sun, J. Tang, and S. Lao, "Collision Avoidance for Cooperative UAVs With Optimized Artificial Potential Field Algorithm," *IEEE Access*, vol. 5, pp. 18382–18390, 2017.

[18] Z. Fu, J. Yu, G. Xie, Y. Chen, and Y. Mao, "A Heuristic Evolutionary Algorithm of UAV Path Planning," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, Sep. 2018.

[19] Vadakkepat, P., Tan, K.C., Ming-Liang, W. "Evolutionary Artificial Potential Fields and Their Application in Real Time Robot Path Planning," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, 2000, vol. 1, pp. 256–263 vol. 1.

[20] C. Chen, J. Tang and Z. Jin, "A path planning algorithm for seeing eye robots based on v-graph," *Mechanical Science and Technology for Aerospace Engineering*, vol. 33, no. 4, pp. 490-495, 2014.

[21] E. Masehian and M. R. Amin - Naseri, ″A voronoi diagram - visibility graph - potential field compound algorithm for robot path planning,*″J. Robotic Syst*., vol. 21, no. 6, pp. 275–300, Jun. 2004.

[22] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, Sep. 2017.

[23] Liu J, Zheng M, Xiong ZJ, Li ZY. "3D dynamic motion of a dielectric micro-sphere within optical tweezers," Opto-Electron Adv 4, 200015 (2021).

[24] J. Liu, L. Long, H. Guo and Z. Li. "Observation of Moon-like Synchronous Revolution and Rotation of Janus Microparticles Trapped in an Annular Optical Trap", ACS Photonics, vol. 11, no. 10, 4027-4035, 2024.

[25] Y. Liu, D. Shangguan, L. Chen, C. Su, and J. Liu, "Prediction of femtosecond laser etching parameters based on a backpropagation neural network with grey wolf optimization algorithm," Micromachines, vol. 15, no. 8, p. 964, 2024, doi: 10.3390/mi1508096.

[26] S. A. Yıldızel and A. U. Öztürk, "A Study on the Estimation of Prefabricated Glass Fiber Reinforced Concrete Panel Strength Values with an Artificial Neural Network Model," 2016.

[27] M. Abdel-Basset, R. Mohamed, M. Jameel, and M. Abouhawwash, "Spider wasp optimizer: a novel meta-heuristic optimization algorithm," *Artif Intell Rev*, vol. 56, no. 10, pp. 11675–11738, Oct. 2023.

[28] L. Van Nguyen, M. D. Phung, and Q. P. Ha, "Game Theory-Based Optimal Cooperative Path Planning for Multiple UAVs," *IEEE Access*, vol. 10, pp. 108034–108045, 2022.

[29] S. Razzaq, C. Xydeas, M. E. Everett, A. Mahmood, and T. Alquthami, "Three-Dimensional UAV Routing With Deconfliction," *IEEE Access*, vol. 6, pp. 21536–21551, 2018.

[30] V. González, C. A. Monje, L. Moreno, and C. Balaguer, "UAVs Mission Planning with Imposition of Flight Level through Fast Marching Square," *Cybernetics and Systems*, vol. 48, no. 2, pp. 102–113, Feb. 2017.

[31] F. Samaniego, J. Sanchis, S. Garcia-Nieto, and R. Simarro, "Smooth 3D Path Planning by Means of Multiobjective Optimization for Fixed-Wing UAVs," *Electronics*, vol. 9, no. 1, 2020.

[32] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Applied Soft Computing*, vol. 107, p. 107376, Aug. 2021.

[33] G. Syswerda, "Simulated Crossover in Genetic Algorithms," in Foundations of Genetic Algorithms, vol. 2, L. D. WHITLEY, Ed., *Elsevier*, 1993, pp. 239–255.

[34] M. Li, Q. Sun, and M. Zhu, "UAV 3-Dimension Flight Path Planning Based on Improved Rapidly-exploring Random Tree," in *2019 Chinese Control And Decision Conference* (CCDC), Nanchang, China: *IEEE*, Jun. 2019, pp. 921–925.

[35] D. Mandloi, R. Arya, and A. K. Verma, "Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3d environment," *International Journal of System Assurance Engineering and Management*, vol. 12, no. 5, pp. 990–1000, Oct. 2021.

Building Intelligent Databases through Similarity: Interaction of Logical and Qualitative Reasoning

José-Luis Vilchis-Medina
ENSTA Bretagne, Lab-STICC, UMR CNRS 6285
Brest, France
jose.vilchis@ensta-bretagne.fr

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899649

***Abstract*—In this article, we present a novel method for assessing the similarity of information within knowledge-bases using a logical point of view. This proposal introduces the concept of a similarity property space $\Xi_{\mathcal{P}}$ for each knowledge $\mathcal{K}$, offering a nuanced approach to understanding and quantifying similarity. By defining the similarity knowledge space $\Xi_{\mathcal{K}}$ through its properties and incorporating similarity source information, the framework reinforces the idea that similarity is deeply rooted in the characteristics of the knowledge being compared. Inclusion of super-categories within the similarity knowledge space $\Xi_{\mathcal{K}}$ allows for a hierarchical organization of knowledge, facilitating more sophisticated analysis and comparison. On the one hand, it provides a structured framework for organizing and understanding similarity. The existence of super-categories within this space further allows for hierarchical organization of knowledge, which can be particularly useful in complex domains. On the other hand, the finite nature of these categories might be restrictive in certain contexts, especially when dealing with evolving or highly nuanced forms of knowledge. Future research and applications of this framework focus on addressing its potential limitations, particularly in handling dynamic and highly specialized knowledge domains.**

***Index Terms*—Knowledge Representation, Similarity, Knowledge-Bases, Qualitative Information**

I. Introduction

In today's information-rich world, the synergy between logic and qualitative reasoning has become increasingly vital. Logic forms the foundation of critical thinking, providing a structured approach to analyzing arguments and solving problems. Simultaneously, qualitative reasoning allows us to grasp the nuances of human experiences, interpreting context and emotions that quantitative data alone cannot capture. This combination proves crucial in decision-making processes [5], [21]. In the realm of knowledge-bases, a logical approach to data management has become increasingly crucial. One key aspect of this approach is the concept of searching data properties, which enables efficient retrieval and utilization of information within the knowledge-base. The foundation of this logical approach lies in the recognition that data, when properly organized and structured, can serve as a powerful tool for knowledge discovery and decision-making [7], [14]. Moreover, similarity logical approach can also help to manipulate the complexities of incomplete data. In a world where information is constantly evolving and new discoveries are made, knowledge-bases can often lag behind, leaving gaps and inconsistencies [22]. By embracing similarity as a guiding principle, we can bridge these gaps, extrapolating from the available data to fill the missing pieces and arrive at a more holistic understanding. So, through the utilization of a similarity-based logical methodology, we can effectively harness the true potential of these repositories, thus transforming them into dynamic and adaptable tools capable of maintaining pace with the perpetually evolving landscape of information.

Additionally, the concept of similarity plays a critical role in how intelligent systems find patterns, make predictions, and enhance user experiences. Positive aspects of similarity can lead to more relevant recommendations and accurate classifications. However, the challenges of similarity come into play when considering superficial comparisons, which can inadvertently reinforce stereotypes or fail to recognize nuanced differences [26]. Similarity transcends mere resemblance, it is a pivotal factor that enables coordination, cooperation, synchronization, and recognition among diverse systems. From a computer science standpoint, the significance of similarity is manifested in its ability to address a wide range of challenges, particularly in domains such as robotics, autonomous vehicles, distributed databases, and networked systems [21]. This relevance is underscored by its capacity to tackle critical issues that arise in these fields, including interoperability obstacles, the necessity for seamless collaboration, and the integration of heterogeneous technologies [17], [31].

As the digital age continues to reshape the way we interact with information, the importance of databases will only continue to grow. This work-in-progress aims to establish the basis for the study of information similarity from a logical point of view.

A. Related works

Intelligent Database Systems (IDS) exhibit several critical issues despite their sophistication. The inherent complexity of these systems often necessitates specialized knowledge and resources, resulting in steep learning curves and increased operational costs. Furthermore, their reliance on artificial intelligence algorithms renders them susceptible to biases and raises privacy concerns, particularly when handling sensitive information. The integration of these systems with machine learning models also heightens their vulnerability to cyber threats [9], [10], [20]. The application of machine learning approaches to similarity measurement, while powerful, is

heavily dependent on the quality and quantity of available data [24]. If the data is poor or biased, the models may fail to generalize correctly. The selection of relevant features is a crucial aspect, and improper execution can lead to unsatisfactory results. Additionally, maintaining efficiency and speed in similarity computation for large datasets remains a significant challenge [13], [19], [30].

The utilization of Natural Language Processing (NLP) techniques for the assessment of semantic similarity is confronted with a set of inherent challenges [6], [11], [39]. Many NLP methodologies rely heavily on the training corpus, and if this corpus is non-representative or biased, the resultant similarity measures may be misleading [23], [28]. The context-dependent similarity of words or phrases presents a particularly arduous obstacle, and models that fail to account for contextual factors may yield inaccurate outcomes.

Probabilistic approaches, including stochastic and Bayesian techniques, frequently depend on specific assumptions regarding the distribution of the data [8], [34], [35], [37]. When these assumptions are not met, the performance of the models can be compromised. The interpretation of the results obtained from these methods can be complex, thereby hindering the comprehension of similarity within specific contexts [29], [40]. Furthermore, the similarity quantified through these methods may not be universally applicable across diverse domains [1], [18], [25], [27].

Graph theory and ontology-based approaches, while offering improved interpretability, can become computationally intensive when dealing with large, complex datasets [2], [12], [36]. Advanced methods in high-performance computing, such as entity alignment from knowledge graphs, require sophisticated computer systems with high memory capacity, limiting their accessibility [16], [32], [33], [38]. Given these limitations, a logic-based approach to knowledge similarity could offer a promising and innovative solution. Logic, with its foundation in formal reasoning and its ability to handle complex relationships, could address many of the shortcomings of current methods [4], [31]. A logic-based system for knowledge similarity could provide a more robust and transparent framework for similarity measurement. Unlike black-box machine learning models, logical systems can offer clear, interpretable reasoning paths [3], [15]. This transparency would be particularly valuable in domains where understanding the basis of similarity is crucial, such as in legal or medical applications. In summary, a logic-based approach to knowledge similarity could serve as a viable alternative to the current computational constraints and lack of transparency in existing methods. This formal reasoning-based system could enhance the interpretability and accessibility of similarity measurement, particularly in critical domains where understanding the underlying reasoning is essential.

Furthermore, logic-based systems have the potential to overcome the data dependency challenges confronted by numerous contemporary methods. By relying on formal rules and relationships rather than extensive training datasets, these systems can provide more consistent and reliable similarity measures, even in domains where data is scarce or subject to bias. The ability of logic to handle incomplete or uncertain information presents a significant advantage. Many real-world scenarios involve partial or uncertain knowledge, and a logic-based system can furnish more robust similarity measures in these situations, unlike probabilistic methods that often struggle with incomplete data.

In this article, we present the fundamental findings of a proposal aimed at evaluating the similarity or relatedness of distinct knowledge domains or concepts within a knowledge-base. The primary objective of this study is to introduce and discuss the core results of a logical approach designed to assess the similarity of knowledge. The work is structured in two principal parts. The first section focuses on the formalization aspect, wherein a detailed explanation and a logical-mathematical model of the proposed methodology for assessing knowledge similarity are provided. The second part outlines the general conclusions drawn from the research findings and highlights the future work to be undertaken.

II. Formalisation

A. Definitions

Definition II.1. *All knowledge or (behavior) $\mathcal{K}$ involves properties $\mathcal{P}_{pro}$ described in First-Order logic (FOL). Formally:* $\{\bigwedge \mathcal{Q} \vdash \mathcal{P}\} \models \mathcal{P}_{pro} \mid \mathcal{P}_{pro} \in \mathcal{K}' \mid \mathcal{K}' \subset \mathcal{K}$.

Example II.2. *We can hypothetically consider two behaviors,* $\{\mathcal{K}_1 \cup \mathcal{K}_2\} \subset \mathcal{K}$*, both of which have properties. Formally,* $\{\mathcal{P}_1^{\mathcal{K}_1} \cup \mathcal{P}_2^{\mathcal{K}_1}\} \subset \mathcal{K}_1^{\mathcal{P}_{pro}}$, $\{\mathcal{P}_1^{\mathcal{K}_2} \cup \mathcal{P}_2^{\mathcal{K}_2} \cup \mathcal{P}_3^{\mathcal{K}_2}\} \subset \mathcal{K}_2^{\mathcal{P}_{pro}}$. *Moreover, properties are expressed in FOL:* $\{q_1 \wedge q_2 \vdash \mathcal{P}_1^{\mathcal{K}_1}\}$, $\{q_3 \wedge \neg q_1 \vdash \mathcal{P}_2^{\mathcal{K}_1}\}$, $\{q_4 \wedge q_5 \vdash \mathcal{P}_1^{\mathcal{K}_2}\}$, $\{\neg q_6 \wedge \neg q_7 \vdash \mathcal{P}_2^{\mathcal{K}_2}\}$. $\{\neg q_4 \wedge q_6 \vdash \mathcal{P}_3^{\mathcal{K}_2}\}$.

Definition II.3. *Any (known or unknown) knowledge $\mathcal{K}$ is expressed in a declarative form.*

Proposition II.4. *For any given knowledge $\mathcal{K}$, there is a similarity property space $\Xi_{\mathcal{P}}$.*

Proof. From definitions II.1 and II.3, we know that any (known or unknown) knowledge $\mathcal{K}$ can be expressed as logical form: $\{\bigwedge \mathcal{Q} \vdash \mathcal{P}\}$. In addition, it is known that a logical form is composed of a body ($\mathcal{Q}$) and a head ($\mathcal{P}$), this form captures properties of an entity or situation. Thus, three groups for identifying similarities are defined. Namely, bodies and heads of proper properties ($\mathcal{P}_{pro}$) of a given knowledge are

compared:

$$\{\bigwedge \mathcal{Q} \vdash \mathcal{P}\} \models \mathcal{P}_{pro} \mid \mathcal{P}_{pro} \subset \mathcal{K}$$
$$\{\mathcal{Q}^{\mathcal{K}_m}, \mathcal{P}^{\mathcal{K}_m}\} \in \mathcal{K}_m,\ \{\mathcal{Q}^{\mathcal{K}_n}, \mathcal{P}^{\mathcal{K}_n}\} \in \mathcal{K}_n \mid \{\mathcal{K}_m \cup \mathcal{K}_n\} \subset \mathcal{K}$$
$$\Xi_{\mathcal{P}} \supseteq \bigcup_{j=1}^{j\leq|\mathcal{P}^{\mathcal{K}_m}|} \bigcup_{i=1}^{i\leq|\mathcal{P}^{\mathcal{K}_n}|} \mathcal{P}_j^{\mathcal{K}_m}/\mathcal{P}_i^{\mathcal{K}_n}$$

where $\mathcal{P}_j^{\mathcal{K}_m}/\mathcal{P}_i^{\mathcal{K}_n}$ can have three cases:

$\mathcal{K}'_{=}$: if $\{\forall\mathcal{Q}^{\mathcal{K}_m} \subseteq \{\mathcal{Q}^{\mathcal{K}_n}, \mathcal{P}^{\mathcal{K}_n}\}\} \wedge \{\forall\mathcal{P}^{\mathcal{K}_m} \subseteq \{\mathcal{Q}^{\mathcal{K}_n}, \mathcal{P}^{\mathcal{K}_n}\}\}$;

$\mathcal{K}'_{\approx}$: if $\{\exists\mathcal{Q}^{\mathcal{K}_m} \subset \{\mathcal{Q}^{\mathcal{K}_n}, \mathcal{P}^{\mathcal{K}_n}\}\} \vee \{\exists\mathcal{P}^{\mathcal{K}_m} \subset \{\mathcal{Q}^{\mathcal{K}_n}, \mathcal{P}^{\mathcal{K}_n}\}\}$;

$\mathcal{K}'_{\neq}$: Otherwise.

$$\text{Then, } \Xi_{\mathcal{P}} \supseteq \{\mathcal{K}'_{=}\}^{|\mathcal{K}'_{=}|} \cup \{\mathcal{K}'_{\approx}\}^{|\mathcal{K}'_{\approx}|} \cup \{\mathcal{K}'_{\neq}\}^{|\mathcal{K}'_{\neq}|} \tag{1}$$

□

Corollary II.0.1. *Similarity knowledge space $\Xi_{\mathcal{K}}$ is defined by its properties.*

Proof. From Proposition II.4, similarity property space is formalized as follows:

$$\bigcup_{j=1}^{j\leq|\mathcal{P}^{\mathcal{K}_m}|} \bigcup_{i=1}^{i\leq|\mathcal{P}^{\mathcal{K}_n}|} \mathcal{P}_j^{\mathcal{K}_m}/\mathcal{P}_i^{\mathcal{K}_n} \subseteq \Xi_{\mathcal{P}} \tag{2}$$

And from Definition II.1, we have that knowledge involves properties: $\mathcal{P}_{pro} \subseteq \mathcal{K}$. Keeping in mind that properties $\mathcal{P}_{pro}$ are fundamental properties of a knowledge $\mathcal{K}$. Then properties $\mathcal{P}_{pro}$ of a given knowledge $\mathcal{K}$ define the behavior of $\mathcal{K}$. Thus for any property $\mathcal{P}_j$ of knowledge $\mathcal{K}_m$ must be the behavior of $\mathcal{K}_m$, in which it is the similarity knowledge space:

$$\bigcup_{i}^{i\leq|\mathcal{K}|} \bigcup_{\substack{j \\ j\neq i}}^{j\leq|\mathcal{K}|} \mathcal{K}_j/\mathcal{K}_i \subseteq \Xi_{\mathcal{K}} \tag{3}$$

□

Example II.5. *Regarding knowledge $\{\mathcal{K}_1 \cup \mathcal{K}_2\} \subseteq \mathcal{K}$ of Example II.2 and using Proposition II.4 in order to evaluate knowledge similarity, in other words, how similar is $\mathcal{K}_1$ to $\mathcal{K}_2$, so $\mathcal{K}_1/\mathcal{K}_2$. We know that knowledges are defined by properties, $\{\mathcal{P}_1^{\mathcal{K}_1} \cup \mathcal{P}_2^{\mathcal{K}_1}\} \subseteq \mathcal{K}_1$, $\{\mathcal{P}_1^{\mathcal{K}_2}, \mathcal{P}_2^{\mathcal{K}_2}, \mathcal{P}_3^{\mathcal{K}_2}\} \subseteq \mathcal{K}_2$. Thus, we can apply the function $\Xi_{\mathcal{P}}$ in order to genera the similarity properties space:*

$$\bigcup_{j=1}^{j\leq2} \bigcup_{i=1}^{i\leq3} \mathcal{P}_j^{\mathcal{K}_1}/\mathcal{P}_i^{\mathcal{K}_2} \subseteq \begin{Bmatrix} \mathcal{P}_1^{\mathcal{K}_1}/\mathcal{P}_1^{\mathcal{K}_2} & \mathcal{P}_2^{\mathcal{K}_1}/\mathcal{P}_1^{\mathcal{K}_2} \\ \mathcal{P}_1^{\mathcal{K}_1}/\mathcal{P}_2^{\mathcal{K}_2} & \mathcal{P}_2^{\mathcal{K}_1}/\mathcal{P}_2^{\mathcal{K}_2} \\ \mathcal{P}_1^{\mathcal{K}_1}/\mathcal{P}_3^{\mathcal{K}_2} & \mathcal{P}_2^{\mathcal{K}_1}/\mathcal{P}_3^{\mathcal{K}_2} \end{Bmatrix} \subseteq \Xi_{\mathcal{P}} \tag{4}$$

Arbitrarily we are going to consider that there are certain properties (both their bodies and their heads) which are the same, similar and different, according to formalization of the Proposition II.4 for these cases.

$$\bigcup_{j=1}^{j\leq2} \bigcup_{i=1}^{i\leq3} \mathcal{P}_j^{\mathcal{K}_1}/\mathcal{P}_i^{\mathcal{K}_2} \subseteq \begin{Bmatrix} \mathcal{K}'_{\neq} & \mathcal{K}'_{\approx} \\ \mathcal{K}'_{\approx} & \mathcal{K}'_{\approx} \\ \mathcal{K}'_{=} & \mathcal{K}'_{\neq} \end{Bmatrix} \subseteq \Xi_{\mathcal{P}} \tag{5}$$

And finally, we can describe the similarity property space $\Xi_{\mathcal{P}}$ in terms of cardinalities:

$$\{\{\mathcal{K}'_{=}\}^1 \cup \{\mathcal{K}'_{\approx}\}^3 \cup \{\mathcal{K}'_{\neq}\}^2\} \subseteq \Xi_{\mathcal{P}} \tag{6}$$

Hence, knowledge $\mathcal{K}_1$ contains at least one equal property, three similar properties and two different properties in relation to knowledge $\mathcal{K}_2$.

Lemma II.1. *Similarity property space $\Xi_{\mathcal{P}}$ contains similarity source information $\Xi_{\mathcal{K}}^{\dagger}$.*

Proof. Given two or more knowledges, $\{\mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3 \cup \ldots\} \subseteq \mathcal{K}$, with a constant (or non-constant) cardinality of properties of each, thus, a Cartesian product is performed in order to search similarities, $\mathcal{K} \times \mathcal{K}$. Consequently, we have a square size set, $|\mathcal{K}| \times |\mathcal{K}|$. In order to discriminate redundancies, we consider half of set: $2 \cdot \Xi_{\mathcal{K}}^{\dagger} \subseteq \Xi_{\mathcal{K}}$. Lastly, we can arbitrarily select the lower triangular space set without considering diagonal information of the space. So, similarity source information $\Xi_{\mathcal{K}}^{\dagger}$ is located in the lower triangular space set of $\Xi_{\mathcal{K}}$. □

Example II.6. *Let's consider 3 different kinds of knowledge $\{\mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3\} \subseteq \mathcal{K}$, and also that whole set of knowledge has equal cardinality of properties: $|\mathcal{K}_1^{\mathcal{P}_{pro}}| \subseteq |\mathcal{K}_2^{\mathcal{P}_{pro}}| \subseteq |\mathcal{K}_3^{\mathcal{P}_{pro}}|$. From Corollary II.0.1, we have:*

$$\bigcup_{i}^{i\leq|\mathcal{K}|} \bigcup_{\substack{j \\ j\neq i}}^{j\leq|\mathcal{K}|} \mathcal{K}_j/\mathcal{K}_i \equiv \bigcup_{j=1}^{j\leq|\mathcal{P}^{\mathcal{K}_m}|} \bigcup_{i=1}^{i\leq|\mathcal{P}^{\mathcal{K}_n}|} \mathcal{P}_j^{\mathcal{K}_m}/\mathcal{P}_i^{\mathcal{K}_n} \subseteq \Xi_{\mathcal{K}} \tag{7}$$

And from Lemma II.1, we know that similarity knowledge space $\Xi_{\mathcal{K}}$ has redundancies that can be expressed as follows: $2 \cdot \Xi_{\mathcal{K}}^{\dagger} \subseteq \Xi_{\mathcal{K}}$. In order to select (lower or upper part) similarity source information part $\Xi_{\mathcal{K}}$ we must not consider the diagonal of the space.

$$2 \cdot \Xi_{\mathcal{K}}^{\dagger} \subseteq |\mathcal{K} \times \mathcal{K}| - |diag(\mathcal{K})| \subseteq \Xi_{\mathcal{K}} \tag{8}$$

Corollary II.1.1. *Similarity knowledge space $\Xi_{\mathcal{K}}$ is bounded by a finite number of categories.*

Proof. We know that similarity knowledge space $\Xi_{\mathcal{K}}$ generates three different sets: $\{\mathcal{K}'_{=}\}, \{\mathcal{K}'_{\approx}\}, \{\mathcal{K}'_{\neq}\}$. Consequently, it will be eight different possible sets as result. Because of all the possible combinations from three sets, initializing with three empty sets up to three sets with information. □

Lemma II.2. *For any knowledge $\mathcal{K}$, similarity identification occurs when knowledge similarity space $\Xi_{\mathcal{K}}$ is strictly non-empty:*

Proof. From Corollary II.1.1, we know that there are eight possible categories. Then, non-empty set can only be obtained when all categories are non-zero cardinality: $\{\mathcal{K}'_{=}\} \cup \{\mathcal{K}'_{\approx}\} \cup \{\mathcal{K}'_{\neq}\} \notin \{\emptyset\}$. Consequently, categories out of this criterion can be considered for similarity identification. □

Lemma II.3. *There are super-categories in the knowledge similarity space $\Xi_{\mathcal{K}}$.*

Proof. From Corollary II.1.1, we know there are eight possible categories of similarities, strictly speaking we can actually consider only 7 by Lemma II.2, where we discard null cardinality for the three categories case. Consequently, we obtain three different super-categories: These super-categories are the consequences of regrouping $\{\mathcal{K}'_{=} \cup \mathcal{K}'_{\approx} \cup \mathcal{K}'_{\neq}\}$ sets, which may be described as follows:

- **Case 1:** It will happen when two sets of $\{\mathcal{K}'_{=} \cup \mathcal{K}'_{\approx} \cup \mathcal{K}'_{\neq}\}$ are empty, thus only $\{\mathcal{K}'_{=}\}$ or $\{\mathcal{K}'_{\approx}\}$ or $\{\mathcal{K}'_{\neq}\}$ will remain.
- **Case 2:** It will happen when one set of $\{\mathcal{K}'_{=} \cup \mathcal{K}'_{\approx} \cup \mathcal{K}'_{\neq}\}$ is empty, then three pairs of sets should be present: $\{\mathcal{K}'_{=} \cup \mathcal{K}'_{\approx}\}$ or $\{\mathcal{K}'_{=} \cup \mathcal{K}'_{\neq}\}$ or $\{\mathcal{K}'_{\approx} \cup \mathcal{K}'_{\neq}\}$.
- **Case 3:** It will occur when no empty set is involved, resulting in a single set: $\{\mathcal{K}'_{=} \cup \mathcal{K}'_{\approx} \cup \mathcal{K}'_{\neq}\}$.

□

B. Discussion

The fundamental assertion that all knowledge involves properties describable in FOL is a powerful starting point. FOL's expressive power allows for the representation of complex relationships and structures, making it a suitable language for capturing the nuances of various forms of knowledge. This approach provides a standardized method for knowledge representation, potentially facilitating easier comparison and analysis across different domains. The concept of a similarity property space $\Xi_{\mathcal{P}}$ for any given knowledge $\mathcal{K}$ is particularly interesting. It suggests that similarity is not an absolute measure but is relative to the specific properties relevant to the knowledge in question. This contextual approach to similarity aligns well with human intuition, what we consider similar in one context may not be in another.

The bounded nature of the similarity knowledge space $\Xi_{\mathcal{K}}$ by a finite number of categories is both a strength and a potential limitation. On one hand, it provides a structured framework for organizing and understanding similarity. The existence of super-categories within this space further allows for hierarchical organization of knowledge, which can be particularly useful in complex domains. On the other hand, the finite nature of these categories might be restrictive in certain contexts, especially when dealing with evolving or highly nuanced forms of knowledge. The condition that similarity identification occurs only when the knowledge similarity space is strictly non-empty is a crucial point. It implies that for any meaningful comparison or similarity assessment to take place, there must be some shared property space between the knowledge entities being compared. This condition helps in avoiding spurious or meaningless similarity assessments.

However, this framework raises several questions and potential challenges:

- Subjectivity in property selection: The selection of properties that define the similarity space could be subjective. Different observers might choose different properties, leading to varying similarity assessments.
- Dynamic nature of knowledge: Knowledge often evolves. How does this framework account for the dynamic nature of knowledge and the potential need for evolving similarity spaces?
- Handling uncertainty: FOL traditionally deals with definite truths. How does this framework handle uncertain or probabilistic knowledge?

III. Conclusion

This paper presented a new proposal for studying knowledge similarity using a logical point of view. The introduction of a similarity property space $\Xi_{\mathcal{P}}$ for each knowledge entity $\mathcal{K}$ represents a nuanced approach to understanding similarity. By recognizing that similarity is context-dependent and relative to specific properties, this framework aligns closely with human intuition about how we perceive and compare concepts. The concept that the similarity knowledge space $\Xi_{\mathcal{K}}$ is defined by its properties and contains similarity source information further reinforces the idea that similarity is not an absolute measure but is deeply rooted in the characteristics of the knowledge being compared.

The bounded nature of the similarity knowledge space $\Xi_{\mathcal{K}}$ by a finite number of categories presents both opportunities and challenges. On one hand, it provides a structured framework for organizing and understanding similarity, which can be especially beneficial in complex domains. The inclusion of super-categories within this space allows for a hierarchical organization of knowledge, facilitating more sophisticated analysis and comparison. This hierarchical structure mirrors many natural and artificial classification systems, potentially making it more intuitive and applicable across various fields.

The condition that similarity identification occurs only when the knowledge similarity space $\Xi_{\mathcal{K}}$ is strictly non-empty is a critical aspect of this framework. This requirement ensures that meaningful comparisons are based on shared properties, preventing spurious or irrelevant similarity assessments. By enforcing this condition, the framework maintains the integrity of similarity comparisons, ensuring that they are grounded in substantive shared characteristics rather than superficial or coincidental similarities.

The existence of super-categories in the knowledge similarity space $\Xi_{\mathcal{K}}$ adds another layer of sophistication to this framework. It allows for a more nuanced understanding of how different knowledge entities relate to each other on multiple levels, potentially revealing higher-order similarities that might not be apparent at more granular levels of comparison.

References

[1] Biazzo, V., Gilio, A., Lukasiewicz, T., Sanfilippo, G.: Probabilistic logic under coherence, model-theoretic probabilistic logic, and default reasoning in system p. Journal of Applied Non-Classical Logics **12**(2), 189–213 (2002)

[2] Blondel, V.D., Gajardo, A., Heymans, M., Senellart, P., Van Dooren, P.: A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM review **46**(4), 647–666 (2004)

[3] Brachman, R., Levesque, H.: Knowledge representation and reasoning. Morgan Kaufmann (2004)

[4] Brachman, R.J., Levesque, H.J.: The tractability of subsumption in frame-based description languages. In: AAAI. vol. 84, pp. 34–37 (1984)

[5] Brannen, J.: Combining qualitative and quantitative approaches: an overview. Mixing methods: Qualitative and quantitative research pp. 3–37 (2017)
[6] Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
[7] Ceri, S., Gottlob, G., Tanca, L., Ceri, S., Gottlob, G., Tanca, L.: Logic Programming and Databases: An Overview. Springer (1990)
[8] Cheng, H., Wang, R.: Semantic modeling of natural scenes based on contextual bayesian networks. Pattern Recognition **43**(12), 4042–4054 (2010)
[9] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
[10] Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Ontology matching: A machine learning approach. Handbook on ontologies pp. 385–403 (2004)
[11] Fellbaum, C.: WordNet: An electronic lexical database. MIT press (1998)
[12] Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Analysis and applications **13**, 113–129 (2010)
[13] Getoor, L., Friedman, N., Koller, D., Taskar, B.: Learning probabilistic models of relational structure. In: ICML. vol. 1, pp. 170–177 (2001)
[14] Goldstone, R.L., Son, J.Y.: Similarity. Oxford University Press (2012)
[15] Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logic. In: Proceedings of the 12th international conference on World Wide Web. pp. 48–57 (2003)
[16] Guarino, N., Welty, C.A.: An overview of ontoclean. Handbook on ontologies pp. 201–220 (2009)
[17] Hahn, U.E., Ramscar, M.E.: Similarity and categorization. Oxford University Press (2001)
[18] Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 50–57 (1999)
[19] Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338 (2013)
[20] Kandola, J., Cristianini, N., Shawe-taylor, J.: Learning semantic similarity. Advances in neural information processing systems **15** (2002)
[21] Kuipers, B.: Qualitative reasoning: modeling and simulation with incomplete knowledge. MIT press (1994)
[22] Libkin, L.: Incomplete data: what went wrong, and how to fix it. In: Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 1–13 (2014)
[23] Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Language and cognitive processes **6**(1), 1–28 (1991)
[24] Minicozzi, E.: Some natural properties of strong-identification in inductive inference. Theoretical Computer Science **2**(3), 345–360 (1976)
[25] Mitra, P., Noy, N.F., Jaiswal, A.R.: Omen: A probabilistic ontology mapping tool. In: The Semantic Web–ISWC 2005: 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005. Proceedings 4. pp. 537–547. Springer (2005)
[26] Noble, C.E.: Psychology and the logic of similarity. The Journal of General Psychology **57**(1), 23–43 (1957)
[27] Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan kaufmann (1988)
[28] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
[29] Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995)
[30] Severyn, A., Moschitti, A.: Learning to rank short text pairs with convolutional deep neural networks. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. pp. 373–382 (2015)
[31] Sheremet, M., Tishkovsky, D., Wolter, F., Zakharyaschev, M.: A logic for concepts and similarity. Journal of Logic and Computation **17**(3), 415–452 (2007)
[32] Shu, G., Rana, O.F., Avis, N.J., Dingfang, C.: Ontology-based semantic matchmaking approach. Advances in engineering software **38**(1), 59–67 (2007)
[33] Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Transactions on knowledge and data engineering **25**(1), 158–176 (2011)
[34] Smith, R.C.: Uncertainty quantification: theory, implementation, and applications, vol. 12. Siam (2013)
[35] Soize, C.: Uncertainty quantification. Springer (2017)
[36] Sowa, J.F.: Conceptual graphs. Foundations of artificial intelligence **3**, 213–237 (2008)
[37] Sullivan, T.J.: Introduction to uncertainty quantification, vol. 63. Springer (2015)
[38] Taieb, M.A.H., Aouicha, M.B., Hamadou, A.B.: Ontology-based approach for measuring semantic similarity. Engineering Applications of Artificial Intelligence **36**, 238–261 (2014)
[39] Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research **37**, 141–188 (2010)
[40] Tversky, A.: Features of similarity. Psychological review **84**(4), 327 (1977)

Optimization of convolution computation based on CPU

1st JunJie Di
Harbin Engineering University
Qingdao Innovation and Development Center
Qingdao, China
junjie.di@foxmail.com

2nd Shuang Li
Harbin Engineering University
Qingdao Innovation and Development Center
Qingdao, China
shuangli@hrbeu.edu.cn

3rd LongXiang Guo
Harbin Engineering University
College of Underwater Acoustic Engineering
Harbin, China
heu503@126.com

4th Wei Ge
Harbin Engineering University
Qingdao Innovation and Development Center
Qingdao, China
gewei@hrbeu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899500

Abstract—In convolutional neural networks (CNNs), convolution is one of the core computations, with most of the network's time spent on convolution calculations. Therefore, optimizing the efficiency of convolution computations is particularly important in resource-constrained environments. CPUs are widely used across various devices, including cloud servers, edge devices, and mobile platforms. These devices do not always have access to dedicated GPUs, so efficient CPU-based convolutions enable the implementation of complex algorithms for low-latency, real-time inference on low-power platforms. In this paper, we propose an improved img2col+GEMM(General Matrix Multiply) convolutional optimization method based on the open-source library OpenBLAS. By leveraging Intel AVX512 vectorization instructions and assembly-level instruction reordering, we present a high-performance micro-kernel to optimize the GEMM operator. Additionally, we implement data reordering and memory alignment operations in the img2col process based on the aforementioned micro-kernel parameters , directly placing them into the buff of the GEMM process to save memory usage and access operations. Combining our micro-kernel with the OpenBLAS library, we achieve 98% of the performance of Intel's MKL on Intel CPUs, with a 10% to 15% performance improvement over OpenBLAS. The improved img2col method shows approximately a 16% enhancement in convolution speed compared to the standard img2col+GEMM approach.

Index Terms—Convolutional neural networks, high-performance matrix multiplication, AVX

I. Introduction

Convolutional Neural Networks (CNNs), a subset of Deep Neural Networks (DNNs), have been widely applied in computer vision models such as object detection [1], semantic segmentation [2], instance segmentation [3], and panoptic segmentation [4]. CNNs typically consist of convolutional layers, pooling layers, activation layers, and fully connected layers. In mainstream CNN models, the computational load of convolutional layers accounts for over 60%-90% of the total computation [5]], making convolution performance a primary bottleneck in CNN training and inference.

The high computational cost of convolutional (CONV) layers can be addressed by: 1) adjusting the structure of convolutional neural network models, such as parameter quantization, network pruning, low-rank decomposition, and knowledge distillation; 2) optimizing the convolution operators within deep learning computation libraries. Current mature convolution operator acceleration algorithms include:

- Fast Fourier Transform (FFT) [6]: Many learning frameworks (e.g., Meta's NNPACK, MATLAB-based LightNet, and CUDA's cuDNN) support FFT-based convolution. However, the acceleration is limited to specific convolution parameters, hindering FFT's use as a default convolution algorithm.
- Winograd Algorithm [7]: Supported by various frameworks such as Facebook's NNPACK, NVIDIA's cuDNN, and Tencent's NCNN, this algorithm accelerates convolution by reducing the number of multiplications at the cost of increased additions and extra preprocessing and transformation operations. As the size of convolution kernels and tiles increases, the costs of additions and transformations must be considered, and precision may be compromised. Some studies integrate Winograd with Strassen's matrix multiplication variants [7], [8] to reduce these costs further, but this integration increases precision loss, making Winograd suitable primarily for smaller kernels and tiles.
- Im2col + GEMM(General Matrix Multiply): This mainstream approach transforms convolution into matrix multiplication by tiling input feature maps and convolution kernels into matrices, leveraging high-performance matrix computation libraries (e.g.,

Intel's oneDNN/MKL). NNPACK and FeatherCNN [9] provide optimized convolution operators for ARM processors. The im2col [10] transformation saves memory access time, while General Matrix Multiply(GEMM) [11]–[13] is the main performance bottleneck due to matrix tiling's spatial storage requirements. As GEMM performance improves, the memory and computational overhead of im2col operations also become apparent.

To further optimize the performance of the im2col + GEMM algorithm, it is crucial to exploit modern hardware architectures effectively.

Modern CPUs offer advanced features such as vectorization instructions (e.g., AVX-512), which can significantly speed up convolution operations when utilized effectively.AVX-512 allows simultaneous processing of multiple floating-point numbers with 512-bit wide vector registers, greatly improving data parallelism. The comprehensive arithmetic, logic, and data manipulation instructions in AVX-512, along with higher parallelism and register counts, help reduce memory access latency and increase data throughput.Optimizing convolutions on CPUs can provide a solution for resource-constrained computing environments. On heterogeneous platforms, the overhead of switching between CPUs and GPUs can also be greatly reduced.

Intel's high-performance computing library, MKL, is optimized for their product architecture but remains largely closed-source, functioning as a black box. This paper leverages the open-source library OpenBLAS to propose two algorithmic improvements for high-performance convolution on CPUs:

- Micro-Kernel Optimization: Utilizing the instruction set to accelerate small matrix operations after segmentation by employing matrix blocking techniques in OpenBLAS to optimize memory access. Instruction reordering at the assembly level is used to enhance hardware utilization.

II. Access Optimisation

In contemporary CPUs, the speed of memory access increases as the memory is located closer to the CPU. When the processor needs to load data required for the current operation, it first attempts to retrieve it from the L1 cache. If the CPU successfully finds the data in the L1 cache, this situation is termed a cache hit. If the data is not found in the L1 cache, the processor then proceeds to search in the L2 and L3 caches. If the data is still not found, the processor will attempt to access it from the main memory. This situation is referred to as a cache miss.

By leveraging the characteristics of multi-level caches, large matrices are partitioned into smaller submatrices. These submatrices are then processed iteratively, utilizing the principles of cache hierarchy during their computation. The submatrices are processed in loops according to the size of the micro-kernel. The results are subsequently merged.

The submatrices are loaded into multiple cache levels based on the frequency of data reuse, significantly increasing the cache hit rate. This process occurs concurrently with the computation performed by the micro-kernel, thereby masking the data loading time with the computation time and achieving a fully pipelined workload.**Algorithm1** illustrates a concept similar to the matrix multiplication optimization methods used in MKL and OpenBLAS. The detailed methodologies are discussed in the work of Goto et al. [14]–[17].

Algorithm 1 GEMM Algorithm

```
for j_c = 0 to n − 1 step n_c do                          ▷ Loop 5
    J_c = j_c : j_c + n_c − 1
    for p_c = 0 to k − 1 step k_c do                      ▷ Loop 4
        P_c = p_c : p_c + k_c − 1
        B(P_c, J_c) → B_c                                 ▷ Pack into B_c
        for i_c = 0 to m − 1 step m_c do                  ▷ Loop 3
            I_c = i_c : i_c + m_c − 1
            A(I_c, P_c) → A_c                             ▷ Pack into A_c
            for j_r = 0 to n_c − 1 step n_r do            ▷ Loop 2
                J_r = j_r : j_r + n_r − 1
                for i_r = 0 to m_c − 1 step m_r do        ▷ Loop 1
                    I_r = i_r : i_r + m_r − 1
                    for k_r = 0 to k_c − 1 do             ▷ Loop 0
                        C_c(I_r, J_r) += A_c(I_r, k_r) · B_c(k_r, J_r)
                    end for
                end for
            end for
        end for
    end for
end for
```

In various contemporary frameworks, researchers have adopted different standards for optimal cache utilization, specifically regarding the settings of parameters such as n_c, m_c, k_c, n_r, and m_r, as outlined in Algorithm 1. Goto [14], Low [15], and Lim [16] have each established different standards based on hardware cache size, the number of registers, and machine bandwidth limitations. Our implementation of the micro-kernel differs from the aforementioned algorithms, particularly in the selection of nr and mr parameters, which will be discussed in Section 2.1, The remaining processes adhere to the standards set forth in the previous works.

Additionally, to achieve optimal performance for the micro-kernel, the data for A and B needs to be reordered and aligned in memory before entering the register computation kernel. This data is then packed into buff A_c and B_c. Reordering the data allows the vector registers to access data from A_c and B_c continuously (which also improves TLB addressing efficiency), and memory align-

ment enables the vector registers to move in unit steps. Our approach integrates this data reordering operation with the im2col process .The specific details are discussed in Section 3.

A. Data vectorisation

The micro-kernel primarily implements lines 9 to 14 of **Algorithm1**, as illustrated in the **Fig1**. In this process, the multiplication of blocks of matrices A and B is performed within a loop of size k_c. Specifically, vectors of size mr from matrix A and vectors of size nr from matrix B are loaded from the cache into registers sequentially, followed by multiplication and accumulation operations. At the end of the loop, the results are summed up.For the memory operations, we can ignore the reads of loop conditions, as these can be provided directly by instruction operands and do not need to be stored in memory. In the innermost loop, the matrix C involves two memory operations (read and write), while matrices A and B each involve one memory read operation. Therefore, the number of memory operations in the innermost loop is $(2+1+1)$. Given that the loop iterates $n_r m_r k_c$ times, the total number of memory operations is $4n_r m_r k_c$ times.

Fig. 1.

We utilize the AVX-512 instruction set to operate on vector registers for parallel data reading. The instruction set includes 32 *zmm* registers, each with a width of 512 bits, capable of holding 8 double-precision float data. Each time we process 8 columns of data, meaning each column can use $32/8 = 4$ *zmm* registers. However, for vectorized operations, additional vector registers are required for data loading and computation. Therefore, as illustrated in the **Fig2**., we allocate 3 *zmm* registers per column, resulting in a total of 24 *zmm* registers for storing results. An additional 4 *zmm* registers are used for data loading and computation. Thus, each iteration within the loop requires 28 *zmm* registers in total, allocated as follows:

- 24 *zmm* registers for storing results (3 registers per column for 8 columns).
- 4 *zmm* registers for loading data and executing computations.

In each iteration of the innermost loop, a column of matrix A and a row of matrix B are utilized to compute a "sector" of the resulting matrix. We allocate 3 *zmm* registers to store a row of matrix B. Subsequently, the first element of the current column from matrix A is read, duplicated 8 times via broadcasting, and stored in a *zmm* register. This broadcasted register is then multiplied

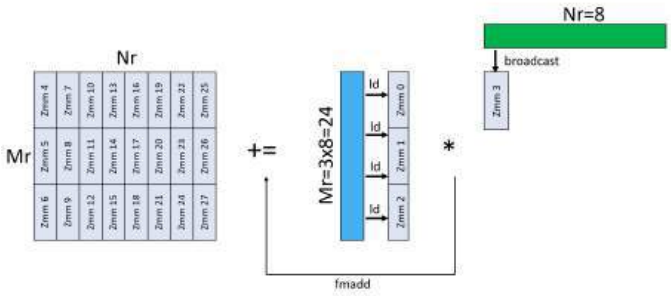


Fig. 2.

element-wise with the 3 *zmm* registers containing the row from matrix B, with the results accumulated into *zmm*4 to *zmm*6.

For the subsequent elements (2 to 8) of the current column from matrix A, the *zmm*3 register is reconstructed with each new element. The data within the *zmm*0 to *zmm*2 registers remains unchanged, thus obviating the need for reloading. Only the target output registers are altered, shifting from *zmm*4 to *zmm*6 to *zmm*7 to *zmm*9 for accumulation purposes.For details on the implementation, please refer to **Algorithm2**.

Algorithm 2 Microkernel Algorithm

1: for $j_r = 0$ to $n_c - 1$ step 8 do
2: for $i_r = 0$ to $m_c - 1$ step 24 do
3: Initialize $Zmm4 \sim Zmm25$ to store the C-matrix calculation results
4: for $k_r = 0$ to $k_c - 1$ do
5: $Zmm0 \sim Zmm2 \leftarrow A_{i_r k_r}, \sim A_{i_r+23k_r}$
6: $Zmm3 \leftarrow B_{k_r j_r}$, broadcast
7: $C_{i_r j_r} \sim C_{i_r+23j_r} \leftarrow$ The $fmadd$ instruction calculates$Zmm3$ with$Zmm0 \sim Zmm2$
8: $Zmm3 \leftarrow B_{k_r j_r+1}$, broadcast
9: $C_{i_r j_r+1} \sim C_{i_r+23j_r+1} \leftarrow$ The $fmadd$ instruction calculates $Zmm3$ with $Zmm0 \sim Zmm2$
10: ⋮
11: $ymm3 \leftarrow B_{k_r j_r+8}$, broadcast
12: $C_{j_r+i_r+8} \sim C_{i_r+23j_r+8} \leftarrow$ The $fmadd$ instruction calculates $Zmm3$ with $Zmm0 \sim Zmm2$
13: end for
14: end for
15: end for

Therefore, in the selection of the micro-kernel parameters m_r and n_r, Low [15], Lim [16] have given an upper bound based on the hierarchical cache size, whereas our micro-kernel based on the $AVX512$ instruction can obtain a smaller upper bound criterion based on its number of vector registers:

$$\frac{M_r}{R_{elm}}(1 + N_r) + 1 < 32 s.t. M_r mod R_{elm} = 0 \qquad (1)$$

R_{elm} denotes the number of elements stored in a single register. Due to SIMD instructions, N_r needs to be a multiple of R_{elm}. Additionally, $M_r/R_{elm}+1$ registers are used for loading data and executing computations, while $M_r/R_{elm} * N_r$ registers are used for storing results. In practice, we also consider the full utilization of the instruction pipeline, which results in the m_r and n_r values for this scheme being smaller than the aforementioned upper limit. This will be discussed in detail in the section 2.2.

B. Instruction Reordering

To prevent the CPU ALU from being underutilized due to I/O wait times, the compute instruction cycles within a single loop of the register micro-kernel should generally be longer than the load instruction cycles. This ensures that the computation time can cover the data loading time, achieving instruction-level parallelism.

In the previous section, each iteration of the innermost loop contains 24 multiply-add instructions, each with a latency of 5 clock cycles, and 11 load instructions, each with a latency of 4 clock cycles (when loading from L1 cache), which satisfies the aforementioned condition. However, there are still two issues with the local instruction arrangement. First, some local computation instructions are densely packed, causing the instruction dispatch unit to generate a large number of compute-related micro-operations in a localized area, leading to idle periods on the load pipeline. This underutilization prevents the pipeline from operating at maximum efficiency.

As shown in **Fig3**, we reuse registers that have already been used in the current loop to preload the data needed for the next loop at the loop exit. These preload instructions are inserted into the current loop's computation instructions, achieving a more balanced distribution of compute and load instructions. This also offsets the startup time of the next loop by hiding it within the computation time of the current loop.

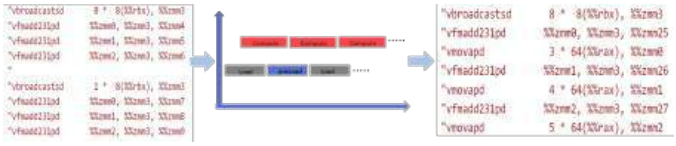


Fig. 3.

The second issue concerns resource dependency. The $fmadd$ instruction uses the $zmm0$ and $mm3$ registers as source registers, and the first two load instructions also utilize the same registers. When the decoder detects this situation, the dispatch stage of the $fmadd$ instruction is delayed until the write-back of the first two source registers is confirmed. This delay creates a significant gap between dispatch and decode stages, leading to substantial pipeline stalls.

To mitigate this issue, as shown in **Fig4**, we altered the register allocation strategy. In the previous scheme, a total of 28 registers were used, which did not reach the upper limit. In the innermost loop, we use two zmm registers for broadcasting, taking advantage of the surplus registers. We insert new operational instructions between the instructions that use the same destination registers, thus offsetting the resource dependency problem. This adjustment also explains why the mr and nr parameters in the previous section are smaller than the upper limit.

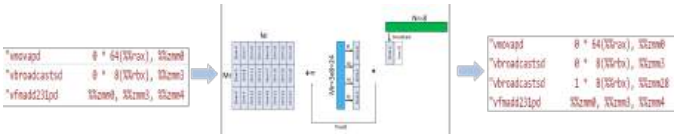


Fig. 4.

In$CPUs$ with strong out-of-order execution capabilities, the CPU reorders the instruction stream during the dispatch stage to optimize execution on the pipeline. However, out-of-order execution can introduce issues such as data dependencies and memory access order uncertainties, especially in multi-threaded execution. To address these problems, we manually optimized the instruction layout of the register calculation kernel using inline assembly. This approach not only enhances performance but also makes the code more portable to $CPUs$ with weaker out-of-order execution capabilities.

III. Experimental

We conducted our tests on an Intel(R) Xeon(R) Platinum processor with a base frequency of 2.5 GHz. The theoretical peak performance for double-precision floating-point calculations on a single core is 80 GFLOPS. Our primary goal was to evaluate the performance of the micro-kernel, so all subsequent tests were performed using a single core and single-thread processing.

A. Micro-kernel Performance Testing

In this subsection, we tested the performance of matrix multiplication for different shapes. Since MKL is not open-source, we replaced the micro-kernel designed in Section 2.1 with the micro-kernel part of the open-source computing library OpenBLAS. We compared the performance of the improved algorithm kernel against OpenBLAS and MKL. Each shape of the matrix was tested 100 times to obtain an average value.

As shown in **Fig5**, when the matrix size is small, all the data involved in matrix multiplication can be loaded into the L1 cache, resulting in high cache hit rates. At this point, the optimizations in the algorithm from Section 2.1 are close to the performance of the OpenBLAS library, but there is still a significant gap compared to MKL. However, as the matrix size increases, the performance of our optimized algorithm gradually approaches that of MKL.

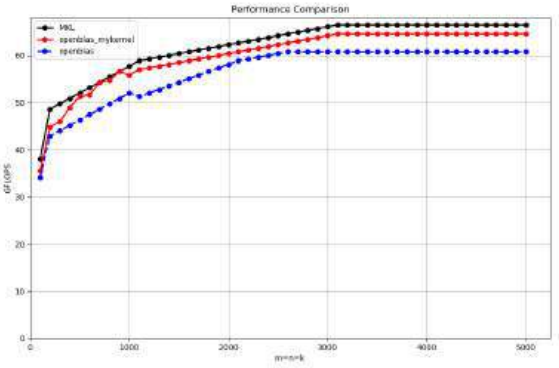


Fig. 5.

B. Instruction Reordering Test

This section tested the effects of instruction reordering described in Section 2.2. We used matrix computation as the test algorithm and implemented two versions of the computation method. The first version used the C++ implementation of the AVX-512 instruction set for the micro-kernel, while the second version used inline assembly with instruction reordering.

We constructed matrices of sizes 1000x1000, 2000x2000, and 5000x5000 for the computations. Each type of computation was run 100 times, and the time taken for each was recorded. Table 1 shows the time consumed by the first computation method, and Table 2 shows the time consumed by the second computation method, with the unit being milliseconds (ms).

TABLE I
The time consumption table for C++ instruction

	1000x1000	2000x2000	5000x5000
max	35.497	269.875	3792.372
min	32.224	242.147	3593.563
mean	33.074	243.915	3694.938

TABLE II
The time consumption table for assembly instruction reordering

	1000x1000	2000x2000	5000x5000
max	28.398	213.092	2844.372
min	25.443	208.270	2695.182
mean	26.359	209.767	2771.204

By comparing the results, we aim to analyze the performance improvements gained from using inline assembly with instruction reordering versus the standard $C++$ AVX-512 implementation. This comparison helps to identify the efficiency of the reordering strategy and its impact on the overall computation time for different matrix sizes.

Acknowledgment

The work of Shuang Li was supported by the Fundamental Research Funds for the Central Universities (XK2050021008)

References

[1] P. Jin, V. Rathod, and X. Zhu, "Pooling pyramid network for object detection," arXiv preprint arXiv:1807.03284, 2018.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[4] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 6399–6408.

[5] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.

[6] M. Mathieu, M. Henaff, and Y. LeCun, "Fast training of convolutional networks through ffts," arXiv preprint arXiv:1312.5851, 2013.

[7] A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4013–4021.

[8] Y. Zhao, D. Wang, L. Wang, and P. Liu, "A faster algorithm for reducing the computational complexity of convolutional neural networks," Algorithms, vol. 11, no. 10, p. 159, 2018.

[9] H. Lan, J. Meng, C. Hundt, B. Schmidt, M. Deng, X. Wang, W. Liu, Y. Qiao, and S. Feng, "Feathercnn: Fast inference computation with tensorgemm on arm architectures," IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 3, pp. 580–594, 2019.

[10] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in Tenth international workshop on frontiers in handwriting recognition. Suvisoft, 2006.

[11] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cudnn: Efficient primitives for deep learning," arXiv preprint arXiv:1410.0759, 2014.

[12] E. Georganas, S. Avancha, K. Banerjee, D. Kalamkar, G. Henry, H. Pabst, and A. Heinecke, "Anatomy of high-performance deep learning convolutions on simd architectures," in SC18: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2018, pp. 830–841.

[13] M. Krainiuk, M. Goli, and V. R. Pascuzzi, "oneapi open-source math library interface," in 2021 International Workshop on Performance, Portability and Productivity in HPC (P3HPC). IEEE, 2021, pp. 22–32.

[14] K. Goto and R. A. v. d. Geijn, "Anatomy of high-performance matrix multiplication," ACM Transactions on Mathematical Software (TOMS), vol. 34, no. 3, pp. 1–25, 2008.

[15] T. M. Low, F. D. Igual, T. M. Smith, and E. S. Quintana-Orti, "Analytical modeling is enough for high-performance blis," ACM Transactions on Mathematical Software (TOMS), vol. 43, no. 2, pp. 1–18, 2016.

[16] D. Yan, W. Wang, and X. Chu, "Optimizing batched winograd convolution on gpus," in Proceedings of the 25th ACM SIGPLAN symposium on principles and practice of parallel programming, 2020, pp. 32–44.

[17] R. Lim, Y. Lee, R. Kim, and J. Choi, "An implementation of matrix–matrix multiplication on the intel knl processor with avx-512," Cluster Computing, vol. 21, pp. 1785–1795, 2018.

Sizing and optimization of a hybrid green energy system for radio sites of mobile telecommunications networks

1st Régis Donald HONTINFINDE
LSIMA[+]
University of Abomey (UNSTIM)
Abomey, Republic of Benin
donald.hontinfinde@yahoo.com

2nd Sylvain TCHIKE
DEE[++]*, EPAC*
University Of Abomey Calavi (UAC)
Abomey-Calavi, Republic of Benin
tchim84@yahoo.fr

3rd Géraud AZEHOUN-PAZOU
LSIMA[+]
University of Abomey (UNSTIM)
Abomey, Republic of Benin
geraud.pazou@gmail.com

4th Audace KOSOKO B. DIDAVI
DEE[++]*, EPAC*
University Of Abomey Calavi (UAC)
Abomey-Calavi, Republic of Benin
didavia@gmail.com

5th Giro Arnaud M. HOUNSOU
LSIMA[+]
name of organization (of Affiliation)
Abomey, Republic of Benin
arnaudhounsous91@gmail.com

6th Christian Djidjoho AKOWANOU
LSIMA[+]
University of Abomey (UNSTIM)
Abomey, Republic of Benin
djidjohoako@yahoo.fr

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899642

Abstract*—In the telecommunications field, the ability of algorithms to transform a sector is no longer in doubt. They could well have major consequences on the way they produce and consume energy, optimize a traffic propagation etc. Between reducing the energy balance of network installations and better use of renewable energies, operators are undertaking major projects. And each time, algorithms are at the heart of these changes. This article seeks to optimize a hybrid system composed of renewable energy (photovoltaic panels, wind turbines) and batteries to power telecommunications base stations (BTS) sites. It proceeds with modeling and optimization based on the non-dominated sorting genetic algorithm (NSGA-II). A multi-objective formulation is proposed to minimize the power loss probability (LPSP) and the levelized cost of energy (LCOE). Input data including ambient temperature, solar irradiation, wind speed and the load profile of the BTS site are considered for simulation in Matlab software. The results obtained are very encouraging for the integration of green energy into modern telecommunications systems. The optimization resulted in a hybrid system composed of 48 photovoltaic modules, each with a capacity of 540 peak watts (Wp), 17 wind turbines, each with a nominal power of one kilowatt, and a storage system with a capacity of 1500 ampere-hours (Ah). The adoption of this system would cost 0.13 euros per kilowatt-hour in our context in Benin with a reliability of 99.2%, providing reliable energy at a lower cost while contributing to achieving the seventh Sustainable Development Goal (SDG).

***Keywords*—Mobile Network, telecommunications, BTS radio site, multi-objective problem, optimization, optimization, green energy**

I. Introduction

The telecommunications sector is experiencing continuous expansion, driven by the growing need to stay connected and access real-time information [1]. To meet this demand, operators are installing numerous radio sites equipped with base transceiver stations (BTS). These stations, equipped with transmitter-receiver antennas, enable communication with mobile devices and requires a constant power supply, efficient cooling, and regular maintenance, leading to high energy consumption. Radio sites alone account for 70% of operators energy bills [2]. Global telecommunications networks electrical consumption is increasing by about 10% per year, according to Lambert et al. (2012) [3].

Many BTS sites are still powered by diesel generators and fossil fuels, thereby contributing to CO2 emissions and climate change. In light of the challenges posed by climate change and the depletion of fossil resources, it is imperative to reduce the use of polluting energy sources. The "Green Power for Mobile" program, launched by the GSMA in 2008, promotes the integration of renewable energy into the telecommunications sector [4].

Fig 1 : Benin Photovoltaic potential [5]

As shown in figure 1, Benin has good solar photovoltaic potential evenly distributed across the country, with annual producibility ranging from 1314 kWh/kWp to 1650 kWh/kWp. This potential represents a valuable resource for BTS, providing them with a clean and sustainable energy

LSIMA[+] : Laboratory of Science Engineering and Mathematics
DEE[++] : Department of Electrical Engineering

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

source through photovoltaic panels. In addition to solar power, coastal regions benefit from regular sea winds, offering a promising wind power potential.

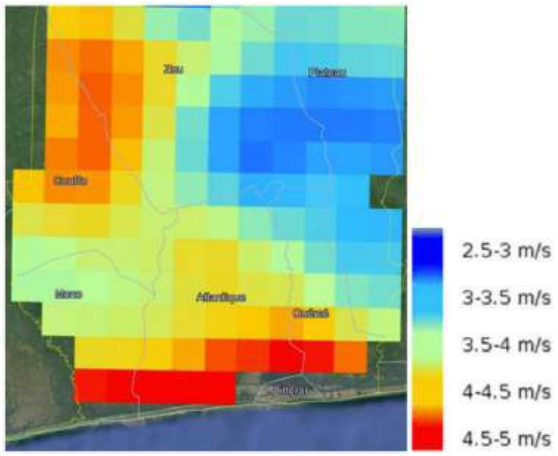


Fig 2 : Benin wind power potential [6]

This potential is particularly strong along the coastal strip, as illustrated in Figure 2, with average wind speeds ranging from 4 to 5 m/s. Harnessing this resource in combination with solar energy could help stabilize the power supply to BTS while reducing dependence on conventional energy sources. However, renewable energy sources have the drawback of being intermittent, as their availability depends on weather conditions. Moreover, electricity demand fluctuates throughout the year and does not always coincide with renewable energy production. Benin's coasts, with their solar and wind potential, represent a unique opportunity for the installation of BTS powered by clean, autonomous, and long-term cost-effective energy sources. These solutions align with the energy transition and the fight against climate change [7]. However, to mitigate the intermittency of these resources, optimizing a hybrid PV/Wind system with batteries appears to be a more viable option to ensure continuous power supply. This study therefore aims to optimize the sizing of a hybrid PV/Wind/Battery system for BTS sites in the coastal regions of Benin.

II. METHODS

A. Data acquisition and pre-processing

The selected site for our study is a rooftop BTS station from the MTN BENIN mobile telecommunications network, located on the roof of the Sun Beach Hotel. This site was chosen due to its proximity to the coastal strip, which allows for taking advantage of the optimal climatic condition characteristic of this region. The electrical consumption data obtained for this BTS site can be represented in Figure 3.

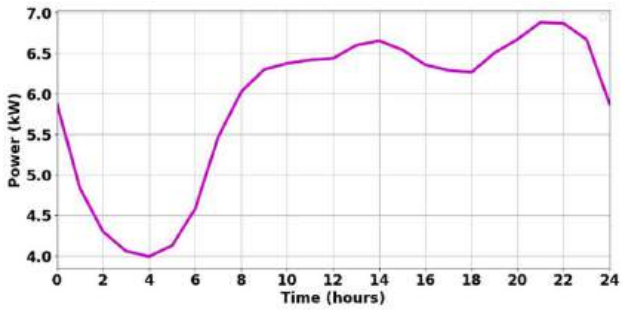


Fig 3 : Load profile of the BTS site under study

Using the geographic coordinates of latitude 6° 21' 27.36" N and longitude 2° 22' 17.76" E for the BTS site under study, the meteorological input data were obtained from the NASA website from 2019 to 2023 [8]. The characteristics of the satellite are as follows:

- Name : NASA/POWER CERES/MERRA2 ;
- Resolution: 0.5 x 0.625 degrees (lat/lon);
- Region: 30.02 meters.

The meteorological parameters used are :

- Ir : Downward shortwave irradiance at the surface for all skies, hourly (Wh/m²) ;
- WS10M: Wind speed at 10 meters (m/s) ;
- T2M: Hourly temperature at 2 meters (°C).

B. Energy Modeling of the Photovoltaic Field

The instantaneous electrical power output of the photovoltaic field at any given time t can be estimated by [9]:

$$P_{PV}(t) = N_{PV}.P_{PVcu}.\left(\frac{G(t)}{G_{STC}}\right).[1+\gamma.(T_{module}(t)-(T_{module_{STC}})] \quad (1)$$

With:

- N_{PV} : the number of installed photovoltaic solar modules;
- $P_{PVcu} = 540Wp$: the peak power of the chosen monocrystalline PV module;
- $G(t)$: global solar irradiation received on the plane of the PV field at time t (Wh/m²);
- $G_{STC} = 1000W/m^2$: solar radiation under STC (Standard Test Conditions);
- $\gamma = 0{,}004°C^{-1}$: the maximum power temperature coefficient (ranging from 0.003 to 0.005 °C−1 for crystalline silicon cells) [10];
- $T_{module_{STC}} = 25°C$: temperature of the PV modules under STC (Standard Test Conditions).

The temperature of the PV module cells at time t is given by [9] :

$$T_{module}(t) = \left(\frac{NOCT-20}{800}\right).G(t) + T_{\alpha}(t) \quad (2)$$

With:

- $T_{module}(t)$: temperature of the PV module cells at time t;
- NOCT: Operating temperature of a module under an irradiance of 0.8 kW/m², an ambient temperature of 20 °C, and a wind speed of 1 m/s. It is often between 40 and 50 °C. For our study, we have chosen a value of 45 °C;
- $T_{\propto}(t)$: ambient temperature of the site at time t.

C. Energy Modeling of the Wind Field

The electrical output power at time t of the wind field can be estimated by [11] :

$$P_E(t) = N_E \times \eta_{gen} \times 0{,}5 \times \pi \times R^2 \rho \times C_p \times (V(t))^3 \quad (3)$$

With:

- P_t : the unit power of the turbine in watts;
- ρ: air density [kg/m³];
- R: the radius of the wind rotor in meters;
- V: wind speed in m/s;
- C_p : power coefficient;
- η_{gen} : generator efficiency.

D. Energy Modeling of Storage

The mathematical formulas for the state of charge of the battery differ depending on whether it is in charging or discharging phase. At each moment t, the difference between the energy produced by the sources and the energy required by the load is determined, which allows for analyzing how the state of charge of the batteries evolves [10].

➢ **Battery in charging phase**

$$SOC(t) = SOC(t-1) + \frac{(P_{PV}(t) \times \eta_{HAC_{PV}} + P_E(t) \times \eta_{HAC_{EOL}} - P_L(t)) \times \eta_{BATcharge})}{U_{bus}} \quad (4)$$

With:

− $SOC(t-1)$: state of charge at time $t-1$;

− P_L : The energy demand at time t;

− $\eta_{HAC_{PV}}$: efficiency of the DC/DC converter for the solar panels;

− $\eta_{\mathbf{HAC_{EOL}}}$: efficiency of the DC/DC converter for the wind system;

− $\eta_{\mathbf{BATcharge}}$: energy efficiency of the battery during charging phase;

− U_{bus}: DC bus voltage on the battery side (48 V in our case).

➢ **Battery in discharging phase**

$$SOC(t) = SOC(t-1) + \frac{P_{PV}(t) \times \eta_{HAC_{PV}} + P_E(t) \times \eta_{HAC_{EOL}} - P_L(t)}{U_{bus} \times \eta_{BATdecharge}} \quad (5)$$

With $\eta_{BATdecharge}$, energy efficiency of the battery during the discharging phase.

➢ **Minimum and maximum state of charge**

To avoid any reduction in lifespan, the value of SOC(t) must be kept within this specified range:

$$20\% \leq SOC(t) \leq 95\% \quad (6)$$

E. System optimization

❖ **Decision variables**

The decision variables for our hybrid PV-wind-battery system are:

- The number of photovoltaic solar panels (N_{PV})
- The number of wind turbines (N_E)
- The number of batteries (N_{BAT})

❖ **Constraints**

The table 1 presents the lower and upper bounds of the decision variables used in the optimization.

Table 1 : Bounds of the optimization variables

Lower bound		Variables		Upper bound
10	≤	N_{PV}	≤	100
5	≤	N_E	≤	80
5	≤	N_{BAT}	≤	60

These bounds define the constraints within which the optimization model must operate.

❖ **Objectives functions**

The choice of objective functions in this study aims to balance the economic profitability and the energy reliability of our system. To achieve this, two objective functions have been established.

- Objective 1 : minimize the probability of power loss of the system described by the following relation [12]:

$$LPSP = \frac{\sum_{t=1}^{8760} LPS(t)}{\sum_{t=1}^{8760} P_L(t)} \quad (9)$$

$$\text{Avec } LPS(t) = P_{PV}(t) + P_E(t) + P_{BAT}(t) - P_L(t) \quad (10)$$

- Objective 2 : Minimize the cost of energy described by the following relation [13] :

$$LCOE = \frac{\text{Total cost over the lifetime}}{\text{Total energy consumed over the lifetime}} \quad (11)$$

❖ Solution approach

The resolution of our optimization problem was carried out using the NSGA II genetic algorithm. The choice of the NSGA-II algorithm is motivated by its effectiveness in multi-objective optimization. NSGA-II is a capable of providing a well-distributed Pareto front, essential for balancing the trade-off between minimizing the LCOE and LPSP. Additionally, its efficient computational performance makes it suitable for complex systems like hybrid PV-wind-battery setups, where multiple conflicting objectives need to be considered simultaneously. Its general principle is based on the execution of a generational loop. In this work, we set the number of parent individuals and the population size to 80. The approach for optimizing the hybrid PV/Wind/Battery system is illustrated by the flowchart in Figure 4.

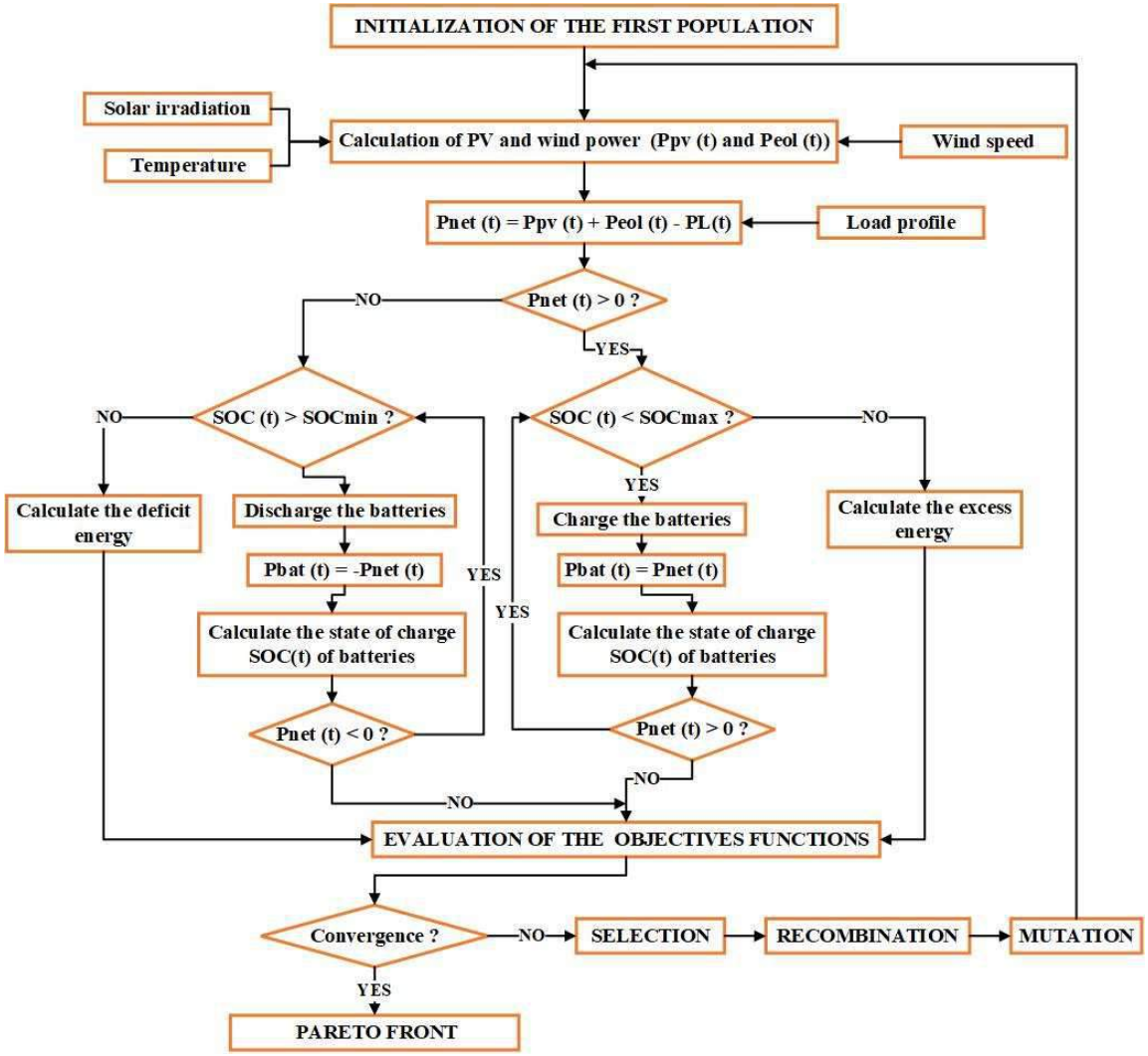


Fig 4 : Flowchart of the proposed optimization process

III. Results

A. Analysis of Pareto points

After implementing the model and optimization, we obtained the Pareto front presented in Figure 5. These points represent the optimal parameters for our hybrid PV/Wind/Battery system in terms of supply reliability, renewable energy production, and cost. Based on the analysis of the different points selected at the elbow of the Pareto front, a point with balanced trade-offs emerges that best meets the requirements of this study, with an LPSP of 0.008 and an LCOE of 84.15 FCFA/kWh. This is therefore the optimal choice for a balanced trade-off between cost and reliability for the hybrid PV-wind-battery system.

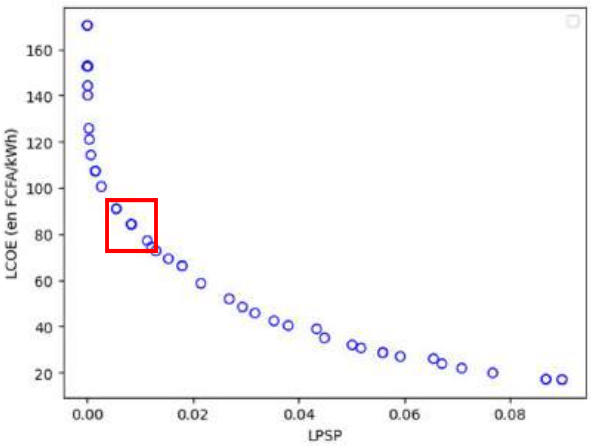


Fig 5 : Pareto front

The characteristics considered result in the hybrid system shown in Figure 6.

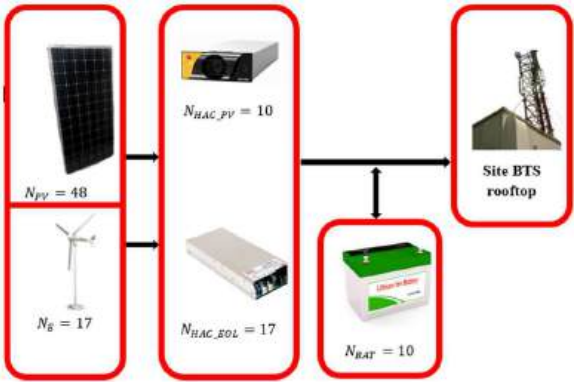


Fig 6 : Characteristics of the optimal point

B. Simulation

Considering the characteristics of the optimal point, we conducted simulations using MATLAB. Thus, the operation (energy flow) of the proposed hybrid system over a full year can be represented by Figure 7.

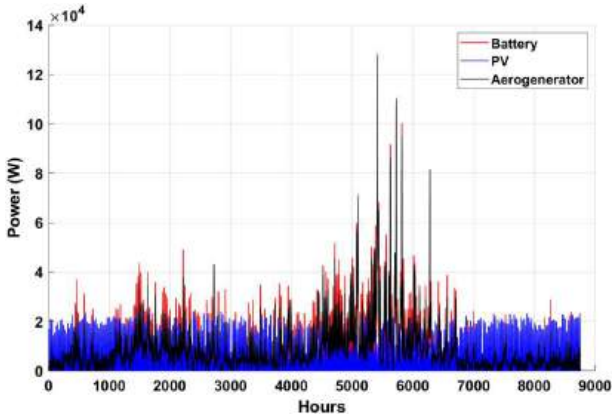


Fig 7 : Energy flow of the system over a year

According to Figure 4, we have:

$$P_{net}(t) = P_{PV}(t) + P_E(t) - P_L(t) \quad (12)$$

— If $P_{net}(t) > 0 : P_{BAT}(t) = P_{net}(t)$ (13)

— If $P_{net}(t) < 0 : P_{BAT}(t) = -P_{net}(t)$ (14)

Thus, for a perfect system, we would obtain:

$$P_{PV}(t) + P_E(t) - P_L(t) - |P_{BAT}(t)| = 0 \quad (15)$$

The simulation conducted at the optimal point over 24 hours can be represented in Figure 8 below.

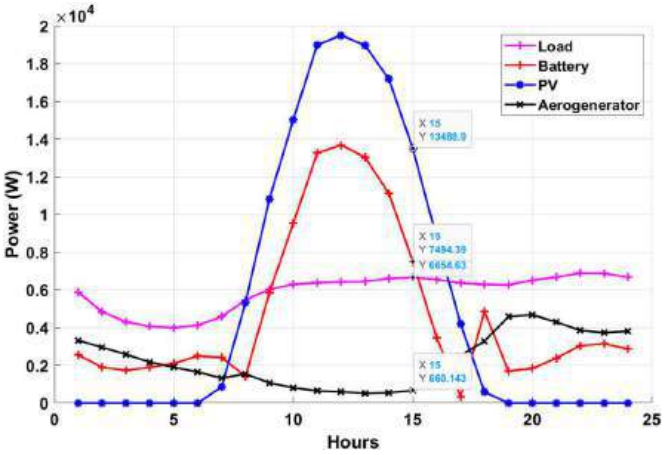


Fig 8 : Energy profile of the hybrid system over 24 hours

By recording the values of each parameter at 15 hours, we have:

$$P_{PV}(t) + P_E(t) - P_L(t) - |P_{BAT}(t)| = 0{,}023\ \text{W} \quad (16)$$

We notice that the difference between the net available power and the power received by the batteries is approximately zero. This corresponds to a very negligible loss.

The simulation with the characteristics of our optimal point was also conducted over different time intervals of 72 hours and one week, and can be represented respectively in Figures 9 and 10.

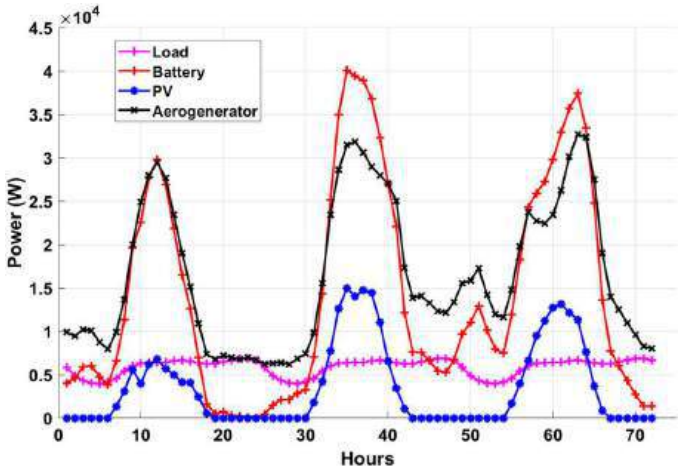


Fig 9 : Energy profile of the hybrid system over 72 hours

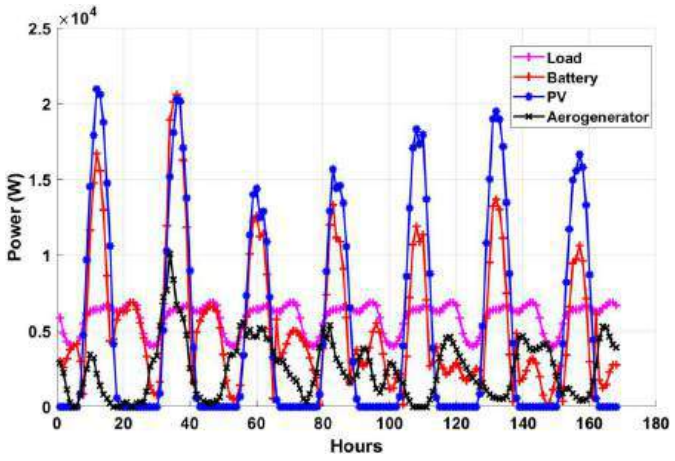


Fig 10 : Energy profile of the hybrid system over one week

For our optimal point, these different curves adequately satisfy the condition:

$$P_{PV}(t) + P_E(t) - P_L(t) - |P_{BAT}(t)| \approx 0 \qquad (17)$$

From all the curves obtained and presented above, it is evident that the proposed system adequately meets the load and utilizes green energy, compared to the current mobile network systems that rely solely on diesel systems with storage and/or conventional grid power.

IV. Conclusion

In summary, this study demonstrates that it is possible to meet the energy needs of BTS sites while reducing the carbon footprint through a hybrid system combining renewable energy sources. The NSGA-II has led to a technically reliable and economically advantageous solution, with a reliability of 99.92% and an energy cost of 0.13 euros per kilowatt-hour. This model provides a sustainable alternative to diesel generators, thus promoting a transition to more environmentally friendly energy sources. Moving forward, on-site experiments and the integration of hybrid systems with the conventional electrical grid could enhance the reliability and efficiency of BTS installations in contexts where space and/or climatic conditions are restrictive.

References

[1] A. Leconte, « Gestion de l'Énergie dans le Secteur des Télécommunications [Étude de Cas - Ooredoo] », Spacewell Energyhttps://www.dexma.com/fr/blog-fr/gestion-de-lenergie-dans-le-secteur-des-telecommunications-etude-de-cas-ooredoo/

[2] « Can the telecoms industry power down its impact on the environment? », https: //inform.tmforum.org/features-and-opinion/can-the-telecoms-industry- power-down-its-impact-on-the-environment/

[3] « Worldwide electricity consumption of communication networks ». https://opg.optica.org/oe/fulltext.cfm?uri=oe-20-26-B513&id=246736

[4] « ENERGIE VERTE POUR LES RESEAUX MOBILES ». Consulté le: 17 août 2024. [En ligne]. Disponible sur: https://www.itu.int/itunews/manager/display.asp?lang=fr&year=2009&issue=04&ipage=32&ext=html

[5] « Global Solar Atlas ». sur: https://globalsolaratlas.info/download/benin

[6] Peter VISSERS et al., « Etude de faisabilité et plan d'actions pour la fabrication des composantes des aérogénérateurs de petite puissance au Bénin », Partners for Innovation, juin 2018.

[7] « Objectifs de développement | Programme De Développement Des Nations Unies », [En ligne]. Disponible sur: https://www.undp.org/fr/sustainable-development-goals

[8] « POWER | DAV ». Consulté le: 24 août 2024. [En ligne]. Disponible sur: https://power.larc.nasa.gov/data-access-viewer/

[9] F. Fodhil, A. Hamidat, et O. Nadjemi, « Potential, optimization and sensitivity analysis of photovoltaic-diesel-battery hybrid energy system for rural electrification in Algeria », *Energy*, vol. 169, p. 613-624, févr. 2019, doi: 10.1016/j.energy.2018.12.049.

[10] A. H. J. Hounnou, « Dimensionnement optimal d'un système hybride hydroélectrique-photovoltaïque-stockage pour une alimentation rurale isolée », 2019.

[11] Ionel Vechiu, « Modélisation et analyse de l'intégration des énergies renouvelables dans un réseau autonome ».

[12] Dhaker ABBES et al, « Etude d'un système hybride éolien photovoltaïque avec stockage : dimensionnement et analyse du cycle de vie ».

[13] « Levelized cost of electricity », *Wikipédia*. 25 avril 2024. Consulté le: 4 août 2024. [En ligne]. Disponible sur: https://en.wikipedia.org/w/index.php?title=Levelized_cost_of_electricity&oldid=1220645980

Transparent Ransomware Detection in Bitcoin Transactions: Leveraging Machine Learning and Explainable AI

*

Roshan Mohammed
Buckinghamshire New University
UK
roshan.mohammed@bucks.ac.uk
ORCID: 0009-0002-9263-8556

Shahadate Rezvy
Buckinghamshire New University
UK
shahadate.rezvy@bnu.ac.uk
ORCID: 0000000226847117

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899728

***Abstract*—Ransomware presents a significant threat within the Bitcoin ecosystem, necessitating robust detection methods. This study leverages machine learning techniques, particularly Random Forest Model, to identify ransomware activities in Bitcoin transactions using the BitcoinHeist dataset. We integrate explainable AI (XAI) tools, specifically LIME, to enhance model transparency and provide insights into predictions.**

Innovative feature engineering techniques are employed to capture critical transaction patterns, such as loops and volume changes, which improve detection accuracy. Our ensemble models achieve high performance metrics, including accuracy, precision, recall, and ROC-AUC scores. The use of XAI mitigates challenges like false positives and adapts to evolving ransomware tactics by elucidating the factors influencing model decisions.

This research underscores the necessity of interpretability in ransomware detection systems, laying the groundwork for future studies focused on real-time detection and adaptive learning in cybersecurity.

***Index Terms*—Ransomware Detection ,BitcoinHeist Dataset ,Machine Learning,Cybersecurity ,Explainable AI,Blockchain Analysis**

I. Introduction

Ransomware has emerged as a major cyber threat, causing severe financial and operational damage by encrypting victims' files and demanding cryptocurrency payments. Detecting ransomware, particularly through patterns on the Bitcoin blockchain, is a complex challenge for cybersecurity professionals, as traditional methods struggle to keep up with rapidly evolving attack tactics [2].

Machine learning (ML) is increasingly applied in cybersecurity to identify ransomware activity by analyzing transaction data, which can reveal patterns too complex for conventional methods to detect [22]. The *BitcoinHeist* dataset [1], which includes features from Bitcoin addresses linked to ransomware, provides valuable data for training and evaluating ML models.

This paper explores the use of a Random Forest classifier for ransomware detection, addressing class imbalance with oversampling techniques such as SMOTE. Additionally, we enhance model transparency using Local Interpretable Model-Agnostic Explanations (LIME) [21], which allows for the interpretation of individual predictions.

The goal of this study is twofold: (1) to assess the performance of a Random Forest model in classifying ransomware addresses, and (2) to highlight the importance of model interpretability in cybersecurity. While based on established ML techniques, this approach offers a practical solution for identifying ransomware activity on the Bitcoin network, contributing to ongoing efforts to improve cybersecurity.

II. Literature Review

A. Introduction

The integration of machine learning (ML) in cybersecurity, particularly for ransomware detection, has led to significant advances. Ransomware, which encrypts data and demands payment, poses serious financial and operational threats to organizations [2]. This review evaluates existing ML techniques for ransomware detection, identifies gaps, and lays the groundwork for future research. It discusses both supervised and unsupervised approaches, with a focus on challenges such as data scarcity, overfitting, and the need for continuous model adaptation to evolving ransomware tactics [23].

B. Background

Ransomware is a major cybersecurity threat, evolving from simple encryption to sophisticated attacks, often involving cryptocurrencies like Bitcoin [24]. High-profile incidents, such as WannaCry and NotPetya, illustrate the growing complexity of these threats [5], [6]. The rise of Ransomware-as-a-Service (RaaS) has made attacks more accessible, increasing their frequency and scale [7]. Consequently, adaptive cybersecurity measures are crucial [8].

C. Machine Learning in Cybersecurity

Machine learning offers promising solutions to traditional security challenges. By analyzing large datasets, ML models can identify patterns and detect threats with minimal human intervention [2]. In cybersecurity, ML has been applied to intrusion detection and phishing detection, with models like Support Vector Machines (SVM) and Convolutional Neural Networks (CNNs) demonstrating success in identifying malicious activities [9], [10].

1) Machine Learning for Ransomware Detection: Several ML techniques are used for ransomware detection:

Anomaly Detection: Identifies unusual patterns indicative of attacks [25]. Pattern Recognition: Uses neural networks to recognize evolving ransomware tactics [11]. Predictive Analytics: Forecasts potential attacks using historical data [12].

2) Challenges and Advancements: While ML shows promise, challenges remain, including limited quality training data and model overfitting. Ongoing research is exploring deep learning, reinforcement learning, and federated learning to improve detection capabilities and maintain data privacy [14], [15].

D. Specific Studies on Machine Learning for Ransomware Detection

Studies highlight the effectiveness of various ML methods for ransomware detection. For example, Masum et al. (2022) achieved 91.63% accuracy using supervised learning through system call analysis, while Sharma et al. (2021) demonstrated 98.08% accuracy in network traffic analysis using unsupervised learning. Recent deep learning advances, such as Convolutional Neural Networks (CNNs), have surpassed 98% detection rates for malware and zero-day attacks [18]. Recurrent Neural Networks (RNNs) are also valuable for detecting ransomware behaviors over time.

1) Future Directions: Hybrid models combining supervised and unsupervised learning are emerging to address the limitations of each approach. Additionally, Generative Adversarial Networks (GANs) and transfer learning hold potential for building more robust detection systems while preserving data privacy [19], [20]. Continuous innovation in ML techniques is essential to keep pace with the evolving nature of ransomware.

E. Evaluation Methods in Machine Learning Models

To ensure reliability, ML models for ransomware detection are evaluated using several metrics:

Accuracy: The overall correctness of the model. Precision: Measures the accuracy of positive predictions. Recall: Assesses the model's ability to identify actual ransomware. F1-Score: Balances precision and recall, especially in imbalanced datasets. ROC-AUC: Summarizes model performance across classification thresholds.

III. Methodology

A. Dataset Description

This study utilizes the BitcoinHeist dataset, which contains Bitcoin transaction data from January 2009 to December 2018, focusing on transactions greater than 0.3 Bitcoin, as lower amounts are less likely to be associated with ransomware. Ransomware addresses were sourced from three established studies: Montreal, Princeton, and Padua, representing various ransomware families.

Key features extracted from the dataset include:

- **Address:** Unique Bitcoin address.
- **Year:** Year of the transaction.
- **Day:** Day of the year of the transaction.
- **Length:** Transaction chain length.
- **Weight:** Merging behavior of the transaction.
- **Count:** Number of transactions for the address.
- **Looped:** Number of transactions looping back to the same address.
- **Neighbors:** Unique addresses associated with the transaction.
- **Income:** Total Bitcoin income (in Satoshis) for the address.
- **Label:** Ransomware family or 'white' for non-ransomware addresses.

Figures 2 and 2 show the distribution of ransomware and non-ransomware addresses, and yearly Bitcoin transaction trends.

B. Data Preprocessing

The dataset was preprocessed by removing outliers using the Interquartile Range (IQR) method on the income feature. Values outside $Q_1 - 1.5 \times \text{IQR}$ and $Q_3 + 1.5 \times \text{IQR}$were excluded to focus on typical transaction behaviors.

A binary label column was created, marking ransomware-associated addresses as '1' and non-ransomware ('white') addresses as '0' for binary classification.

C. Model Training

The Random Forest classifier was chosen for its effectiveness with high-dimensional and imbalanced data. The dataset was split into 80% training and 20% testing sets. To address class imbalance, Synthetic Minority Over-sampling Technique (SMOTE) was applied, generating synthetic samples for the minority class. Class weights were adjusted, and hyperparameters were optimized using cross-validation.

D. Model Evaluation

Model performance was evaluated using several metrics:

- **Accuracy:** Overall correctness of the model.
- **Precision:** Ability to correctly identify ransomware transactions.
- **Recall:** Ability to identify all actual ransomware transactions.
- **F1-Score:** Balances precision and recall, useful for imbalanced datasets.
- **ROC-AUC:** Summarize model performance across thresholds.

To improve interpretability, the LIME (Local Interpretable Model-agnostic Explanations) framework was applied, providing insights into the most influential features for individual predictions.

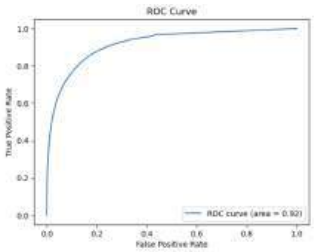


Fig. 1. ROC curve of the Random Forest model with an AUC score of 0.92.

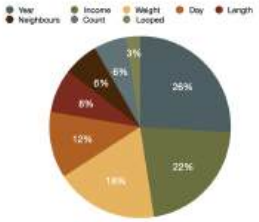


Fig. 2. Feature importance of the Random Forest model, showing the impact of each feature.

E. Implementation

Data processing, model training, and evaluation were carried out using Python. Pandas was used for data manipulation, Scikit-learn for machine learning, and Imbalanced-learn for handling class imbalance. The LIME library was used for visualizing model predictions, ensuring transparency.

IV. Findings

The analysis of ransomware detection using the Bitcoin-Heist dataset yielded significant insights into the effectiveness of the Random Forest model and the impact of various features on detection performance. The findings are presented below, highlighting model performance metrics, feature importance, and the interpretability of model predictions.

A. Model Performance

The Random Forest model demonstrated a high level of efficacy in detecting ransomware transactions. The performance metrics obtained from the evaluation of the test set are as follows:

- **Accuracy:** The model achieved an accuracy of 97%, indicating that a substantial majority of transactions were classified correctly.
- **F1-Score:** The F1-score for the ransomware class was found to be 0.34, reflecting a balance between precision and recall but indicating room for improvement in detection capabilities.
- **ROC-AUC Score:** The ROC-AUC score was measured at 0.92, as shown in Fig. 1, suggesting that the model has a high capability to distinguish between ransomware and non-ransomware transactions across different thresholds.

B. Classification Report

The classification report for the model's performance is summarized in Table I.

TABLE I
Classification Report

Class	Precision	Recall	F1-Score	Support
Non-Ransomware (0)	0.99	0.98	0.99	575057
Ransomware (1)	0.25	0.52	0.34	8283
Accuracy			0.97	583340
Macro Avg	0.62	0.75	0.66	583340
Weighted Avg	0.98	0.97	0.98	583340

C. Confusion Matrix Analysis

The confusion matrix shown in Table 2 provides further insights into the model's performance. The matrix displayed the following results:

TABLE II
Confusion Matrix

	Non-Ransomware	Ransomware
Non-Ransomware	562527	12530
Ransomware	4012	4271

The matrix indicates that while the model effectively identified a high number of non-ransomware transactions (true negatives), it struggled with ransomware detection, leading to a significant number of false positives (12530) and false negatives (4012). This imbalance highlights the need for ongoing improvements in detecting ransomware.

D. Feature Importance Analysis

Feature importance analysis(Fig. 2.) revealed the following rankings of features contributing to the model's decision-making process:

- **Year:** 25.9%
- **Income:** 21.6%
- **Weight:** 18.3%
- **Day:** 11.8%
- **Length:** 7.8%
- **Neighbors:** 6.2%
- **Count:** 5.8%
- **Looped:** 2.5%

The analysis indicates that the 'Year' and 'Income' features are the most influential in predicting ransomware transactions, followed by 'Weight' and 'Day.' This suggests that temporal factors and transaction volumes play a critical role in ransomware detection.

E. Interpretability of Model Predictions

The application of the LIME framework enhanced the interpretability of the model's predictions. By analyzing individual predictions, LIME provided insights into the specific features influencing the model's classifications. This interpretative approach identified that higher transaction volumes and specific transaction patterns were associated with a higher likelihood of being labeled as ransomware.

These findings emphasize the importance of explainability in machine learning models, particularly in cybersecurity contexts, where understanding the rationale behind predictions can guide further investigation and preventive measures.

F. LIME Interpretation

To enhance the interpretability of the Random Forest model's predictions, we utilized the Local Interpretable Model-agnostic Explanations (LIME) framework. LIME provides insights into how individual features influence model predictions, allowing us to better understand the decision-making process,as illustrated in Fig. 3.

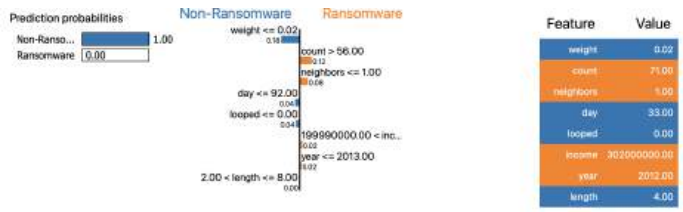


Fig. 3. LIME interpretation of model predictions showing feature contributions for a specific instance.

The LIME results indicate that features such as 'income' and 'year' significantly influenced the model's decision to classify a transaction as ransomware. This emphasizes the importance of these features in distinguishing between ransomware and non-ransomware activities within Bitcoin transactions.

The study highlights the potential of machine learning, specifically Random Forest algorithms, for ransomware detection within the Bitcoin ecosystem. While the model exhibits promising accuracy and ROC-AUC scores, challenges remain in reducing false positives and improving recall rates. Future work should focus on refining feature engineering techniques and exploring additional machine learning models to enhance detection capabilities further.

These findings lay the groundwork for developing more effective real-time ransomware detection systems that leverage advanced machine learning techniques and explainable AI, ensuring both robustness and transparency in decision-making processes.

V. Discussion

The findings of this study highlight the potential of machine learning, particularly the Random Forest model, in detecting ransomware transactions within the Bitcoin ecosystem. Using the BitcoinHeist dataset, the model achieved a high accuracy of 97

A key issue is the model's performance disparity between the two classes. While it accurately identified non-ransomware transactions (precision of 0.99, recall of 0.98), its performance on ransomware transactions was subpar, with a precision of 0.25 and recall of 0.52. This indicates a struggle with false positives and false negatives, which is concerning in cybersecurity contexts where misclassifications could lead to alarm fatigue and missed threats.

This poor performance on ransomware transactions underscores the impact of class imbalance in the dataset, where non-ransomware transactions vastly outnumber ransomware ones. Future research should explore methods to address this imbalance, such as using SMOTE or cost-sensitive learning approaches that prioritize the minority class.

Additionally, the model's high false positive rate suggests challenges in its real-world applicability, particularly in real-time systems. Reducing false alarms without compromising recall will be crucial to improving trust in automated detection tools.

While feature engineering improved detection, the completeness of the feature set remains uncertain. Future studies should investigate additional features, such as temporal or behavioral characteristics, to further refine the model. Moreover, as ransomware tactics evolve, continuous retraining with updated data will be necessary to maintain the model's effectiveness.

In conclusion, this study demonstrates the promise of machine learning for ransomware detection but also highlights critical areas for improvement. Addressing class imbalance, enhancing model precision, and ensuring adaptability to new threats will be essential for deploying effective, real-world detection systems.

VI. Limitations and Future Work

This study demonstrates the potential of machine learning in detecting ransomware within Bitcoin transactions. However, the proposed approach has notable limitations, which also guide future research directions.

Simplified Approach: The methodology presented in this study employs a Random Forest model combined with synthetic oversampling techniques. While effective for proof of concept, this approach remains relatively simple and does not fully capture the complexity of real-world ransomware behaviors. Future research should explore more advanced techniques, such as deep learning or hybrid models, to better handle the intricacies of evolving ransomware patterns.

Proof of Concept Limitations: Although the results show the model's potential in detecting ransomware transactions, the proof of concept is limited by the dataset and evaluation metrics used. The reliance on historical data from the BitcoinHeist dataset (2009–2018) may not accurately represent current ransomware trends. Additionally, the evaluation primarily focuses on accuracy, which does not fully address practical application challenges. Future work should incorporate more recent datasets and additional evaluation metrics, such as detection rates in real-world environments and the impact of false positives. **Data Challenges:** The dataset exhibits class imbalance, with fewer ransomware transactions compared to non-ransomware transactions. While techniques like SMOTE were used, synthetic samples may not capture the full complexity of ransomware activity. Future studies should explore more advanced methods for handling class imbalance and consider incorporating real-time data streams to improve model training. **Model Development and Generalization:**

The feature set, drawn from existing literature, may overlook critical factors such as temporal patterns and advanced network characteristics. Additionally, the Random Forest model may not generalize well to new datasets or emerging ransomware types. Future work should focus on expanding the feature set and testing the model across diverse, real-world datasets to ensure robustness. **Interpretability and Usability:** While explainable AI techniques such as LIME were used to enhance model transparency, there are still challenges in ensuring complete interpretability and user-friendliness. Future research should prioritize developing models that balance predictive accuracy with interpretability and usability to foster trust among cybersecurity professionals. **Real-Time Adaptation:** Future work should aim to develop real-time ransomware detection systems capable of analyzing continuous data streams and adapting to new ransomware tactics. This would enable a shift from reactive to proactive defense strategies, improving the overall effectiveness of cybersecurity efforts.

VII. Conclusion

This study investigates the application of machine learning techniques for ransomware detection within the Bitcoin ecosystem, focusing on the BitcoinHeist dataset. The results demonstrate the efficacy of employing Random Forest classifiers, which achieved high accuracy, precision, and recall in distinguishing between ransomware and non-ransomware transactions. Specifically, the model exhibited an overall accuracy of 97%, underscoring its potential as a viable tool for cybersecurity practitioners in combating ransomware threats.

Despite the promising outcomes, several limitations were identified, including dataset constraints and class imbalance, which may affect the generalisability of the findings. Addressing these limitations through future research will be essential in enhancing the robustness of the proposed models. Additionally, the integration of explainable AI tools like LIME has highlighted the importance of transparency in model predictions, facilitating a better understanding of the underlying factors contributing to ransomware detection.

In conclusion, this research underscores the critical role of machine learning in the ongoing battle against ransomware, particularly in the context of Bitcoin transactions. The study not only establishes a foundation for future exploration of more sophisticated models and datasets but also emphasizes the need for adaptive learning systems that can keep pace with the rapidly evolving landscape of cyber threats. Continued innovation in this field will be vital for developing effective defenses against ransomware and improving overall cybersecurity measures.

References

[1] C. G. Akcora, Y. Li, Y. R. Gel, and M. Kantarcioglu, "BitcoinHeist: Topological data analysis for ransomware detection on the Bitcoin blockchain," IJCAI-PRICAI 2020, 2019.

[2] J. B. Fraley and J. Cannady, "The promise of machine learning in cybersecurity," in SoutheastCon 2017, IEEE, 2017.

[3] C. G. Akcora, A. Bakar, and A. Kadir, "Ransomware detection via machine learning," *Computers & Security*, vol. 90, p. 101674, 2019.

[4] S. Corbet and J. W. Goodell, "The reputational contagion effects of ransomware attacks," Finance Research Letters, vol. 47, p. 102715, 2022.

[5] S.-C. Hsiao and D.-Y. Kao, "The static analysis of WannaCry ransomware," in 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon, Korea (South), 2018, pp. 153–158.

[6] R. A. Lika, D. Murugiah, S. N. Brohi, and D. Ramasamy, "NotPetya: Cyber attack prevention through awareness via gamification," in 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), Shah Alam, Malaysia, 2018.

[7] P. H. Meland, Y. F. Bayoumy, and G. Sindre, "The Ransomware-as-a-Service economy within the darknet," *Computers & Security*, vol. 92, p. 101762, 2020.

[8] L. Bekkers, S. van 't Hoff-de Goede, E. Misana-ter Huurne, Y. van Houten, R. Spithoven, and E. R. Leukfeldt, "Protecting your business against ransomware attacks? Explaining the motivations of entrepreneurs to take future protective measures against cybercrimes using an extended protection motivation theory model," *Computers & Security*, vol. 127, p. 103099, 2023.

[9] S. Zhang, Y. Li, Y. Shi, and M. Hua, "Application of machine learning algorithms in network intrusion detection," in 2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 2022.

[10] M. Hussain, C. Cheng, R. Xu, and M. Afzal, "CNN-Fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet," Information Sciences, vol. 631, pp. 328–345, 2023.

[11] G. Keren and B. Schuller, "Convolutional RNN: An enhanced model for extracting features from sequential data," in 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 3412–3419.

[12] M. Xue, C. Yuan, H. Wu, Y. Zhang, and W. Liu, "Machine learning security: Threats, countermeasures, and evaluations," IEEE Access, vol. 8, pp. 74720–74742, 2020.

[13] S. Verma, M. Ernst, and R. Just, "Removing biased data to improve fairness and accuracy," arXiv preprint arXiv:2102.03054, 2021.

[14] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, IEEE, 2018, pp. 356–363.

[15] H. N. Nguyen, H. T. Nguyen, and D. Lescos, "Detection of ransomware attacks using federated learning based on the CNN model," arXiv preprint arXiv:2405.00418, 2024.

[16] M. Masum, M. J. H. Faruk, H. Shahriar, K. Qian, D. Lo, and M. I. Adnan, "Ransomware classification and detection with machine learning algorithms," in 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2022, pp. 0316–0322.

[17] S. Sharma, C. R. Krishna, and R. Kumar, "RansomDroid: Forensic analysis and detection of Android Ransomware using unsupervised machine learning technique," Forensic Science International: Digital Investigation, vol. 37, p. 301168, 2021.

[18] Q. Wen and K. P. Chow, "CNN based zero-day malware detection using small binary segments," Forensic Science International: Digital Investigation, vol. 38, p. 301128, 2021.

[19] V. Bok and J. Langr, GANs in Action: Deep Learning with Generative Adversarial Networks. Simon and Schuster, 2019.

[20] S. Saha and T. Ahmad, "Federated transfer learning: concept and applications," Intelligenza Artificiale, vol. 15, no. 1, pp. 35–44, 2021.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[22] J. Saxe and K. Berlin, "Deep neural network-based malware detection using two dimensional binary program features," in Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Denver, CO, USA, Oct. 2015, pp. 1–9.

[23] K. Lee and E. Koraye, "AI and machine learning in cybersecurity: Leveraging technology to combat threats," EasyChair Preprint 11610, 2023.

[24] M. Paquet-Clouston, B. Haslhofer, and B. Dupont, "Ransomware payments in the Bitcoin ecosystem," Journal of Cybersecurity, vol. 5, no. 1, p. tyz003, 2019.

[25] N. A. Rosli, W. Yassin, M. A. Faizal, and S. R. Selamat, "Clustering analysis for malware behavior detection using registry data," Int. J. Adv. Comput. Sci. Appl., vol. 10, p. 12, 2019.

New Parallelized Greedy Soup Algorithm in the End-to-end Automatic Speech Recognition

Wei Liu
School of Computer Science and Technology
Changchun University of Science and Technology
Changchun City, China
liuwei@cust.edu.cn

Chunyi Chen*
School of Computer Science and Technology
Changchun University of Science and Technology
Changchun City, China
* Corresponding author: sunny_file@126.com

Yuxin Wei
School of Computer Science and Technology
Changchun University of Science and Technology
Changchun City, China
ccccyy_888@126.com

Zhuo Shao
School of Computer Science and Technology
Changchun University of Science and Technology
Changchun City, China
1084578436@qq.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899738

***Abstract*—This paper proposed a new algorithm in the end-to-end automatic speech recognition. For the end-to-end speech recognition model, we select greedy soup instead of the average model parameters in WeNet. We proposed a dynamic parallel greedy soup optimization algorithm to increase computational speed. The experiments show the importance of proposed method and optimization algorithm. The effectiveness is also proved on multiple corpora.**

Keywords-end-to-end, greedy soup, Transformer, self-attention mechanism

I. INTRODUCTION

In recent years, the end-to-end speech recognition has large improvement.[1] Speech recognition is to classify the input speech according to a certain pattern, and then find the best matching result according to the judgment criterion. Similarity judgment between features is an important basis for classification. The purpose of this study is to improve the classification ability by improving the similarity determination method between features. All methods want to achieve the purpose of fast and accurate matching.

Most of approaches use a Transformer to replace recurrent neural network or its improved method. Transformer can better extract the time-dependent features between the phones of pronunciations. Compared with recurrent neural network (RNN), Transformer solves the long sequence dependence between speech features. In addition, Transformer has better parallel computing ability than the recurrent neural network, that is, Transformer has a faster computing speed. Multi-head self-attention is the core module of Transformer, which plays an important role in extracting long-term speech dependency features.[2]

The Bayesian learning framework that introduces language model estimation can also reduce the word error rate of the model.[3] Aiming at the excessive parameters problem of Transformer model, reduce the weight accuracy and model parameters in the model so that the model can run in an embedded system. Reduce the self-attention in the encoder and increase the feedforward layer encoder to downgrade the model complexity.

The self-attention mechanism in the Transformer model simulates human attention to important things. Various improvement approaches based on the Transformer model have become a hot topic in the research field.[4] Aiming at the problem of excessive calculation of speech sequence input, a method of time-sharing calculation showed that the input sequence features are limited from the time interval to calculate the feature sequence of input speech. The attention heads are randomly removed during training, and all attention heads are retained during testing to obtain the best model structure. By adding a memory bank and a short segment of an input sequence to self-attention, this method can reduce the model size of Transformer and RNN-T. The purpose of the mentioned methods is to reduce the attention model space of streaming speech recognition by enhancing the convolution kernel memory.

Aiming at the acceleration problem of Transformer model, a method by using the combination of CTC and attention is proposed. And a truncated attention decoder is used to reduce the time delay of the model. The CTC module is introduced to predict the length of the target sequence, accelerate the convergence speed of the model.[5] A one-step non-autoregressive method used CTC alignment and the acoustic feature representation of each label was extracted in parallel to accelerate the inference speed of the model. [6,7]

When modeling and testing the proposed method on the WeNet[8] platform, it is found that although the proposed method improves the overall recognition accuracy, the model test still has the problems of slow speed and low resource utilization. Therefore, the greedy soup method is introduced to improve the model test speed and resource utilization efficiency. Therefore, the greedy soup method is introduced to improve the model test speed and resource utilization efficiency.

II. TRADITIONAL METHOD

The greedy soup method is a model parameter averaging method proposed by Google in reference [9]. The general idea is: Firstly, all models are arranged in descending order according to the accuracy of the validation set. Then, the parameters of all models are averaged according to the order of permutation and combination of accuracy from high to low. Finally, whether the effect of the model after averaging the obtained parameters is improved or not, if it is improved, the model obtained by the combination is added, otherwise the model combination is excluded. The specific algorithm is as algorithm 1:

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

Algorithm 1 greedy soup

Input: Potential soup ingredients $\{\theta_0, \cdots, \theta_{k-1}\}$ (optionally sorted in decreasing order of ValAcc(θ_i)).
ingredients ← {}
for i=0 **to** k-1 **do**
  if ValAcc(average(ingredients ∪ $\{\theta_i\}$)) ≥ ValAcc(average(ingredients))
  then
    ingredients ← ingredients ∪ $\{\theta_i\}$
return ingredients

Average model parameter method can improve the performance of model generalization. In experiments, the greed soup method is used to replace the original average model parameter method in WeNet. The experiment results show that the performance of test model is improved. In WeNet, a model and its loss information on the validation set are saved after each epoch training. Using the n model parameters with the lowest loss in the validation set, the average result is obtained as the final model. However, directly averaging the best n models does not always yield satisfactory results. In order to further improve the efficiency, we proposed a parallelized greedy soup and the acceleration method.

III. PROPOSED METHOD

In practical applications, it has been found that traditional greedy soup in speech recognition can significantly increase the benefits from model averaging operation. The traditional method can significantly improve the performance of the final model. However, in speech recognition, the model testing step has the characteristics of slow speed and low resource utilization. In order to improve resource utilization and implement multi-threaded computing, we named the parallelized method as parallelized greedy soup.

A. Parallelized greedy soup

The first step in traditional greedy soup is to sort the model performance according to the model performance from best to worst. The proposed method tests the performance of each model on the validation set. In order to test the performance of each model on the validation set better and faster. We choose the attention rescoring of smaller beam size as testing mode. And use the attention rescoring as the sorting standard. Proved by experiments, greedy soup can get a better average model than the original model.

In order to accelerate the speed of greedy soup and improve the GPU resource utilization, we proposed the parallelized greedy soup method. The general idea is to average all possible combinations of the current model set, the latter one model, and the latter two models first. Then test the possible combinations in parallel. Finally, choose the best combination with the test results to compare with the current model set. Update the current model set by using model combination with best result if best result is obtained. Otherwise, the current model combination is not updated.

To illustrate the steps in detail, the process of selecting the optimal model is showed by Figure 1. In the figure, the soup column represents the current best model set. The best(m_1, m_2,m_3)=m_1 represents the best result selected from the current set m_i, m_i means list of ordered models. The final_index presents the total model list number. The step t means the number of times to select the best model

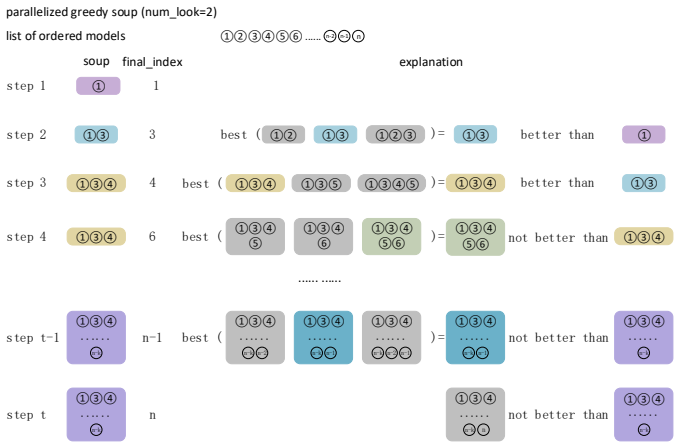


Figure 1. parallelized greedy soup diagrammatic sketch (num_look=2)

If we use $soup$ to represent the current set of models, where 1 represents the model with index final_index+1, 2 represents the model with index final_index+2, and $soup \cup \{1\}$, $soup \cup \{2\}$, and $soup \cup \{1,2\}$ represent all possible combinations between them that must include $soup$ and one other element.

Step 1: Sort all models from good to bad, and record subscripts in turn for sorted models: 1, 2, ⋯ , n, Put the first model into the current model set $soup$, the subscript final_index of the model is marked as 1, The model in $soup$ is the best model set.

Step 2: Averages the parameters of the model in the model collection $soup \cup \{1\}$, $soup \cup \{2\}$, $soup \cup \{1,2\}$. And test the average models obtained in parallel on the validation set.

Step 3: Result judgment. Observe the model average parameters of $soup \cup \{1\}$, $soup \cup \{2\}$ and $soup \cup \{1,2\}$ set. Select the model set with the best average model performance among the $soup \cup \{1\}$, $soup \cup \{2\}$ and $soup \cup \{1,2\}$. If the average model performance of the model set is better than that of $soup$, the current model set $soup$ is replaced with $soup \cup \{1\}$. The replaced model is renamed $soup$, otherwise it will not be replaced. Update the subscript final_index of the model (final_index add 1).

Step 4: If final_index ≥ n-1, return the best model set and end the algorithm, otherwise repeat step 2 and step 3.

Through the above four steps, it can be seen that each cycle of greedy soup can handle one new model that has not been judged, while each cycle of parallelized greedy soup (num_look=2) can handle up to two new models that have not been judged. Therefore, in the case of all average model tests are parallel, the theoretical speed of parallelized greedy soup (num_look=2) is about twice that of traditional greedy soup.

For the follow algorithm, parallelized greedy soup (num_look=2), If soup represents the current model set, 1 Represents a model with the subscript final_index+1, 2 Represents a model with the subscript final_index+2, $soup \cup \{1\}$, $soup \cup \{2\}$ and $soup \cup \{1,2\}$ represent all possible

combinations of the current set soup with model 1 and model 2. The traditional greedy soup can also be described as follows:

Algorithm 2 parallelized greedy soup (num_look=2)

Input: Potential soup ingredients $\{\theta_1, \cdots, \theta_k\}$ (optionally sorted in decreasing order of ValAcc(θ_i) or increasing order of loss or WER).
ingredients ← $\{\theta_1\}$
do_val_test(ingredients)
final_index ← 1
while True:
  ingredients1 ← ingredients ∪ $\{\theta_{final\_index+1}\}$
  if final_index + 2 <= k **then**
    ingredients2 ← ingredients ∪ $\{\theta_{final\_index+2}\}$
ingredients3 ← ingredients ∪ $\{\theta_{final\_index+1}, \theta_{final\_index+2}\}$
    ingredients_list ← [ingredients1, ingredients2, ingredients3]
  else
    ingredients_list ← [ingredients1]
  parallelized_do_val_test(ingredients_list)
  # ingredients_best is the best ingredients in ingredients_list
  ingredients_best ← select_best(ingredients_list)
  # best_index is the index of ingredients_best in ingredients_list, start from 0
  best_index ← select_best_index(ingredients_list)
  if ingredients_best **better than** ingredients **then**
    ingredients ← ingredients_best
    if best_index == 0 **then**
      final_index += 1
    elif best_index <= 2 **then**
      final_index += 2
  else
    final_index += 2
  if final_index >= k **then**
    break
return ingredients

B. Acceleration of parallelized greedy soup

Can each cycle of parallelized greedy soup handle more new models have faster speeds? If the hardware conditions can support parameter averaging for N sets simultaneously, we define parallelized greedy soup (num_look=N, N is a natural number)

Algorithm 3 parallelized greedy soup (num_look=N) (N is a natural number)

Input: Potential soup ingredients $\{\theta_0, \cdots, \theta_{k-1}\}$ (optionally sorted in decreasing order of ValAcc(θ_i) or increasing order of loss or WER).
ingredients ← $\{\theta_0\}$
do_val_test(ingredients)
final_index ← 0
num_look ← N
while True:
  ingredients_list ← [all combinations of (ingredients, $\theta_{final\_index+1}$, $\theta_{final\_index+2}$, … , $\theta_{final\_index+num\_look}$) including ingredients and at least one other element]
  parallelized_do_val_test(ingredients_list)
  # ingredients_best is the best ingredients in ingredients_list
  # last_model_index is the maximal index of the model in ingredients_best
  ingredients_best, last_model_index ← select_best(ingredients_list)
  if ingredients_best **better than** ingredients **then**
    ingredients ← ingredients_best
    final_index ← last_model_index
  else
    final_index += num_look
  if final_index >= k - 1 **then**
    break
return ingredients

For the above algorithm, if $soup$ represents the current set of models, k denotes the model with final_index+k (k is an integer and $2 \le k \le N$), it can also be expressed as the following steps:

Step 1: Sort all models from good to bad, the ordered models are recorded as 0, 1, ... , k-1, Put the 0th model into the current model set $soup$, the model subscript final_index is marked as 0. The model in $soup$ is the best model set.

Step 2: All possible combinations are composed of $soup$, each of which includes soup and at least one element other than soup. By the binomial theorem, there are a total of $C_N^1 + C_N^2 + \cdots + C_N^N = 2^N - 1$ combinations that meet the conditions. So these different combinations are denoted as $soup_1$, $soup_2$,⋯, $soup_{2^N-1}$. Average the parameters for all models in each combination, the average models obtained are tested on the validation set in parallel.

Step 3: The model set with the best average model effect in $soup_1$, $soup_2$,⋯, $soup_{2^N-1}$ is selected, note $soup_i$ (i is an integer and $1 \le i \le 2^N - 1$). Record the maximum value of the model subscript in $soup_i$ is j. If the average model of $soup_i$ is better than the average model of the current model set $soup$, $soup$ is updated to $soup_i$ and final_index is updated to j, otherwise, $soup$ is not updated and final_index is updated to final_index+N.

Step 4: If final_index ≥ n-1, return the best model set and end the algorithm, otherwise repeat step 2 and step 3.

Each cycle of parallelized greedy soup (num_look=N) can handle up to N new models that have not been judged, so theoretically up to N times the speed of greedy soup, the occupied resources also increase exponentially with the increase of N. Due to resource constraints, in practical applications, the num_look usually equals to an integer, the integer range is the two-thirds to three-quarters for the number of the actual utilization GPU cores.

IV. EXPERIMENTS

The comparative experiments using traditional greedy soup and parallelized greedy soup (num_look=2) method with Conformer structure. In the early stage of model training, the model trained in each epoch has not yet converged. So the models from the last 80 epochs (from 50th to 129th epochs) are selected and sorted in ascending order of the validation set loss. Through experiments, it has been found that the models with the highest loss are usually not included in the final results by the traditional greedy soup. Therefore, these 80 models are not directly listed as a sorted model list in the parallelized greedy soup. Instead, the top 10 models with the highest loss were removed, and the top 70 models were selected for the next

greedy soup step. Removing a portion of models with high loss can not only linearly reduce the time spent on greedy soup, but also prevent bad models from being added to the final model average results. Use the decoding results of CTC greedy search on the validation set as the standard for judging the quality of the average model.

A. Data set

AISHELL-1 includes 178 hours of voice data, recorded by 400 people from different accent areas in China. Recording takes place in a quiet indoor environment, use a high-fidelity microphone, down sampling to 16 kHz. The ratio of training set, validation set and test set in the data set is 7:2:1. Voice data for 340 people is used in the training set, 40 people for the validation set and 20 for the test set. In addition, the AISHELL-1 training set is used to annotate the text as the corpus of the language model.

Wenetspeech[10] is a data set jointly released by the Audio Speech and Language Processing Research Group (ASLP Lab) of Northwestern Polytechnical University and Hill Shell. It contains about ten thousand hours of speech data sets with a confidence level of [0.95, 1.0]. In these voice data sets with confidence of 1.0, there is a subset L of Wenetspeech-L, totaling 10000 hours. Part as a subset of Wenetspeech-S, a total of 100 hours.

The validation and testing sets of Wenetspeech are more diverse and challenging compared to AISHELL-1, and can better represent the recognition performance of the model in real-world scenarios.

B. Experiment setting

We select 80-dimensional Fbank as speech feature. The frame length is 25ms and the frame shift is 10ms. For all models, the number of encoder blocks (num_block) is 12, the number of decoder blocks is 6, the number of attention heads is 4, and the dimension of attention output is 256. The CTC weight for training is 0.3.

In the Transformer model, AISHELL-1 data set, python 3.8 and WeNet-2.0.0 tool are used to test the unit dot product similarity method. The two cases of adding absolute position encoding and not adding position encoding are tested. In the experiment, sort_size is set to 6400, accum_grad is set to 2. That is each GPU is updated every 2 batch gradients. Warmup_steps for the Adam optimizer is set to 25000 and gradient clipping to 5.0. 250 epochs were used in training. When decoding, the lowest 25 models in loss are averaged for parameters. The parameters for attention re-scoring are ctc weight of 5.0 and beam size of 10. The speed perturb parameter is also used, that is before the audio feature extraction, the audio was randomly adjusted to 0.9 or 1.0 or 1.1 times the original speed.

C. Effects of greedy soup and parallelized greedy soup

We performed traditional greedy soup and parallelized greedy soup on Conformer and recorded the consumed time. All experimental results of WER and time are shown in Table 1, Table 2 and Table 3. The '+' in tables means using method based on Conformer.

First, we compared the proposed parallelized greedy soup with traditional greedy soup method on AISHELL-1 dataset. In Table 1, column rescoring means decoding using rescore method, column with LM means using language model, column time means the running time of training one epoch.

It can be seen that based on the Conformer experiment using AISHELL-1 as the dataset, parallelized greedy soup improved speed by 23.22% while maintaining comparable performance to traditional greedy soup.

TABLE I. COMPARE RESULTS ON AISHELL-1

Model	**rescoring**	**with lm**	**Time**
Conformer	4.420	4.166	/
+ greedy soup	**4.395**	**4.140**	63.08 min
+ parallelized greedy soup	4.405	4.165	48.43 min

Due to the small amount of testing data in the validation and test sets in AISHELL-1, performing greedy soup on the validation set can easily result in an average model that is too adapted to the validation set and deviates from the distribution of the test set data. Therefore, the parallelized greedy soup used here does not have a significant advantage over the default model parameter averaging method.

To verify the effectiveness of the algorithm, we test the proposed method on the large dataset of Wenetspeech-S and Wenetspeech-L respectively. Column dev, test_net and test_meeting denote one validation set and two test sets which decoding method used rescoring method.

TABLE II. COMPARE RESULTS ON WENETSPEECH-S

model	**dev**	**test_net**	**test_ meeting**	**time**
Conformer	14.778	20.167	29.056	/
+ greedy soup	14.401	**19.772**	**28.369**	84.65 min
+ parallelized greedy soup	**14.397**	19.875	28.408	**59.68 min**

From Table 2, the parallelized greedy soup method achieved best result. The training time is almost the same.

Table 1 and Table 2 shows the proposed algorithm gets the lowest word error rate (WER) in different dataset, and the running time of training one epoch is a slight increase in Wenetspeech-S.

The proposed parallelized greedy soup can effectively reduce training time. Each epoch can be reduced by a maximum of 26.68 minutes. To further demonstrate the effectiveness of parallel algorithms on large dataset, we test the proposed method in Wenetspeech-L dataset.

TABLE III. COMPARE RESULTS ON WENETSPEECH-L

model	**dev**	**test_net**	**test_ meeting**	**time**
Conformer	8.87	8.75	17.09	/
+ greedy soup	8.25	8.53	16.04	72.25 min
+ parallelized greedy soup	**8.25**	**8.53**	**16.04**	**45.57 min**

From Table 3, although parallelized greedy soup method achieved the same result compared with traditional greedy soup method, the training speed of parallelized greedy soup is 1.585 times more than greed soup.

On the Wenetspeech-S and Wenetspeech-L datasets, the proposed parallelized greedy soup method is effective. It can effectively avoid over fitting the model to the validation set, thereby obtaining a model that performs better in real-world scenarios.

V. Conclusions

The parallelized greedy soup algorithm proposed in this paper has achieved better results in the experiment of Wenetspeech-S and Wenetspeech-L datasets. In the experiment, the speed of parallelized greedy soup is decreased by more than 40% compared with greedy soup, the experimental accuracy is comparable to that of the original greedy soup. Similarly, the parallelized greedy soup algorithm is also applied on the AISHELL-1 dataset. Although there is no improvement in WER, it basically maintains the original accuracy with faster training speed.

Acknowledgment

This work was supported by the Jilin Provincial Science and Technology Development Plan Project, Grant number 20220201149GX.

References

[1] Jiang Y, Yu J, Yang W, et al. Nextformer: A ConvNeXt Augmented Conformer For End-To-End Speech Recognition[J]. arXiv preprint arXiv:2206.14747, 2022.

[2] Chen X, Wu Y, Wang Z, et al. Developing real-time streaming transformer transducer for speech recognition on large-scale dataset[C] ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 5904-5908.

[3] Xue B, Yu J, Xu J, et al. Bayesian transformer language models for speech recognition[C] ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 7378-7382.

[4] Li J. Recent advances in end-to-end automatic speech recognition[J]. APSIPA Transactions on Signal and Information Processing, 2022, 11(1).

[5] Yoshimura T, Hayashi T, Takeda K, et al. End-to-end automatic speech recognition integrated with ctc-based voice activity detection[C] ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 6999-7003.

[6] Song X, Wu Z, Huang Y, et al. Non-Autoregressive Transformer ASR with CTC-Enhanced Decoder Input[C] ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 5894-5898.

[7] Higuchi Y, Inaguma H, Watanabe S, et al. Improved Mask-CTC for Non-Autoregressive End-to-End ASR[C]//ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 8363-8367.

[8] Yao Z, Wu D, Wang X, et al. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit[J]. arXiv preprint arXiv:2102.01547, 2021.

[9] Xue B, Yu J, Xu J, et al. Bayesian transformer language models for speech recognition[C] ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 7378-7382.

[10] Zhang B, Lv H, Guo P, et al. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition[C] ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022: 6182-6186.

Cross-Subject Drowsiness Recognition Based on EEG Signals of Frontal Area

1st Jinbiao Ren
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
37697210@qq.com

2nd Tao Deng
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
122202972@qq.com

3rd Yanlin Huang
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
634529848@qq.com

4th Da Qu*
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
1243306704@qq.com

5th Jianqiu Su
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
53430419@qq.com

6th Bingen Li
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
1248015370@qq.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899448

***Abstract*—For EEG-based drowsiness recognition, cross-subject recognition is desirable since calibrating for each subject is time-consuming. Although EEG provides objective measures of fatigue with very high temporal resolution, its practical application is often limited by the complexity of multichannel systems. Although frontal EEG signals are more practical and portable, they may suffer from higher levels of noise and less robust signal quality compared to full-cap setup. In this paper, we introduce a Squeeze and Excitation Convolutional Neural Network (SECNN) aimed at improving cross-subject recognition of drowsiness utilizing solely forehead EEG signals. The network features a compact structure and employs separable convolutions specifically to reduce the number of parameters and computational complexity, thereby improving the network's convergence and efficiency. Furthermore, the use of separable convolutions aids in further noise and artifact removal, enhancing the richness of the extracted features from the forehead EEG signals. Results show that the model achieves an average accuracy of 80.68% for cross-subject drowsiness recognition on a public dataset using only three forehead EEG channels. This performance significantly exceeds that of traditional baseline methods (53.40%-72.68%) and surpasses state-of-the-art deep learning methods (68.66%-78.03%) that utilized 30 EEG channels.**

***Index Terms*—Electroencephalogram (EEG), cross-subject, depthwise separable convolution, driver drowsiness recognition, Squeeze and Excitation**

I. Introduction

In high-risk occupational settings like construction, high-altitude work, and other intensive environments, drowsiness can lead to decreased vigilance, attention, and reaction time, posing severe safety hazards. Fatigue detection systems are vital not only in driving but also in these industrial scenarios, where maintaining alertness is critical to prevent accidents and ensure worker safety. In construction sites or high-altitude operations, the risk factors related to drowsiness can be especially pronounced due to long working hours, physically demanding tasks, and exposure to challenging environmental conditions. Continuous drowsiness monitoring could thus play a significant role in these industries by providing real-time assessments of workers' cognitive states and enabling timely interventions to prevent accidents.

Electroencephalography (EEG) technology has emerged as an effective non-invasive method for monitoring fatigue and drowsiness in real time, as it captures brain activity with high temporal resolution. While traditional full-scalp EEG systems provide comprehensive brain signal coverage, their cumbersome setups make them impractical for mobile or field applications, where lightweight, comfortable, and less obtrusive systems are required. In such work settings, a portable and frontal EEG-based system, which relies on fewer electrodes, offers a feasible and cost-effective solution. However, creating an effective drowsiness detection system based solely on frontal EEG signals presents significant challenges, especially in environments with high movement and noise interference, such as construction sites.

The primary difficulty in using only frontal EEG signals lies in distinguishing relevant fatigue patterns from low signal-to-noise ratio data derived from a limited number of electrodes. Differences in EEG signals among individuals can further complicate detection, as these variations may arise from individual factors like electrode position shifts, variations in skin-electrode impedance, head shape and size differences, unique brainwave patterns, and interference from unrelated brain activities. Traditional methods based on manual feature extraction are often constrained to specific EEG features, which can lead to the exclusion of other important information necessary for accurate drowsiness detection.

Moreover, research on drowsiness detection that specifically focuses on using only frontal EEG signals remains limited. Most existing studies utilize comprehensive full-cap systems, raising concerns about the adaptability of these methods in practical, simplified setups for construction or high-risk work

* Corresponding author

environments. Deep learning offers a powerful alternative with end-to-end learning capabilities, eliminating the need for manual feature extraction. Deep learning models can learn essential features directly from raw, high-dimensional data by transforming it into a cascade of hierarchical representations optimized through backpropagation.

In this paper, we propose an innovative Separable Convolutional Neural Network (SECNN) designed to detect drowsiness across various industrial and driving contexts by identifying common patterns in frontal EEG signals from multiple subjects. The SECNN features a simplified architecture that utilizes separable convolutions to capture spatiotemporal information from frontal EEG signals, effectively addressing the challenges of low signal quality and inter-subject variability. Our method surpasses leading deep learning models that require 30-channel EEG input, demonstrating robust performance with only three electrodes positioned in the frontal area.

II. Related Work

As a cost-effective brain imaging method, EEG detects voltage fluctuations on the scalp using a series of electrodes, which are caused by ionic currents in the cerebral cortex. Numerous studies have established a robust relationship between drowsiness and the oscillation patterns detected in EEG signals [1]–[3]. Traditional methods for recognizing drowsiness from EEG signals have been extensively researched [4], [5], where features are extracted from raw data and combined with machine learning models for classification.

A. Current Monitoring Systems

To address fatigue detection in high-risk work environments, a variety of monitoring systems and wearable technologies have been explored across fields such as construction, healthcare, and transportation. In construction settings, fatigue management often involves wearable sensors and integrated monitoring software. For example, a study using forearm EMG and IMU wearable sensors combined with recurrent neural networks has shown promising results in continuously tracking worker fatigue on construction sites, providing alerts when workers exhibit signs of excessive fatigue [6].

In healthcare, where long shifts and circadian disruptions are common, fatigue management systems have been critical for improving both worker safety and patient outcomes. Healthcare fatigue management often includes risk-based approaches such as shift maxima and rest scheduling to manage the high demands on healthcare professionals, particularly for roles involving night shifts [7].These systems are structured to prevent fatigue-related errors, which can impact both staff well-being and patient care quality.

The transportation industry, especially sectors like long-haul trucking and aviation, also extensively relies on fatigue detection to enhance safety. Wearable systems capturing both physiological and behavioral data are used to monitor fatigue among drivers, employing a range of sensors to track sleep and cognitive alertness, aiming to prevent accidents linked to drowsiness [8].

Incorporating EEG-based monitoring for these environments—similar to methods used in driver drowsiness detection—can improve the accuracy of fatigue detection by providing real-time cognitive fatigue indicators. However, adopting EEG in such settings would require compact, portable EEG devices that can work effectively with fewer electrodes, comparable to advancements in portable frontal EEG technology for drivers. This adaptation could significantly enhance safety by detecting fatigue before physical symptoms become evident.

B. EEG-based drowsiness recognition

Electroencephalography (EEG) records voltage changes originating from ionic currents caused by synaptic activity in the brain's pyramidal neurons [9], particularly in the outer cortical layer EEG systems typically use between 1 to 256 electrodes following the international 10-20 system, where electrodes are labeled by region, such as AF3 (frontal), Cz (central), and T4 (temporal), facilitating targeted data collection from specific brain areas related to drowsiness.

In recent years, deep learning models, particularly Convolutional Neural Networks (CNNs), have been applied to EEG-based drowsiness recognition due to their ability to learn complex patterns directly from raw data. Lawhern et al. [10] introduced EEGNet, a compact CNN model that effectively extracts spatial and temporal features from multi-channel EEG signals, achieving competitive accuracy across various brain-computer interface (BCI) tasks. Nissimagoudar et al. [11] extended CNN use to singlEG for driver assistance, while Ding et al. implemented a CNN model on mobile devices to classify drowsiness in real time, outperforming traditional machine learning models such as support vector machines (SVM) and linear discriminant analysis (LDA).

For multi-channel EEG analysis, Gao et al. [12] proposed thorating convolutional layers, activation functions, and batch normalization to improve feature extraction and stability in drowsiness detection. Furthermore, Zeng et al. [13] developed two advanced CNN models, EEG-Conv and EEGth the latter integrating deep residual learning to enhance performance and convergence rate, outperforming classifiers based on long short-term memory (LSTM) networks and SVM.

III. Methods

A. Separable Convolution

Our EEG processing approach uses depthwise separable convolutions to streamline feature extraction while reducing the model's complexity [14]. First, a pointwise convolution layer isolates or "demixes" relevant channel features, followed by depthwise convolution to independently extract temporal features across channels [15]. This arrangement enables efficient learning of spatial and temporal EEG patterns with fewer parameters than traditional 2D convolutions, which improves the model's performance and processing efficiency. Although the structure differs slightly from conventional applications, it achieves similar benefits to those seen in lightweight models like MobileNet [16], [17] and Xception [18].

B. SE Block

We adopted the squeeze-and-excitation (SE) approach introduced by Jie Hu et al. [19] in 2017 and adapted it for the domain of EEG drowsiness recognition. The SE algorithm offers specific advantages when analyzing frontal EEG channels in drowsiness detection. EEG signals, especially those from a few frontal electrodes, often contain noise and lack the rich spatial information found in full-cap setups. By applying the SE block, our model can adaptively emphasize task-relevant EEG features, thus strengthening the discriminative power of key channels and improving the overall signal quality.

The SE block consists of three main steps:

Squeeze: Through global average pooling, the SE block captures each channel's global spatial information, compressing it into a single scalar descriptor. This condenses complex EEG patterns into a channel-wise summary, making it easier to highlight relevant brain activity linked to drowsiness. Specifically, given an input feature map $X \in \mathbb{R}^{H\times W\times C}$,the vector $z \in \mathbb{R}^C$ obtained through global average pooling can be expressed as:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c} \tag{1}$$

Excitation: By using a bottleneck network to recompute each channel's weight, the SE block models non-linear dependencies across channels. This operation allows the model to selectively enhance or suppress specific electrode channels based on their relevance, which is crucial when working with limited electrodes in the frontal area. The specific formula is:

$$s = \sigma(W_2 \cdot \mathrm{ReLU}(W_1 \cdot z)) \tag{2}$$

where $W_1 \in \mathbb{R}^{C/r\times C}$ and $W_2 \in \mathbb{R}^{C/r\times C}$ are the weights of the two fully connected layers, r is the dimension reduction factor, and σ is the sigmoid activation function.

Scale: The final scaling step applies the recalibrated channel weights back to the original feature map. This targeted scaling ensures that the model focuses on the most informative EEG signals, reducing the impact of noise and irrelevant data, and improving the robustness of drowsiness detection across subjects. The specific operation is:

$$\hat{X}_{i,j,c} = X_{i,j,c} \cdot s_c \tag{3}$$

In summary, the SE block enables efficient utilization of the sparse information from frontal EEG electrodes, enhancing signal relevance and improving drowsiness classification accuracy by dynamically adjusting channel importance in response to the specific needs of EEG-based drowsiness recognition.

C. Network Design

As shown in Figure 1, the network consists of four main modules to process EEG signals effectively.

Channel Expansion Module: The first module applies a pointwise convolution to the input EEG signals, increasing the number of channels to N_1 (set to 8). This expansion enhances the feature richness of the EEG signal and accelerates network convergence by establishing a broader foundation for feature extraction across channels. The output at this stage, shown in Equation 4, provides a baseline for further feature extraction.

$$h_{i,j}^{(1)} = \sum_{p=1}^{m} w_{i,p}^{(1)} x_{p,j} + b_i^{(1)} \tag{4}$$

Independent Feature Extraction Module: In the second module, depthwise convolution is applied to independently capture spatial and temporal characteristics for each channel created in the first module. This approach efficiently extracts distinct patterns relevant to drowsiness recognition from each channel. The depthwise convolution for a channel $h_{i,j}^{(1)}$ is given by Equation 5, producing a refined feature representation for each channel.

$$h_{i,j}^{(2)} = \begin{cases} \sum_{r=1}^{l} h_{\frac{i+1}{2},j+r-1}^{(1)} w_{i,r}^{(2)} + b_i^{(2)}, when\ i\ is\ odd. \\ \sum_{r=1}^{l} h_{\frac{i}{2},j+r-1}^{(1)} w_{i,r}^{(2)} + b_i^{(2)}, when\ i\ is\ even. \end{cases} \tag{5}$$

Recalibration and Residual Enhancement Module: The third module includes activation and batch normalization layers, followed by the SE (squeeze-and-excitation) block and a pointwise convolution. This module enables feature recalibration to emphasize task-relevant information, with the SE block adaptively adjusting channel weights to improve model focus on key EEG features. The SE layer, specifically applied here, recalibrates features (Equation 6)) to enhance discriminative power. The SE and pointwise convolution layers are integrated via residual connection, which helps maintain essential information while allowing for feature recalibration, reducing redundancy, and enhancing overall model performance.

$$h_{i,j}^{(3)} = SE(h_{i,j}^{(2)}) \tag{6}$$

Classification and Final Recalibration Module: The final module includes activation, SE, global average pooling (GAP), a dropout layer, and fully connected layers with Softmax for classification. The GAP layer, which aggregates global feature information across the EEG time series, significantly reduces parameter count and prevents overfitting. The SE layer in this stage ensures the most relevant feature channels are preserved, while dropout regularizes the model. Finally, a fully connected layer followed by Softmax produces the classification output.

The SE layers are strategically positioned in modules 3 and 4, specifically where critical recalibration of channel importance can enhance EEG feature representation. Including SE in other sections would increase computational cost without a clear performance gain, as early and later-stage features do not require the same level of adaptive recalibration. The design ensures efficient feature extraction, effective recalibration, and precise classification, while avoiding redundancy and maintaining network simplicity.

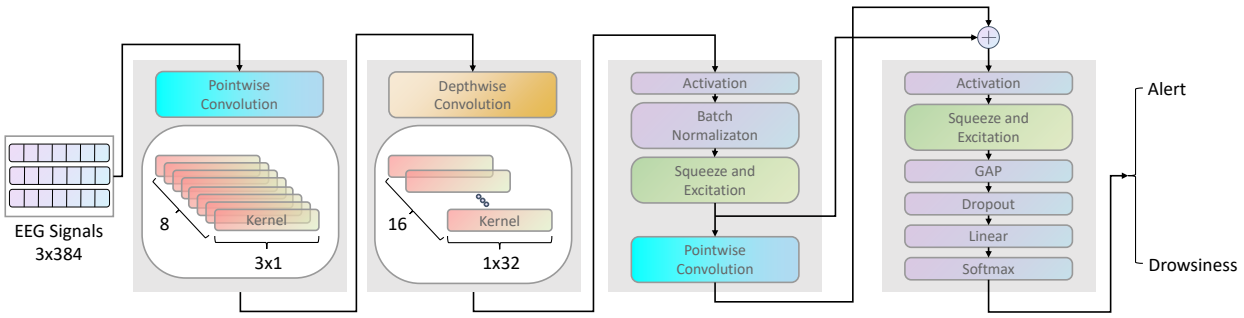


Fig. 1. The architecture of the proposed model.

IV. Experiments

A. Data Preparation

In this study, we utilized a public EEG dataset collected from 27 participants aged between 22 and 28 years. The data were obtained during a 90-minute continuous attention virtual driving task [20]. Participants were instructed to remain focused and respond quickly to randomly introduced lane departure events that caused the vehicle to deviate from the center lane. The start time of the event, reaction time, and end time of the reaction were recorded and synchronized with the EEG data timeline. Drowsiness levels were assessed using local Reaction Time (local-RT), defined as the interval between the start of each lane departure event and the start of the reaction [21].

EEG signals were recorded using a wired EEG cap equipped with 32 Ag/AgCl electrodes (30 EEG electrodes and 2 reference electrodes), following a modified international 10-20 electrode placement system. The dataset wera extracted to a balanced and an unbalanced version by Cui et al. [15]. The balanced dataset contains a total of 2022 samples from 11 different subjects. The number of samples per subject is detailed in Table I. The extracted balanced dataset is available online [22]. The unbalanced dataset contains 2,952 samples from 11 different subjects, with the number of samples per subject listed in Table II.This unbalanced dataset is available online [23]. In this study, we focused on EEG data from three channels (Fp1, Fp2, and Fz). The EEG signals were downsampled to 128 Hz, and 3-second samples prior to each lane departure event were extracted, forming vectors of size 3 (channels) × 384 (sampling points) [14].

TABLE I
NUMBER OF EXTRACTED SAMPLES FROM EACH SUBJECT (BALANCED DATASET)

Subject ID	Sample Number	
	Alert	Drowsiness
1	94	94
2	66	66
3	75	75
4	74	74
5	112	112
6	83	83
7	51	51
8	132	132
9	157	157
10	54	54
11	113	113
Total	1011	1011

TABLE II
NUMBER OF EXTRACTED SAMPLES FROM EACH SUBJECT (UNBALANCED DATASET)

Subject ID	Sample Number	
	Alert	Drowsiness
1	94	96
2	363	66
3	75	180
4	118	74
5	161	112
6	83	116
7	51	103
8	238	132
9	243	157
10	192	54
11	113	131
Total	1731	1221

B. Baseline

1) Deep Learning Methods: For comparison, we evaluate three deep learning models commonly used in EEG signal processing and drowsiness detection. The first model is EEGNet-4,2, an established approach for EEG classification and a baseline for brain-computer interface (BCI) tasks, introduced by Lawhern et al. [10]. The second model, SincShallowNet, incorporates a sinc convolutional layer and serves as a comparison in the time-space context of raw EEG data, as proposed by Davide et al. [24]. Finally, we assess ICNN, an interpretable convolutional neural network by Cui et al. [15], which represents a state-of-the-art (SOTA) approach for cross-subject EEG analysis in drowsiness detection.

2) Machine Learning Methods: In this study, we evaluate five manually extracted feature types and eight classifiers as baselines for EEG-based drowsiness detection. The selected feature types included relative band power features (RelativePower), logarithmic band power features (LogPower) [25], band power ratio features (PowerRatio) [26], wavelet entropy features (WaveletEntropy) [27], and four entropy measures (FourEntropies) including sample entropy, fuzzy entropy, approximate entropy, and spectral entropy [28].

For classification, we used eight classifiers: Decision Trees (DT), Random Forests (RF), k-Nearest Neighbors (KNeighbors), Gaussian Naive Bayes (GNB), Logistic Regression (LR), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Support Vector Machines (SVM). Each classifier was evaluated to determine its performance on the EEG dataset for detecting drowsiness.

C. Implementation Details

The comparison was conducted on a Linux server system equipped with an Intel(R) Xeon(R) Gold 6230R CPU (2.10 GHz) and an NVIDIA GeForce RTX 3090 GPU. The code was implemented and tested on the Python 3.9.0 platform. Although various optimizers, including the Sobolev gradient optimizer [29], [30], were used across different deep learning architectures, we chose Adam [31] for its computational efficiency and significant effectiveness in our approach. The batch size was set to 50, and the default parameters of the Adam optimizer were used ($\eta = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$).

Regarding traditional methods, the band power features were derived using the Welch technique provided by the SciPy library [32]. Classifiers were executed using the sklearn library [33], with default settings applied throughout the process.

D. Mean Accuracy Comparison on the Balanced Dataset

TABLE III
THE MEAN CROSS-SUBJECT CLASSIFICATION ACCURACIES (%) OF THE FIVE BASELINE METHODS COMBINED WITH DIFFERENT CLASSIFIERS.

Classifiers	Relative-Power	Log-Power	Power-Ratio	Wavelet-Entropy	Four-Entropies
DT	60.61	64.30	60.27	53.40	58.16
RF	64.67	69.54	63.39	56.69	61.82
KNeighbors	62.66	71.77	61.62	57.44	61.95
GNB	64.76	**72.68**	58.75	56.34	62.96
LR	68.58	70.24	63.17	**60.40**	60.62
LDA	66.29	70.44	64.19	59.71	60.98
QDA	65.00	61.62	59.19	59.37	57.40
SVM	**68.64**	71.95	**64.24**	60.18	**66.49**
Mean	65.16	69.07	61.85	57.94	61.30

In this section, we present the average accuracy results for different classifiers and models on a balanced dataset. As shown in Table III, traditional baseline methods with classifiers achieve accuracy ranging from 53.40% to 72.68%, with the best result obtained using the LogPower feature with Gaussian Naive Bayes (GNB). Band power-based methods (RelativePower, LogPower, PowerRatio) generally outperform entropy-based methods in classification accuracy.

TABLE IV
COMPARISON OF THE MEAN CROSS-SUBJECT ACCURACIES (%) ON THE BALANCED DATASET BETWEEN THE PROPOSED MODEL AND FOR BASELINE METHODS.

ID	EEGNet-4,2	Conv-ShallowNet	ICNN	Ours
1	85.43	82.54	85.64	**86.70**
2	21.34	80.73	69.70	**84.85**
3	58.19	65.77	78.67	**85.33**
4	**89.74**	70.64	78.38	79.73
5	44.24	82.97	87.95	**91.96**
6	48.76	78.64	**84.34**	78.31
7	**88.85**	72.53	66.67	64.71
8	76.43	70.06	**78.79**	76.14
9	71.94	**90.24**	89.18	88.22
10	**90.21**	82.78	73.15	85.19
11	**80.17**	65.10	65.93	66.37
Ave.	68.66	76.54	78.03	**80.68**

Table IV displays the accuracy of the deep learning methods. The average accuracy of the proposed model is 80.68%, surpassing that of EEGNet-4.2 (68.66%), Conv-ShallowNet (76.54%), and ICNN (78.03%). Individual subject accuracy varies, with the highest accuracy reaching 91.96% for subject 5. Some subjects, however, perform better with EEGNet-4.2, possibly due to the limited coverage of three frontal electrodes.

The average inference time for the proposed algorithm and baseline methods for each sample is shown in Table V. Although our model has a slightly longer inference time (4.86×10^{-6} seconds) compared to EEGNet-4.2 and other baselines, it offers improved accuracy and robustness, making it suitable for high-precision fatigue monitoring applications.

E. Individual Comparison Results on the Unbalanced Dataset

In this section, we compare the proposed model's performance with the best baseline methods on an unbalanced dataset using leave-one-out cross-validation, where one subject's unbalanced EEG data serves as the test set, and the balanced data from all other subjects are used for training. This ensures an unbiased performance assessment.

We evaluate accuracy, precision, and recall. Precision measures the correctly classified drowsy samples out of all samples classified as drowsy, while recall measures the correctly classified drowsy samples out of the actual drowsy samples. Low precision or recall indicates misclassifications between awake and drowsy states.

For comparison, we selected three baseline deep learning models: EEGNet4.2, Conv-ShallowNet, and ICNN, along with one traditional baseline model: RelativePower+SVM. The deep learning models were trained for 11 epochs after convergence.

As shown in Table VI, Our model achieved the highest average accuracy (76.62%), precision (73.85%), and recall (76.03%) among all methods, showing superior performance across challenging subjects and proving more reliable for EEG-based drowsiness detection by effectively minimizing false positives.

TABLE V
AVERAGE INFERENCE TIME(S) PER SAMPLE

Method	EEGNet-4,2	Conv-ShallowNet	ICNN	Ours
Average Inference Time(s)	2.43×10^{-6}	3.35×10^{-6}	3.04×10^{-6}	4.86×10^{-6}

TABLE VI
COMPARISON OF THE MEAN CROSS-SUBJECT ACCURACIES (%) ON THE UNBALANCED DATASET BETWEEN THE PROPOSED MODEL AND FOR BASELINE METHODS. THE PRECISION, RECALL, AND ACCURACIES OBTAINED FOR EACH SUBJECT ARE SHOWN IN THE TABLE.

ID	EEGNet-4.2			Conv-ShallowNet			RelativePower+SVM			ICNN			Ours		
	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.
1	87.36	80.85	**84.57**	82.21	78.13	78.63	58.55	92.71	63.16	77.51	74.39	74.21	82.0	80.13	80.0
2	1.87	100.0	20.45	32.21	31.82	76.89	15.52	78.79	30.77	52.35	54.13	62.23	71.45	86.23	**80.89**
3	0.0	0.0	58.00	80.33	62.78	61.49	71.83	56.67	53.73	65.33	65.44	54.50	67.97	69.67	**72.55**
4	85.59	90.77	**89.19**	53.50	93.24	66.83	44.87	47.30	57.29	69.18	67.70	70.83	77.20	75.90	78.12
5	96.0	16.11	43.75	72.61	84.82	79.15	82.67	55.36	76.92	91.76	88.35	**90.11**	89.83	86.98	88.64
6	34.92	91.67	49.19	94.68	64.66	74.89	81.62	95.69	**84.92**	78.94	76.38	72.86	71.85	72.26	70.85
7	100.0	81.03	**89.22**	76.20	78.64	68.73	78.26	87.38	75.32	67.16	69.31	66.23	66.92	68.37	62.99
8	94.44	45.54	75.76	51.05	90.91	66.27	57.53	81.06	71.89	75.11	73.88	**77.03**	69.82	71.02	71.62
9	93.06	62.33	71.02	85.03	87.90	**87.0**	81.25	74.52	83.25	87.22	83.09	85.75	84.90	81.85	84.25
10	26.67	100.0	89.81	100.0	22.22	78.93	85.37	64.81	89.84	91.76	73.81	88.21	87.43	81.51	**89.84**
11	69.75	90.22	**80.09**	66.18	61.07	61.16	60.80	92.37	63.93	69.90	67.41	65.98	62.93	62.37	63.11
Ave.	62.69	68.96	68.28	72.18	68.74	72.72	65.30	75.15	68.28	75.11	72.17	73.45	75.66	76.03	**76.62**

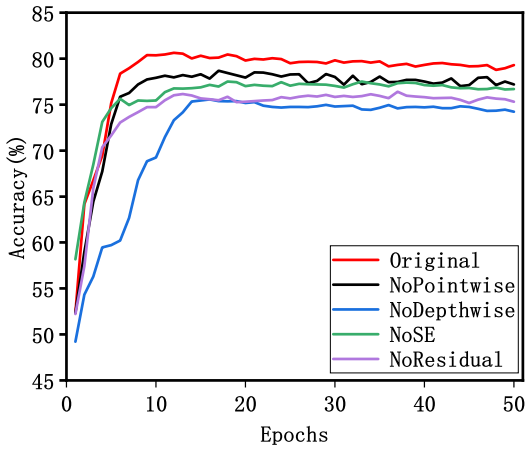


Fig. 2. Comparison of the mean cross-subject classification accuracies (%) between the proposed model and its variations, against training epochs from 1 to 50. The standard errors and accuracies averaged over 10 iterations for 11 subjects of each model are shown.

F. Ablation Study

We evaluated the model and its variants, as shown in Fig. 2. The full model, with all modules included, achieved the highest accuracy at nearly 80%. Removing the pointwise convolution (NoPointwise) caused a slight accuracy drop, indicating its supportive role in model performance. The exclusion of depthwise convolution (NoDepthwise) led to a more significant accuracy decrease, emphasizing its essential role in feature extraction. Omitting the squeeze-and-excitation (SE) blocks (NoSE) and residual connections (NoResidual) reduced accuracy to 78% and 77%, respectively, showing that both SE blocks and residual connections enhance recognition and stability. Overall, depthwise convolution and SE blocks contribute most substantially, with pointwise convolution and residual connections aiding stability, showcasing the compact and effective design of the model.

V. Conclusion

In this paper, we presented a novel model for cross-subject drowsiness recognition using forehead EEG signals. Our approach employs a compact convolutional neural network structure and separable convolutions to effectively process the spatial-temporal sequences of EEG data. Experimental validation on a public dataset demonstrated notable performance improvements, with our model achieving an average accuracy of 80.68% during leave-one-out cross-validation. This performance surpasses traditional baseline methods and state-of-the-art deep learning approaches, highlighting the practical feasibility of our method, which does not require individual subject calibration. The compact architecture and efficient processing of forehead EEG signals through separable convolutions facilitate lightweight and real-time applications, enhancing both practicality and portability.Despite these advancements, there are still opportunities for further enhancement. Future research may focus on optimizing the network architecture, exploring alternative neural network models, or extending investigations to other bio-signals or multimodal fusion approaches. Our study provides an effective method for cross-subject fatigue recognition using forehead EEG signals, carrying significant implications for practical applications. We hope our findings inspire new insights and directions in related fields.

VI. Acknowledgments

This work is supported by the Technology Project of China Southern Power Grid under Grant 030900KC23070004 (GDKJXM20230850).

References

[1] Ruilin Li, Zirui Lan, Jian Cui, Olga Sourina, and Lipo Wang. Eeg-based recognition of driver state related to situation awareness using graph convolutional networks. In *2020 International Conference on Cyberworlds (CW)*, pages 180–187. IEEE, 2020.

[2] Yisi Liu, Zirui Lan, Jian Cui, Olga Sourina, and Wolfgang Müller-Wittig. Inter-subject transfer learning for eeg-based mental fatigue recognition. *Advanced Engineering Informatics*, 46:101157, 2020.

[3] Yisi Liu, Zirui Lan, Jian Cui, Olga Sourina, and Wolfgang Müller-Wittig. Eeg-based cross-subject mental fatigue recognition. In *2019 international conference on cyberworlds (cw)*, pages 247–252. IEEE, 2019.

[4] Jianfeng Hu. Automated detection of driver fatigue based on adaboost classifier with eeg signals. *Frontiers in computational neuroscience*, 11:72, 2017.

[5] Haowen Luo, Taorong Qiu, Chao Liu, and Peifan Huang. Research on fatigue driving detection using forehead eeg based on adaptive multi-scale entropy. *Biomedical Signal Processing and Control*, 51:50–58, 2019.

[6] Srikanth Sagar Bangaru, Chao Wang, and Fereydoun Aghazadeh. Automated and continuous fatigue monitoring in construction workers using forearm emg and imu wearable sensors and recurrent neural network. *Sensors*, 22(24):9729, 2022.

[7] M Sprajcer, A Robinson, MJW Thomas, and D Dawson. Advancing fatigue management in healthcare: risk-based approaches that enhance health service delivery. *Occupational Medicine*, 73(8):459–463, 2023.

[8] Neusa R Adão Martins, Simon Annaheim, Christina M Spengler, and René M Rossi. Fatigue monitoring through wearables: A state-of-the-art review. *Frontiers in physiology*, 12:790292, 2021.

[9] German Torres, Michael P Cinelli, Alexander T Hynes, Ian S Kaplan, and Joerg R Leheste. Electroencephalogram mapping of brain states. *Journal of Neuroscience and Neuroengineering*, 3(2):73–77, 2014.

[10] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

[11] Prabhavathi C Nissimagoudar, Anilkumar V Nandi, and HM Gireesha. Deep convolution neural network-based feature learning model for eeg based driver alert/drowsy state detection. In *Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019) 11*, pages 287–296. Springer, 2021.

[12] Zhongke Gao, Xinmin Wang, Yuxuan Yang, Chaoxu Mu, Qing Cai, Weidong Dang, and Siyang Zuo. Eeg-based spatio–temporal convolutional neural network for driver fatigue evaluation. *IEEE transactions on neural networks and learning systems*, 30(9):2755–2763, 2019.

[13] Hong Zeng, Chen Yang, Guojun Dai, Feiwei Qin, Jianhai Zhang, and Wanzeng Kong. Eeg classification of driver mental states by deep learning. *Cognitive neurodynamics*, 12:597–606, 2018.

[14] Jian Cui, Zirui Lan, Yisi Liu, Ruilin Li, Fan Li, Olga Sourina, and Wolfgang Müller-Wittig. A compact and interpretable convolutional neural network for cross-subject driver drowsiness detection from single-channel eeg. *Methods*, 202:173–184, 2022.

[15] Jian Cui, Zirui Lan, Olga Sourina, and Wolfgang Müller-Wittig. Eeg-based cross-subject driver drowsiness recognition with an interpretable convolutional neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[17] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[18] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[20] Zehong Cao, Chun-Hsiang Chuang, Jung-Kai King, and Chin-Teng Lin. Multi-channel eeg recordings during a sustained-attention driving task. *Scientific data*, 6(1):19, 2019.

[21] Jian Cui, Zirui Lan, Tianhu Zheng, Yisi Liu, Olga Sourina, Lipo Wang, and Wolfgang Müller-Wittig. Subject-independent drowsiness recognition from single-channel eeg with an interpretable cnn-lstm model. In *2021 international conference on cyberworlds (CW)*, pages 201–208. IEEE, 2021.

[22] J. Cui. Eeg driver drowsiness dataset. Available from: https://figshare.com/articles/dataset/EEG_driver_drowsiness_dataset/14273687. 2019.

[23] J. Cui. Eeg driver drowsiness dataset (unbalanced). Available from: https://figshare.com/articles/dataset/EEG_driver_drowsiness_dataset_unbalanced_/16586957. 2021.

[24] Davide Borra, Silvia Fantozzi, and Elisa Magosso. Interpretable and lightweight convolutional neural network for eeg decoding: Application to movement execution and imagination. *Neural Networks*, 129:55–74, 2020.

[25] Nikhil R Pal, Chien-Yao Chuang, Li-Wei Ko, Chih-Feng Chao, Tzyy-Ping Jung, Sheng-Fu Liang, and Chin-Teng Lin. Eeg-based subject-and session-independent drowsiness detection: an unsupervised approach. *EURASIP Journal on Advances in Signal Processing*, 2008:1–11, 2008.

[26] Budi Thomas Jap, Sara Lal, Peter Fischer, and Evangelos Bekiaris. Using eeg spectral components to assess algorithms for detecting fatigue. *Expert Systems with Applications*, 36(2):2352–2359, 2009.

[27] Qingjun Wang, Yibo Li, and Xueping Liu. Analysis of feature fatigue eeg signals based on wavelet entropy. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(08):1854023, 2018.

[28] Jianfeng Hu and Jianliang Min. Automated detection of driver fatigue based on eeg signals using gradient boosting decision tree model. *Cognitive neurodynamics*, 12:431–440, 2018.

[29] Evgin Goceri. Capsnet topology to classify tumours from brain images and comparative evaluation. *IET Image Processing*, 14(5):882–889, 2020.

[30] Evgin Goceri. Diagnosis of alzheimer's disease with sobolev gradient-based optimization and 3d convolutional neural network. *International journal for numerical methods in biomedical engineering*, 35(7):e3225, 2019.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Scipy.org. Available from: https://scipy.org/.

[33] scikit-learn machine learning in python. Available from: https://scikit-learn.org/stable/.

Compressed Vision Transformer for Scene Text Recognition

1st Jinbiao Ren
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
37697210@qq.com

2nd Tao Deng
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
122202972@qq.com

3rd Yanlin Huang
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
634529848@qq.com

4th Da Qu*
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
1243306704@qq.com

5th Jianqiu Su
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
53430419@qq.com

6th Bingen Li
Maoming Power Supply Bureau
Guangdong Power Grid Co.,LTD
Maoming, China
1248015370@qq.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899477

***Abstract*—With the advancement of scene text recognition and deep learning, an increasing number of models have been proposed and applied to scene text recognition tasks. However, deploying these powerful yet computationally intensive models on resource-constrained devices is challenging. Model pruning is one of the most effective methods for compressing and accelerating these models, as it reduces the number of parameters and computational load by removing less critical parameters or structures. In ViT models, each parameter influences its neighboring parameters locally. Therefore, rather than pruning solely based on parameter magnitude, we propose selecting parameters for removal based on their local influence. By calculating the combined impact of each parameter along with its neighbors, we identify and prune those with minimal overall influence on the model, achieving compression and acceleration without significantly compromising accuracy. Our pruning method substantially reduces parameter count and computational cost while preserving accuracy, as demonstrated across seven test datasets and in comparison with more than five similar STR algorithms.**

***Index Terms*—ViT, Scene Text Recongnition, Pruning, Compression**

I. Introduction

Text serves as a vital medium for conveying, recording, and transmitting information about human experiences, making it indispensable across various fields [1]–[5]. In recent years, advancements in artificial intelligence, image search, multimodal learning, and embodied intelligence have underscored the growing importance of accurately recognizing text [6]–[8].

Despite the maturity of the OCR field, recent advancements in deep learning techniques have enabled mainstream OCR frameworks to effectively recognize most types of text, including improved recognition of certain ancient scripts [9]. However, not all text is printed or handwritten on paper. In real-world applications, visual models must be capable of recognizing text in various forms, including occluded, distorted, mosaicked, rotated, and even weather-affected conditions. For instance, while OCR can readily recognize the word 'shakeshack' when printed on paper, accurately identifying is challenging in scenario. Consequently, scene text recognition presents a newer and more challenging topic. Unlike OCR applied to scanned documents, text in natural scenes can appear anywhere, often against complex backgrounds, which introduces additional recognition challenges.

Early research in scene text recognition (STR) heavily relied on hand-designed features [10]–[12], which often struggled to encompass a broad range of patterns, resulting in subpar performance. The advent of deep learning has significantly enhanced STR capabilities through neural networks. However, this improvement has come with increasingly demanding hardware requirements, particularly the need for high-performance GPU devices. Many scene text recognition scenarios necessitate compression and acceleration to meet real-time and computational requirements.

Therefore, we propose a pruning method for the Transformer family in STR. By selecting the less informative parameters in the Transformer model for pruning, the model can use fewer parameters to achieve almost unaffected accuracy. At the same time, we test the effect of pruning on the accuracy of different modules in the model, and the degree of its recognition of different categories in the dataset.

Our pruning method enables efficient scene text recognition on Transformers, significantly reducing computation and parameter counts, especially in real-time demanding scenarios, adding considerable value.

II. Methdology

A. ViT Models in Scene Text Recognition

The ViT model architecture, introduced by Dosovitskiy [13], is an image classification model built on the Transformer framework. It processes images by partitioning them into

* Corresponding author

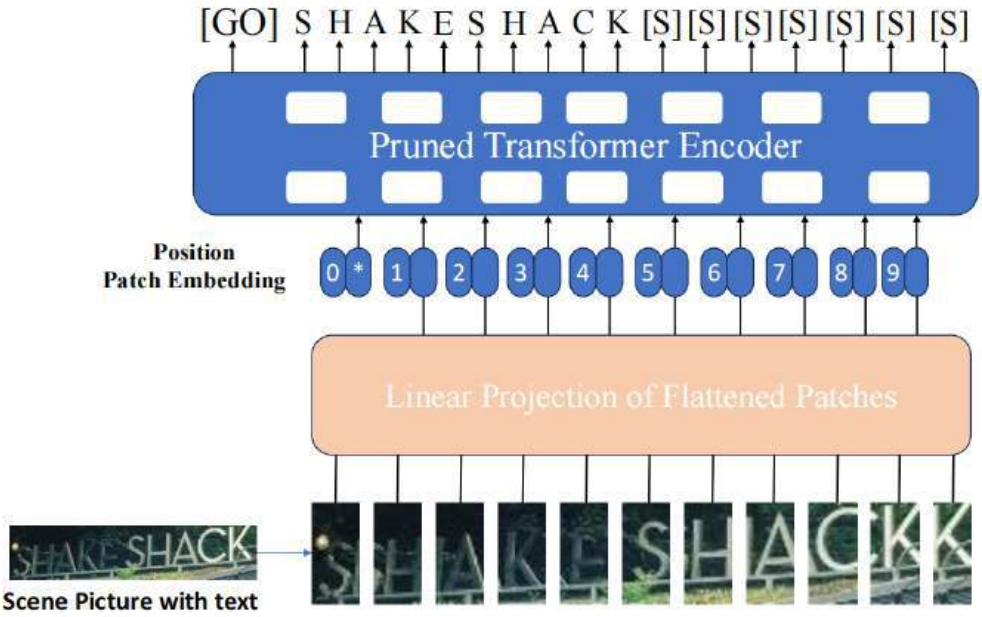


Fig. 1. Network architecture of ViTSTR with a pruned encoder. Firstly convert an image to patches, then making it 1D vector embeddings as the input of pruned encoder. [GO] is the start symbol while [s] is the space symbol and end symbol.

fixed-size patches, which are then flattened and transformed into high-dimensional vectors. These patches are input to the model in a sequence, enabling ViT to leverage the Transformer's self-attention mechanism.

In ViT [13], positional encoding is added to each block's embedding to retain their positional information within the original image, ensuring the model captures spatial relationships. The self-attention mechanism, central to ViT, enables the model to focus on the relationships between different patches in the input sequence, dynamically adjusting the emphasis on each block. ViT comprises multiple Transformer encoder layers, each featuring multi-head self-attention and feed forward neural networks. These layers allow the model to progressively extract more complex features and generate a high-level representation of the image. At the end, ViT processes all the image patches through multiple layers of encoding and adds a special classification token to the representations. This classification token is then passed through a fully connected layer for the final image classification. When applying ViT to scene text recognition (STR), the prediction header is modified [14]. Unlike the traditional ViT, which employs separate target classification, the ViT model for STR must accurately distinguish the length and order of multiple characters to recognize text in images. We refer to ViTSTR [14], proposed by Rowel [14], which utilizes a [GO] token as the initial token. Instead of extracting a single output vector at a time, ViTSTR extracts multiple feature vectors from the encoder. The number of vectors corresponds to the maximum length of the text in the dataset, plus the [GO] and [s] tokens. The [GO] token signifies the start of text prediction, while the [s] token indicates the endpoint or space.

B. Compress ViTSTR Models with Pruning

Model pruning is a technique for achieving model compression and acceleration by removing unnecessary or low-impact parameters. In Convolutional Neural Networks (CNNs) and Transformer models, pruning typically involves identifying and eliminating connections, entire channels, or attention heads with low weights, thereby reducing computational complexity and memory footprint. While magnitude pruning relies on predefined thresholds or sensitivity analysis, structured pruning aims to maintain the overall architectural integrity of the model. This approach not only significantly enhances inference speed but also optimizes model performance in resource-constrained environments while ensuring accuracy. By judiciously applying pruning strategies, the efficiency of deep learning models in practical applications can be substantially improved.

Magnitude pruning [15] assumes that the smaller the absolute value of the weight, the less impact it has on the model, and therefore can be preferentially removed. Magnitude pruning $\mathrm{T(x)} = x \cdot \mathbf{1}_{-t \le x \le t}$ will set weights less than threshold t to zero. The expected error of pruning can be expressed as follows [16]:

$$\mathbb{E}\left[(\mathrm{T(w)} - 0)^2\right] = \int_{-t}^{t} w^2 \mathrm{g(w)} dw, \tag{1}$$

where $\mathrm{g(w)}$ is the distribution of parameters, w is an arbitrary weight in the model. If the pruning rate p is fixed, the threshold t can be calculated from the pruning rate p and the total number of weights N. First, all weights $\mathbf{W}$ are sorted in ascending order, and then the $\lceil Np \rfloor$-th parameter is selected as the pruning threshold t. While weighted pruning assesses the importance of a parameter solely based on the

TABLE I
MODEL ACCURACY(%) IN DIFFERENT DATASETS.

Dataset	IIIT	SVT	IC03		IC13		IC15		SVTP	CT
Sample number	3000	647	860	867	857	1015	1811	2077	645	288
CRNN [17]	81.8	80.1	91.7	91.5	89.4	88.4	65.3	60.4	65.9	61.5
R2AM [18]	83.1	80.9	91.6	91.2	90.1	88.1	68.5	63.3	70.4	64.6
GCRNN [19]	82.9	81.1	92.7	92.3	90.0	88.4	68.1	62.9	68.5	65.5
Rosetta [20]	82.5	82.8	92.6	91.8	90.3	88.7	68.1	62.9	70.3	65.5
RARE [21]	86.0	85.4	93.5	93.4	92.3	91.0	73.9	68.3	75.4	71.0
STAR-Net [22]	85.2	84.7	93.4	93.0	91.2	90.5	74.5	68.7	74.7	69.2
SAR [23]	90.2	84.2	-	-	91.0	-	69.2	-	76.4	81.5
Real [24]	89.8	84.3	-	-	90.9	90.6	73.1	-	74.6	82.3
ICRNN [25]	83.9	83.0	-	-	89.7	-	71.9	-	69.6	60.4
ViTSTR-Tiny [14]	83.7	83.2	92.8	92.5	90.8	89.3	72.0	66.4	74.5	65.0
ViTSTR-Tiny-10%	83.4	**83.9**	**93.5**	**93.3**	**91.4**	**90.4**	**73.7**	**68.1**	**76.6**	68.4
ViTSTR-Tiny-20%	85.0	**84.7**	**93.3**	**92.9**	**91.7**	**90.3**	**73.0**	**67.5**	**74.0**	68.4
ViTSTR-Tiny-30%	84.4	**84.5**	**93.0**	**92.7**	**90.7**	**89.6**	**72.3**	**66.7**	**74.4**	67.0
ViTSTR-Small [14]	85.2	85.1	93.3	93.2	91.3	90.6	75.3	69.5	78.1	71.3
ViTSTR-Small-10%	85.2	**85.3**	**93.8**	**93.8**	**91.0**	**89.9**	**74.6**	**68.9**	**80.2**	69.4
ViTSTR-Small-20%	85.4	**83.9**	**94.0**	**93.1**	**90.3**	**89.8**	**73.6**	**67.6**	**76.9**	68.4

magnitude of its current value to determine which parameters can be removed or retained, the interconnectivity of image information in Scene Text Recognition (STR) complicates this approach. In this context, each parameter and its neighboring parameters influence one another, meaning that modifying a single parameter can impact the overall model. Therefore, in addition to evaluating a parameter's importance based on its absolute value, we also consider its neighboring parameters to calculate an average importance score. Specifically, we use the average of the absolute values of each parameter and its neighboring parameters as an indicator of the parameter's influence, which guides the decision on whether the parameter should be removed. This approach allows us to measure the significance of a parameter in conjunction with its neighbors, guiding our decision on whether it should be retained or removed.

In the ViTSTR model, parameters are primarily concentrated in the embedding layer, attention layer, feed-forward neural network, and residual connections. The attention layer has the greatest impact on model performance. To achieve a lightweight ViTSTR model, we focus on pruning parameters with smaller influence in the attention layer, embedding layer, and feed-forward neural network, removing them according to the required pruning percentage. This approach minimizes the impact on accuracy while reducing model parameters and computational load.

Since the ViTSTR model is trained from scratch on multiple datasets, pruning can impact its performance and requires further fine-tuning. To preserve the model's accuracy, we prune parameters with influence smaller than a predefined threshold before each training round, removing them from the model. This allows the pruned model to recover its accuracy through iterative training, ensuring minimal loss of performance.

The pruned model will have fewer parameters and reduced floating-point computations. We tested the ViT family of models at various pruning percentages. Due to the differing parameter distributions across layers, the thresholds calculated for each layer vary, ensuring stable pruning percentages. If the threshold is determined based on the overall model's parameter distribution, increasing the pruning percentage could result in excessively high pruning rates in certain layers, leading to overly sparse parameters and potential instability in the pruned model. For each layer of ViT, we first calculate the threshold based on the designated pruning percentage p.

$$t_l = \text{Quantile}(|W_l|, p), \tag{2}$$

where the W_l is the L-layer parameters, the quantile function represents the t_l that satisfies $\text{P}(t_l \leq |W_l|) = p$ for $0 < p < 1$.

Based on the calculated thresholds, we then choose to prune the parameters that local influences are smaller than the thresholds.

$$W_l' = W_l \cdot \mathbb{I}(I_l \geq t_l), \tag{3}$$

where I_l represents the local influence of each parameter of the layer. The local influence of each parameter is calculated by averaging the absolute values of that parameter with the parameters before and after it.

In model compression and acceleration, we commonly use Floating-point computation (FLOPs) to measure the computational requirements and efficiency of a model. FLOPs represent the number of floating-point operations needed for single-batch inference. A high FLOPs value indicates that the model requires substantial computational resources, leading to increased hardware demands and reduced real-time capability. Conversely, a low FLOPs value suggests that the model can execute tasks more quickly, with reduced computational load

and improved real-time performance. It is often necessary to strike a balance between FLOPs and model accuracy, aiming to minimize FLOPs while maintaining acceptable accuracy levels.

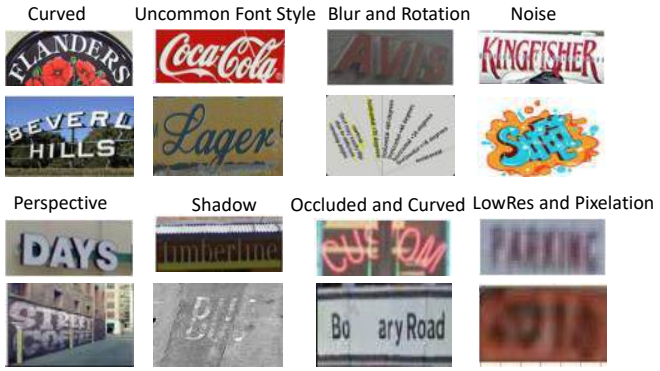


Fig. 2. Text in different scenes, including various distortions and variations.

III. Experiment

TABLE II
Comparison of different models in terms of accuracy, parameters number and FLOPs.

Model	Accuracy (%)	Parameters (1×10^6)	FLOPS (1×10^9)
CRNN [17]	76.7	8.5	1.4
R2AM [18]	78.4	2.9	2.0
GRCNN [19]	78.3	11.2	1.8
Rosetta [20]	78.4	5.3	10.1
RARE [21]	82.1	10.8	2.0
STAR-Net [22]	81.8	48.9	10.7
ViTSTR-Tiny [14]	80.3	5.4	1.3
ViTSTR-Tiny-20%	**81.2**	**4.3**	**1.0**
ViTSTR-Small [14]	82.4	21.5	4.6
ViTSTR-Small-20%	**81.5**	**17.2**	**3.7**

In this section, we experimentally validate the ViTSTR pruning method by selecting various scene text recognition datasets and using models of Tiny and Small sizes. We aim to recover the accuracy lost due to pruning through additional training, all while avoiding an increase in computational effort.

A. Datasets

We use the MJSynth(MJ) [26] and SynthText(ST) [27] datasets for training. The MJSynth dataset contains approximately 8.9 million synthesized scene text images, while SynthText includes around 5.5 million images from various scenes.

We adopt seven datasets for testings:

1) **IIIT5K** [28]: Contains approximately 5,000 images extracted from natural scenes, such as billboards, store signs, and road signs, where training set has 2,000 images and testing set has 3,000.
2) **SVT** [29]: Comprises about 647 images sourced from Google Street View maps, featuring a variety of fonts, sizes, and signposts.
3) **IC03** [30]: Includes 1,110 test images from ICDAR2003, with 867 images filtered to remove those containing fewer than three characters. However, the first version often have 860 images, 7 images have been missed. Both of them are often assessed separately.
4) **IC13** [31]: An extension of the IC03 dataset from ICDAR2013, with two versions containing 1,015 and 857 scene text images respectively.
5) **IC15** [32]: Derived from ICDAR2015 and captured by Google Glass, this dataset consists of images that are often blurry, noisy, rotated, or low-resolution, with two versions containing 1,811 and 2,077 images.
6) **SVTP** [33]: Comprises 645 images from Google Street View.
7) **CT** [34]: Contains 288 images featuring T-shirt and product logos.

With over 10,000 images, this comprehensive selection allows us to test the performance of the pruned model on real-world scene text, validating the effectiveness of our algorithm.

B. Setting

To compare the baseline results across different scene text recognition, we adopt the frameworks of Baek [35] and Rowel [14], ensuring a fair and consistent configuration for comparison. Parameter pruning is determined by evaluating the individual and local importance of each parameter. For training, we follow Rowel's settings, using a batch size of 192, 300 epochs, and the Adadelta optimizer with a learning rate of 1.0. All input images for pruned model are resized to 224x224. All experiments are conducted in NVIDIA RTX 3090.

C. Results

To demonstrate that the pruned model outperforms its counterparts in both accuracy and computational efficiency, we conducted extensive comparative experiments.

As shown in Table I, we achieve an accuracy of 81.2% in both the Tiny-ViTSTR models pruned by 10% and 20%, which is even higher than the accuracy of the full model. In the Small-ViTSTR model, pruning results in less than a 1% decrease in accuracy. Compared with other methods, pruned ViTSTR has almost achieved the highest level of results on SVT, IC03, IC13, IC15 and SVTP datasets. Especially on the 10% pruned small ViT model of SVTP datasets, which is more than 4% higher than other methods. The 10% pruned small ViT model on the IC15 dataset is more than 1.5 percent higher than the other methods.

As shown in Table II, the pruned ViTSTR model exhibits the smallest number of parameters and FLOPs for comparable accuracy, while also achieving the highest accuracy among models with similar parameter counts and FLOPs. In particular, the 10% tiny model is the only one with less than 5×10^6 parameters and less than or equal to 1×10^9 floating point calculation among all methods with the accuracy more than 80%. The accuracy here represents the average precision of the testing datasets.

IV. Conclusion

In this paper, we propose a pruning-based compression method for the ViTSTR model, aimed at reducing model size and accelerating computation by removing the least influential parameters. Compared with other methods, the proposed algorithm incurs low additional cost, achieves a significant reduction in parameter count, and maintains better accuracy retention.

V. Acknowledgments

This work is supported by the Technology Project of China Southern Power Grid under Grant 030900KC23070004 (GDKJXM20230850).

References

[1] G. Nagy, "Twenty years of document image analysis in pami," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 22, no. 01, pp. 38–62, 2000.

[2] H. Guo, J. Liu, H. Huang, J. Yang, Z. Li, D. Zhang, Z. Cui, and F. Wei, "Lvp-m3: language-aware visual prompt for multilingual multimodal machine translation," *arXiv preprint arXiv:2210.15461*, 2022.

[3] H. Guo, B. Wang, J. Bai, J. Liu, J. Yang, and Z. Li, "M2c: towards automatic multimodal manga complement," *arXiv preprint arXiv:2310.17130*, 2023.

[4] M. Li, B. Fu, H. Chen, J. He, and Y. Qiao, "Dual relation network for scene text recognition," *IEEE Transactions on Multimedia*, vol. 25, pp. 4094–4107, 2022.

[5] M. Talarmain, C. Boned, S. Biswas, and O. Ramos Terrades, "Recurrent few-shot model for document verification," in *International Conference on Document Analysis and Recognition*. Springer, 2024, pp. 51–62.

[6] Y. Zhou, S. Liu, Y. Zhang, Y. Wang, and W. Lin, "Perspective scene text recognition with feature compression and ranking," in *Computer Vision-ACCV 2014 Workshops: Singapore, Singapore, November 1-2, 2014, Revised Selected Papers, Part II 12*. Springer, 2015, pp. 181–195.

[7] T. Kawakatsu, "Multi-cell decoder and mutual learning for table structure and character recognition," in *International Conference on Document Analysis and Recognition*. Springer, 2024, pp. 389–405.

[8] X. Chen, B. Chen, C. Qu, D. Peng, C. Liu, and L. Jin, "Dtsm: Toward dense table structure recognition with text query encoder and adjacent feature aggregator," in *International Conference on Document Analysis and Recognition*. Springer, 2024, pp. 438–452.

[9] H. Guan, H. Yang, X. Wang, S. Han, Y. Liu, L. Jin, X. Bai, and Y. Liu, "Deciphering oracle bone language with diffusion models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 15 554–15 567.

[10] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, 2012, pp. 3304–3308.

[11] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *2012 IEEE conference on computer vision and pattern recognition*, 2012, pp. 3538–3545.

[12] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representation*, 2021, pp. 1–22.

[14] R. Atienza, "Vision transformer for fast and efficient scene text recognition," in *International Conference on Document Analysis and Recognition, ICDAR 2021*, vol. 12821. Springer, 2021, pp. 319–334.

[15] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *Proceedings of the International Conference on Learning Representation*, 2016, pp. 1–14.

[16] A. Kuzmin, M. Nagel, M. Van Baalen, A. Behboodi, and T. Blankevoort, "Pruning vs quantization: Which is better?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 414–62 427, 2023.

[17] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[18] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2231–2239.

[19] J. Wang and X. Hu, "Gated recurrent convolution neural network for ocr," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[20] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 71–79.

[21] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4168–4176.

[22] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, and J. Han, "Star-net: a spatial attention residue network for scene text recognition." in *BMVC*, vol. 2, 2016, p. 7.

[23] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8610–8617.

[24] J. Baek, Y. Matsui, and K. Aizawa, "What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3113–3122.

[25] W. Yu, M. Ibrayim, and A. Hamdulla, "Scene text recognition based on improved crnn," *Information*, vol. 14, no. 7, p. 369, 2023.

[26] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *arXiv preprint arXiv:1406.2227*, 2014.

[27] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2315–2324.

[28] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *BMVC-British machine vision conference*, 2012.

[29] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *International conference on computer vision*, 2011, pp. 1457–1464.

[30] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto *et al.*, "Icdar 2003 robust reading competitions: entries, results, and future directions," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 7, pp. 105–122, 2005.

[31] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *International conference on document analysis and recognition*, 2013, pp. 1484–1493.

[32] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *international conference on document analysis and recognition (ICDAR)*, 2015, pp. 1156–1160.

[33] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 569–576.

[34] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8027–8048, 2014.

[35] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4715–4723.

IDEA: Intelligent Diffusion Model for Edge Cache System in Content Delivery Network

1st Yicong Zhao*
China Mobile Communications Group Co.,Ltd.
Beijing, China
zhaoyicong@chinamobile.com
*Corresponding author

2nd Zhi Li
China Mobile Communications Group Co.,Ltd.
Beijing, China
lizhi@chinamobile.com

3rd Lingshan Kong
China Mobile Communications Group Co.,Ltd.
Beijing, China
konglingshan@chinamobile.com

4th Song Guo
China Mobile Communications Group Co.,Ltd.
Beijing, China
guosong@chinamobile.com

5th Longfei Jin
China Mobile Communications Group Co.,Ltd.
Beijing, China
jinlongfei@chinamobile.com

6th Zhao Yang
China Mobile Communications Group Co.,Ltd.
Beijing, China
yangzhao@chinamobile.com

7th Jing Yu
China Mobile Communications Group Co.,Ltd.
Beijing, China
yujing@chinamobile.com

8th Zichen Wang
China Mobile Communications Group Co.,Ltd.
Beijing, China
wangzichen@chinamobile.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899569

***Abstract*—Content Delivery Network is an overlay network, which distributes the content from the origin servers to distributed edge service nodes, enabling users to get the required content from a nearby location, therefore, improving the perceived user experience. In order to accelerate content transmission speed and reduce network latency and congestion, CDN load balancing technology was developed. This technology optimizes the allocation of incoming traffic across multiple servers, ensuring equitable workload distribution and mitigating risks of server overload or underutilization. Therefore it improves the responsiveness of the website, reduces communication overhead, and maximizes the efficiency of computational resources. However, the balancing methods proposed by academia use various algorithms or approaches, but these algorithms often fail to account for the trade-off between communication costs and latency especially when the service machine is down. In recent years, most load balance techniques in the industry have primarily used edge node storage resources while neglecting computational resource utilization. To reduce both communication costs in routing and the waiting times of end-users, it is essential to direct user requests to the most optimal servers available.**

This study presents the implementation and design of IDEA, an Intelligent Diffusion Model for Edge Cache System in CDN, which reduces the cost of communication link and the time required to fetch distributed contents by introducing Long-Short Term Memory Network combined with a new meta-heuristic algorithm. We then compare it to other algorithms available in this paper. Experimental results show that this algorithm exceeds the performance of current methods across the key metrics outlined above.

***Index Terms*—CDN, Edge Computing, Load Balance, LSTM, Optimization**

I. Introduction

CDN (Content Delivery Network) adds a new layer of network architecture to the existing Internet, enabling users to retrieve the necessary content from proximate locations. This provides high-performance, scalable, and cost-effective content distribution services to users[1, 2, 3]. The conventional approach involves deploying multiple edge nodes across different regions. When a user requests a resource, if the edge node does not have the cached resource, it will request the content from the content center. If the content center cannot fulfill the request, it will initiate a back-to-origin request to retrieve the files from the user's source server. In the case of a problem at the edge node or data center, the request will be redirected to an alternative healthy service node in proximity, ensuring the continuity of accelerated services. This approach enhances the efficiency of resource distribution in the corresponding regions[4, 5].

A prominent challenge in Content Delivery Networks is balancing load distribution against communication costs[6, 9]. Users often face restricted choices when it comes to being directed to the most suitable server. When a server reaches its capacity limits, users are automatically redirected to other available servers. However, when traffic congestion occurs in high-demand regions, it may become more efficient to route users to a more distant server [7], which increases communication costs. Researchers have proposed various algorithms to enhance both balancing load and the communication

efficiency[8]. This study evaluates these approaches in real-world settings, incorporating scenarios like server failures. We propose an algorithm to address the problem.

Despite progress in CDN load balancing technologies, several challenges persist [13, 14]. Approaches proposed by the academic community often overlook the trade-off between communication costs and latency [15]. The bee approach[7] offers a more efficient choice to the Weighted Metrics Combination [6], which was developed for application in Content Delivery Networks. It showed shorter waiting times between user clients and node servers when operating under specific conditions during testing. Algorithms such as the Joint Shortest Queue approach prioritize server congestion as the key decision-making parameter for redirection. In subsequent work, the Control-Law Load Balancing algorithm [10] improved performance upon the Joint Shortest Queue approach. Nevertheless, the Control-Law Load Balancing algorithm overlooks communication expenses and presupposes that replica sets are situated together.

In recent years, most load balance techniques in the industry have primarily used edge node storage resources, however, these techniques neglect computational resource utilization. Articles in industry show that, in CDN, it is evitable that communication cost against response time. Three fundamental algorithms address this trade-off, with each generating a distinct trade-off curve[6, 9]. The concept of multiple choices approach[11] represents a well-established paradigm for load balancing, frequently employed in practical implementations of algorithms. The primary advantages of this paradigm are its simplicity and its ability to yield significantly better results than a basic randomized method. The authors in [12] introduced this demo an extension, referred to as load sharing with next-neighbor. In this approach, each server is capable of receiving client requests and directing them to the most suitable server. The optimal server is selected from it‘s next neighbor and a random node server. The authors argued that the next-neighbor load-sharing approach results in improved response time. These approches usually based on predefined variables and heuristics, without using computing resources to gather current network conditions parameters for making decision in real-time.

In this work, we present an algorithm on the basis of LSTM and heuristics, which has high tolerance of real network conditions, inhance the performance by using the computing resources, and make better trade-off between communication cost and response time. Our modifications enable us to showcase the algorithm's versatility across various scenarios, highlighting its overall capabilities and providing a clearer understanding of its potential in real-world, high-traffic CDNs.

II. Method

The innovation of the method proposed in this paper lies in addressing two key issues within the CDN load balance scenario: the insufficiency utilization of the computing resource and the trade-off between communication costs and latency. By tackling these challenges, the method effectively enhances model performance and metrics.

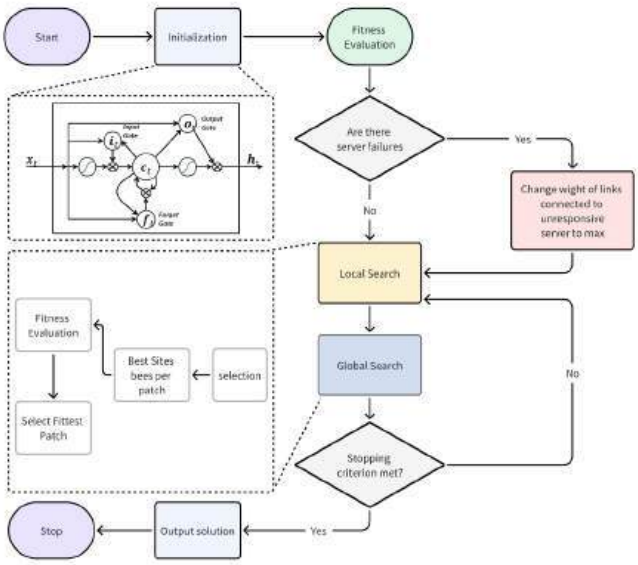


Fig. 1. IDEA, Intelligent diffusion model for edge cache system in CDN.

Classified as a metaheuristic method, we can try a population-based search technique which mimics the bees feeding behavior, and it provides shorter response times than the common WMC algorithm in the industry[7], therefore the optimal solution to an optimization problem can be determined by identifying the most suitable approach. Bees are distributed randomly throughout high-dimension space with equal probability and evaluate the suitability of the environment at their points of arrival. The core of the algorithm is a loop consisting of several steps. The subsequent step, referred to as the fitness assessment, requires bees to identify potential solutions and recruiting foragers to further explore these regions. These hunters are tasked with revisiting the optimal neighbors. Foragers from Team B search for the best solutions in Part A's neighborhoods, while other foragers explore the remaining promising solutions. In part of the local search, individuals assigned to forage inspect the floral areas that had been initially located by the guides who recruited them. If a forager finds a more optimal choice within a patch than the guide, the forager becomes the new scout for that patch. The most optimal solution at the conclusion of each round encountered thus is selected as the ouystanding outcome for the entire floral areas to precisely mirror the real-world scenario that will arise in a CDN. Comparing approaches under these situations may not yield results that accurately reflect their performance in real-world Content Delivery Network.

To solve the main issue: the test situation with server downtime which was not studied in school. We choose our real CDN server and data, to make a restoration that certain servers in specific regions are expected to undergo periods of downtime, which may result from either system failures or scheduled maintenance activities.

Our objective is to introduce IDEA and benchmark it against WMC approach and the standard Honeybee or called the LDEA_without_LSTM method, which is typically considered. The WMC approach will be configured in two ways: ideal, that is, yielding optimal results with real-time updates, which, however, leads to extended waiting times. Nevertheless, this algorithm is unsuitable for deployment in real-world networks, as it must make decisions within significantly shorter time-frames. We will also assess the Periodic_WMC approach with update interval, as this duration is considered the standard for providers.

The LSTM network [16] can prevent the mentioned problems and through training, the optimal parameters will be learned out, and used for parameter initialization of subsequent heuristic algorithms. The ability of memory units to manage long-range dependencies is the key factor. Specifically, an LSTM is composed of multiple control gates and a single memory unit. We set input vector, hidden state vector, and memory cell state denote the x_t, h_t, and c_t, respectively, at time t. The LSTM processes them to generate a corresponding sequence of hidden states and memory cells as its output when given the input vector. The procedure can be represented as a formula, as follows [17].

$$\begin{aligned} i_t &= \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \\ f_t &= \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ c_t &= f_t c_{t-1} + i_t \tanh\left(W_j x_t + U_j h_{t-1} + b_j\right) \\ o_t &= \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \\ o_t &= \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \\ h_t &= o_t \tanh\left(c_t\right) \end{aligned} \tag{1}$$

Here, the input gate, forget gate, and output gate correspond to i, f, and o, display, W and b represent the layer's parameters of network, respectively. the symbol σ denotes the sigmoid function. A lot of tasks can be addressed using the LSTM network.

We evaluate the IDEA and its benchmark versions using two metrics, for which the first is the communication expense for single content which CDN delivered.

$$\overline{Cost}(\boldsymbol{\varphi}) = \mathbb{E}\left[\frac{1}{\sum_{i=1}^{K} n_i} \sum_{i=1}^{K} \sum_{j=1}^{n_x} c_i d_{i,j}\right], \tag{2}$$

Here, $S_{i,j}$ indicates the server responsible for handling the jth request of client i, and n_i represents the number of requests made by the client i during the time period T.

For about the Server Latency of Response is the second metric:

$$\overline{Dealy}(\boldsymbol{\varphi}) = \mathbb{E}\left[\frac{1}{\sum_{i=1}^{K} n_i} \sum_{i=1}^{K} \sum_{j=1}^{n_i} \tau_{i,j}\right], \tag{3}$$

Let τ_i, j represent the time duration from when the j_{th} request is sent by user i to the moment the first packet of the response is received by the client.

The comparison of these two metrics is addressed in the following optimization problem:

$$\min_{\boldsymbol{\varphi}} \left[\alpha \overline{Cost}(\boldsymbol{\varphi}) + (1-\alpha)\overline{Delay}(\boldsymbol{\varphi})\right], \tag{4}$$

In this case, we vary α between 0 and 1, aiming to minimize two essential metrics of the server, the Communication Expense and the Response Latency. Then we proposed our IDEA, The steps are presented in Fig. 1

III. Experiment

A. Dataset and Experimental Settings

To validate the performance of IDEA in practical business scenarios, we constructed a dataset based on CDN real network business data over a three-month period. The dataset underwent cleaning and sampling, and sample labels were built for both positive and negative samples. The data was sorted chronologically, with the first 60 days used for training, two weeks for model validation, and the remaining two weeks for testing. In order to enable a comparison of the existing scheme and proposed approaches, the parameters are set as follows: the value of both the number of node servers and content files is set to 100, with each machine containing 10 distinct document cases representing the size of cached content. When it comes to each result presented, the final result is derived from a set of poisson distributed requests, furthermore, a collection of real-world distributed files is included in the CDN. The quantity of simulation epoches is 1000, depending on the required precision of the results. Table I shows the statistics of the dataset. The training work of LSTM can be modeled as a classification task, evaluated using machine learning metrics, primarily ACC (Accuracy) and AUC (Area Under the ROC Curve), to represent the model's predictive accuracy. Both metrics span from 0 to 1, with values approaching 1 signifying better performance, and values approaching to 0 signifying lower performance. Define TP as the quantity of resources predicted correctly, FP the quantity of resources predicted correctly, TN the quantity of false labeled resources predicted correctly, and FN the quantity of false labeled resources predicted incorrectly. So the accuracy metric (ACC), the True Positive Rate (TPR), and the False Positive Rate (FPR) is calculated using the following formula:

TABLE I
STATISTIC DATA OF THE DATASET AND EXPERIMENT.

# Servers	100	# Clients	1,000
# Files	1,000,000	# Requests	10,000
# Cache Size	10 M	# Sites	25
# Bandwidth	2 MB/s	# Propagation Delay	0.1 ms
# Request Size	0.5 KB	# Epoches	1000

$$\begin{aligned} \text{ACC} &= \frac{TP+TN}{TP+TN+FP+FN} \\ \text{TPR} &= \frac{TP}{TP+FN} \\ \text{FPR} &= \frac{FP}{FP+TN} \end{aligned} \tag{5}$$

For our experiments, hyperparameter tuning is performed on the validation set. We employ a batch size of 64 and choose the Adam optimizer. Each trial is repeated five folds, and the mean scores, as mentioned earlier, are reported.

For the heuristic approach, it identifies a set of servers containing the desired file and then selects one at random without performing additional verification. This approach is effective in an ideal scenario where all servers are functioning properly; however, it may lead to the selection of a non-operational server. To address potential server failures, we make a minor adjustment to the redirection algorithm by adding a step to test the server connection prior to finalizing its use. If the node server is found to be faulty, the closest available alternative server is chosen based on its proximity to that user client.

To achieve our testing objectives, we conduct simulations using parameters that closely mimic a practical CDN. Most of the values differ from those presented in the baseline approach paper. This approach allows us to represent practical and attainable data points derived from pictures.

B. Performance Evaluation

In this section, we compare the performance of the algorithms. Our primary focus is on comparing the IDEA algorithm introduced in this paper with proposed methods. This study illustrates that our proposed method outperforms alternative approaches in optimizing the balance between transmission expenses (which is also called communication cost) and mean latency (which is also called response time).

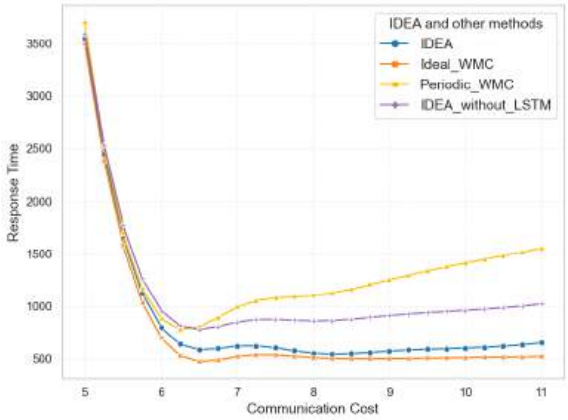


Fig. 2. The trade-off between response time and communication costs varies across different methodologies

We compare the approaches mentioned above in Fig. 2, which include IDEA, Ideal_WMC, IDEA_without_LSTM and Periodic_WMC, and the results demonstrate that our proposed method yields the curve most closely approximating the ideal scheme, while effectively managing the trade-off in comparison to the other approaches.

Except the optimal solution(Ideal_WMC) which is impossible to implement in the real world, and a baseline solution(Periodic_WMC). The plot above compares the performance of the algorithms under standard situations while different types of server malfunctions take place, by averaging the performance in three situations - 0,1,20 server failure in the whole network.

Under these conditions, the Periodic_WMC performs marginally worse than the other times, meanwhile, the mean communication cost of the IDEA_without_LSTM methods is slightly higher. After intropducing LSTM, the mean communication expense and waiting latency are reduced for our IDEA method in comparison to the IDEA_without_LSTM approach. Therefore, the IDEA outperforms the original in the real world.

In the real world, the communication expense per unit of data is lower on average with the innovation introduced in this work, but the waiting latency performs worse than the ideal condition. This phenomenon demonstrates that the additional overhead associated with verifying the server's connection status, along with the incorporation of the neural network, leads to an enhance in the mean response latency each time. However, this increase is justifiable when considering the communication expense and the waiting latency involved.

C. Hyperparameter Analysis

This section, we primarily analyzes the impact of a crucial hyperparameter on the experiments, which is essential for evaluating the balance between communication expense and load balancing. By comparing the schemes at their optimal performance, we can determine the optimal value of these hyperparameters.

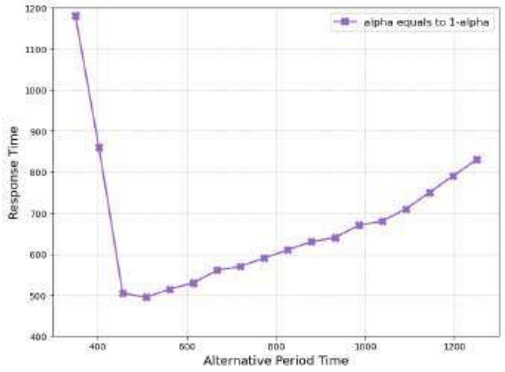


Fig. 3. Relationship between update step and optimum value.

We first discuss the update step of the periodic algorithm, we can see the Fig. 3. The performance results, presented in Fig. 4, demonstrate how adjusting this hyperparameter influences the mean response latency. The optimal step update value is also found to be 500 ms. The curve demonstrates worse when the hyper-parameter become lower or higher. The reason responding to the two phenomenon maybe because the sources from other servers is updated more often, or the network becomes fully congested, with the negative effects of congestion outweighing the benefits of more frequent updates.

It is also valuable to illustrate how changing the cache size affects the balance in the proposed scheme. In Fig. 4, the cache size ranges from 20M to 80M, and its impact on the give-and-take curve is depicted. The curve shown in Fig.2 represents

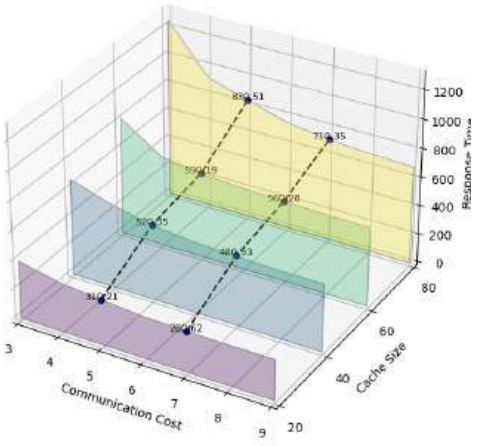


Fig. 4. Relationship between the response time and communication cost for different cache sizes.

the scenario with a cache capacity of 10, whereas the curve in Fig. 4 illustrates the trade-off when the cache capacity is varied. As expected, increasing the cache capacity assigned to each server results in lower communication costs and response times. However, the overall pattern of the trade-off remains unchanged.

IV. Conclusion

To further enhance customer experience and decrease the time delay at CDN, we introduce artificial intelligence into CDN load balance technology. We design a efficient algorithm combined with LSTM and heuristic algorithm to strike an optimal give-and-take between communication overhead and delay times, so that we subsequently evaluate its performance against several other approaches discussed in the existing papers. Our approach achieves rapid computational performance, enabling seamless deployment and efficient utilization of online computing resources.

In this paper, we proposed IDEA, a kind of Intelligent Diffusion Model for Edge Cache System in Content Delivery Network technique, which has been validated on real-world datasets. Compared to other techniques, our method demonstrates effectiveness at the business level. Specifically, PIPM achieves performance improvements significantly in terms of time latency, communication cost, and tested on different conditions such as cache size and alpha value, respectively, compared to the baseline strategy. This demonstrates cost-efficiency, market demand, and extensive applicability in current network environments.

References

[1] Li Z, Zhao Y, Kong L, et al. PIPM: Practical Intelligent Prefetch Model for Content Delivery Network[C]//Proceedings of the 2024 4th International Conference on Artificial Intelligence, Automation and Algorithms. 2024: 145-149.

[2] Abolhassani B, Tadrous J, Eryilmaz A, et al. Fresh caching of dynamic content over the wireless edge[J]. IEEE/ACM Transactions on Networking, 2022, 30(5): 2315-2327.

[3] Buyya, Rajkumar, Mukaddim Pathan, and Athena Vakali, eds. Content delivery net-works. Vol. 9. Springer Science & Business Media, 2008.

[4] Yao J, Han T, Ansari N. On mobile edge caching[J]. IEEE Communications Survscholaeys & Tutorials, 2019, 21(3): 2525-2553.

[5] Abolhassani B, Tadrous J, Eryilmaz A. Optimal load-splitting and distributed-caching for dynamic content over the wireless edge[J]. IEEE/ACM Transactions on Networking, 2023.

[6] M. J. Siavoshani, S. P. Shariatpanahi, H. Ghasemi and A. Pourmiri, "On communication cost vs. load balancing in Content Delivery Networks", Proceedings - IEEE Symposium on Computers and Communications, 2017.

[7] H. Ghasemi, M. J. Siavoshani and S. Hadadan, A Novel Communication Cost Aware Load Balancing in Content Delivery Networks using Honeybee Algorithm, pp. 1-9, 2019.

[8] M. Aibin, M. Kantor, P. Borylo, H. Niedermayer, P. Cholda and T. Braun, "Resilient SDN CDN and ICN Technology and Solutions" in Guide to Disaster-resilient Communication Networks, Springer, pp. 1-22, 2020.

[9] M. J. Siavoshani, A. Pourmiri, and S. P. Shariatpanahi, "Storage, communication, and load balancing trade-off in distributed cache networks,"IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 4, pp. 943–957, 2018.

[10] S. Manfredi, F. Oliviero, and S. P. Romano, "A distributed control law for load balancing in content delivery networks," IEEE/ACM Transactions on Networking (TON), vol. 21, no. 1, pp. 55–68, 2013.

[11] M. Mitzenmacher, "The power of two choices in randomized load balancing," IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 10, pp. 1094–1104, 2001. [12] C.-M. Chen, Y. Ling, M. Pang, W. Chen, S. Cai, Y. Suwa, and

[12] O. Altintas, "Scalable request routing with next-neighbor load sharing in multi-server environments," in null. IEEE, 2005, pp. 441–446.

[13] V. Mathew, R. K. Sitaraman and P. Shenoy, "Energy-aware load balancing in content delivery networks", Proceedings - IEEE INFOCOM, 2012.

[14] Xu D, Liu X, Fan B. Efficient server provisioning and offloading policies for internet data centers with dynamic load-demand[J]. IEEE Transactions on Computers, 2013, 64(3): 682-697.

[15] George D A S, George A S H. The evolution of content delivery network: How it enhances video services, streaming, games, ecommerce, and advertising[J]. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), 2021, 10(07): 10435-10442

[16] Viola R, Martin A, Morgade J, et al. Predictive CDN selection for video delivery based on LSTM network performance forecasts and cost-effective trade-offs[J]. IEEE Transactions on Broadcasting, 2020, 67(1): 145-158.

[17] Wang X, Du X, Li W L, et al. A Bandwidth Prediction Method Based on Hybrid LSTM for Content Delivery Network[C]//2022 IEEE 7th International Conference on Smart Cloud (SmartCloud). IEEE, 2022: 206-211.

A Multi-Angle Encoding Spiking Convolutional Neural Network for Remote Sensing Classification

1st Xiang Li
School of Physical Science and Technology
Lanzhou University
Lanzhou, China
220220938961@lzu.edu.cn

2nd Jingwei Zhang
School of Integrated Circuits
Southeast University
Nanjing, China
zhangjingwei@seu.edu.cn

3rd Peng Wang*
School of Physical Science and Technology
Lanzhou University
Lanzhou, China
wangpeng@lzu.edu.cn
ORCID:0000-0003-3167-1547

4th Yanrong Wang
School of Physical Science and Technology
Lanzhou University
Lanzhou, China
wyr@lzu.edu.cn

5th Meng Zhang
School of Integrated Circuits
Southeast University
Nanjing, China
zmeng@seu.edu.cn

6th Feng Xu
School of Integrated Circuits
Southeast University
Nanjing, China
fxu@seu.edu.cn

7th An Jing
Nanjing Farad Technology Ltd.
Nanjing, China
jafld2024@163.com

8th Lizi Zhang
Brown University
Providence, USA
lizi_zhang1@alumni.brown.edu

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899489

***Abstract*—Spiking Convolutional Neural Networks (SCNNs), known as the third generation of neural networks, are favored for their low energy consumption and biological plausibility, making them ideal for energy-limited applications like satellite remote sensing image classification. Traditional Convolutional Neural Networks (CNNs) consume significant energy, prompting a shift towards more efficient architectures like binary and adder neural networks. However, SCNNs have been overlooked due to their binary information transmission, which typically results in lower accuracy. This paper introduces the Multi-Angle Encoding Spiking Convolutional Neural Network (MASCNN), featuring a Multi-Angle Encoding Layer and a Deep Feature Extraction Module to enhance input information and improve classification accuracy. A new Multi-Angle Loss Function is also proposed to enrich learning. Testing on various datasets shows that MASCNN outperforms other low-energy networks in accuracy while maintaining minimal energy use.**

Keywords—Deep learning, Spiking Convolutional Neural Networks, Remote Sensing Images Classification, Low Energy Consumption

I. Introduction

In Spiking Convolutional Neural Networks (SCNNs), the spiking refers to spiking neurons, an abstracted model that better aligns with the human brain's mechanism for generating judgments[1]. When the human brain receives a stimulus, many biological neurons generate spike signals. As these spike signals propagate, neurons with less relevance to the stimulus cease transmitting signals (state 0), while the remaining few continue (state 1). Ultimately, only a few neurons may fire (state 1) to make the final judgment. Spiking neurons model this process by converting input into electrical membrane potential. Each input-related potential accumulates over time in the neuron. When the accumulated membrane potential exceeds a set threshold, the neuron fires a spike (output 1). When it does not reach the threshold, the neuron outputs 0[2]. This characteristic of having time accumulation is the biggest difference between SCNN and Binary Neural Networks (BNN)[3]. In artificial intelligence models, spiking neurons alone are not sufficient to achieve desired outcomes. In tasks like computer vision classification, inputs still need hidden layers to extract features before yielding correct classification results. Current mainstream practice is to integrate CNN's convolutions with spiking neurons, forming SCNNs. SCNNs consume less energy than traditional CNNs because the data transmitted by spiking neurons is in binary form. This avoids floating-point multiplications in convolution calculations, primarily involving multiplication and addition with convolution weights and binary data, significantly reducing energy storage and computation costs[4].

However, SCNNs also have the evident disadvantage of lower accuracy than CNNs with the equivalent parameter and computation counts. Due to the binary nature of spiking neurons, feature information is not adequately transmitted, worsening through deeper neural network layers, leading to lower final classification accuracy. To address this, SCNNs' event-driven nature can be leveraged by adding a time dimension, allowing membrane potential to gradually increase over time, firing more spike signals to transmit information. Conventional cameras' images lack event characteristics, thus requiring encoding before feeding into SCNNs. This encoding imparts temporal dynamics, helping SCNNs capture more information. The event-driven nature translates the network's input from the conventional four dimensions [batch size, channel, height, width] to five dimensions [T, batch size, channel, height, width], where T is the encoding timestep[5]. Typically, a single image is encoded over various time steps, each containing different information. Larger T retains more information, enhancing accuracy but also increasing computation and time.

Common encoding methods address general computer vision tasks, but remote sensing images of the same class can have varying orientations. Conventional encodings lack specificity in remote sensing contexts. Therefore, encoding remote sensing images should incorporate rotation characteristics, aiding the network in learning varying angles and directions, ultimately boosting classification accuracy.

* Corresponding author

This work was supported by the Key R&D Program of Guangdong Province (Project No. 2021B1101270006) and the Fundamental Research Funds for the Central Universities.

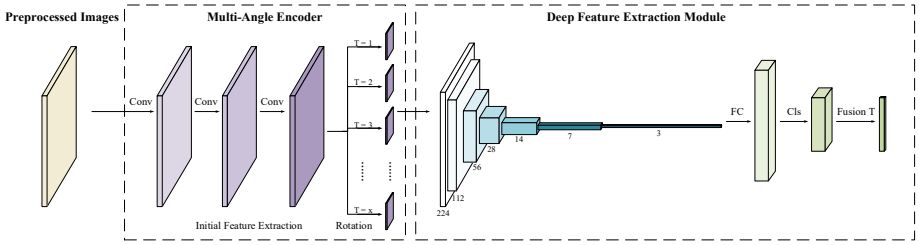


Fig. 1. The figure of the overall structure of MASCNN.

In this paper, a Multi-Angle Encoding Spiking Convolutional Neural Network (MASCNN) is proposed to leverage these characteristics for high-precision, low-energy remote sensing image classification. First, input images are encoded through a multi-angle encoding layer. These encoded feature maps then feed into a deep feature extraction module to extract deep-level features from multi-angle information, fusing predictions over all time steps for the final classification result. A novel multi-angle loss function evaluates the loss at each timestep, performing weighted backpropagation. Across multiple remote sensing image classification datasets, MASCNN demonstrated superior results and the lowest energy consumption compared to previously proposed low-energy neural networks for remote sensing image classification.

II. Related Works

A. Spiking Convolutional Neural Networks

Since the 0 and 1 signals produced by pulse neurons do not have gradients, making backpropagation impossible, researchers introduced surrogate gradients, enabling Spiking Convolutional Neural Networks (SCNN) to learn similarly to Artificial Neural Networks (ANN)[6]. Following the success of Spiking VGG, it was discovered that SCNN could achieve comparable accuracy to ANN. Subsequently, Zheng et al.[7] introduced residual structures into SCNN, allowing for deeper network architectures. In the last two years, research on the vision transformer structure of SCNN has also become increasingly mature. SCNNs are now widely applied in various computer vision tasks.

B. Low-Energy Neural Networks for Remote Sensing Image Classification

Chen et al. proposed an Adder Neural Network (ANN), which converts most multiplication operations in convolutional layers to addition operations[8]. Since addition operations consume less power than multiplication, this reduces the overall power consumption of convolutions, resulting in a lower energy yet high-accuracy ANN. Building on ANN, Zhang et al. proposed an even more extreme All Adder Neural Network (A^2NN), replacing all multiplication operations in CNN with addition, resulting in lower energy consumption[9].

III. Methodology

A. Spiking Neuron Models

Neurons generally consist of three processes: charging, discharging, and resetting. Common neuron models include the Integrate-and-Fire (IF) neuron and the Leaky Integrate-and-Fire (LIF) neuron[10].

1) IF Neuron: The IF neuron is characterized by a constant membrane potential when there is no input. During the charging process, the membrane potential $V[t]$ at the current time t is related to the previous membrane potential $V[t-1]$ and the current input $X[t]$. During discharging, $V[t]$ is compared with a set discharge threshold $V_{threshold}$, and the output spike signal at the current time is $S[t]$. The resetting process can be either hard or soft reset. Hard reset means directly resetting $V[t]$ to a set original membrane potential V_{reset}, while soft reset means resetting $V[t]$ to $V[t]-V_{threshold}$. The entire process can be described by the following equations.

$$V[t] = V[t-1] + X[t] \tag{1}$$

$$S[t] = \begin{cases} 0 & if\ V[t] < V_{threshold} \\ 1 & if\ V[t] \geq V_{threshold} \end{cases} \tag{2}$$

$$V[t] = \begin{cases} V_{reset} & if\ HardReset \\ V[t] - V_{threshold} \times S[t] & if\ Soft\ Reset \end{cases} \tag{3}$$

2) LIF Neuron: Compared to the IF neuron, the main difference in LIF neurons is the introduction of a leakage factor τ during the charging process, causing the membrane potential to naturally decrease when there is no input. The charging process can be described by the formula:

$$V[t] = V[t-1] - \frac{1}{\tau}(V[t-1] - V_{reset}) + X[t] \tag{4}$$

B. MASCNN Design

The conventional encoding method groups the pixel values of images based on their size over various time steps T. This approach lacks rotational information, preventing the network from learning rotation details and achieving high accuracy on remote sensing image datasets. Therefore, we propose a Multi-Angle encoding layer. First, let $T = 1$, and the input image I has a size of [1, bs, c, h, w], where bs means batch size, c means channels and h and w means height and width. We perform initial feature extraction to obtain I'. Here, $I' = \Phi(I)$, where $\Phi(\cdot)$ includes the stacking of three convolutional layers, one batch normalization layer and one activation layer, which do not change the number of channels or the size of the feature maps, and are used to initially abstract features. The weights of the convolutional kernels are still updated through backpropagation, which helps continually optimize the feature extraction results. Neurons are not used here because at this stage, T is equal to 1, meaning there is no accumulation of membrane potential over different time steps. Using neurons in this case would result in a significant loss of information, potentially leading to all pixel

values being zero, which would cause the network to lose information and produce erroneous outputs.

Next, let $T = x$, where x represents the total number of rotation angles. To uniformly cover the range from 0 to 360 degrees without any fractional angles, x can take values from 1 to 6 and 8. Although larger values of x are possible, they increase computational load significantly, leading to low efficiency, so larger values are not considered. The feature map I' obtained from initial feature extraction is rotated by $\alpha = \frac{360i}{x}$ degrees and fed into the network at each time step $i = 1, \dots, x$. Finally, the outputs are concatenated in sequence to form a tensor of size [x, bs, c, h, w] for the subsequent deep feature extraction module. This is more intuitively expressed in the following formula:

$$output = R\big(\Phi(I)\big) \tag{5}$$

where $R(\cdot)$ represents uniform rotation x times to cover the range from 0 to 360 degrees. For instance, when $x = 4$, the α are 90, 180, 270, and 360 degrees, resulting in an output tensor size of [4, bs, c, h, w] after the operations of $\Phi(\cdot)$ and $R(\cdot)$ functions.

The deep feature extraction module is designed as a stack of six convolution layers, LIF neurons, and max pooling layers. Each convolution ensures that the feature map's scale remains invariant while expanding the width of the network, and max pooling is used for down sampling. After the last max pooling layer, global average pooling is applied. This process allows a 224×224 image to rapidly extract features and downsample across six convolutions to produce abstract features of size 3×3. Subsequently, the features are flattened and passed through the fully connected layer and the Classification layer, followed by the fusion of classification results from all time steps T to obtain the final classification prediction.

The main design concept of the deep feature extraction module is to gradually reduce the size while increasing the width layer by layer, aiming to quickly extract abstract features from the input image with fewer layers. This approach helps to avoid data loss or failure of information transmission when the pulse signal data, consisting only of 0s and 1s, is passed through deeper layers of the network. Moreover, a smaller network reduces energy consumption and achieves faster image processing speed. The overall network structure is shown in the Fig. 1, where Conv represents convolutional layers, FC represents fully connected layers, and Cls represents the Classification layer.

C. Multi-Angle Loss Function

Before fusing all time steps T, the network produces a tensor A of size [x, bs, cls], which contains the classification results at each time step, where cls means the number of classifications. This tensor is used for calculating the multi-angle loss function. However, the size of the label is [bs, cls], so we first expand the label into a tensor l_T of size [x, bs, cls], and then calculate the loss for each time step. The loss is defined as follow:

$$L_T = MSE(A, l_T) \tag{6}$$

with L_T having a size of [x, 1]. Next, all losses are sorted in ascending order and assigned weights from 2 to 1 for backpropagation. The specific formula is as follows:

$$L_{bp} = AO(L_T) \times LS(2,1,x) \tag{7}$$

where L_{bp} represents the backpropagation loss, AO denotes the ascending order rearrangement of L_T, and LS uses the linspace function to generate an arithmetic sequence of x values from 2 to 1 as weights for L_T. This design leverages the fact that different time steps carry information from different perspectives, and since the membrane potential accumulated by neurons differs across time steps, the output results vary. By utilizing the loss from each time step, the network can learn information from different angles at different moments, which is beneficial for improving classification accuracy. The calculation process of L_{bp} and prediction accuracy is shown schematically in Fig. 2.

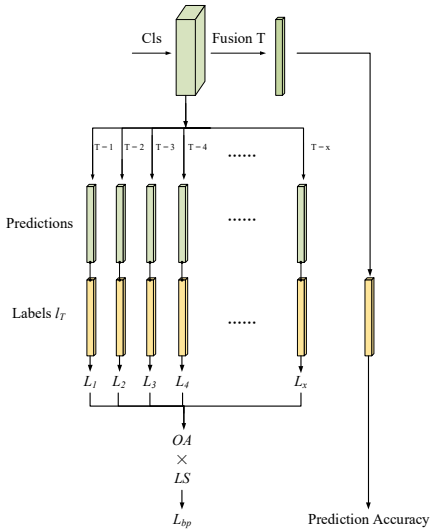


Fig. 2. The figure of calculation process of multi-angle loss function and prediction accuracy.

IV. Experiment

We tested the capabilities of MASCNN on multiple datasets, including the RS-C11[11], RSSCN7[12], SIRI-WHU[13], and WHU-RS19[14]. All these datasets are related to environmental category classification, with RS-C11 containing 11 categories with a total of 1232 images. RSSCN7 consists of 7 categories, each with 400 images. SIRI-WHU contains 12 categories with 200 images each and WHU-RS19 includes 19 categories with a total of 1005 images. We will compare our proposed MASCNN with two other low-energy neural networks, ANN and A^2NN, across these four datasets.

To ensure fairness, we preprocessed the datasets following the methodology outlined in the [9]. For RS-C11 and WHU-RS19, 80% of the images were randomly selected for training, while the remaining 20% were used for validation. For SIRI-WHU, the training and validation ratio was 4:6, while RSSCN7 had a 2:8 ratio. Additionally, we resized all images to 256×256 pixels. During training, we used random cropping to input images of size 224×224, along with random horizontal

TABLE I.
ACCURACY OF MASCNN VS. OTHER LOW ENERGY NEURAL NETWORKS WITH VGG OR RESNET AS BACKBONE ON MULTIPLE DATASETS.

Networks		Param(k)	Accuracy (%)			
			RS-C11	RSSCN7	SIRI-WHU	WHU-RS19
ResNet18		11690	94.94	84.16	91.90	92.58
ResNet18-ANN		11690	91.97	79.98	86.22	87.22
ResNet18-A^{2}NN		11690	91.73	78.63	85.59	79.12
VGG11		46900	94.46	82.12	87.53	91.90
VGG11-ANN		46900	93.41	79.98	86.68	89.46
VGG11-A^{2}NN		46900	92.05	78.92	85.90	87.22
VGG13		48100	94.38	81.94	87.99	90.83
VGG13-ANN		48100	91.00	77.16	86.06	87.22
VGG13-A^{2}NN		48100	90.62	76.86	85.57	86.83
MASCNN	x=1	393	76.95	75.71	81.25	74.23
	x=1 (Aug)	393	82.30	77.63	83.54	80.93
	x=2	393	85.19	80.76	85.83	85.57
	x=3	393	88.89	83.08	87.36	88.14
	x=4	393	90.77	82.46	88.68	90.72
	x=5	393	92.18	83.08	88.75	92.27
	x=6	393	93.42	83.75	87.64	93.81
	x=8	393	**94.65**	**84.51**	**89.17**	**95.36**

and vertical flips for data augmentation. For validation, we applied center cropping to produce 224×224 input images.

We conducted experiments with x values ranging from 1 to 6 and 8, where $x = 1$ serves as the baseline for MASCNN, as it does not use any angle encoding and directly inputs images into the networks.

To demonstrate the effectiveness of multi-angle encoding layer, we added random rotation data augmentation to the input images at $x = 1$, labeled as (Aug). The comparison results are shown in the Tab. I below, indicating that when $x = 8$, our proposed MASCNN outperforms both ANN and A^{2}NN across the four datasets, while also having fewer parameters and lower computational costs. Compared to other low-energy neural networks that use traditional CNN structures like VGG[15] and ResNet[16]as backbones, MASCNN achieves high accuracy with a relatively small number of parameters primarily due to the incorporation of rotation information. This involves rotating the initial features before passing them into the deeper network, which is a unique optimization method well-suited for the application of remote sensing image classification. Because of this, our proposed MASCNN even outperforms traditional CNNs on some datasets. In contrast, other networks do not incorporate rotation information. Despite having a large number of parameters, their accuracy remains low.

Furthermore, it is evident that the network's accuracy is low at $x = 1$, and while data augmentation improves accuracy, it still does not match the performance when x is greater than 1. As x increases, the accuracy of MASCNN improves, which aligns with expectations. A larger x means more refined rotational encoding, and the accumulated membrane potential increases over time, leading to more information being transmitted through spikes. However, achieving higher accuracy comes at the cost of increased energy consumption and slower computation, as shown in the Tab. II and Tab. III.

TABLE II.
FRAME RATE PERFORMANCE OF MASCNN WHEN X TAKES DIFFERENT VALUES.

MASCNN	Performance (fps)
x=1	218.77
x=2	143.72
x=3	123.44
x=4	102.31
x=5	81.15
x=6	70.41
x=8	56.61

We referred to [17] to compute the power consumption of all the networks mentioned above. We assume that all operations are performed using 32-bit float calculations implemented with 45-nanometer technology[18]. In this context, the energy cost of each multiplication operation E_{mul} is 3.7 pJ, the energy cost of each addition operation E_{ac} is 0.9 pJ, and the energy cost of each multiply-accumulate operation E_{mac} is 4.6 pJ. Additionally, because MASCNN includes rotation operations, each pixel requires 64 multiplications and 48 additions for each rotation. The rotated feature map contains a total of 224×224 pixels, so this energy cost also needs to be taken into account. The total power consumption is shown in the Tab. III. It can be observed that our proposed MASCNN also has the lowest power consumption, demonstrating that MASCNN is a low-energy neural network suitable for remote sensing image classification.

TABLE III.
COMPARISON OF MASCNN WITH OTHER LOW ENERGY NEURAL NETWORKS IN TERMS OF ENERGY CONSUMPTION.

Networks		FLOPs (M)	Operations (M)					Energy (mJ)
			Convolution / Adder Layers		Rotation		Other Layers	
			Mul	Add	Mul	Add	MACs	
ResNet18		36300	18147.51	18147.51	0	0	4.98	83.50
ResNet18-ANN		36300	59.01	36236.01	0	0	4.98	32.85
ResNet18-A^{2}NN		36300	0	36295.02	0	0	4.98	32.69
VGG11		15000	7492.58	7492.58	0	0	14.85	34.53
VGG11-ANN		15000	43.36	14941.79	0	0	14.85	13.68
VGG11-A^{2}NN		15000	0	14985.15	0	0	14.85	13.55
VGG13		22420	11197.76	11197.76	0	0	24.48	51.62
VGG13-ANN		22420	43.36	22352.16	0	0	24.48	20.39
VGG13-A^{2}NN		22420	0	22395.52	0	0	24.48	20.27
MASCNN	x=1	953	0	940.81	0	0	12.19	0.9
	x=1 (Aug)	953	0	940.81	0	0	12.19	0.9
	x=2	1912	0	1881.61	3.51	2.61	24.39	1.82
	x=3	2871	0	2822.42	7.02	5.22	36.58	2.74
	x=4	3830	0	3763.23	10.54	7.83	48.77	3.66
	x=5	4789	0	4704.31	14.05	10.44	60.69	4.57
	x=6	5749	0	5644.84	17.56	13.05	73.16	5.49
	x=8	7661	0	7526.46	21.07	15.65	97.54	**7.31**

V. CONCLUSION

This paper introduces the Multi-Angle Encoding Spiking Convolutional Neural Network (MASCNN), a low-energy neural network designed for classifying remote sensing images. MASCNN features a multi-angle encoding layer to extract and concatenate features from multiple rotations, followed by a deep feature extraction module comprising convolutional layers, LIF neurons, and a max pooling layer. This configuration effectively reduces the feature map size while increasing channel depth. A novel multi-angle loss function, which aggregates weighted losses from different angles and time steps, optimizes the training process. MASCNN achieved the best results across multiple remote sensing datasets with the lowest energy consumption.

REFERENCES

[1] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," Brain Sciences, vol. 12, no. 7, p. 863, 2022.

[2] S. Sonoda and N. Murata, "Neural network with unbounded activation functions is universal approximator," Applied and Computational Harmonic Analysis, vol. 43, no. 2, pp. 233–268, 2017.

[3] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," Advances in neural information processing systems, vol. 28, 2015.

[4] J. Zhang et al., "A comprehensive analysis of DAC-SDC FPGA low power object detection challenge," Science China Information Sciences, 2024, 67(8): 182401.

[5] W. Fang et al., "Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence," Science Advances, vol. 9, no. 40, p. eadi1480, 2023.

[6] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," Neural computation, vol. 30, no. 6, pp. 1514–1541, 2018.

[7] H. Zheng, Y. Wu, L. Deng, Y. Hu, and G. Li, "Going deeper with directly-trained larger spiking neural networks," in Proceedings of the AAAI conference on artificial intelligence, 2021, pp. 11062–11070.

[8] H. Chen et al., "AdderNet: Do we really need multiplications in deep learning?" in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 1468–1477.

[9] N. Zhang, G. Wang, J. Wang, H. Chen, W. Liu, and L. Chen, "All adder neural networks for on-board remote sensing scene classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1–16, 2023.

[10] E. Hunsberger and C. Eliasmith, "Training spiking deep networks for neuromorphic hardware," arXiv preprint arXiv:1611.05141, 2016.

[11] L. Zhao, P. Tang, and L. Huo, "Feature significance-based multibag-of-visual-words model for remote sensing image scene classification," Journal of Applied Remote Sensing, vol. 10, no. 3, pp. 035004–035004, 2016.

[12] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," IEEE Geoscience and remote sensing letters, vol. 12, no. 11, pp. 2321–2325, 2015.

[13] B. Zhao, Y. Zhong, G.-S. Xia, and L. Zhang, "Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery," IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 4, pp. 2108–2123, 2015.

[14] G. Sheng, W. Yang, T. Xu, and H. Sun, "High-resolution satellite scene classification using a sparse coding based multiple feature combination," International journal of remote sensing, vol. 33, no. 8, pp. 2395–2412, 2012.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[17] M. Yao et al., "Spike-driven transformer," Advances in neural information processing systems, vol. 36, 2024.

[18] G. Li et al., "An analysis of TinyML@ ICCAD for implementing AI on low-power microprocessor", Science China Information Sciences, 2024, 67(4): 149402.

ChatGPT as a Negotiator: An Analysis of Its Adherence with Proportionality and Equality

1st Meltem Aksoy
Computer Science
Research Center Trustworthy Data Science and Security
Technical University Dortmund
Dortmund, Germany
meltem.aksoy@tu-dortmund.de

2nd Veronika Tsishetska
Computer Science
Technical University Dortmund
Dortmund, Germany
veronika.tsishetska@tu-dortmund.de

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899509

***Abstract*—This study examines the adherence of ChatGPT (GPT-3.5 and GPT-4) to fairness principles, specifically proportionality and equality, in negotiation scenarios. Three distinct negotiation contexts were explored: work-study program funding, company sale proceeds division, and employee bonus allocation. The models were prompted to prioritize proportionality, equality, or no specific fairness principle to assess how they respond in different ethical frameworks. A combination of qualitative and quantitative methods, including dialogue analysis, sentiment tracking, and fairness scoring, was used to evaluate their negotiation behaviors. Results indicate that both models tend to favor proportionality by default, with GPT-4 showing greater adaptability compared to GPT-3.5. When explicitly directed to prioritize equality, the models followed these instructions but maintained a largely assertive negotiation style with limited engagement in dynamic exchanges. Sentiment analysis revealed that both models adopted increasingly positive tones as negotiations progressed. However, both versions were susceptible to prompt manipulation, potentially compromising fairness in some outcomes. This research highlights the potential of LLMs like ChatGPT in automating fair negotiations, though improvements in consistency, adaptability, and safeguards against manipulation are necessary for more robust, ethical applications in real-world scenarios.**

***Index Terms*—Human-AI interaction, large language model, ChatGPT, fairness**

I. Introduction

In recent years, advances in Artificial Intelligence (AI) have transformed various industries, particularly with the rise of Large Language Models (LLMs). These models are now being used in areas such as contract negotiation, bargaining and e-commerce. Meta's research has shown that it's possible to train end-to-end models for negotiation, where LLMs learn both language and reasoning skills without needing annotated dialogue [1]. These models improve their negotiation abilities by simulating conversations and planning responses through dialogue rollouts. For instance, the Luminance AI Autopilot tool uses a legal-specific LLM, trained on millions of legal documents, to assist legal professionals by simplifying the negotiation process for standard contracts [2]. Additionally, AI-powered negotiation systems, like those used in e-commerce, enable shoppers to negotiate prices with virtual agents, creating more interactive and personalized experiences [3].

These innovations demonstrate the growing potential of LLMs to optimize and automate negotiation tasks, which have traditionally been the domain of humans. However, as LLMs gain more autonomy, concerns about bias, fairness, and ethical decision-making become increasingly important. Ensuring that these systems operate ethically and fairly, particularly in competitive negotiating environments, is essential to their wider acceptance and effectiveness.

This research aims to explore how ChatGPT, specifically its two versions GPT-3.5 and GPT-4, aligns with fairness principles in human-AI negotiations. By focusing on the models' behavior in terms of compromise, assertiveness, or neutrality, we seek to understand whether ChatGPT consistently applies fairness foundations, such as equality and proportionality, in its interactions. Furthermore, this study examines how negotiation outcomes are influenced when ChatGPT is explicitly directed to prioritize these fairness principles.

Addressing these questions is crucial to advancing our understanding of AI-driven negotiation and ensuring that such systems are transparent, accountable, and capable of generating fair outcomes. Ultimately, this research contributes to the development of trustworthy AI systems that can not only participate in negotiations but also lead them in a manner that aligns with ethical standards and human expectations.

The structure of this paper is as follows: Section 2 presents an overview of the literature. Section 3 outlines the proposed research methodology, while Section 4 presents the results. Section 5 presents conclusions, and Section 6 gives the limitations and suggests directions for future research.

II. Related Works

The exploration of LLMs in human negotiation settings has gained increasing attention, especially with the development of models like ChatGPT. Previous studies have examined the application of LLMs in non-collaborative and competitive contexts, revealing both their strengths and limitations in human-AI interactions.

Reference [4] investigated LLMs in price negotiations. The authors showed that models can be easily manipulated

by prompt hacks, leading to irrational agreements despite structured negotiation processes. The authors also identified a significant gap in the ability of humans to negotiate effectively with LLMs. Reference [5] evaluated the GPT-4 in negotiation tasks and found it to be strong in objective reasoning but weak in subjective judgment and strategic thinking. Reference [6] compared ChatGPT and Claude in a ransomware negotiation task, highlighting hallucinations and the need for human supervision in a high-stakes scenario.

Reference [7] revealed the vulnerabilities of GPT-4's APIs, particularly focusing on fine-tuning, function calling, and knowledge retrieval functionalities. These findings have important implications for using LLMs in negotiations where safeguarding against adversarial use is crucial. Moreover, the authors highlighted the susceptibility of GPT-4's knowledge retrieval system to prompt injections, a concern that could easily manifest in high-stakes negotiation scenarios where accuracy and fairness are imperative. Reference [8] noted that while LLMs maintain internal logic, they struggle to maximize cooperative outcomes. Reference [9] introduced NEGOTIATIONARENA, showing that while certain tactics improved negotiation results, models still exhibited irrational behaviors. Reference [10] emphasized the need for robust evaluation frameworks for LLMs in real-world negotiations, calling for long-term, adaptive strategies. Reference [11] assessed the moral reasoning of LLMs, revealing significant limitations in handling complex ethical dilemmas, and highlighting the need for ongoing evaluation to align with human ethics.

Despite the growing body of research on the negotiation skills of LLMs, there remains a gap in understanding how these models apply fairness principles, such as proportionality and equality, in negotiation scenarios. This study addresses this gap by systematically analyzing the behavior of ChatGPT models in three predefined negotiation settings. By focusing on ChatGPT's adherence to fairness principles, we provide a structured evaluation of its performance in human-AI negotiations. In addition, by using the Moral Foundations Questionnaire (MFQ) to assess model fairness, our research provides a comprehensive analysis of ChatGPT's ethical behavior in negotiation.

Table I provides an overview of studies evaluating the negotiation abilities of LLMs, detailing the models used, scenarios, methodologies, and the involvement of human participants in each study.

TABLE I
STUDIES ON NEGOTIATION ABILITIES OF LLMS

Study	LLM Evaluated	Scenario/Context	Methodology	Human Participants in Negotiations
[1]	GPT-3.5	Price negotiations	User study, thematic content analysis and quantitative analysis of negotiation dialogues	
[2]	GPT-4, GPT-3.5, Mistral-7B, Wizard-13B, Vicuna, Flan-T5	Multi-issue bargaining	Evaluating LLMs on 35 negotiation tasks to assess comprehension, partner modeling, strategy, and response generation	No
[3]	GPT-3.5, Claude	Ransomware negotiations	For simulated ransomware scenarios, evaluating LLMs on efficacy, accuracy, adaptability, ethics and manipulation resistance	No
[4]	GPT-3.5, GPT-4, Claude-2, Cohere, LLaMA 2	Structured negotiation games	Testing LLMs in structured negotiation games to evaluate alignment, performance, and instruction-following	No
[5]	GPT-3.5, GPT-4, Claude 2, Claude 2.1	Allocating shared resources, aggregate resources and buy/sell goods	Evaluating their negotiation skills, rationality, and social behaviors across various game types	No
Our study	GPT-3.5, GPT-4	Work-study program funding allocation, company sale proceeds division, and employee bonus distribution	Human participants negotiating with LLMS in scenarios focused on fairness principles, using both qualitative and quantitative analyses to evaluate the models' adherence to proportionality and equality	Yes

A. Large Language Models

LLMs represent a significant breakthrough in Natural Language Processing (NLP), with the transformer architecture at the forefront of this transformation. Traditionally, NLP tasks relied heavily on recurrent neural networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs). These models were designed to handle sequential data, allowing them to capture temporal dependencies in language. However, despite their successes, RNNs and their derivatives faced inherent limitations that constrained their performance, particularly when dealing with long-range dependencies and complex contextual relationships in text.

The introduction of the transformer architecture [12] marked a paradigm shift in NLP. Transformers eliminated the need for sequential processing by leveraging the self-attention mechanism, allowing models to consider all elements in a sequence simultaneously. This innovation addressed the key limitations of RNNs and enabled the development of more powerful and efficient language models.

OpenAI's development of the GPT (Generative Pre-trained Transformer) models built on this foundation, with each new version introducing substantial advancements. The original GPT [13] was a specialized adaptation of the transformer, incorporating essential components such as embedding algorithms, positional encoding, self-attention mechanisms, and softmax layers. GPT-2 [14] followed, with 1.5 billion parameters and significantly improved text generation capabilities. In 2020, GPT-3 [15] was introduced, featuring 175 billion parameters, a milestone in generating highly convincing, human-like text and even code based on detailed instructions.

Building on GPT-3, OpenAI launched ChatGPT [16], based on GPT-3.5 and fine-tuned using Reinforcement Learning from Human Feedback (RLHF). In April 2023, OpenAI released GPT-4 [17], its most advanced model to date, surpassing GPT-3.5 in output quality, accuracy, and contextual understanding.

With each iteration, GPT models have increased in both the number of parameters and the size and diversity of training data. A notable trend throughout this evolution has been the power of scaling—larger models with more extensive training data generally deliver better performance. However, this scaling also brings challenges, including the risk of harmful outputs, the increasing demand for computational resources, and the need for effective strategies to control and guide the models' behavior.

In this study, we utilized two specific versions of the GPT models: GPT-3.5-turbo and GPT-4-turbo, which represent key stages in the evolution of ChatGPT. These versions were accessed through the OpenAI API and Python bindings and were consistently used throughout our analysis.

B. Moral Foundations Theory and Questionnaires

Moral Foundations Theory (MFT) [18] was developed to explore the origins of morality, the diversity in moral judgments across cultures, and whether morality functions as a singular construct or a complex system [19], [20]. Grounded in evolutionary psychology and anthropology, MFT suggests that humans possess a set of innate moral foundations that are expressed differently based on individual and cultural contexts. These core foundations include care, fairness, loyalty, authority, and purity, which influence ethical decision-making and social behavior. For a more in-depth exploration of MFT, see [20].

To measure how individuals prioritize these moral values, the Moral Foundations Questionnaire (MFQ) was created. The original MFQ [21], [22] consists of 30 items rated on a six-point Likert scale, divided into two sections: one assessing the importance of moral judgments and the other gauging agreement with moral statements. Each of the five moral foundations is represented by six items, and scores for each foundation are averaged. The MFQ has been widely translated and is extensively used in cross-cultural research on morality.

However, critics have argued that the original MFQ's fairness foundation lacked nuance in capturing the complexities of distributive justice across different cultures [23]. To address this issue, [23] introduced the MFQ-2, which refines the fairness foundation by splitting it into two components: Equality (treating everyone the same) and Proportionality (rewarding people based on their contributions). The MFQ-2 includes 36 items across different foundations, with each item rated on a five-point Likert scale, from 1 (”does not describe me at all”) to 5 (”describes me extremely well”). This version provides a more detailed assessment of moral foundations, with average scores calculated for each foundation. More details on MFQ-2 can be found in [23].

In this study, we conducted two experiments using the MFQs to evaluate ChatGPT's adherence to fairness principles. The first experiment, based on the fairness foundation of the original MFQ, assessed two key aspects: the role of fairness in the models' moral judgments and their agreement or disagreement with fairness-related statements. Each model rated three statements in both categories, with multiple iterations performed to ensure consistency and reliability. The second experiment focused on the equality and proportionality foundations of MFQ-2. Both GPT-3.5 and GPT-4 rated fairness-related statements on a Likert scale from 1 to 5, where 1 indicated that the statement did not describe the model at all, and 5 indicated it described the model extremely well. These ratings were repeated 100 times, and the mean and standard deviation were calculated for both equality and proportionality.

III. Research Methodology

We designed a web platform to efficiently collect participant data in a structured and user-friendly manner. Initially, participants provided basic demographic information, including age, gender, academic qualifications, and languages spoken. Following this, they rated how well the statements from the MFQ-2 related to equality and proportionality aligned with their views. They also responded to two open-ended questions to explain their understanding of these principles.

We developed two chatbots and integrated them into the web platform, allowing participants to engage in negotiation scenarios using GPT models. In one chatbot, participants conducted a negotiation using GPT-3.5 within a selected scenario, and in the other, they repeated the negotiation using GPT-4. Each negotiation was structured around a specific context—proportionality, equality, or a default setting. This setup enabled us to collect data on how the models interacted with humans across these different negotiation strategies.

A. Negotiation Scenarios

We developed three distinct scenarios to simulate realistic negotiation settings:

- **Work-Study Program (WSP):** Participants negotiated how to allocate funds among three candidates, taking into account their varying academic achievements and financial needs.
- **Company Selling (CS):** Participants negotiated the division of proceeds from a company sale, with consideration given to each party's level of investment.
- **Bonus Allocation (BA):** A negotiation scenario focused on distributing a bonus among three employees, based on their performance and contributions.

B. Negotiation Strategies

In each scenario, the model's behavior was directed by specific prompt instructions based on one of three negotiation strategies: Proportional, Equal, or a Default setting. These strategies allowed us to evaluate how language models negotiate under different ethical and fairness guidelines:

- **Proportional**: The model was prompted to adjust its responses or resource allocations according to the specific needs or contributions of the parties involved, ensuring decisions were tailored to varying circumstances.
- **Equal**: The model was instructed to treat all parties equally, regardless of differences in their circumstances, prioritizing uniformity and non-discrimination.

- **Default**: The model received no specific instruction, allowing it to negotiate without following any particular strategy, serving as a neutral baseline for comparison.

C. Prompting

Prompts are the key inputs that guide LLMs in generating responses. Crafting these input sequences, known as prompting, allows for control over the model's output. Prompts offer flexibility through techniques such as zero-shot, one-shot, and few-shot prompting. In zero-shot prompting, the model generates responses based solely on the provided prompt without any examples. One-shot prompting includes a single example, while few-shot prompting provides several examples to improve accuracy. In our research, we utilized zero-shot prompting.

To evaluate ChatGPT's adherence to fairness principles, we used items from the fairness foundation of the MFQ and the equality and proportionality foundations of the MFQ-2. Both GPT-3.5 and GPT-4 rated a series of fairness-focused items from these questionnaires. Each item was rated 100 times by both models to ensure reliability and consistency in the results. Additionally, a system prompt was included for each model to ensure their responses followed the Likert scale format for each item.

- **System prompt for questionnaire**: For each statement, please indicate how well it describes you or your opinions. Select one of the options: Does not describe me at all, Slightly describes me, Moderately describes me, Describes me fairly well, Describes me extremely well.

For effective human-GPT interaction, we designed another system prompt to ensure that the GPT models negotiate as intended for each scenario and strategy, establishing clear rules to guide their responses during the negotiation process.

- **System prompt for negotiations**: Strategy type[strategy], scenario instructions[scenario], Respond concisely and briefly in no more than three sentences following these rules: 1. Do not apologize. 2. Do not include the prompt in your answers. 3. Act according to the given principle, but do not mention that it is given to you. 4. Do not use the following words in your answers: principle, proportionality, equality. 5. Support your opinions with reasoning rather than simply listing numbers.

D. Research Evaluation

We evaluated the performance of GPT-3.5 and GPT-4 in negotiations using both qualitative and quantitative analyses to provide a thorough assessment of their negotiation strategies and adherence to fairness principles.

For the qualitative analysis, we examined how closely the models followed the instructed fairness principles of equality and proportionality, identifying any deviations or compromises. We also analyzed their negotiation styles, noting patterns such as a tendency toward assertiveness, compromise, or neutrality. Additionally, we assessed how well the models maintained politeness and stayed focused, even when confronted with inappropriate language or off-topic behavior during the negotiations.

The quantitative analysis provided statistical insights into the models' negotiation behavior. We analyzed participant demographics and tracked the number of completed negotiation rounds. We classified dialogue acts (such as statements, offers, and requests) to quantify the models' communication styles, and we used sentiment analysis to monitor the emotional tone of their responses, categorizing them as positive, neutral, or negative. We also examined concession patterns to measure how frequently the models made compromises. Finally, we conducted a statistical analysis of negotiation outcomes to determine whether the results aligned with the fairness principles the models were instructed to follow.

Additionally, we introduced two new indices to evaluate the negotiation outcomes: proportionality index (I_p) and equality index (I_e). I_p measures how closely the actual distribution of resources matches the ideal proportional share for n participants, while the I_e measures how evenly resources were distributed among participants.

Let's assume that x_i represents the actual amount distributed to participant i, T represents the total amount distributed, i.e., $T = \sum_{i=1}^{n} x_i$, w_i represents the proportional weights for each participant, such that $\sum_{i=1}^{n} w_i = 1$. The proportional share for participant i is $p_i = T \times w_i$, and the I_p is shown as in (1):

$$I_p = 1 - \frac{\sum_{i=1}^{n} |x_i - T \times w_i|}{T} \tag{1}$$

The equality index for n participants measures how closely the actual distribution matches an equal share for all. Assuming each participant should receive an equal share $E = \frac{T}{n}$, the I_e is shown as in (2):

$$I_e = 1 - \frac{\sum_{i=1}^{n} \left|x_i - \frac{T}{n}\right|}{T} \tag{2}$$

Higher values for these indices reflect stronger alignment with the principles of proportionality or equality, providing a measure of the overall fairness of the negotiation outcomes. A score of 1 on either index indicates perfect alignment with the corresponding principle.

IV. Results

A total of 150 participants took part in our study. Following data cleaning to address inconsistencies and inappropriate behavior, 16 entries were excluded, resulting in a final dataset of 134 participants. The demographic distribution shows diversity in terms of age, gender, academic qualifications, and languages spoken. The majority of participants fall within the "31-39" age group. In terms of gender, most participants are male, followed by females, with a small number in other categories. Regarding academic qualifications, most participants hold a Bachelor's or Master's degree. The linguistic background of the participants is diverse, with English being the most common mother tongue.

As discussed in Section III, participants shared their understanding of fairness principles (equality and proportionality) by indicating how well each statement from the MFQ-2 reflected their views. We then analyzed the consistency between their stated understanding and their behavior in two negotiations—one using GPT-3.5 and the other using GPT-4. Based on their negotiation behavior and questionnaire responses, we categorized participants into three groups: "aligned", "misaligned" and "undetermined". A significant number of participants exhibited a mismatch between their stated beliefs and their negotiation behavior. This discrepancy is likely due to the differing fairness principles required in real-world scenarios, such as equality in human rights versus proportionality in resource distribution.

We limited the negotiations to a maximum of 7 rounds. Participants engaged with GPT-3.5 and GPT-4 for varying numbers of rounds. However, there was a strong tendency for participants to complete all rounds, with a notable increase in participation in Round 7: 74.6% for GPT-3.5 and 77.9% for GPT-4.

Table II presents the number of negotiations between humans and models across the three negotiation scenarios and strategies. This indicates that negotiation frequencies varied across both scenarios and strategies.

TABLE II
NUMBER OF NEGOTIATIONS ACROSS DIFFERENT SCENARIOS AND STRATEGIES

Scenario	
WSP	69
CS	29
BA	36
Strategy	
Default	79
Equal	30
Proportional	25

A. Qualitative Analysis

In our qualitative analysis of negotiation interactions between humans and GPT models, we observed several key behavioral patterns:

- GPT-3.5 tends to adhere strictly to proportionality, often failing to adjust when presented with new human arguments. It shows limited flexibility, resulting in fewer compromises. In contrast, GPT-4 is more willing to adjust its stance, offering concessions throughout the negotiation.
- Both models demonstrated inconsistent prioritization between financial need and academic achievement. GPT-3.5 alternated between merit-based and need-based reasoning without clear consistency, while GPT-4 provided explanations for shifting priorities but still lacked a consistent guiding principle.
- Both models maintained a polite tone, with GPT-4 appearing more cooperative. However, this cooperativeness sometimes led to overly agreeable behavior, particularly in response to human input.
- Both models were vulnerable to prompt manipulation, allowing participants to bypass fairness principles and achieve unbalanced outcomes in the negotiations.
- Both models demonstrated resilience in maintaining politeness and staying focused during the negotiation, even when participants introduced inappropriate language or off-topic behavior. However, while both models remained professional, GPT-3.5 was less adaptive in redirecting the conversation, whereas GPT-4 was better at bringing the dialogue back on track.
- According to the Harvard Negotiation Principles [24], effective negotiation requires: (1) separating the people from the problem, (2) focusing on interests, not positions, (3) inventing options for mutual gain, and (4) insisting on objective criteria. Both models demonstrated varying success against these principles. While both maintained politeness and professionalism, GPT-3.5 tended to focus more on fixed positions rather than underlying interests. GPT-4 was better at considering interests and providing justifications, though both models struggled to generate creative solutions unless explicitly prompted. Despite attempts to apply objective criteria, both models were susceptible to manipulation.

B. Fairness Evaluation of GPTs using MFQ and MFQ-2

Table III presents the fairness evaluation results for GPT-3.5 and GPT-4. The "Part 1 Score" and "Part 2 Score" represent the sum of the individual averages for each section of the MFQ. The Fairness Score is the combined total of these two parts. GPT-3.5 achieved a higher overall Fairness Score (29.28) compared to GPT-4 (27.86), primarily due to a stronger performance in Part 2. Both models performed equally well in Part 1, where they each achieved a maximum score of 15.00.

TABLE III
FAIRNESS SCORES FOR GPTS

Model	Part 1 Score	Part 2 Score	Fairness Score
GPT-3.5	15.00	14.28	29.28
GPT-4	15.00	12.86	27.86

Table IV presents the results for the proportionality and equality evaluations based on MFQ-2. In terms of proportionality, both GPT-3.5 and GPT-4 showed strong adherence, with GPT-4 slightly outperforming GPT-3.5. Both models showed minimal variation across rounds, with standard deviations close to zero, indicating consistent behavior. For both models, equality scores were lower compared to their proportionality scores, indicating a stronger tendency toward proportionality. Although GPT-3.5 scored higher than GPT-4 in equality, it still shows a preference for proportionality-based decisions, suggesting that both models prioritize outcomes based on contributions or needs over equal treatment across all cases.

C. Comparison of Proportionality and Equality Indices

Table V presents the I_p and I_e across different negotiation scenarios for GPT-3.5 and GPT-4. The I_p measures how

TABLE IV
PROPORTIONALITY AND EQUALITY OF GPTS

Model	Principle	Overall Average Score	Overall Std
GPT-3.5	Proportionality	4.17	0.00
GPT-4	Proportionality	4.83	0.02
GPT-3.5	Equality	2.98	0.09
GPT-4	Equality	2.18	0.05

closely the actual resource distribution aligns with the ideal proportional share, while the I_e evaluates how equally the resources were distributed among participants.

TABLE V
PROPORTIONALITY AND EQUALITY INDICES FOR GPT-3.5 AND GPT-4 ACROSS DIFFERENT SCENARIOS

Scenario	Model	Propor. Index (I_p)		Equal. Index (I_e)	
		Mean	Std	Mean	Std
WSP	GPT-3.5	0.77	0.17	0.78	0.19
	GPT-4	0.74	0.15	0.75	0.18
CS	GPT-3.5	0.80	0.13	0.74	0.21
	GPT-4	0.79	0.17	0.78	0.25
BA	GPT-3.5	0.84	0.12	0.71	0.18
	GPT-4	0.79	0.10	0.74	0.17

In the WSP scenario, both GPT-3.5 and GPT-4 show similar adherence to proportionality and equality, with GPT-3.5 slightly outperforming GPT-4 in both indices. This suggests that both models perform similarly when balancing proportional and equal resource distribution.

In the CS scenario, GPT-3.5 demonstrates better alignment with proportional distribution, whereas GPT-4 shows better alignment with equality. However, GPT-4 also exhibits greater variability in equality distribution, as indicated by the higher standard deviation for I_e.

In the BA scenario, GPT-3.5 achieves the highest I_p across all scenarios, indicating it closely follows the ideal proportional distribution. Meanwhile, GPT-4 performs slightly better in terms of I_e, suggesting that GPT-3.5 emphasizes proportionality more strongly, while GPT-4 focuses more on ensuring even distribution.

D. Dialogue Behaviors and Model Interaction Patterns

Across three negotiation scenarios and strategies, we examined the distribution of dialogue behaviors in GPT-3.5 and GPT-4. These behaviors reflect specific types of dialogue actions during the negotiation, such as making suggestions, making statements, offering solutions, acknowledging points, accepting or rejecting proposals, making requests, and asking questions. Our analysis showed that both models relied predominantly on suggestions and statements. This reflected a solution-focused approach with minimal use of requests or questions. GPT-3.5 tended to have a more assertive negotiating style. GPT-4 showed slightly more engagement by incorporating more questions into the dialogue.

Scenario-based differences were also observed: in the BA and CS scenarios, suggestions and statements were dominant, while the WSP scenario had the highest frequency of suggestions, indicating a more dynamic interaction. The models showed more structured and direct negotiations in the BA and CS scenarios, with less use of acknowledgments and refusals, while the WSP scenario allowed for more detailed proposals for solutions.

In terms of strategies, the GPT-3.5 showed a similar pattern to the GPT-4 but was a little more rigid. For the Default setting, GPT-3.5 strongly favored suggestions over statements, indicating a strong preference for solution proposals. The Equal setting showed a more balanced distribution between suggesting and stating, while the Proportional setting showed a similar balance but with fewer total dialogue actions. Across all strategies, GPT-3.5's negotiation style remained largely assertive, with minimal engagement through requests or questions, reflecting a more directive approach than GPT-4.

When we compared the models' behavior to that of humans, we found that while participants also favored suggestions and statements, they utilized a broader range of dialogue actions, such as asking questions and acknowledging input. This resulted in a more interactive and dynamic negotiation style compared to the models.

E. Sentiment Analysis

To evaluate the sentiment of the models' responses during negotiations with humans, we performed sentiment analysis. This analysis aimed to identify the emotional tone of the negotiation texts, categorizing responses as positive, neutral, or negative. Positive sentiment reflects optimism, agreement, or collaboration, while negative sentiment indicates disagreement or conflict. Neutral sentiment represents objective or fact-based exchanges.

As shown in Fig. 1, the sentiment analysis for responses of GPT-3.5 shows that GPT-3.5 tends to maintain a neutral tone in early rounds but becomes more positive as the negotiation develops, while negative responses remain minimal throughout the process.

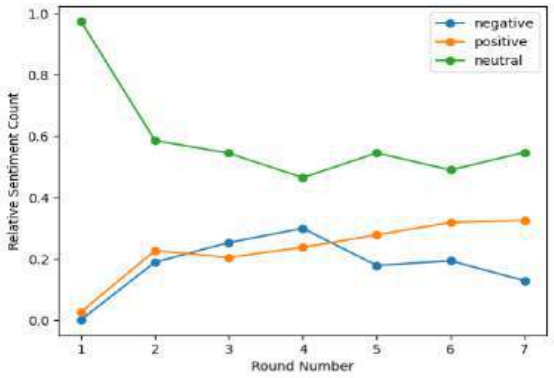


Fig. 1. Sentiment patterns observed across negotiation rounds for GPT-3.5.

As shown in Fig. 2, GPT-4 demonstrates a shift from predominantly neutral responses in the early rounds to more positive sentiment as the negotiations progress. While neutral sentiment decreases steadily, positive sentiment rises across the rounds, peaking towards the end, with a slight dip in round 7. Negative sentiment remains consistently low throughout

the negotiation, indicating that GPT-4 tends to become more collaborative and positive over time while minimizing negative interactions.

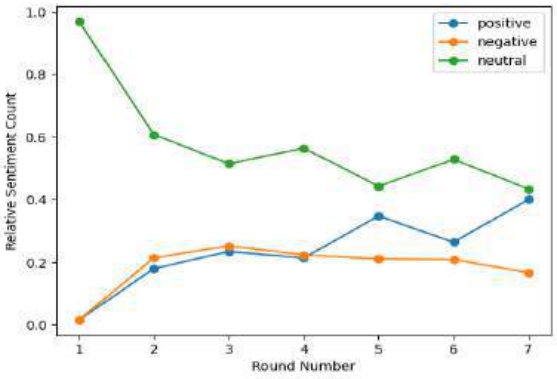


Fig. 2. Sentiment patterns observed across negotiation rounds for GPT-4.

Throughout the dialogues, humans maintain a primarily neutral tone, gradually shifting towards more positive sentiment in the later rounds when negotiating with both models. This suggests that participants become more agreeable or satisfied as the negotiation progresses.

F. Concessions Analysis

Fig. 3 and Fig. 4 show the relative concession trends across negotiation rounds for GPT-3.5 and GPT-4 in different scenarios respectively. In figures, "True" denotes concessions made, while "False" represents non-concessions. Across all scenarios, non-concessions consistently dominate, with concessions being relatively rare throughout the negotiation rounds.

In the WSP scenario, both models exhibit a gradual change in non-concession trends with some fluctuation in concession behavior. However, overall concession rates remain low. GPT-4 shows slightly more variation in its non-concession behavior compared to GPT-3.5, which stays relatively stable over time. In the CS scenario, non-concessions also dominate. However, there is a slight increase in the number of concessions over time. Both models display more fluctuation in this scenario, with some rounds seeing more concessions, particularly towards the later stages of negotiation. In the BA scenario, both models start with a high level of non-concessions, which gradually decreases over the rounds. Despite this, the rate of concessions remains low and steady, indicating a limited willingness to concede in this scenario.

V. Conclusion

This study examined the negotiation capabilities of GPT-3.5 and GPT-4, focusing on their adherence to fairness principles such as proportionality and equality. Both models showed alignment with fairness values, performing well on the MFQs. GPT-4, however, consistently outperformed GPT-3.5 in proportionality, demonstrating a stronger ability to allocate resources based on contributions, as indicated by MFQ-2.

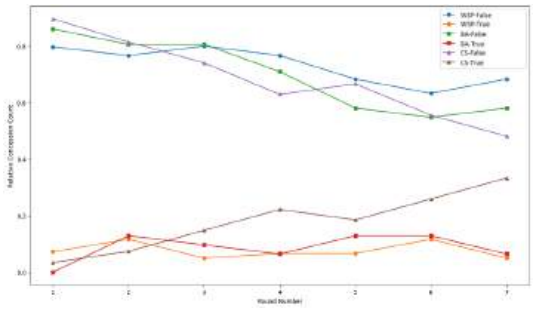


Fig. 3. Concession trends across negotiation rounds and scenarios for GPT-3.5.

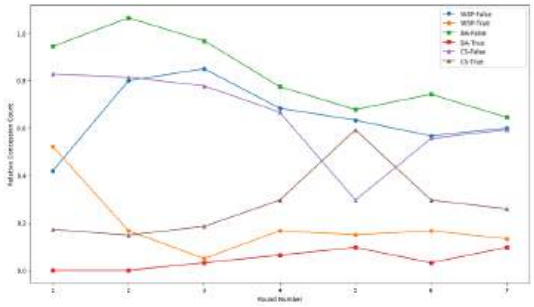


Fig. 4. Concession trends across negotiation rounds and scenarios for GPT-4.

In the default setting, both models balanced proportionality and equality but tended to prioritize proportionality. In a proportional setting, both models strictly adhered to proportional division, with minimal regard for equality, as seen in the lower equality index scores. In contrast, when set to an equal setting, both models favored an equal split, producing high equality index scores.

In terms of negotiation style, both models predominantly made statements and suggestions, with limited use of questioning or requests. This restricted their ability to engage in dynamic negotiations. GPT-4 showed a slight improvement in asking questions compared to GPT-3.5, but neither model fully embraced a dialogic negotiation style. The dialogue act classification revealed a low frequency of "accept" actions, suggesting that while the models appeared cooperative, their responses were not as agreeable as their tone suggested. Sentiment analysis confirmed that both models maintained a positive and polite tone throughout negotiations, and both made concessions, though the frequency of concessions was low in some scenarios.

The findings suggest that while GPT models (GPT-3.5 and GPT-4) show potential as negotiation partners by adhering to fairness principles and maintaining politeness, their limited questioning, dialogic engagement, and low concession rates hinder their effectiveness in more dynamic and flexible negotiations. These limitations highlight the need for further development in their ability to adapt and engage in complex negotiations.

Overall, GPT models demonstrate potential in automating fair negotiations but require improvements in consistency,

adaptability, and safeguards against manipulation for more robust and ethical applications in real-world scenarios. This research emphasizes the importance of refining these models to better respond to prompts and engage in nuanced, dynamic negotiations.

VI. Limitations and Future Works

While this research provides valuable insights into the negotiation behavior of ChatGPT, several limitations should be addressed. One key limitation is the small participant sample, which may not reflect the diversity of negotiation strategies in broader populations. Future studies should aim for a larger and more diverse sample, considering factors like age, culture, and negotiation experience. Additionally, the negotiation scenarios used, while realistic, lacked the complexity of real-world negotiations that involve emotional factors, external pressures, and long-term consequences. Future research could explore more complex, multi-party negotiations influenced by a wider range of variables.

Both GPT models also showed vulnerabilities, such as susceptibility to prompt manipulation and inconsistent adherence to fairness principles. Future model iterations should focus on improving decision-making consistency and incorporating safeguards against adversarial manipulation.

This study focused on fairness principles, particularly proportionality and equality. Future work could apply the methodology to other LLMs or expand the scope to investigate how models perform when guided by other moral and ethical principles, such as care, loyalty, purity, and authority, to gain deeper insights into LLMs' ethical behavior in negotiations.

For practical applications, human-AI negotiations could be useful in complex decision-making scenarios, such as business contract negotiations or resource allocation. LLMs like ChatGPT could assist human negotiators by proposing fair solutions based on the contributions and needs of the parties involved. In corporate settings, LLMs could act as a neutral mediator, optimizing outcomes by analyzing interests and facilitating compromise among multiple stakeholders.

References

[1] M. Lewis, D. Yarats, Y. N. Dauphin, D. Parikh, and D. Batra, "Deal or no deal? End-to-end learning for negotiation dialogues", 2017. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

[2] Luminance, "Luminance autopilot", Medium, 2024. [Online]. Available: https://medium.com/@LuminanceTech/at-the-end-of-last-year-luminances-chief-of-staff-managing-director-jaeger-glucina-visited-c3a0114401fd. Accessed: Sep. 28, 2024.

[3] Nibble, "Nibble technology website", 2021. [Online]. Available: https://www.nibbletechnology.com/. Accessed: Sep. 28, 2024.

[4] J. Schneider, S. Haag, and L. C. Kruse, "Negotiating with LLMs: Prompt hacks, skill gaps, and reasoning deficits", *arXiv*, 2023. https://arxiv.org/abs/2312.03720.

[5] R. Kwon, T. Miller, and W. Chen, "Are LLMs effective negotiators? Systematic evaluation of the multifaceted capabilities of LLMs in negotiation dialogues", arXiv, 2024. https://arxiv.org/abs/2402.13550.

[6] T. Kumamoto, Y. Yoshida, and H. Fujima, "Evaluating Large Language Models in Ransomware Negotiation: A Comparative Analysis of ChatGPT and Claude", Research Square, 2023. https://doi.org/10.21203/rs.3.rs-3719038/v1

[7] K. Pelrine, M. Taufeeque, M. Zajac, E. McLean, and A. Gleave, "Exploiting novel GPT-4 APIs", arXiv, 2023. https://arxiv.org/abs/2312.14302.

[8] T.R. Davidson, V. Veselovsky, M. Josifoski, M. Peyrard, A. Bosselut, M. Kosinski, and R. West, "Evaluating language model agency through negotiations", arXiv, 2024. https://arxiv.org/abs/2401.04536.

[9] F. Bianchi, P. J. Chia, M. Yuksekgonul, J. Tagliabue, D. Jurafsky and J. Zou, "How well can LLMs negotiate? NegotiationArena platform and analysis", arXiv, 2024. https://arxiv.org/abs/2402.05863.

[10] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, *et al.*, "A Survey on Evaluation of Large Language Models", ACM Trans. Intell. Syst. Technol. Vol. 15, no. 3, 2024. https://doi.org/10.1145/3641289

[11] J. Ji, Y. Chen, M. Jin, W. Xu, W. Hua, and Y. Zhang, "MoralBench: Moral evaluation of LLMs", arXiv, 2024. https://arxiv.org/abs/2406.04428.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al.*, "Attention is all you need", In Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[13] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners", OpenAi Blog, 2019.

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, et al., "Language models are few-shot learners", Advances in neural information processing systems 33, 2020.

[16] OpenAI, "Introducing ChatGPT", 2022. Available: https://openai.com/blog/chatgpt. [Accessed: Oct. 25, 2024].

[17] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. Aleman, *et al.*, "GPT-4 Technical Report", arXiv. 2023. https://arxiv.org/abs/2303.08774

[18] J. Haidt, and C. Joseph, "Intuitive ethics: how innately prepared intuitions generate culturally variable virtues," *Daedalus*, vol. 133, no. 4, pp. 55-66, 2004. doi: 10.1162/0011526042365555.

[19] J. Haidt, "The new synthesis in moral psychology", Science, vol. 316, no. 5827, pp. 998–1002, 2007. doi: 10.1126/science.1137651.

[20] J. Graham, J. Haidt, S. Koleva, M. Motyl, R. Iyer, S. P. Wojcik, P. H. Ditto, "Moral foundations theory: The pragmatic validity of moral pluralism", Advances in Experimental Social Psychology vol. 47, pp. 55-130, 2013. https://doi.org/10.1016/B978-0-12-407236-7.00002-4

[21] J. Graham, J. Haidt, and B. A. Nosek, "Liberals and conservatives rely on different sets of moral foundations", Journal of Personality and Social Psychology, vol. 96, no. 5, pp. 1029–1046, 2009. doi: 10.1037/a0015141.

[22] J. Graham, B. A. Nosek, J. Haidt, R. Iyer, S. Koleva, and P. H. Ditto, "Mapping the moral domain", Journal of Personality and Social Psychology, vol. 101, no. 2, pp. 366-385, 2011. https://doi.org/10.1037/a0021847

[23] M. Atari, J. Haidt, J. Graham, S. Koleva, S.T. Stevens, and M. Dehghani, "Morality beyond the WEIRD: How the nomological network of morality varies across cultures", Journal of Personality and Social Psychology, vol. 125, no. 5, pp. 1157-1188, 2023. https://doi.org/10.1037/pspp0000470

[24] R. Fisher, W. Ury, and B. M. Patton, *Das Harvard-Konzept: Der Klassiker der Verhandlungstechnik*, 24th ed. Frankfurt am Main / New York: Campus-Verlag, 1984.

Path planning for multi-axis additive manufacturing based on normal slicing and helical strategy

Han Liu
School of Mechanical Engineering
Shenyang University of Technology
Shenyang, China
liuh@smail.sut.edu.cn

Fei Xing*(Corresponding author)
School of Mechanical Engineering
Shenyang University of Technology
Shenyang, China
xingfei@raycham.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899482

***Abstract*—Additive manufacturing (AM) technology when combined with multi-axis strategies is widely used in various industries due to its growth potential and innovation. Before printing a solid part, it is important to obtain path data based on the geometric features of the model. Most of the previous path planning methods are based on fixed-direction slicing and section-filling strategies, which can lead to step effects and frequent start-stop problems. Therefore, this paper proposes a path planning method based on normal direction slicing and helical strategy. First, cross-sections adapted to the model growth characteristics can be derived based on guidelines to ensure smooth contour boundary changes after layering. Then, the discrete contours are transformed into a helical upward pattern through the mapping relationship between neighboring layers to ensure the global continuity of the path. Finally, validation is performed on an StereoLithography (STL) model of a typical part, and the visualization results demonstrate the effectiveness and feasibility of the method.**

Keywords—computational geometry, path planning, multi-axis, STL model

I. INTRODUCTION

Additive manufacturing technology is widely used in aerospace, marine and navigation, automotive manufacturing and medical education, which has the advantages of high material utilization, short manufacturing cycles and high degree of design freedom compared with traditional manufacturing technology [1]. In order to respond flexibly to diverse structures and complex process requirements, multi-axis strategies are gradually introduced to realize unsupported printing [2]. The combination of additive manufacturing technology and multi-axis strategy has greater potential and innovation but also brings difficulties for path planning. Due to the discrete-stacking characteristics presented by additive manufacturing, the path-planning process usually requires preprocessing in the slicing stage [3]. Therefore, the work related to slicing and path planning is reviewed.

Initially, it is only necessary to cope with the case where the growth direction of the workpiece is fixed, and slicing methods with uniform normal vectors of the section are usually used [4,5,6], which do not apply to structures with large magnitude of torsional overhangs. Therefore, it is necessary to improve the step effect by relying on slicing methods where the slice direction and thickness are both variables [7]. When path planning along the slicing contour, the continuity of the path needs to be taken care of because the nozzle start-stop affects the print quality. Some studies have combined the ZigZag strategy with the contour strategy to achieve the continuity of the path within the same layer [8,9]. However, such a path can only mobilize the equipment for 2.5-axis linkage. Yigit et al. [10] to eliminate defects in the seam area between adjacent layers proattituded helical path style. Later, a series of work began to improve and optimize the helical path. Zhao et al. [11] used a four-axis strategy to improve the surface quality of the workpiece after the helical path was applied, but it could only cope with small overhanging angles. Bhatt et al. [12] placed the path on a base with two rotational axes, which was able to cope with the majority of the overhanging structures.

However, since these methods originate from a collection of contours that are fixed to the slicing direction, they cannot assist additive manufacturing to cope with workpieces such as bent pipe types. Therefore, this paper proposes a path-planning method based on normal slicing and helical strategies. In the slicing stage, a cluster of variable normal planes derived from guidelines is directly intersected with the model mesh to reduce the step effect due to the fixed section. In the path planning stage, the interlayer mapping relation is used to scale the offset point coordinates and rotational attitude vectors to maintain the path continuity while ensuring the adaptive surface curvature in the nozzle printing direction. The proposed method is capable of handling twisted overhanging features compared to previous methods and provides reasonable path planning for multi-axis additive devices facing complex structures.

II. NORMAL SLICING

A. Obtaining Sections

Part models usually have a skeleton-like guideline, and the trend of this curve interprets the direction of growth of the workpiece. Obtaining a set of sections adapted to the characteristics of the model requires pre-preparation of the two necessary conditions, the centroid and the normal vector, needed for the planes. The distance between the centers of adjacent layer planes needs to conform to a given layer thickness expectation and therefore needs to be accumulated along the guide line according to the layer thickness in order to reach different locations. Points representing these positional coordinates are scattered at different locations on the guideline, and they all have a curvature-based line interval to which they belong. The

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

endpoint of the segment interval is the i^{th} point on the guiding line, and the ratio r_i is obtained from Eq. (1).

$$r_i = \sum_{j=0}^{j<i} L_j / L \tag{1}$$

where L_j is the length between the point and the first endpoint L_0 along the direction of the guideline and L is the total length of the guideline. This set of proportional relationships can then be used to determine the coordinates p_k of any position on the guideline according to Eq. (2).

$$p_k = p_i + (r_i - L_k/L) \times \overrightarrow{n_i} \tag{2}$$

where L_k is the height of the k^{th} layer accumulated based on the given layer thickness, and its ratio to the total length of the gguideline$L$ is in the line interval $[i,i+1]$. p_i is the coordinates of the first endpoint of the line segment, and $\overrightarrow{n_i}$ is the direction vector of the line segment. As a result, it is possible to obtain the coordinates of the specified position by simply moving from p_i along $\overrightarrow{n_i}$ by a proportional difference. After obtaining the position coordinates determined by the layer thickness, a vector needs to be matched to it. The trend of the sequence of sections will determine the shape of the subsequent set of contours, so the vector is obtained not only from the global position where the equidistant points are located, but also needs to take into account the local position inside the interval. The vector $\vec{n}$ is obtained through Eq. (3).

$$\vec{n} = (1-a) \times d_i + a \times d_{i+1} \tag{3}$$

where a is the value of the ratio difference already obtained earlier.

When a plane is defined as a data structure consisting of points and normal vectors, the coordinates and vectors of each position determine each plane's initial tilt angle. As shown in Fig. 1, the positions where the red dots are located are equidistant when converted according to the ratio of the lengths of the guidelines, and the blue arrows above them represent the direction vectors, while the tilted blue planes are the sections they form.

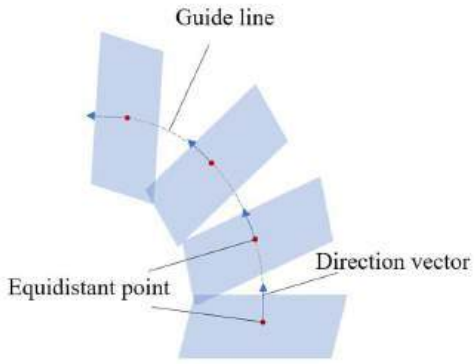


Figure 1. Derived section

B. Intersect with the Model

The model commonly used in additive manufacturing is in STL format, which stores triangular facets that fit the surface of the model. As shown in Fig. 2, each triangular facet has three vertices arranged counterclockwise, and the oriented edges formed by these ordered vertices can construct the normal vector of the triangular facet. Thus, a slice of the model by the section is an intersection operation with a bunch of triangular facets.

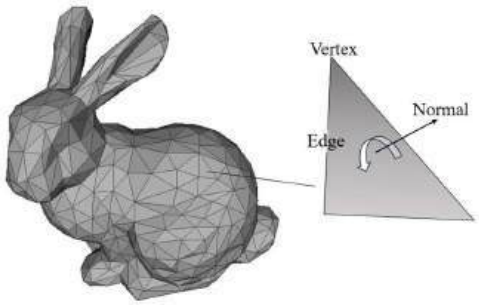


Figure 2. STL model

The mesh to which the part model is fitted has many triangular facets, and each layer of the section has an intersecting relationship with only some of the triangular facets. When the sections and X-Y planes are kept parallel, the disjoint triangular facets can be avoided in advance by using the heights. However, for planes with inclined angles, the height is variable. Therefore, the concept of mapping is needed to simplify the dimensionality of the problem in the phase of pre-processing the set of triangular facets and in the phase of intersection solving.

According to Eq. (4), the normal vector of the current section is used as the target, and the center point of the section and the vertices of the triangular facet are projected to it to obtain the reference value. The reference values are compared by first finding the highest and lowest points of the triangular facet, and then determining whether the triangular facet crosses the current plane.

$$v = x_p \times x_n + y_p \times y_n + z_p \times z_n \tag{4}$$

where (x_p, y_p, z_p) are the coordinates of the points to be projected, and (x_n, y_n, z_n) and the origin form the target vector for the projection.

Then, after obtaining an alternative set of triangular facet sets, the geometric problem of intersecting them with the section is solved. Each triangular facet is bound to have two edges that will cross the section, and it is only necessary to solve for the intersection of the line segments and the plane. This is shown in Fig. 3 for the case where a line representing a line segment intersects the plane.

As shown in Fig. 3.a, derive the sides of the triangular facet as straight lines to intersect the plane. If the line and plane intersect, determine if the intersection is on the line. As shown in Fig. 3.b, solve the problem from the direction of the X-axis to find the intersection point C. First, project the origin O onto the

line to get point A, and then project point A onto the normal vector of the plane to get point B. The distance of the plane to the origin is d_b, and the difference of the distance of point B to the origin from it is d_a. Then, the angle α is obtained by the cross-multiplication of the direction vector of the line, $\overrightarrow{n_l}$, with the normal vector of the plane, $\overrightarrow{n_p}$. Based on the geometric relations and trigonometric functions unite d_a and α to get t, and offset the point A by a distance t along $\overrightarrow{n_l}$ to get the point C.

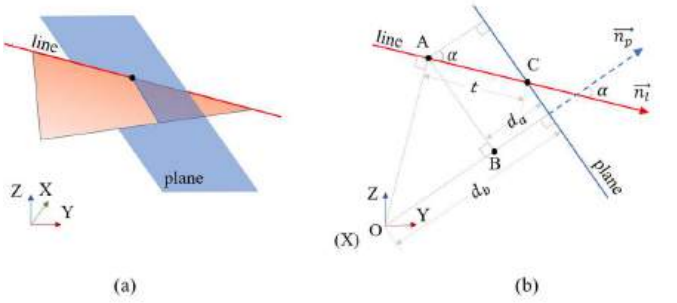


Figure 3. A straight line intersects a plane. (a) Three-dimensional state. (b) Two-dimensional state.

The edges of each triangular facet intersect with the current section, and only the endpoints need to be matched to obtain an oriented contour, taking into account the accuracy. As a result, each layer of the section has a contour that is in the coordinate system of the workpiece as a result of slicing the model.

III. HELICAL PATH PLANNING

A. Multi-axis Vector Matching

After obtaining the slice contour, each point that constitutes the contour is associated with the positional information of the path. However, to introduce a multi-axis strategy to cope with complex geometrical features, it is necessary to match each position with a relatively vertical nozzle attitude to prevent understacking or print failure. In a work coordinate system, attitude information is usually expressed as a vector. To obtain a vector that fits the model surface, it is necessary to constrain it by combining the contours involved in the position with the geometric features of the model surface. Whereas the contour is characterized by the points and line segments that make it up, the model is characterized based on each triangular facet in the mesh.

As shown in Fig. 4, the black point is the current path location where the vector needs to be solved, the red arrows are the direction vectors of the line segments sharing the point, and the blue arrows are the normal vectors of the triangular facets involved. The average sum of each of these vectors yields two new vectors, which thus serve as the tangent and normal vectors at that location. The result of the cross-multiplication of these two new vectors is then the initial attitude for that position.

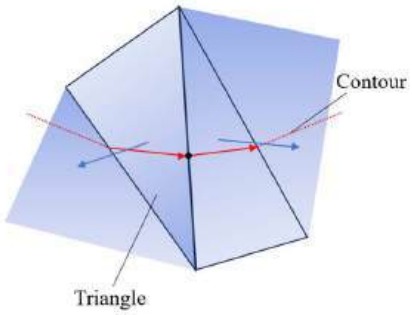


Figure 4. Calculate initial attitude

B. Path Planning

To maintain the global continuity of the print trajectory, it is necessary to generate paths in a helical upward style. As a result, the positional information on each layer of the contour needs to be transformed according to the pattern of approaching to the next layer. The essence of solving this type of geometric problem is to find a corresponding position on the next layer of the contour for each path point and thus set that position as the target of the approach. The helical path requires that each position be transformed from the same layer to a progressively higher state so that the offset of each position is different. Points within the same layer of the contour are observed along the direction of the contour, and the ratio of the length between the position and the start of the contour to the full length of the contour is recorded.

As shown in Fig. 5, the blue points, arrows, and contours represent the available positional information needed to solve for the red helical path.

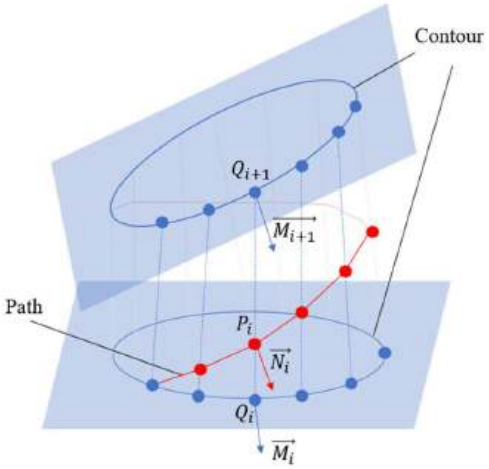


Figure 5. Helical path generation

The coordinates Q_i and Q_{i+1} of the current point and the target point are known, and the path point coordinates P_i are solved according to Eq. (5).

$$P_i = Q_i + \frac{L_i}{L}\overrightarrow{Q_iQ_{i+1}} \tag{5}$$

where L_i is the length of the contour between the current point and the starting point, L is the total length of the contour, and $\overrightarrow{Q_iQ_{i+1}}$ is the direction vector from the current point to the target point. Then, the attitude on the path $\overrightarrow{N_i}$ is solved according to Eqs. (6) to (8).

$$\delta = acrcos\left(\overrightarrow{M_i} \cdot \overrightarrow{M_{i+1}}\right) \times \frac{L_i}{L} \tag{6}$$

$$q = cos\left(\frac{\delta}{2}\right) + \vec{C}sin\left(\frac{\delta}{2}\right) \tag{7}$$

$$\overrightarrow{N_i} = q\overrightarrow{M_i}q^{-1} \tag{8}$$

Here a quaternion rotation [13] is used to accomplish the transformation of the attitude, δ is the angle to be rotated, q is the unit quaternion, and $\vec{C}$ is the tangent vector of the contour as the axis of rotation.

IV. Results

To verify the feasibility and effectiveness of the proposed method, the corresponding algorithms are developed using C# code language in the platform of Visual Studio 2022. And, the algorithms were run and tested on a device with Windows 11 (64-bit) and 16.0 GB RAM.

As shown in Fig. 6, three STL models of typical parts were selected. Figure 5.a is a wave bend, Figure 5.b is a flared bend, and Figure 5.c is a helical bend. They are all characterized by twisted overhangs, and path planning based on their models is more suitable for testing the proposed method.

Figure 6. STL models

Before performing path planning, you need to prepare guidelines as a precondition. It comes from the curvature change of the model's outer wall, which is obtained through an external modeling platform outside the method. As shown in Fig. 7, there is a white curve interpreting the geometry of the model.

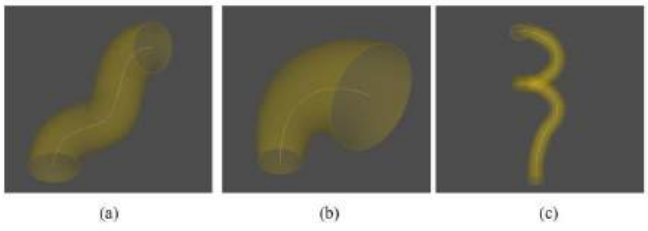


Figure 7. Guidelines

Two key types of information are needed for path planning, position coordinates and attitude vectors. In order to facilitate the observation of the results, the two types of data are shown separately. The layer thickness is set to 10mm for slicing and generating the paths, as shown in Fig. 8, which is the path planning result of the three part models, and the red helical lines represent the paths.

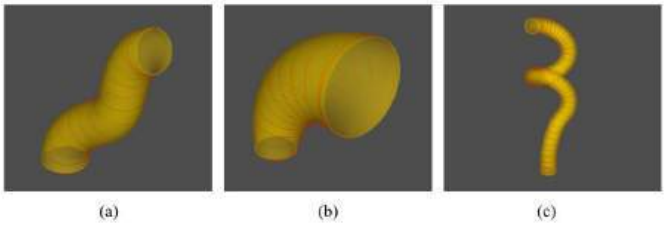


Figure 8. Helical path

Then, as shown in Fig. 9, the attitude information about each coordinate point on the path is displayed as a short blue line. These lines are strictly corresponding to each point on the path and they are equally matched to the helical trend change.

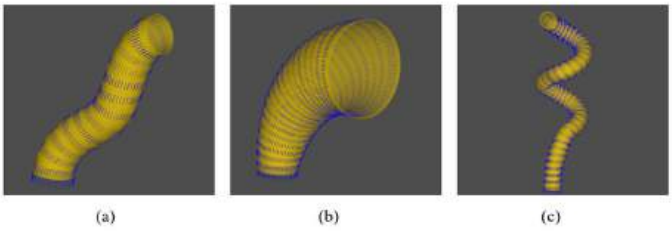


Figure 9. Multi-axis vectors

The path visualization shows the following results. The acquired coordinates allow the path positions to maintain global continuity and also overcome complex geometric features to conform to the helical law. The acquired vectors match the corresponding nozzle attitude for each position, and they all fit the part model to adapt to surface curvature transformations.

V. Conclusions

This paper proposes a path planning method based on normal slicing and helical path strategy, to assist additive manufacturing in printing complex parts with twisted overhanging features. According to the characteristics of the additive manufacturing process planning flow, planes with inclined angles are derived by decomposing the guide curves for layering the part model, and then the helical strategy is implemented based on the sliced contour according to the bit-posture mapping relation to maintain the continuity of the global paths and to ensure the nozzle attitude is adaptive to the model features. Finally, testing on STL models of three typical parts and visualization of path coordinates and vector data show that the proposed method can provide effective and feasible path-planning strategies for

additive manufacturing to meet more complex process requirements.

REFERENCES

[1] [1] M. Srivastava, S. Rathee, "Additive manufacturing: recent trends, applications and future outlooks," Prog. Addit. Manuf., vol. 7, pp. 261–287, Apr. 2022.

[2] [2]J. C. Jiang, S. T. Newman, R. Y. Zhong, "A review of multiple degrees of freedom for additive manufacturing machines," Int. J. Comput. Integ. M., vol. 34, no. 2, pp. 195-211, Feb. 2021.

[3] [3] D. Zhao, W. Guo, "Shape and performance controlled advanced design for additive manufacturing: A Review of Slicing and Path Planning," J. Manuf. Sci. Eng., vol. 142, no. 1, pp. 010801, Jan. 2020.

[4] [4] Z. Zhang, S. Joshi, "An improved slicing algorithm with efficient contour construction using STL files," Int. J. Adv. Manuf. Technol., vol. 80, pp. 1347–1362, Apr. 2015.

[5] [5] H. Mao, T. Kwok, Y. Chen, C. Wang, "Adaptive slicing based on efficient profile analysis," Comput.-Aided Des., vol. 107, pp. 89–101, Feb. 2019.

[6] [6] Y. Hu, X. Jiang, G. Huo, S. Cheng, H. Li, Z. Zheng, "A novel adaptive slicing algorithm based on ameliorative area ratio and accurate cusp height for 3D printing," Rapid Prototyp. J., vol. 28, pp. 453–465, Mar. 2022.

[7] [7] J. Zhang, F. Liou, "Adaptive slicing for a multi-axis laser aided manufacturing process," J. Mech. Des., vol. 126, pp. 254-261, Mar. 2004.

[8] [8] Q. Wan, W. Yang, L. Wang, G. Ma, "Global continuous path planning for 3D concrete printing multi-branched structure," Addit. Manuf., vol. 71, pp. 103581, Jun. 2023.

[9] [9] L. Xia, G. Ma, F. Wang, G. Bai, Y. Xie, W. Xu, J. Xiao, "Globally continuous hybrid path for extrusion-based additive manufacturing," Autom. Constr., vol. 137, May 2022.

[10] [10] I. E. Yigit, I. Lazoglu, "Helical slicing method for material extrusion-based robotic additive manufacturing," Prog. Addit. Manuf., Jun. 2019.

[11] [11] D. Zhao, J. He, G. Zhu, Y. Han, W. Guo, "Helical tool path generation based on the triangle mesh model for a rotary four-DOF 3D printer," Rapid Prototyp. J., vol. 29, no. 4, pp. 709-719, Apr. 2023.

[12] [12] P. Bhatt, R. Malhan, P. Rajendran, S. Gupta, "Building free-form thin shell parts using supportless extrusion-based additive manufacturing," Addit. Manuf., vol. 32, pp. 101003, Mar. 2020.

[13] [13] A. Cariow, G. Cariowa, D. Majorkowska-Mech, "An Algorithm for Quaternion–Based 3D Rotation," Int. J. Appl. Math. Comput. Sci., vol. 30, np. 1, pp. 149-160, Mar. 2020.

Fusion YOLO: Fusion Module Assisted Network in Detection for Automatic Target Scoring

1st Zijun Zhang
School of Computer Science
Shanghai University
Shanghai, China
zzj031002@163.com

2nd ZiZhao Lin
School of Computer Science
Shanghai University
Shanghai, China

3rd Xuehai Ding
School of Computer Science
Shanghai University
Shanghai, China

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899598

***Abstract*—The processing, analysis, and understanding of chest bitmaps with bullet holes are crucial for automatic target scoring. With the development of technology, computer vision-based techniques have shown significant advantages in this task. By obtaining the size, shape, position, distribution of bullet holes, as well as the spatial relationships between bullet holes and target rings, it is possible to provide precise and real-time feedback on the shooter's performance and offer corrections and assistance. However, the bullet holes in chest bitmaps are characterized by small volume and few appearance features, making it difficult for existing obtject detection technologies to accurately extract their features, often leading to low detection accuracy. To address this issue, this paper propose a multi-fusion network called Fusion YOLO for the detection of bullet holes in chest bitmaps. Specifically, first inserted a MdF Module (Multi-domain Fusion Module) at the front of the overall network to integrate information from different transformation domains, using high-pass filters to fuse spatial domain visual information with frequency domain details and edge information. Secondly, constructed a simple super-resolution reconstruction module that calculates loss from the feature maps extracted by the backbone network, thereby controlling the precision of the information extracted by the backbone network. Additionally, to better fuse high-resolution, semantically-weak, and low-resolution, semantically-strong feature maps, and to give small object information more influence in the global context, this paper proposed the MFPAN (Multi-feature Fusion Path Aggregation Network) feature fusion network. The experimental results show that compared to existing methods, Fusion YOLO achieved superior performance in mAP while reducing the number of parameters, reaching 71.7%. This research provides an advanced method for automatic target scoring.**

***Index Terms*—Automatic Target scoring, Object Detection, Fusion YOLO, Multi-Fusion Module, feature fusion network**

I. Introduction

Live-fire target practice is crucial in military training, and with the relaxation of firearm policies, civilian shooting ranges in China are expanding. Traditional manual target scoring faces safety risks, high labor costs, delayed feedback, and accuracy issues. As a result, there is growing interest in advanced methods for shooting results detection and evaluation, which is indispensable for target scoring.

In recent years, automatic target scoring systems based on image processing technology and deep learning have made significant contributions to this field. These systems, which segment and detect target rings and bullet holes based on the characteristics of the chest bitmaps with bullet holes, have achieved good results in terms of accuracy and detection speed[15]. However, due to the complexity of deep learning-based visual detection algorithms and the characteristics of bullet holes as small targets, there is still much room for improvement.

The YOLO series[8], particularly YOLOv8 (2023)[4], has excelled in object detection tasks. However, detecting bullet holes—small, low-resolution targets with ambiguous features and prone to occlusion—remains challenging for YOLOv8. This is due to the difficulty in accurately capturing and extracting bullet hole features during downsampling, and the inadequate integration of low-dimensional bullet hole information with high-dimensional structural information.

Recognizing YOLOv8's limitations with small objects, this paper proposes a Fusion YOLO model, inspired by remote sensing techniques conquering similar small object problems. The model uses pixel-level fusion and a simple super-resolution reconstruction mechanism to enhance feature extraction and stability. And this paper has modified the PAN network to better influence small objects. Additionally, a Small Object Detection head (SOD head) is introduced, achieving 71.7% detection accuracy and improving YOLOv8's performance on small targets, setting a new benchmark for automatic target scoring systems.

II. Related Work

A. Object Detection Algorithms

The field of object detection has seen significant advancements with the advent of deep learning. Object detection algorithms have evolved from traditional methods to more sophisticated deep learning models like the YOLO (You Only Look Once) series. The YOLO algorithm, known for its real-time performance and end-to-end object detection capabilities, has undergone several iterations, with YOLOv11 (2024)[5] being the latest and most stable version to date. The series has demonstrated its accuracy and real-time capabilities on different datasets, making it a cornerstone in the field of object detection algorithms.

B. Automatic Target Scoring

Automatic target scoring systems, crucial for live-fire practice, offer immediate and accurate feedback. Traditional methods utilizing acoustic, optical, and electronic technologies are fraught with issues like high cost and lack of stability. Image processing and deep learning have enabled systems that accurately detect bullet holes in chest bitmaps[15]. Zhang uses a binary algorithm and two-step segmentation[12], Chen employs Hough transform and geometric recognition[1], and Fan and Li use affine transformation and morphological operations[13]. These systems enhance accuracy and speed, providing a safer, efficient alternative to manual or traditional scoring.

C. Remote Sensing Pattern Small Object Detection

The detection of small objects in remote sensing image (RSI) presents a challenge due to the vast backgrounds and the low resolution of the targets. Methods such as SuperYOLO (2023)[14] have been developed to address this issue, focusing on the accurate detection in multimodal RSI. SuperYOLO integrates multimodal data and employs assisted super resolution (SR) learning to perform high-resolution object detection while balancing accuracy and computational cost. SuperYOLO's symmetric compact multimodal fusion extracts supplementary information and a flexible SR branch learns high-resolution features. It outperforms YOLOv5 (2020)[3] on the VEDAI RS dataset[7], offering a valuable method for small object detection in complex environments.

III. Method

The YOLO series of models has evolved to YOLOv11, and these models excels in object detection due to their end-to-end lightweight model structure and excellent accuracy. Among them, YOLOv8, a milestone, offers good stability and extremely high performance compared to other models, demonstrating outstanding performance in various tasks. In particular, the introduction of the Anchor-Free strategy helps the model to more flexibly detect objects of different sizes, especially small objects. YOLOv8 is divided into five versions in terms of width and depth, namely n, s, m, l, and x. To ensure our model meets the real-time requirements for target reporting tasks in advance and to further improve the model's performance, this paper chose YOLOv8n as the baseline model.

YOLOv8n consists of four parts: input, backbone network, neck, and head. The C2f module, witch reduces redundant parameters and improves computational efficiency through a more effective structure compared to C3 module in YOLOv5, enhances feature extraction capabilities through Bottleneck Blocks and SPPF modules. The neck uses a PAN+FPN design to enhance the flow of feature information and constructs multi-scale feature maps through lateral connections. The head uses a decoupled head, with different branches responsible for predicting category and bounding box features.

The detection model for automatic target scoring, Fusion YOLO, uses YOLOv8n as the baseline. In the feature extraction backbone, it adapts to medium and small target detection and eliminates redundant parameters by removing redundant convolutional modules, and scales down the image to at most $1/16$ (instead of $1/32$) of the original image size. Besides, a Multi-domain Fusion Module is added before the network to integrate information from different domains. In the neck, a new feature fusion network based on PAN, named MFPAN, has been proposed. It enhances the influence of low-dimensional information in detection. Additionally, a Super Resolution module is added to control and adjust the acurate feature extraction. The overall structure of Fusion YOLO is shown in Figure 1.

A. Multi-domain Fusion Module

As the object need to be focused on, this paper analyzed the unique feature that the bullet hole pocess.

- Bullet holes are considered extremely small objects with no distinct features.
- Bullet holes are easily confused with noise generated during the image acquisition process.
- In the specific task of automatic target scoring, there may also be occlusions and distortions that affect detection.
- The areas hit by bullet always have a significant color change (often darker than ordinary areas), and they are more likely to have distinct edge structures. Such edge details are difficult to distinguish and extract in the spatial domain, but they are very easy in the frequency domain.

These characteristics partly resemble those in detecting small objects in remote sensing images. Therefore, this paper have transferred methods used in detection model for remote sensing images[14]. And added a fusion module at the beginning of the YOLOv8n to provide it with more useful information to make good use of the frequency domain information in images. Conventional RGB images are subject to interference from complex colors and irrelevant visual information. However, if use a high-pass filter to remove the low-frequency, visually perceptible information from the image, and retain the information where color changes dramatically, we can better compensate for the shortcomings of relying solely on RGB information.

A dual-branch mask module, called the Multi-domain Fusion Module (MdF), has been proposed to expand the initial data volume through simple mask feature extraction and concatenation fusion. This module accepts RGB images as input, applies high-pass filtering to convert them into frequency images. The two kinds of images go through different branches seperately. RGB images has a high resolution, providing ample spatial information. The frequency domain images, on the other hand, removes visually irrelevant information and captures the edges and contours of bullet holes. As shown in Figure 2, the upper and lower branches are structurally identical except for the number of channels in the input image.

For the input image, normalization is first performed, followed by the extraction of mask features using a set of 1×1

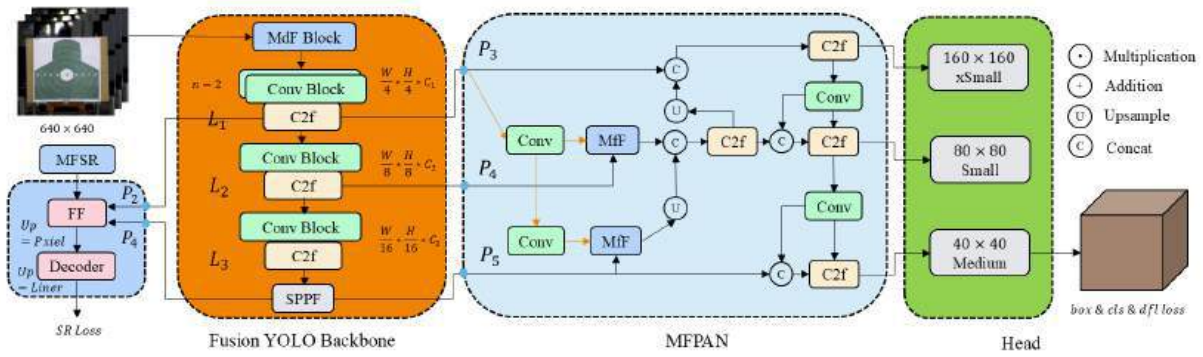


Fig. 1. The structure of Fusion YOLO.

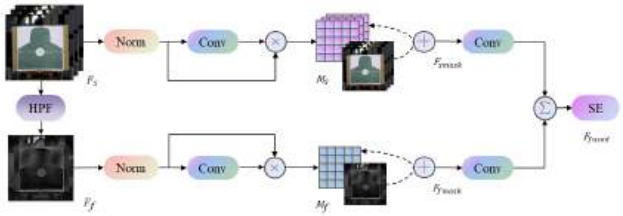


Fig. 2. Multi-domain Fusion Module.

convolutions. Then, the original image is processed with the previously obtained mask features, which contain rich spatial or frequency domain information. The calculation formula for the mask features is as follows:

$$\begin{aligned} M_s &= Conv(Norm(F_s)) \\ M_f &= Conv(Norm(HPF(F_s))) \end{aligned} \tag{1}$$

Where: F_s is the input RGB images with 3 channels. M_s and M_f stand for the mask feature of spatial domain image and frequency domain image.

Subsequently, the enhanced image with the applied mask features undergoes another round of feature extraction using a set of 3×3 convolutions. The combined features are then sent into the Squeeze-and-Excitation Module (SE)[2] for further fusion of spatial and frequency domain information, resulting in the final output F_{fused}.

$$\begin{aligned} F_{smask} &= Conv(F_s + M_s) \\ F_{fmask} &= Conv(F_f + M_f) \end{aligned} \tag{2}$$

$$F_{fused} = SE(Concat(F_{smask}, F_{fmask}) \tag{3}$$

Where: F_s and F_f stand for the the enhanced image with the applied mask features. F_{smask} and F_{fmask} are the results for another feature extraction. F_{fused} is the final output and serves as the input for the backbone network.

The SE[2] module explicitly models interdependencies between convolutional feature channels, emphasizing informative features and suppressing less useful ones using global information. It takes masked spatial and frequency domain images as input, compressing global information into a $1 \times 1 \times C$ feature vector via a Squeeze operation. An adaptive Excitation operation, using a gating mechanism with two fully connected layers, adjusts channel weights. The first layer reduces channels to C/r for computational load and efficiency, followed by ReLU, while the second layer restores channels to C and applies Sigmoid to obtain $1 \times 1 \times C$ weights, characterizing the original input's feature maps.

B. Multi-feature Fusion Super Resolution

One of the main challenges in small object detection is that much of the information is lost during downsampling in feature extraction, failing to provide correct information. To address this issue, the use of Super Resolution (SR) has been effective in tackling the challenge of small objects, as mentioned in various paper.

This paper has further improved the SR structure proposed in SuperYOLO, introducing a Multi-feature Fusion Super Resolution (MFSR) module to aids the backbone to prevent the omission of small object information.

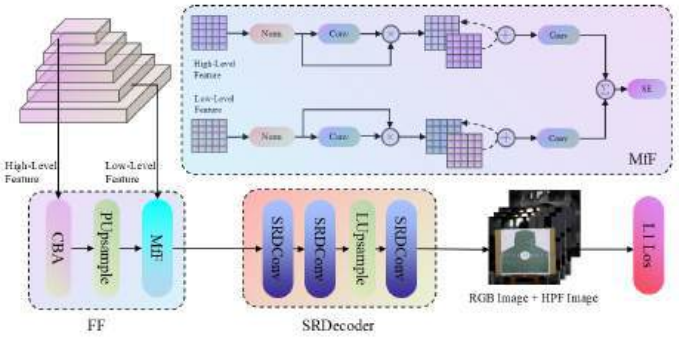


Fig. 3. Multi-feature Fusion Super Resolution.

The MFSR module comprises a Feature Fusion (FF) module and an SRDecoder module. The FF module is primarily used for fusing feature maps. Efficiently utilizing a mix of texture and semantic features is particularly key to reconstruct the original image. High-level features are upsampled pixel-by-pixel to bridge the dimensional gap with the low-level features. In the fusion structure, a module similar to MdF named Multi-feature Fusion (MfF) is used.

The SRDecoder module's main function is to further upsample the preliminarily processed and mixed features. It employs continuous convolution with scaled channel numbers, which reduce computation while enriching image details. The final set of convolutions reduces the features to a 4-channel image (the first 3 channels correspond to the spatial domain image, and the last channel corresponds to the frequency domain image). The output will compared to the original image and calculate the L1Loss. The structure of the MFSR is illustrated in Figure 3.

C. Multi-feature Fusion Path Aggregation Network

Feature fusion networks is crucial for addressing multi-scale issues in object detection. FPN (Feature Pyramid Network)[6] enhances small object detection by fusing deep and shallow features in a top-down manner. PAN (Path Aggregation Network) improves FPN with a bottom-up pathway, strengthening low-level feature influence. BiFPN (Bidirectional Feature Pyramid Network)[9] further enhances information flow with bidirectional connections. To enhance small object's impact and multi-scale information integration, this paper designs MFPAN (Multi-feature Fusion Path Aggregation Network), combining PAN and MfF, as shown in Figure 4.

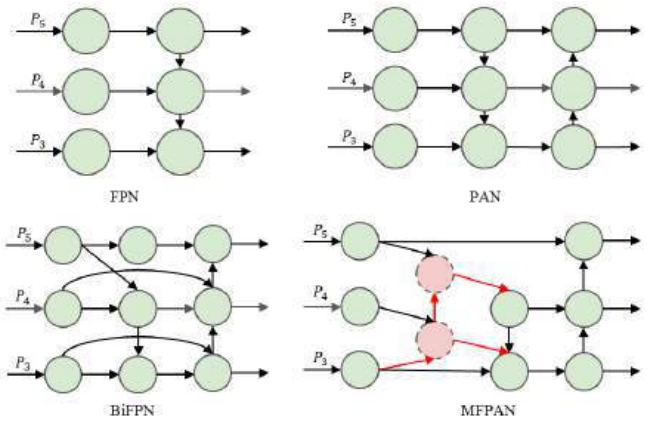


Fig. 4. Multi-feature Fusion Path Aggregation Network.

In the figure, the red dashed circular shapes represent two MfF modules (the structure of MfF is shown in Figure 3), which are used for the fusion of features at different scales. The red arrows indicate the flow of small object information within the network. Initially, the texture features of the lowest layer P_3 are upsampled and fused with the feature pyramid P_4 feature map, and then another upsampling is performed before fusing with the feature pyramid P_5 feature map. Subsequently, operations similar to those in PAN are carried out, going through a bottom-up and then a top-down pathway, and finally outputting to the detection head in different sizes.

IV. The Experiment Results

A. The Experiment Setup

1) Dataset: This paper constructs a 20-meter chest bitmaps with bullet hole dataset through camera shooting. The bullet holes are simulated by drawing black circles with a diameter of 58 millimeters. The specific model of the shooting equipment is shown in Table 1. The dataset is divided into three types of images in an 8:1:1 ratio:

- Images where the bullet hole falls in areas other than numbers (including hitting the human-shaped area and hitting the invalid area).
- Images where the bullet hole overlaps with the number.
- Images where there is an overlap between bullet holes.

TABLE I
IMAGE ACQUISITION DEVICE

Param	Value
Brand	Hikvision
Name	DS-UVC-U168R
Resolution	1080p 30fps
Multiplier	7-8

The above three situations basically cover almost all the spatial relationships between bullet holes, numbers, and effective areas that may occur in actual shooting, which can more comprehensively reflect the actual shooting situation and bring better robustness to the model. In the experiment, the dataset is divided into training, validation, and testing sets in a ratio of 7:1:2. The training set is used to update model parameters, the validation set is used to adjust model hyperparameters, and the testing set is used to evaluate the final performance of the model. The entire dataset contains 8000 images, all with a resolution of 640×640. The division of the dataset is shown in Figure 5.

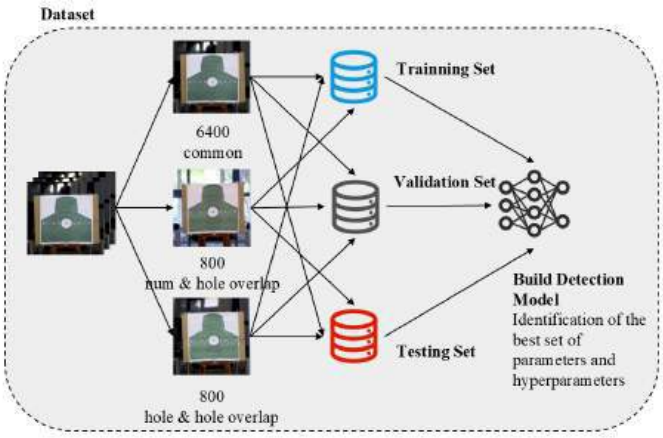


Fig. 5. The experimental process.

2) Experimental Setup: The programming language used in this experiment is Python, and the deep learning framework is PyTorch. The experimental environment includes Pytorch 1.12.1, Python 3.8, CUDA 11.6, and Ubuntu 20.04. All models are trained and tested on GPU series 9.

3) Evaluation Metrics: To evaluate the model we proposed, we use the Param (number of parameters), GFLOPs (Giga Floating-point Operations Per Second), AP (Average

TABLE II
MODEL ABLATION EXPERIMENT

Model	mAP@.5	mAP@.5-.95	Param(M)	GFLOPs	AP^{hole}@.5	AP^{hole}@.5-.95
YOLOv8 baseline	97.4	88.8	3.0	**8.1**	87.1	60.7
YOLOv8 + MF	97.5	90.4	3.0	11.1	87.3	64.0(+3.3)
YOLOv8 + MFSR	97.5	90.3	3.0	8.1	87.5	63.5(+2.8)
YOLOv8 + MFPAN	97.5	90.3	4.2	9.6	87.4	63.6(+2.9)
YOLOv8 + all	97.5	**91.3**(+2.5)	4.2	12.8	87.7	66.6(+5.9)
Fusion YOLO(+ SOD)	**99.3**(+1.9)	89.9	**2.1**	15.7	**98.2**(+11.1)	**71.7**(+11.0)

TABLE III
MODEL COMPARISON EXPERIMENT

Model	mAP@50	mAP@.5-.95	Param(M)	GFLOPs	AP^{hole}@50	AP^{hole}@.5-.95
YOLOv5(2020)	99.1	89.2	**1.7**	**4.2**	97.4	67.7
YOLOv8(2023)	97.4	88.8	3.0	8.1	87.1	60.7
YOLOv10(2024)	98.5	89.5	2.6	8.2	93.6	63.5
YOLOv11(2024)	97.1	89.8	2.5	6.3	85.1	61.7
Mamba-YOLO(2024)	97.7	**91.1**	5.6	13.6	88.7	66.5
Fusion YOLO	**99.3**	89.9	2.1	15.7	**98.2**	**71.7**

Precision), and mAP (mean Average Precision) as evaluation metrics. AP is a metric for measuring the performance of object detection models, calculated as the average precision at different recall levels. mAP is the average of AP across all categories. The formulas for calculating AP and mAP are as follows:

$$AP = \frac{1}{\sum_{k=1}^{n} p_k} \sum_{k=1}^{n} p_k \cdot \Delta r_k \tag{4}$$

Where: p_k is the precision at the k-th decision threshold. Δr_k is the change in recall between the k-th and (k+1)-th decision thresholds. n is the number of different decision thresholds.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{5}$$

Where: AP_i is the AP value for the i-th class. N is the total number of classes.

B. Ablation Experiment

To analyze the importance of each component designed in Fusion YOLO, this paper separately applies MdF, MFSR, and MFPAN to the baseline model to verify their effectiveness. Table 2 shows the impact of adding different modules on the evaluation metrics. The "+all" row indicates the performance of the model after integrating all three modules. The "+SOD" row represents the final performance of Fusion YOLO on the evaluation metrics after compressing the model hierarchy and adding a layer for small object detection.

From Table 2, it can be seen that the addition of MdF, MFSR, and MFPAN can significantly improve the evaluation metrics. Compared with the YOLOv8 baseline, with little change in the model's parameters and GFLOPS, the mAP@.5-.95 increased by about 1.5%. AP^{hole}@.5-.95, which is particularly crucial for bullet hole detection, saw even greater improvements: the model with MF increased by 3.3%, the model with MFSR increased by 2.8%, and the model with MFPAN increased by 2.9%.

TABLE IV
COMPARISON OF DIFFERENT HIGH-PASS FILTER

Model	AP^{hole}@50	AP^{hole}@.5-.95
YOLOv8 + ideal MdF	87.3	64.0
YOLOv8 + ButterWorth MdF	87.6	63.4
YOLOv8 + Gaussian MdF	88.0	64.1

Furthermore, we compared the impact of using different high-pass filtering methods in MdF on the model. According to Table 4, the MdF module using an ideal high-pass filter achieved an accuracy of 64% on AP^{hole}@.5-.95, while the MdF module using a Butterworth filter achieved 63.4%. The MdF module using a Gaussian filter achieved the best results, reaching 64.1%, and improved by about 0.5% on AP^{hole}@.5 compared to the former two.

C. Contrast Experiment

To further verify the effectiveness of the model proposed in this paper, this paper conducted contrast experiments on the designed dataset. The evaluation metrics for comparison include Param, GFLOPs, AP, mAP. We compared our model with a series of existing object detection models, including YOLOv5[3], YOLOv8[4], YOLOv10[10], and the recently proposed YOLOv11[5], as well as Mamba-YOLO[11], which has gained popularity by incorporating the Mamba Vision module[16]. As shown in Table 3, it can be seen that Fusion YOLO achieved the best results in almost all metrics while maintaining a relatively lightweight model, especially in terms of AP for bullet holes. This indicates that our model has a certain advantage in the detection of bullet holes.

V. Conclusion

The bullet holes on the chest bitmaps, as small targets, consist of relatively few pixels and cannot provide effective and sufficient visual information. Existing object detection algorithms struggle to accurately grasp their features and recognize bullet holes. This paper proposes a detection algorithm for chest bitmaps with bullet holes based on multiple fusion modules, named Fusion YOLO. Specifically, on the basis of the YOLOv8 baseline model, three modules—MdF, MFSR, and MFPAN are introduced to enhance model information richness, feature extraction precision and stability, and the integration of multi-scale feature maps. The experimental results demonstrate that the three proposed modules are effective and can significantly enhance the model's detection performance. Compared with other models in the YOLO series, including YOLOv5, YOLOv8, YOLOv10, YOLOv11, and Mamba-YOLO, the Fusion YOLO proposed in this paper also shows outstanding advantages. Fusion YOLO demonstrates advantages on the chest bitmaps with bullet hole dataset constructed in this paper, achieving a very small number of parameters (only 2.1M), with global mAP.5 and mAP.5-95 reaching 99.3% and 89.9% respectively, while the AP^{hole}@.5 and AP^{hole}@.5-.95 for bullet holes can reach 98.2% and 71.7% respectively, making it valuable for practical application in this downstream task. In the future, efforts will be made to further balance the model's accuracy and real-time performance, and to research a time-series-based bullet hole tracking model oriented towards video streams.

Acknowledgment

This work was supported by the 2023 Shanghai Industrial Development Project Agreement under project number HCXBCY-2023-050.

References

[1] Chen Haifeng. "Study on Automatic Scoring System for Shooting Sports Based on Image Processing Technology (in Chinese)". MA thesis. Nanjing University of Aeronautics and Astronautics, 2005.

[2] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[3] Glenn Jocher. *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*. https : / / github . com / ultralytics/yolov5. Version v3.1. Oct. 2020. DOI: 10.5281/zenodo.4154370. URL: https://doi.org/10.5281/zenodo.4154370.

[4] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLO*. Version 8.0.0. Jan. 2023. URL: https://github.com/ultralytics/ultralytics.

[5] Rahima Khanam and Muhammad Hussain. "YOLOv11: An Overview of the Key Architectural Enhancements". In: *arXiv e-prints*, arXiv:2410.17725 (Oct. 2024), arXiv:2410.17725. DOI: 10.48550/arXiv.2410.17725. arXiv: 2410.17725 [cs.CV].

[6] Tsung-Yi Lin et al. "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[7] Sebastien Razakarivony and Frederic Jurie. "Vehicle detection in aerial imagery : A small target detection benchmark". In: *Journal of Visual Communication and Image Representation* 34 (2016), pp. 187–203. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2015.11.002. URL: https://www.sciencedirect.com/science/article/pii/S1047320315002187.

[8] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Computer Vision Pattern Recognition*. 2016.

[9] Mingxing Tan, Ruoming Pang, and Quoc V. Le. "EfficientDet: Scalable and Efficient Object Detection". In: *arXiv e-prints*, arXiv:1911.09070 (Nov. 2019), arXiv:1911.09070. DOI: 10.48550/arXiv.1911.09070. arXiv: 1911.09070 [cs.CV].

[10] Ao Wang et al. "YOLOv10: Real-Time End-to-End Object Detection". In: (2024).

[11] Zeyu Wang et al. "Mamba YOLO: SSMs-Based YOLO For Object Detection". In: *arXiv e-prints*, arXiv:2406.05835 (June 2024), arXiv:2406.05835. DOI: 10 . 48550 / arXiv . 2406 . 05835. arXiv: 2406 . 05835 [cs.CV].

[12] Hang Gao Wei Zhang. "Design and Realization of Auto-Counting System in Automatic Target Scoring Based on Image Processing (in Chinese)". In: *Journal of Nanjing University of Aeronautics Astronautics* 6 (2000), pp. 691–695. ISSN: 1005-2615.

[13] Weiqi Yuan and Mengqi Li. "Research on Target of Automatic Scoring System Based on Visual Inspection (in Chinese)". In: *Computer Technology and Development* 29.02 (2019), pp. 147–151. ISSN: 1673-629X.

[14] Jiaqing Zhang et al. "SuperYOLO: Super Resolution Assisted Object Detection in Multimodal Remote Sensing Imagery". In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), pp. 1–15. DOI: 10.1109/TGRS.2023.3258666.

[15] Jun Zhang, Shuhua Yan, and Yan Xu. "Automatic Target-reading System Research Progress (in Chinese)". In: *Laser & Infrared* 36.12 (2006), pp. 1152–1154+1164. ISSN: 1001-5078.

[16] Lianghui Zhu et al. "Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model". In: *arXiv e-prints*, arXiv:2401.09417 (Jan. 2024), arXiv:2401.09417. DOI: 10.48550/arXiv.2401.09417. arXiv: 2401.09417 [cs.CV].

MSADeepLoc: Subcellular Localization Prediction Using MSA and Protein Language Model

1st Wenhui Zhao
Artificial Intelligence Institute
University of Jinan
Jinan, China
zhao_wh0205@163.com

2nd Yixin Zhong
Artificial Intelligence Institute
University of Jinan
Jinan, China
zyx@bupt.edu.cn

3rd Yi Cao
Artificial Intelligence Institute
University of Jinan
Jinan, China
ise_caoy@ujn.edu.cn

4th Wenxing He
School of Biological Science and Technology
University of Jinan
Jinan, China
chm_hewx@ujn.edu.cn

5th Yaou Zhao
Artificial Intelligence Institute
University of Jinan
Jinan, China
ise_zhaoyo@ujn.edu.cn

6th Yuehui Chen
Artificial Intelligence Institute
University of Jinan
Jinan, China
yhchen@ujn.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899712

***Abstract*—Protein subcellular localization (PSCL) prediction is a key research area in bioinformatics. However, most existing models have ignored the homology and diversity among protein sequences, which can reveal the functional sequence segments and indirectly reflect the structure of the protein. Consequently, the achieved performance is limited. To break through this limitation, we propose an end-to-end deep learning model for PSCL prediction, namely, MSADeepLoc. In this model, the protein features are extracted by a protein language model, the MSA Transformer. The MSA Transformer can learn the evolutionary conservation and variability among protein sequences from extensive multiple sequence alignments, effectively capturing and refining key biological information within the sequence structure. Additionally, we apply a pre-trained Conditional Generative Adversarial Network (CGAN) to solve the data imbalance issue. Extensive experiments demonstrate that our proposed MSADeepLoc outperforms the state-of-the-art models in the task of PSCL prediction.**

***Index Terms*—deep learning, protein subcellular localization, protein language model, generative adversarial networks.**

I. Introduction

Understanding protein subcellular localizations (PSCLs) is crucial for gaining insights into protein functions [1]. With the discovery of the proteins with multiple PSCLs, traditional experimental methods have become inadequate to reveal such complex mechanisms. As an appealing alternative, machine learning techniques have emerged for PSCL prediction. Currently, most machine learning-based prediction methods rely on a set of handcrafted features extracted from protein sequences. These features cover physicochemical properties [2], evolutionary information [3], and pseudo amino acid composition (PseAAC) [4]. However, the information carried by these features is limited, which may not be sufficient for sequence reconstruction, nor be beneficial to the specific task. Therefore, for most of the protein property prediction tasks, the results cannot be accurate.

Recently, the end-to-end deep learning models have become the mainstream for predicting PSCLs, where abstract features can be extracted automatically. DeepLoc [5] and MULocDeep [6] utilize deep neural networks to capture information from protein sequences to predict PSCL. However, most PSCL datasets are relatively small, and large models are prone to get overfitted. To address this issue, the widely applied pretraining techique has been introduced into this area from natural language processing (NLP). Several large protein language models (PLMs) have been pretrained on millions of protein sequences and then got fine-tuned on the downstream datasets. Models like DeepLoc 2.0 [7] and LAProtT5 [8] leverage those PLMs to predict PSCLs and have achieved signifcant improvements.

Biologically, the protein structure information is the pivot for predicting various properties of proteins. As reported in [9]–[11], the correlated mutation which can be revealled from multiple sequence alignments (MSAs) implies the key amino acids for protein folding. However, most PLMs are pretrained using masked single sequences, which is inefficient in extracting correlated mutation patterns. Therefore, advanced PLMs, such as AlphaFold [12] and MSA Transformer [13], have been proposed by mining the correlated mutations from MSAs and achieved remarkable success in protein structure prediction. The features extracted by these models provide a solid foundation to a variety of downstream tasks. For protein subcellular localization prediction, two challenges should be addressed. First, the evolutionary information in MSAs is often overlooked. Second, the datasets are usually small and highly imbalanced. Concerning these issues, we have propose a novel end-to-end deep learning framework, namely, MSADeepLoc. This framework introduces the MSA Transformer as the core

This work was supported in part by the University of Jinan Disciplinary Cross-Convergence Construction Project 2023 (XKJC-202308), and in part by Shandong provincial Natural Science Foundation, China (ZR2021MF036).

feature extractor, which is responsible for extracting evolutionary information from MSAs. Specially, we further incorporate row attention maps as supplementary features to strengthen the evolutionary information. For the data imbalance issue, we train a conditional generating adversarial network (CGAN) to generate synthesized samples for the minority classes. The CGAN is first trained on a large dataset to learn the general patterns of protein sequences, and then is fine-tuned by the minority classes so that high quality samples can be generated. Our main contributions are summarized as follows:

- We propose a novel and effective feature extraction method based on the MSA Transformer, via which the evolutionary information is obtained and enhanced by utilizing the correlations captured by row attention maps.
- We are the first to introduce the pre-training techique to generate high quality samples by a CGAN and augment the minority classes.
- We constructed a new archaeal dataset for independent testing. Our method is extensively tested on the PSCL dataset and achieved state-of-the-art results.

II. Method

The architecture of the proposed model is illustrated in Fig. 1. The MSA Transformer is trained as a universal feature extracter. Two types of features are extracted from the MSA Transformer, which are the embedding feature and the row attention feature map. These two features are then fused to produce a full representation of a protein. The data augmentation for the minority classes is performed by pre-training a CGAN. Inspired by [14], [15], the CGAN is trained to generate the protein features, instead of protein sequences. Finally, the above features are passed into the multi-label predictor.

A. Feature Extraction

Proteins are aligned by HHblits [16], which is an open-source tool for fast and accurate alignment of multiple protein sequences. Each protein sequence is aligned against the Uniclust30_2023_02 database to generate an MSA. For each alignment, the number of iterations is set to 3 for efficiency and alignment quality. We limit the maximum number of homologous sequences to 256. When exceeding this limit, a diversity-minimizing strategy is applied to select 256 representative sequences to maintain diversity while avoiding redundancy.

The MSA Transformer [13] is a large protein language model trained on 26 million MSAs using Masked Language Modeling. It contains 12 layers, 12 attention heads with 768 embedding dimensions and 100 million parameters in total. It can learn evolutionary and deep information from MSAs, encompassing structure features, function informations and potential biological activities.

We utilize the MSA Transformer to extract features from multiple sequence alignments of protein sequences. The 768-dimensional embeddings, extracted from the last attention block of the MSA Transformer, serve as the representation of each amino acid and are referred to as the MSA features. All 144 row attention maps are collected from 12 attention layers,

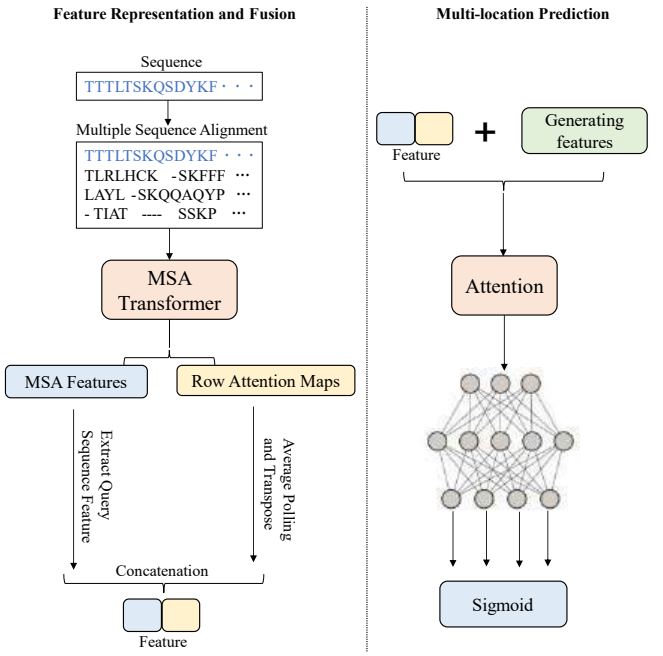


Fig. 1. The workflow of MSADeepLoc. The input amino acid sequences are first encoded using an MSA-based protein language model. The resulting MSA features and row attention maps are combined for feature fusion. Sequence representations of the fused and enhanced features are subsequently generated using the attention pooling mechanism and then processed by the prediction head MLP to predict subcellular localization.

each with 12 heads, to form the row attention map features, which encompass contextual and inter-sequence information to strengthen the evolutionary information. These features will be concatenated to generate the full representation of a protein.

B. Feature Fusion

For an input protein sequence, its MSA with S sequences at the maximun length of L is denoted by a matrix with the dimension of $[S, L]$. Its embedding feature generated by the MSA Transformer is an $[S, L, 768]$ tensor, one 768-dimensional vector for each amino acid. We extract features of the MSA query sequence (the input sample) from this tensor (the first matrix by default), obtaining a tensor with the dimension of $[1, L, 768]$ as the MSA feature.

A single-channel row attention map is a $L \times L$ matrix. In a typical MSA Transformer setup, there are 12 attention layers with 12 heads each. All the row attention maps are stacked into a tensor with dimensions $[L, L, 144]$. After average pooling along both row and column axes, two tensors with dimension $[1, L, 144]$ and $[L, 1, 144]$ are produced as the row attention map features. By transposing the second row attention map feature (from $[L, 1, 144]$ to $[1, L, 144]$) and concatenating the above three tensors along their last dimension, a comprehensive feature representation is constructed with the dimension of $[1, L, 1056]$.

This feature representation not only contains detailed information about each position in the query sequence but also

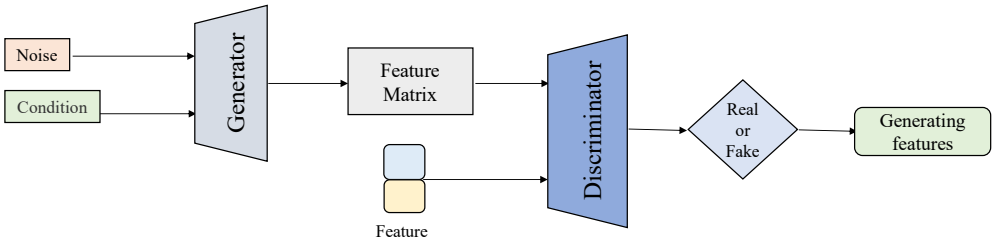


Fig. 2. Pseudo-feature generation is a component of MSADeepLoc. It generates pseudo-features for data balancing by utilizing specified label categories, employing labels as conditional variables for the CGAN.

integrates evolutionary relationships and similarities between different sequences, thereby providing robust support for subsequent PSCL prediction analysis.

C. Data Balancing by CGAN

Concerning that most of the datasets are highly imbalanced, especially for multi-label PSCL prediction, we train a CGAN [17] to supplement minority samples. By a CGAN, additional information, such as labels, can be treated as conditional variables so that it can be finely controlled to generate a specific types of samples. Similar as traditional the GAN model, the CCGAN contains a generator G and a discriminator D. G produces fake samples and tries to confuse the discriminator D, while D tries to discern the real samples from the fake ones, as shown in Figure.2.

The objective function L is defined as a minimax game between the generator G and the discriminator D, expressed as follows:

$$\begin{aligned} L = \min_G \max_D & E_{x\sim P_{\text{data}}(x)}[\log D(x \mid y)] \\ & + E_{z\sim P_z(z)}[\log(1 - D(G(z \mid y)))] \end{aligned} \tag{1}$$

where for a given label y, x is the real sample and $G(z \mid y)$. is a fake sample generated from the noise z.

Inspired by [15], the CGAN is not trained to generate protein sequences, instead, protein sequence pseudo-features are generated. In order to obtain high quality synthesized samples, the CGAN cannot be trained on minority classes directly due to the limited data size. Hence, we adopt a two-stage training strategy. First, we pre-train the CGAN on a large-scale dataset to ensure that it can learn a broad and rich feature representation. We have constructed a pre-training dataset of 12,450 PSCL sequences from the UniProt (2024_02) database [18], which are rigorously validated through experiments. These sequences broadly cover twelve subcellular localization categories: Cell Wall, Cytoplasmic, Extracellular, Outer Membrane, Periplasmic, Flagellar, Fimbrial, Nucleus, Chloroplast, Endoplasmic Reticulum, Golgi Apparatus, and Mitochondrion. After completing the pre-training, we then fine-tune the model using the limited minority class data to enhance the generator's for producing high-quality synthetic features of the target classes.

D. Prediction Layer

We employ an attention mechanism to process the sequence feature encoding. This is followed by a multi-layer perceptron (MLP) to make the final PSCL prediction. The output threshold for each label is determined by maximizing the Matthews correlation coefficient (MCC) value on the training data, ensuring that the model produces stable and reliable predictions. During training, we use weighted focal loss (WFL) [19] in combination with the binary cross-entropy objective function to optimize model parameters. The label weights are adjusted to the inverse frequency of their occurrence in the training dataset. This strategy further mitigates potential prediction bias due to dataset imbalance.

$$p_l = yp + (1 - y)(1 - p) \tag{2}$$

$$L_{ML} = \sum_{l\in N} -w_l \left(1 - p_l\right)^{\gamma} \log\left(p_l\right) \tag{3}$$

As shown in Eq.(2) and Eq.(3), p is the predicted probability for the label l, while y is the actual label value, p_l is the adjusted probability for each label l, calculated based on the actual label y and the predicted probability p. N is the set of all labels w_l is the weight for the label l, and γ is focal loss parameter, typically set to 1, used to adjust the contribution of hard-to-classify examples.

III. Experiments

A. Experiment Setups

In this section, we select the PSCL dataset adopted in DeepLoc 2.0 [7] for five-fold cross-validation. The dataset contains a total of 28,303 protein sequences and tagged partitions are created for the five cross-validations following the methodology outlined in Gíslason et al. [20]. Additionally, the sequence homology between the training and validation sets is less than 30%.

Given the significant evolutionary differences between archaea and bacteria/eukaryotes, as well as the relative scarcity of subcellular localization data for archaeal proteins in the current literature, we further created an independent test set to evaluate the proposed model. Specifically, we selected 587 archaeal protein sequences from the PSORTdb 4.0 [21] and UniProt (2024_02) databases [18]. These sequences have

all been experimentally validated to ensure their reliability. Furthermore, they are over 60 amino acids in length and exhibit less than 20% homology with any sequences in the previously mentioned training dataset, thereby ensuring the independence of the test results. The considered metrics include Overall Accuracy (OAA), F1, Ranking Loss(RL) and Hamming Loss (HL). We use the AdamW optimizer and the CosineAnnealingLR learning rate scheduler. The training is conducted with an initial learning rate of 0.001 and a dropout rate of 0.3 to prevent overfitting.

B. Experiment Results

In this study, we selected two different PSCL prediction methods as baselines, the LAProtT5 [8] and DeepLoc 2.0 [7]. The experiment results on the DeepLoc 2.0 dataset are summarised in Table I. Compared to DeepLoc 2.0, our model has improved performance on most metrics. The experiment results on the test dataset are summarised in Table II. It can be seen that the performance of our model is significantly better than that of Deeploc 2.0 and LAProtT5. Compared with Deeploc 2.0, our method improves 2.0% and 3.4% on the evaluation metrics of OAA and F1, respectively.

The performance improvement of the MSADeepLoc model can be attributed to the selection of features and generation strategies. To deeply analyze this improvement mechanism, three model variants were designed for comparison, as shown in Table III. The first variant examined the impact of feature encoding strategies on model performance. We explored the integration of various traditional feature encoding techniques—including encoding based on group weights (EBGW) [22], combined triplet encoding (CT) [23], dipeptide composition (DC) [24], and combined (C), transformed (T), and distributed (D) [25] feature encoding, and compared with the feature encoding of the protein language model ESM-1b. The experimental results show that using the conventional feature encoding method results in a reduction of OAA by about 13.4%. Using a single sequence-based encoding approach for the protein language model also results in a decrease in model performance.

Additionally, we assessed the impact of using a single feature dimension on the model's performance. Specifically, when only the 768-dimensional MSA feature was utilized, the model's performance declined. Furthermore, an experiment with a model variant that removed the feature generation layer resulted in a 2.3% decrease in OAA. In conclusion, the performance improvement of the MSADeepLoc model is the result of multiple factors working in synergy, particularly the optimized selection of feature encoding and the effective application of the feature generation layer, which lays a solid foundation for the model's application in the field of bioinformatics.

TABLE I
RESULTS ON THE DEEPLOC 2.0 DATASET

Method	OAA	Jaccard	F1	RL	HL
DeepLoc 2.0 (esm-1b)	0.9105	0.6657	0.7045	0.0670	0.0795
DeepLoc 2.0 (ProtT5)	0.9237	0.6721	0.7118	0.0614	0.0703
MSADeepLoc	**0.9373**	**0.6784**	**0.7207**	0.0617	0.0697

TABLE II
RESULTS ON THE ARCHAEAL INDEPENDENT TEST SET

Method	OAA	Jaccard	F1	RL	HL
LAProtT5	0.8140	-	0.5408	0.0908	0.1184
DeepLoc 2.0 (esm-1b)	0.8805	0.5316	0.6460	0.0786	0.0820
DeepLoc 2.0 (ProtT5)	0.8895	0.5537	0.6671	0.0721	0.0773
MSADeepLoc	**0.9093**	**0.5690**	**0.7014**	0.0683	0.0723

TABLE III
ABLATION STUDY RESULT

Method	OAA	Jaccard	F1	HL
Default	0.9373	0.6784	0.7207	0.0697
Traditional Feature	0.8037	0.4723	0.5337	0.4089
ESM-1b Feature	0.9213	0.6573	0.6954	0.0738
w/o Feature Fusion	0.9224	0.6570	0.6937	0.1012
w/o Feature Generation	0.9145	0.6417	0.6780	0.1077

IV. CONCLUSIONS

Accurate prediction of protein subcellular localization is crucial for understanding protein functions and interactions. Leveraging protein language models, we introduce MSADeepLoc, a novel multi-label prediction method. MSADeepLoc utilizes a protein language model with multiple sequence inputs to extract protein features and attention maps, improving performance and reducing manual feature design. To address the class imbalance issue, we incorporate CGAN to perform data augmentation at the feature level. An MLP trained with weighted focal loss performs multi-label classification. As shown in the extensive experiments, our model outperforms the existing methods, demonstrating its effectiveness. In the future work, we will focus on the

improvement of the multi-label classification method and develop new techniques to address the dataset imbalance issue.

REFERENCES

[1] K.-C. Chou, "Advance in predicting subcellular localization of multi-label proteins and its implication for developing multi-target drugs." *Current medicinal chemistry*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:146809486

[2] A. Sharma, K. K. Paliwal, A. Dehzangi, J. G. Lyons, S. Imoto, and S. Miyano, "A strategy to select suitable physicochemical attributes of amino acids for protein fold recognition," *BMC Bioinformatics*, vol. 14, pp. 233 – 233, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:1979815

[3] S. Wang, W. Li, Y. Fei, Z. Cao, D. Xu, and H. Guo, "An improved process for generating uniform pssms and its application in protein subcellular localization via various global dimension reduction techniques," *IEEE Access*, vol. 7, pp. 42 384–42 395, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:115196721

[4] K.-C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 273, pp. 236 – 247, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:35827803

[5] A. Armenteros, "Deeploc: prediction of protein subcellular localization using deep learning," *Bioinformatics*, vol. 33, p. 3387–3395, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3789627

[6] Y. Jiang, D. Wang, Y.-M. Yao, H. Eubel, P. Künzler, I. M. Møller, and D. Xu, "Mulocdeep: A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 4825 – 4839, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:234645420

[7] V. Thumuluri, J. J. A. Armenteros, alexander rosenberg johansen, H. Nielsen, and O. Winther, "Deeploc 2.0: multi-label subcellular localization prediction using protein language models," *Nucleic Acids Research*, vol. 50, pp. W228 – W234, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248477735

[8] H. Stärk, C. Dallago, M. Heinzinger, and B. Rost, "Light attention predicts protein location from the language of life," *Bioinformatics Advances*, vol. 1, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233449747

[9] F. Graner and J. A. Glazier, "Simulation of biological cell sorting using a two-dimensional extended potts model." *Physical review letters*, vol. 69 13, pp. 2013–2016, 1992. [Online]. Available: https://api.semanticscholar.org/CorpusID:20538720

[10] U. Göbel, C. Sander, R. Schneider, and A. Valencia, "Correlated mutations and residue contacts in proteins," *Proteins: Structure*, vol. 18, 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:14978727

[11] D. Altschuh, T. Vernet, P. J. Berti, D. Moras, and K. Nagai, "Coordinated amino acid changes in homologous protein families." *Protein engineering*, vol. 2 3, pp. 193–9, 1988. [Online]. Available: https://api.semanticscholar.org/CorpusID:24208963

[12] J. M. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, pp. 583 – 589, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235959867

[13] R. Rao, J. Liu, R. Verkuil, J. Meier, J. F. Canny, P. Abbeel, T. Sercu, and A. Rives, "Msa transformer," *bioRxiv*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231939146

[14] C. Wang, Y. Wang, P. Ding, S. Li, J. X. Yu, and B. Yu, "Ml-fgat: Identification of multi-label protein subcellular localization by interpretable graph attention networks and feature-generative adversarial networks," *Computers in biology and medicine*, vol. 170, p. 107944, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:266849455

[15] Y. Yang, H. Wang, W. Li, X. Wang, S. Wei, Y. Liu, and Y. Xu, "Prediction and analysis of multiple protein lysine modified sites based on conditional wasserstein generative adversarial networks," *BMC Bioinformatics*, vol. 22, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232434407

[16] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment," *Nature Methods*, vol. 9, pp. 173–175, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:205420247

[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *ArXiv*, vol. abs/1411.1784, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:12803511

[18] T. U. Consortium, "Uniprot: the universal protein knowledgebase," *Nucleic Acids Research*, vol. 45, pp. D158 – D169, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:214935664

[19] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:47252984

[20] M. H. Gíslason, H. B. Nielsen, J. J. A. Armenteros, and alexander rosenberg johansen, "Prediction of gpi-anchored proteins with pointer neural networks," *bioRxiv*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:209573642

[21] W. Y. V. Lau, G. Hoad, V. Jin, G. L. Winsor, A. Madyan, K. L. Gray, M. R. Laird, R. Lo, and F. S. L. Brinkman, "Psortdb 4.0: expanded and redesigned bacterial and archaeal protein subcellular localization database incorporating new secondary localizations," *Nucleic Acids Research*, vol. 49, pp. D803 – D808, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229175035

[22] Z.-H. Zhang, Z. Wang, Z.-R. Zhang, and Y.-X. Wang, "A novel method for apoptosis protein subcellular localization prediction combining encoding based on grouped weight and support vector machine," *FEBS Letters*, vol. 580, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:21328447

[23] Y. Wang, X. Wang, C. Chen, H. Gao, A. Salhi, X. Gao, and B. Yu, "Rpi-capsulegan: Predicting rna-protein interactions through an interpretable generative adversarial capsule network," *Pattern Recognit.*, vol. 141, p. 109626, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258393974

[24] B. Yu, X. Wang, Y. Zhang, H. Gao, Y. Wang, Y. Liu, and X. Gao, "Rpi-mdlstack: Predicting rna-protein interactions through deep learning with stacking strategy and lasso," *Appl. Soft Comput.*, vol. 120, p. 108676, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247353953

[25] C. Chen, Q. Zhang, B. Yu, Z. Yu, P. J. Lawrence, Q. Ma, and Y. Zhang, "Improving protein-protein interactions prediction accuracy using xgboost feature selection and stacked ensemble classifier," *Computers in biology and medicine*, vol. 123, p. 103899, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221085564

Enhancing Building Information Extraction from Remote Sensing Images through Reinforcement Learning from AI Feedback

1st Wenjing Cai
School of Cybersecurity
Northwestern Polytechnical University
Xi'an, China
Research & Development Institute
Northwestern Polytechnical University
Shenzhen, China
caiwenjing@nwpu.edu.cn

2nd Huiqun He
Satellite Research Department
Shanghai Academy of Spaceflight Technology
Shanghai, China
hehuiqun13@163.com

3rd Jiakun Yao
School of Software
Northwestern Polytechnical University
Xi'an, China
yaojiakun@mail.nwpu.edu.cn

4th Lipeng Gao
School of Software
Northwestern Polytechnical University
Xi'an, China
gaolipeng@nwpu.edu.cn

5th Liang He
School of Software
Northwestern Polytechnical University
Xi'an, China
2021050018@nwpu.edu.cn

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899536

***Abstract*—Deep learning is a standard approach in remote sensing for building information extraction, but it faces challenges like sparse data, large obstacles, and complex light-shadow interactions. To address these, we propose a novel reinforcement learning approach, Reinforcement Learning from AI Feedback (RLAIF), which enhances extraction by using a customized reward network with a deep multi-scale attention mechanism. We also improve the deep deterministic policy gradient algorithm with AI feedback, reducing iterations and boosting robustness. Experimental results show our approach reduces cross-entropy loss per pixel by 3.52% compared to supervised and 7.43% compared to unsupervised methods, while improving recall and accuracy, solidifying its competitive advantage.**

***Index Terms*—Deep Learning , Remote Sensing Images , Building Information Extraction , Reinforcement Learning from AI Feedback (RLAIF) , Deep Deterministic Policy Gradient.**

I. Introduction

Accurate building data have been essential since the 1920s with reinforced concrete and glass curtain walls [1]. Advances in GIS and high-resolution satellite imagery from China enable more precise building monitoring [2]. Insufficient research could lead to inaccurate urban mapping, hindering planning and ecological decisions [3].

Over the past two decades, deep learning has revolutionized building information extraction from remote sensing imagery. However, integrating AI feedback into reinforcement learning for extraction remains a challenge. Our research advances deep learning in remote sensing with a novel deep hybrid multiscale attention method. Building information extraction algorithms are classified into supervised and unsupervised types. Supervised methods, like Vision Transformer [4], UNet [5], and DeepLab [6], depend on large datasets and costly human annotations. To reduce this, they use pre-training [7], fine-tuning with limited samples [8], and transfer learning.

Unsupervised techniques for building extraction include diffusion models [9], GANs for image generation and segmentation [10], and traditional methods like SVM with wavelet transforms [11], SIFT-based or custom edge detection features [12], and random forests with image vectorization [13]. These methods require significant computational resources, limiting accessibility for small enterprises, and suffer from low accuracy and poor generalization due to manual feature design and sensitivity to disturbances. In contrast, models such as OpenAI's ChatGPT [14], GPT-4 [15], Google's Bard [16], and Meta's LLaMA [17], which use Reinforcement Learning from Human Feedback (RLHF) [18], requiring less annotation, they offer small firms a solid foundation for basic tasks.

RLAIF, introduced by Lee et al. [19], reduces reliance on large datasets and human annotations by automating the annotation process using large models. Unlike previous reinforcement learning applications in image segmentation, such as Duan's co-segmentation [20], Park's semi-supervised method [21], and Abo-eleneen's medical image algorithm [22], RLAIF avoids ineffective manually constructed or deep network-predicted reward functions for closed-loop systems.

Pre-trained segmentation networks can be viewed as agents in information extraction. After analyzing building photos, these agents generate segmentation results as actions, learning to achieve optimal outcomes through AI interaction.

The main benefits and contributions of the algorithm pro-

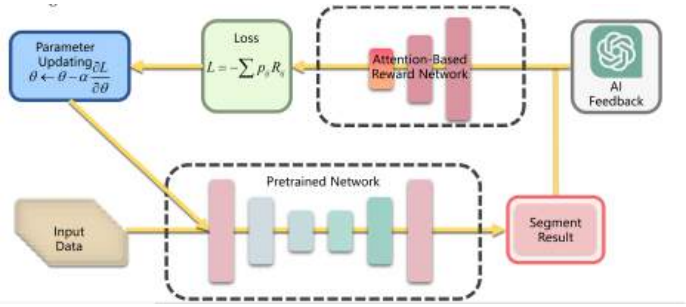


Fig. 1. The Framework of Attention-Based Reinforcement Learning from AI Feedback.

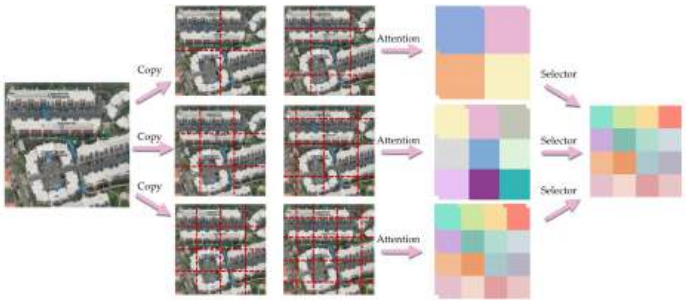


Fig. 2. Hybrid Spatial Multi-Scale Attention Mechanism

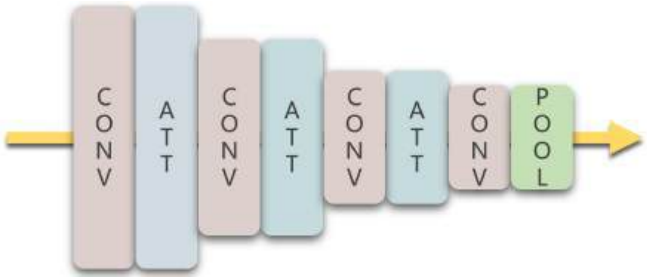


Fig. 3. The Structure of the Reward Network

posed in this study, based on the previous analysis, are as follows:

a) We propose an RLAIF-based algorithm for more accurate information extraction that requires less data and human annotation than other techniques.

b) Using attention, we propose a reward network design that reduces training and enhances feature capture over existing methods.

II. Method

The attention-based AI feedback reinforcement learning algorithm, shown in Fig. 1 and resembling DDPG [23], includes five modules: segmentation training, AI feedback design, attention-based reward construction, loss formulation, and parameter updates. It uses the pre-trained DeepLab V2 [24] from Pascal VOC [25] and the Adam optimizer.

A. AI Feedback Design

The latest GPT-4V, integrated with DALL·E 3, serves as the AI supervisor for feedback reinforcement learning, ensuring segmentation accuracy and correcting discrepancies between initial and actual segmentation results [26]. The original image weight is set to 0.95, with 50 fine-tuning steps for high-quality output. The reward network combines outputs from GPT-4V and a pre-trained segmentation network. The AI Feedback System (AIFS) processes remote sensing data, identifies key features like terrain and buildings, and generates feedback to adjust the incentive system, modifying parameters for image processing to enhance accuracy and adaptability in complex image analysis with real-time adjustments.

B. Reward Network Design Based on Attention Mechanism

Stacking layers in deep reinforcement learning can cause inefficiencies due to redundant parameters. To solve this, we propose a reward network with an attention mechanism inspired by PSPNet [27] and Swin-Transformer [28]. As shown in Fig 2, this hybrid multi-scale attention reduces GPU time, boosts precision, and cuts costs.

This section explains the reward network's role in interpreting AI feedback and adjusting rewards, emphasizing attention's impact on feature extraction and learning efficiency.

Fig 2 illustrates the attention mechanism where images are divided into n segments, each producing feature maps of size $\left(\frac{W}{i}, \frac{H}{i}\right)$, $i = 1, 2, 3, \ldots, n$. This method, inspired by PSPNet's pyramid pooling, addresses diverse spatial scales in building extraction. Using Swin-Transformer's cross-window partitioning, the feature maps are further segmented into units of size $\left(\frac{W}{2i}, \frac{H}{2i}\right)$, $i = 1, 2, 3, \ldots, n$, enhancing feature connectivity across spatial regions. These segmented maps then undergo self-attention [29], optimizing feature weighting and improving data extraction from each small map.

$$\mathbf{W}_i = N\left(\mathbf{Q}_i \mathbf{K}_i^T\right) \mathbf{V}_i \tag{1}$$

Here, W_i denotes the attention distribution in the i-th feature map, and $N(\cdot)$ is the normalization function to prevent excessive values. The matrices $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i$ represent the query, key, and value matrices, respectively, all derived from a single convolutional layer applied to the inputs.

Finally, small feature maps are concatenated by position into a complete map and weighted via a fully connected layer.

$$\mathbf{x}_o = \sum_{i=1}^{n} N(w_i \mathbf{x}_i) \tag{2}$$

The weight of the i-th scale is denoted by w_i. Summing values from different scales, the item-by-item normalization in Equation (2) prevents excessive output values.

An attention-based reward network is created by stacking multi-scale spatial attention mechanisms Fig. 3.

Fig 3 illustrates the reward network, consisting of four modules: pooling (POOL), attention (ATT), convolutional (CONV), and reward vector output. The CONV module extracts local features through activation, batch normalization, and convolution layers. The ATT module, utilizing a hybrid

multi-scale spatial attention mechanism, focuses on global information. The POOL module adjusts the feature map size, and the reward vector is generated.

A key aspect is integrating the LLM output into the reward network, where high-quality annotations adjust rewards, linking annotation quality to reinforcement learning for better segmentation accuracy.

C. Loss Function Design

The reinforcement learning loss function based on AI feedback has two components: the reward network loss and the pre-trained segmentation network loss. The reward network generates a reward that approximates the actual AI-derived reward, guiding network updates. The reward network loss is defined as the Mean Squared Error (MSE) function:

$$L_R = \frac{1}{n}\sum_{i=1}^{n}\left(R_i - \hat{R}_i\right)^2 \tag{3}$$

In this model, R_i represents the observed reward for the i-th instance, and $\hat{R}_i$ denotes the predicted reward before feedback. The difference between the sum of classification probabilities in the pre-trained output and the adjusted probabilities after feedback is expressed as follows:

$$R_i = \frac{|\hat{p} - p|}{HW} \tag{4}$$

The expression $|\hat{p} - p|$ measures the total variation between the pre-trained network's output, $\hat{p}$, and the adjusted output after AI feedback, p, for the i-th data point. Here, H and W represent the image dimensions. If the loss is below the threshold $\mathcal{E}$, training stops, and the predicted reward becomes the actual reward, reducing GPU training time.

The pre-trained segmentation network's loss function, inspired by DDPG, is as follows:

$$L_S = -\hat{R}\hat{p} \tag{5}$$

The loss function L_S represents the loss of the pre-trained segmentation network, where $\hat{P}$ is the per-pixel correct classification probability. The negative sign (-) indicates that the goal is to minimize the loss, which corresponds to maximizing the positive reward, aligning with reinforcement learning's reward maximization. Equation (5) shows that a smaller loss indicates a higher per-pixel classification probability, with the loss decreasing as the reward network's output increases.

III. Experiment and Result

A. Experiment Dataset

This study used four Nvidia RTX 3090 Ti GPUs. The Adam optimizer, with a learning rate of 1e-4 and 150 epochs, was employed. Human feedback dynamically adjusted the learning rate, and the Swish activation function was used with a batch size of 16.

To address the limitations of satellite deployment and dataset segmentation, which typically result in small datasets, we used the Stable Diffusion network to generate synthetic high-resolution remote sensing images. This approach created a large, diverse dataset of over 15,000 geotagged images (0.2 m/pixel resolution), covering various architectural scenarios, weather conditions, times of day, and perspectives. These synthetic images improved architectural diversity and enhanced the accuracy and generalization of our deep learning models, as shown in Fig 4 [30], mitigating overfitting and challenges posed by complex environments with shadows and overlaps.

Fig. 4. The Experiment Dataset

To prevent overfitting, the dataset was divided as follows: 10% for testing, 20% for validation, and 70% for training. The collection includes the following building types: commercial, industrial, and residential.

B. Assessment Indicators

Common metrics in supervised segmentation, such as pixel-wise cross-entropy loss (CE), accuracy (Acc), precision (Pre), recall (Rec), F1-Score (F1), and intersection over union (IoU), are used to evaluate the defect segmentation results in this study.

C. Results and Analysis

For qualitative comparison of RLAIF with enhanced UNet, Swin-Transformer, DeepLab, and Meta's SAM (April 5, 2023), segmentation difference maps are shown in Fig. 5.

In Fig 5, red and green areas represent misclassified backgrounds and structures, respectively. While various algorithms yield similar results, RLAIF outperforms SAM in accurately segmenting buildings, thanks to its AI feedback mechanism that refines its architecture through continuous learning. This enables RLAIF to handle complex contexts and obstacles, such as shadows, better than SAM. SAM excels as a general segmenter but struggles with specialized tasks like fault segmentation. UNet, typically used for medical image segmentation, may misclassify similar fault locations due to its fine granularity. RLAIF's multi-scale attention mechanism excels in distinguishing buildings in crowded areas, improving precision in tasks like building information extraction. This contrasts with SAM's universal approach, which leads to lower accuracy and recall, showing how algorithm design, training goals, and datasets impact performance.

While DeepLab and Swin-Transformer avoid significant under- or over-segmentation, they show larger discrepancies,

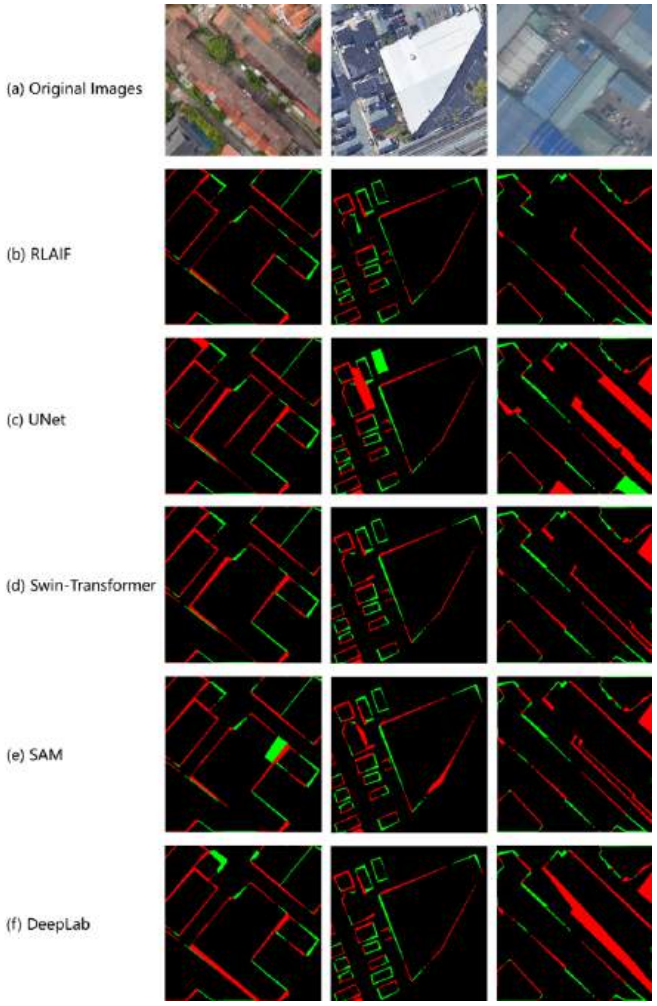


Fig. 5. The Different Plot of the Segmentation

TABLE I
QUANTITATIVE COMPARISON OF SEGMENTATION PERFORMANCE

Algorithm	CE	Acc	Pre	Rec	F1	IoU
SAM	0.598	0.872	0.686	0.852	0.772	0.629
UNet	0.573	0.851	0.692	0.803	0.743	0.591
DeepLab	0.383	0.957	0.974	0.870	0.919	0.850
Swin Trans	0.398	0.954	0.970	0.862	0.913	0.839
RLAIF	**0.372**	**0.963**	**0.978**	**0.883**	**0.928**	**0.866**

especially at defect boundaries, compared to the proposed algorithm. DeepLab, using image pyramid pooling and dilated convolution, extracts information at multiple spatial scales, performing closer to the proposed algorithm than Swin-Transformer, which relies on a cross-window attention mechanism. To objectively compare these algorithms, we used established metrics, presenting the results in Table 1 for both qualitative and quantitative assessment.

Table 1 shows that the proposed method achieves the lowest cross-entropy (CE), indicating better fitting to training data than SAM and UNet. RLAIF and DeepLab demonstrate strong pixel classification with accuracies of 0.963 and 0.957, outperforming SAM (0.872) and UNet (0.851). In precision,

TABLE II
SEGMENTATION PERFORMANCE QUANTIFICATION ABLATION EXPERIMENT

Algorithm	CE	Acc	Pre	Rec	F1	IoU
Sigmoid	0.390	0.954	0.971	0.861	0.913	0.839
ReLU	0.374	0.962	0.976	0.881	0.926	0.862
Tanh	0.401	0.952	0.970	0.856	0.909	0.834
NO ATT	0.452	0.915	0.911	0.785	0.843	0.792
NO S-ATT	0.399	0.958	0.968	0.875	0.919	0.851
S-ATT	0.419	0.934	0.924	0.832	0.876	0.770
O-RLAIF	0.372	0.963	0.978	0.883	0.928	0.866
0.5-RLAIF	0.406	0.946	0.953	0.849	0.898	0.815
2-RLAIF	**0.341**	**0.967**	**0.984**	**0.894**	**0.937**	**0.881**

RLAIF, DeepLab, and Swin-Transformer excel with scores of 0.978, 0.974, and 0.970, while UNet and SAM lag with precision scores of 0.692 and 0.686. For recall, the proposed method (0.883) and SAM (0.852) perform well, but SAM's higher recall sacrifices precision, resulting in a lower F1-Score (0.772). The proposed method leads in Intersection over Union (IoU) with 0.866, followed by DeepLab (0.850) and Swin-Transformer (0.839). SAM and UNet show lower IoUs of 0.629 and 0.591, indicating significant deviations from actual segmentation. This data suggests the proposed algorithm outperforms others in overall segmentation, but there are still gaps in recall compared to other advanced methods.

In addition to evaluating segmentation performance, we explored how AI Feedback enhances building extraction precision. This feedback significantly improved recall and precision through continuous integration of high-quality AI-generated annotations. Our approach achieved a 3.52% reduction in per-pixel cross-entropy loss compared to supervised segmentation networks and a 7.4% reduction compared to unstructured models, emphasizing the key role of AI Feedback in improving building delineation accuracy.

D. Ablation Study

We conducted an ablation study to assess the impact of various components on RLAIF's performance, testing variations such as 0.5-RLAIF (half network depth), 2-RLAIF (double depth), S-ATT and NO-ATT attention mechanisms, and activation functions (Sigmoid, ReLU, Tanh). Each variant's performance was evaluated based on Precision, Recall, F1 Score, and IoU, as shown in Table 2.

Table 2 shows that 2-RLAIF outperforms the original algorithm in accuracy (Acc), precision (Pre), recall (Rec), F1 score, and IoU, indicating that expanding the network improves segmentation performance. In contrast, 0.5-RLAIF performs worse on all metrics, with IoU dropping from 0.866 to 0.815, likely due to reduced network parameters diminishing model expressiveness.

The O-RLAIF algorithm, which combines self-attention and multi-scale attention, achieved an IoU of 0.866, outperforming versions using only self-attention (IoU 0.770) or no attention mechanism (IoU 0.792). This indicates that while self-attention has a smaller impact, it still contributes to performance, with the absence of attention significantly reducing results.

Additionally, the original RLAIF, using the Swish activation function, achieved an IoU of 0.866. Variations with Sigmoid, ReLU, and Tanh performed worse, with IoUs of 0.839, 0.862, and 0.834, respectively. This underscores the effectiveness of Swish, with ReLU showing similar performance due to its strong gradient propagation and better model generalization compared to Sigmoid and Tanh.

In summary, the original RLAIF algorithm performs significantly better with increased parameters, the use of a hybrid spatial multi-scale attention mechanism, and the Swish activation function.

IV. Conclusions

This study presented the RLAIF method for information extraction, effectively addressing challenges such as small datasets, occlusions, and shadows in traditional deep learning approaches. By developing a reward network combined with a deep mixed multi-scale attention mechanism, we significantly improved recall and precision, thereby enhancing remote sensing image analysis, particularly for building extraction.

In the future, we aim to improve RLAIF, explore its scalability for broader remote sensing applications, and integrate it with multispectral and hyperspectral datasets to enhance accuracy. Additionally, we plan to enhance its real-time processing for dynamic tasks like environmental monitoring and disaster response, collaborating with domain experts to tailor the model for specific applications and further advance AI-driven remote sensing solutions.

V. ACKNOWLEDGMENT

This work was supported in part by the Aeronautical Science Foundation of China under Grant 2023Z034053001, in part by the National Natural Science Foundation of China under Grant 42101469, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515111175.

References

[1] A. B. Özkaya, "Encyclopedia of twentieth century architecture," 2004.

[2] C. Yu, D. Hu, M. Liu, S. Wang, and Y. Di, "Spatio-temporal accuracy evaluation of three high-resolution satellite precipitation products in china area," *Atmospheric research*, vol. 241, p. 104952, 2020.

[3] J. E. Goldstein, B. Neimark, B. Garvey, and J. Phelps, "Unlocking "lock-in" and path dependency: A review across disciplines and socio-environmental contexts," *World Development*, vol. 161, p. 106116, 2023.

[4] M. Chen, H. Peng, J. Fu, and H. Ling, "Autoformer: Searching transformers for visual recognition," pp. 12 270–12 280, 2021.

[5] S. Liu, H. Ye, K. Jin, and H. Cheng, "Ct-unet: Context-transfer-unet for building segmentation in remote sensing images," *Neural Processing Letters*, vol. 53, pp. 4257–4277, 2021.

[6] Y. Nishida, Y. Li, and T. Kamiya, "Environment recognition from a spherical camera image based on deeplab v3+," pp. 2043–2046, 2021.

[7] M. Macary, M. Tahon, Y. Estève, and A. Rousseau, "On the use of self-supervised pre-trained acoustic and linguistic features for continuous speech emotion recognition," pp. 373–380, 2021.

[8] Y. Saleh and G. Issa, "Arabic sign language recognition through deep neural networks fine-tuning," 2020.

[9] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," pp. 22 500–22 510, 2023.

[10] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," pp. 6199–6203, 2020.

[11] J. Yan *et al.*, "Noncontact defect detection method of automobile cylinder block based on svm algorithm," *Mobile Information Systems*, vol. 2022, 2022.

[12] S. Li, X. Fu, and J. Dong, "Improved ship detection algorithm based on yolox for sar outline enhancement image," *Remote Sensing*, vol. 14, no. 16, p. 4070, 2022.

[13] J. Zapata, R. Vilar, and R. Ruiz, "Automatic inspection system of welding radiographic images based on ann under a regularisation process," *Journal of Nondestructive Evaluation*, vol. 31, pp. 34–45, 2012.

[14] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[15] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[16] M. S. Rahaman, M. Ahsan, N. Anjum, M. M. Rahman, and M. N. Rahman, "The ai race is on! google's bard and openai's chatgpt head to head: an opinion article," *Mizanur and Rahman, Md Nafizur, The AI Race is on*, 2023.

[17] O. Analytica, "Meta llama leak raises risk of ai-linked harms," *Emerald Expert Briefings*, no. oxan-es, 2023.

[18] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan *et al.*, "Training a helpful and harmless assistant with reinforcement learning from human feedback," *arXiv preprint arXiv:2204.05862*, 2022.

[19] H. Lee, S. Phatale, H. Mansoor, K. Lu, T. Mesnard, C. Bishop, V. Carbune, and A. Rastogi, "Rlaif: Scaling reinforcement learning from human feedback with ai feedback," *arXiv preprint arXiv:2309.00267*, 2023.

[20] X. Duan, X. Liu, X. Gong, and M. Han, "Rl-coseg: A novel image co-segmentation algorithm with deep reinforcement learning," *arXiv preprint arXiv:2204.05951*, 2022.

[21] J. Park, Y. Seo, J. Shin, H. Lee, P. Abbeel, and K. Lee, "Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning," *arXiv preprint arXiv:2203.10050*, 2022.

[22] A. Abo-Eleneen and A. Mohamed, "Mmrl: A multi-modal reinforcement learning technique for energy-efficient medical iot systems," pp. 2026–2031, 2021.

[23] Z. Wei, Z. Quan, J. Wu, Y. Li, J. Pou, and H. Zhong, "Deep deterministic policy gradient-drl enabled multiphysics-constrained fast charging of lithium-ion battery," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2588–2598, 2021.

[24] J. Czajkowska, P. Badura, S. Korzekwa, and A. Płatkowska-Szczerek, "Automated segmentation of epidermis in high-frequency ultrasound of pathological skin using a cascade of deeplab v3+ networks and fuzzy connectedness," *Computerized Medical Imaging and Graphics*, vol. 95, p. 102023, 2022.

[25] K. Tong and Y. Wu, "Rethinking pascal-voc and ms-coco dataset for small object detection," *Journal of Visual Communication and Image Representation*, vol. 93, p. 103830, 2023.

[26] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, "The dawn of lmms: Preliminary explorations with gpt-4v (ision)," *arXiv preprint arXiv:2309.17421*, vol. 9, no. 1, p. 1, 2023.

[27] L. Yan, D. Liu, Q. Xiang, Y. Luo, T. Wang, D. Wu, H. Chen, Y. Zhang, and Q. Li, "Psp net-based automatic segmentation network model for prostate magnetic resonance imaging," *Computer Methods and Programs in Biomedicine*, vol. 207, p. 106211, 2021.

[28] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," pp. 10 012–10 022, 2021.

[29] J. Zhang, Y. Jiang, S. Wu, X. Li, H. Luo, and S. Yin, "Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism," *Reliability Engineering & System Safety*, vol. 221, p. 108297, 2022.

[30] R. Tang, L. Liu, A. Pandey, Z. Jiang, G. Yang, K. Kumar, P. Stenetorp, J. Lin, and F. Ture, "What the daam: Interpreting stable diffusion using cross attention," *arXiv preprint arXiv:2210.04885*, 2022.

Leveraging Neural Networks to Locate Short-lived Anomalies in gas consumption

Sahil Varma Penmetsa
Computer Information Systems & Information Technology
University of Central Missouri
Austin, USA
sahilvrp9@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899681

***Abstract*—The core objective of this research revolves around the creation of a robust system designed to identify outliers with a novel approach. This approach combines the strengths of both Artificial Neural Networks (ANNs) and Auto-regressive Integrated Moving Averages (ARIMA) models. While ARIMA excels in linear prediction, ANNs demonstrate their prowess in tackling nonlinear prediction challenges. Through their symbiotic integration, they effectively encapsulate the intricate and intricate nonlinear relationships that underlie the interaction between meteorological variables and gas consumption patterns. The system excels at labeling and pinpointing outliers, allowing these anomalies to be promptly communicated to building managers for further scrutiny and potential rectification measures within HVAC systems. This intervention, in turn, contributes to the reduction of energy wastage. The system functions within a dual-phase framework: first, it forecasts short-term (hourly) gas consumption by tapping into historical time-series data; second, it employs a deviation-based strategy to identify anomalies from projected or expected values. Notably, the identification of these anomalies is accomplished without the requirement for pre-annotated instances. This is attributed to the remarkable precision of the predictions, characterized by a Root Mean Square Error (RMSE) that spans from 8 cubic meters to 2.5 cubic meters.**

***Keywords*:** *Anomaly Detection, Gas Consumption, Neural Networks, ARIMA Model*

I. Introduction

Energy utilization in structures is perhaps of the quickest developing area. Roughly 41% of the complete energy in Europe is consumed by structures (families and administrations) [1]. Studies and states' orders about limiting energy utilization and utilizing sustainable power expanded consistently with the decrease of petroleum products, the boundary grindings with eastern nations like Russia, and the increment of different ecological issues. Considering this, the European Association, with a new order [2], has the objective to raise EU energy utilization delivered from sustainable assets to 20%, to decrease by 20% the EU ozone harming substance emanations, and to improve by 20% the EU's energy productivity. This implies speculations to re-qualify old structures, new nation regulations, energy finding, yet additionally new effectiveness frameworks from the pre-owned apparatuses.

Determining energy requests has become one of the significant examination fields in the energy divisions since it can assist with gassing utilities yet in addition organizations and families. Gas utilities purchase gas from pipeline organizations consistently, so they need to know the requirements ahead of time to be serious. Organizations and families expect to diminish energy utilization and increment productivity.

Of late, huge organizations like Google have additionally shown their premium in this new market, creating indoor regulators that consequently control the house environment by putting together the choices with respect to the timetable of the clients. Home, an organization obtained by Google, pronounced that clients saved the 11.3% of AC-related energy utilization without compromising solace [3], because of the programmed learning executed in their indoor regulators. On the off chance that, from one viewpoint, the programmed indoor regulator program setting in light of individuals' way of behaving can assist them with setting aside cash, abnormality discovery can diminish significantly more energy utilization. It is displayed by [4] and [5], that business structures consume from 15% to 30% more energy than needed due to inadequately kept up with, corrupted, and inappropriately controlled hardware. These irregularities can turn out to be effectively fixable issues with a solid shortcoming location and determination (FDD) framework.

In this paper, a programmed exception identification framework is proposed, where days/hours with strangely high and low gas utilization are marked and answered to the structure supervisor. He can additionally investigate and fix the central air framework, limiting the energy squander brought about by the exceptions. Gas utilization is exceptionally sporadic and not effectively unsurprising with exemplary strategies. The exception discovery framework introduced depends on forecasts made by a cross breed ARIMA-ANN, which can show direct and non-straight way of behaving of the information with entirely solid outcomes and an examination between the anticipated worth/pattern and the real one to track down anomalies.

Since the meaning of exception is profoundly application-reliant, in section II they are characterized. In a similar segment, ANNs and ARIMA are momentarily made sense of on the grounds that they will be of late utilized in the proposed arrangement (section IV), in light of a gas utilization forecaster. In section III, a few related works are examined. In section V, a few examinations on engineered and genuine information are shown. section VI presents a few future thoughts for the perusers in light of procedures that the creator

lacked opportunity and energy to apply.

II. Background

A. What is an outlier

An exception, by definition [6], is a perception that goes astray altogether from different perceptions so it makes doubt that various elements made it. In spite of this overall definition, the more suitable approach to characterizing anomalies is profoundly application-subordinate since even similar situations might require various conclusions of exceptions.

In this paper, exceptions are firmly connected with the issue of time-series estimating since anomalies are announced based on deviations from expected (or figure) values. In this specific situation, a worth is viewed as an exception as a result of its relationship to its connected information (*contextual* exception [7] or restrictive oddities [8]). An unexpected pinnacle (fig. 1) in a period series is a *contextual* exception in light of the fact that its worth is totally different from the upsides of its nearby items.

Whenever a gathering of focuses are proclaimed exceptions, it is alluded to as *collective* inconsistency or anomaly [7]. It initially shows up at a point, and afterward it influences the qualities promptly close to it. Sooner or later, this impact vanishes, passing on the time series to typical way of behaving. This situation is normally difficult to recognize.

Exceptions can have unmistakable primary reasons:

1) Blemished framework (for example a damaged radiator in a room).
2) Terrible human way of behaving (e.g., individuals who leave open the window in a room while the framework is attempting to warm it).
3) Faulty observing framework, where the framework screens various qualities from the genuine one because of a glitch, figuring process blunders, or a recording carelessness.

In writing, exceptions are likewise alluded to as irregularities, freaks, oddities or peculiarities.

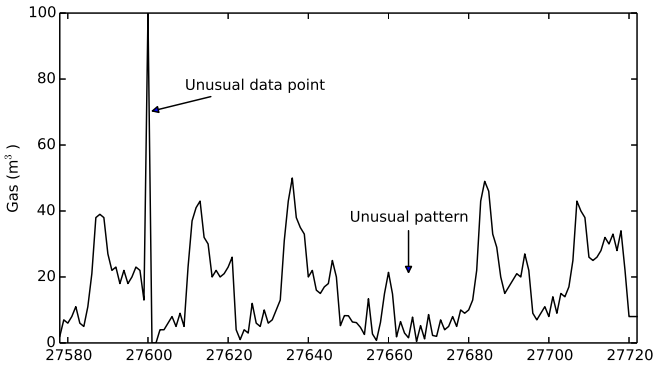


Fig. 1. Different types of outliers. On the left an unusual data point is presented, on the right an unusual pattern of changes can be recognized if compared to the other days shape.

B. Artificial Neural Networks

Artificial Neural Networks(ANNs) were initially evolved to imitate cerebrum usefulness. There is definitely not a generally acknowledged definition, however by [9]: "A brain network is a circuit made out of an extremely huge number of straightforward handling components that are neurally based. Every component works just on neighborhood data. Besides, every component works nonconcurrently; accordingly, there is no general framework clock. " From **??**, it is feasible to see a completely associated ANN with five sources of info, three neurons on the secret layer (purported on the grounds that the ANN resembles a black box), and one result. The data sources are additionally called highlights, and they address the attributes that depict the result. The secret neurons will plan this connection. Every association has an actuation that represents the significance (weight) of the associated neuron. ANNs with somewhere around 1-stowed away layer can plan non-direct relations.

The ANNs are generally applied with the *Stochastic Slope Descent* calculation, which attempts to track down the right loads of every association with have the right result esteem. It is generally joined with the *Backpropagation* calculation, which computes the mistake in the result layer and afterward back-engenders it to the past layers to change the loads [10]. More data can be found in [11].

These days, ANNs are a best in class procedure that has numerous applications.

C. Autoregressive models

Let $X_1, X_2, \ldots X_t$ be the values in an univariate time-series. In the *Auto-Regressive Moving Average* model, the value of X_t is defined in terms of the values of the last window of length p and q moving average terms.

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + c + \varepsilon_t.$$

The left-hand part is called the auto-regressive part because it depends on the previous (lagged) values $X_{t-1}, X_{t-2}, \ldots X_{t-p}$, the right-hand part is called moving average because the error at time t is the linear combination of the previous errors $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$.

These techniques are applied to *stationary* time-series, purported when the mean, fluctuation, and autocorrelation structure don't change after some time. Unfortunately, many time series make occasional impacts or patterns. Specifically, irregular strolls, which describe many kinds of series, are non-fixed. Differencing the information focuses can frequently change a non-fixed time series into a fixed one. In view of the Crate Jenkins models of the $1970s$, ARIMA models separate where series with deterministic patterns ought to be differenced first, then an ARMA model applied. ARIMA models are normally referenced as ARIMA(p, d, q), to show the ARMA boundaries and the d request of difference.ARIMA models are likewise fit for demonstrating a large number of occasional information.

$$ARIMA(p, d, q)(P, D, Q)_m$$

where m is the quantity of periods per season. The capitalized documentation is utilized for the occasional pieces of the model, and the lower-case documentation for the non-occasional pieces of the model.

The decision of the boundaries p, d, q is profoundly application ward and it depends on hypothesis that is past the extent of this paper. More data can be found in [12].

III. Related work

Since this paper declares outliers based on deviations from the expected (or forecast) value, this section is divided into related work in forecasting and outlier detection.

A. Outlier detection

Outlier detection systems are a wide range of areas, from introduction detection systems to fraud detection systems, law enforcement systems to earth science anomaly detection systems.

Outlier detection can be supervised when available data is labeled indicating previously known examples of anomalies, semi-supervised, where only examples of normal data or anomalies are available, or unsupervised, where previous examples of interesting anomalies are not available. Typically, most of the unsupervised outlier mechanisms use a measure of *outlierness* of a data point, such as sparsity of underlying region, nearest neighbor distance, or the fit to underlying distribution [7]. In these cases, a data point is unusual due to one or more variables rather than a specific one (like in the supervised methods).

In energy consumption outlier detections, literature is usually based on the Gaussian error theory, stating that when the measurement accords with normal distribution, the probability that the residual falls in three times the variance is more than 99.7%. Therefore the residuals falling outside it can be considered outliers. In [13] the author further improved this system considering a rolling window median which seems to improve the results when the distribution is not fixed. Supervised methods are usually based on classifications using trees, ANNs, and other different algorithms, thanks to the presence of previous examples of anomalies. In the energy consumption field, unsupervised methods are usually based on clustering, where an algorithm tries to find similarities between points/trends and cluster them into groups, calculating the distance between them. A cluster is considered good when the intra-cluster distance is minimized, and the intra-cluster distance is maximized. Popular methods in this group are k-means, one-class SVM and self-organizing maps.

For example, in [14], some clustering methods, like CART, k-means, and DBScan, were applied to detect outliers in the office lighting energy consumption. The author showed different techniques applied with the Generalized Extreme Studentized Deviate (GESD) and listed some irregularities found. He also stated that the clustering methods were not able to detect faults strongly related to time variables.

Clustering methods are very difficult to apply in time-series data, and the results are not usually excellent. For this reason, this paper will build a prediction algorithm, where outliers are declared based on deviations from the expected (or forecast) value. The more accurate the predictor, the more it will detect abnormal data points.

B. Forecasting

Traditionally, several techniques have been used for energy use forecasting, but short-term, medium-term, and long-term energy forecasting needs to be differentiated. The former usually refers to prediction with a horizon of hours or days, the second refers to weeks, the latter refers to monthly or annual horizon. Long-term forecasting usually deals with data that rarely presents significant distortions and irregularities, so they have a small effect on the overall value. On the contrary, short-term forecasting has to deal with irregularities and sudden changes in values (due to weather changes, human behavior, etc.).

There are essentially five types of prediction models [15]: Engineering methods, Statistical methods, Artificial Neural networks, Support Vector Machines, and Grey models. Engineering methods use physical principles to calculate thermal dynamics and energy behavior of the building, Statistical methods build empirical models to apply regression to a time series of values, Neural networks try to predict energy using an artificial intelligence network of interconnected neurons, Support vector machines are based in a machine learning algorithm and Grey models apply a mixture of the models. All the principal methods are extensively reviewed in [15] and [16].

Several techniques have been traditionally applied for energy use forecasting, and among the statistical methods, Kalman filtering and ARIMA/ARMAX time-series techniques are the most famous.

The first reports about applications of Artificial Neural Networks (ANNs) were published in the early 1990s [17]. Since then, the number of publications increased steadily. Kalogirou et la. [18] used back propagation neural networks to predict the required heating load of 225 buildings, Ekici and Aksoy used the same model t predict building heating loads in three-buildings. Nizami and Al-Garni [19] tried a simple feed-forward NN and related the electric energy consumption to weather data and population, Taylor and Buizza [20] used an ANN with weather data (51 variables) to predict load of 10 days ahead. Gonzales [21] built an ANN to predict hourly energy consumption. Some researchers tried to specialize the ANNs: Neto and Fiorelli [22] compared generic ANNs with working days ANNs and weekend ANNs, Lazzerini and Rosario [23] specialized them to predict electric lighting with weather data.

Some researchers have also tried to apply a hybrid model to increase the performance of the ANN. One example above all is [24] which applied a hybrid ARIMA and neural network model to forecast electricity use, another one is [25] who improved the previous one. This paper is based also on his work.

Until now, only electric forecasting was presented because the majority of the existing forecasters are related to electric forecasting. There are only a few of them are about natural gas forecasting: Brown et al.[26] built one of the first predictors for natural gas consumption, and Khotanzad et al. [27] developed a two-stage system ANN with very good results.

Even if ANNs might outperform traditional methods, the researchers are still not convinced about the results of ANNs in this field. Nevertheless, it is also stated that "a significant portion of the ANN research in forecasting and prediction, lacks validity" [28] and that most of the papers seem misspecified models that had been incompletely tested (no standard benchmarks, no synthetic data, etc.) [16]. This paper will try to avoid these mistakes.

It also needs to be pointed out that ANNs are *multi-step ahead* forecasters, while Auto-Regressive methods are potentially useless in long ahead data points.

IV. Proposed solution

As stated before, this paper proposes a regression algorithm in which outliers are declared based on deviations from the expected (or forecast) value.

A time-series is a sequence of data points typically measured at successive points of a uniform time interval t (eq. (1)).

$$\{x(t_0), x(t_1), \dots x(t_i), x(t_{i+1}) \dots\} \tag{1}$$

where x is the value and t the time.

Time-series forecasting is about predicting future values given past data (eq. (2)).

$$\hat{x}(t+s) = f\left(x(t), x(t-1) \dots\right) \tag{2}$$

where s is the step size. A *multivariate* time-series is a $(n \times 1)$ vector of n time-series variables.

It can be seen that in academic and industry research, linear regression-based systems are the standard "de facto" of energy forecasting. In recent works, this problem is treated by combining weather forecast data. However, this relationship is clearly non-linear [16]. Consequently, even if some papers have acceptable results with measured datasets, these systems cannot adequately capture the relationship in all the situations and data. Since ANNs are the state-of-the-art technique of many machine learning problems where there is complex non-linear hypotheses, the proposed solution is composed of a *multilayer feed-forward* neural network with *backpropagation*.

A. Experimental data

Ebatech collects the energy consumption datasets used in different buildings of the Hogeschool van Amsterdam (as shown in TABLE I). The Universiteit van Amsterdam provided these datasets to this project. These buildings are located in Amsterdam, the capital city of The Netherlands. This city has a maritime climate similar to that of England, which the North Sea strongly influences. Winters are fairly cold, and summers are rarely hot by European standards. Amsterdam is characterized by the common presence of rain and wind, and the weather conditions vary frequently.

Building name	Date interval	Number of rows
HvA 740 - NTH	01/2008 - 03/2014	54.725
Hva 761 - KMH	01/2009 - 09/2013	40.407
Hva 882 - WBW	01/2008 - 03/2014	54.647

TABLE I
BUILDINGS USED.

Ebatech collected different types of features in each building, with different granularity. For this project, three buildings are used: *HvA 740 - NTH*, *Hva 882 - WBW* and *Hva 761 - KMH*. In these buildings, the company collected the energy consumption and the gas consumption as other variables. It needs to be noted that gas is used only to heat the buildings.

The weather data was collected by KNMI[1] in Schipol, the Amsterdam airport 16 km away from the tested buildings. The dataset, findable on the website, consists of over 21 variables collected hourly. The proposed solution only uses a few of them, as explained in the section IV-B, and they are used as forecast values: the measured weather conditions are linked to the previous hour of energy consumption. It is necessary to consider that there will be an error in the built model since the weather data is collected in a different location from the building's positions, and in practice, the error will be larger than those obtained in this simulation due to the effect of the weather forecast uncertainty [29], [30]. The advice is to keep it in mind before applying the methods contained in this paper with days forecasting.

1) Data analysis: The energy consumption dataset covers a very large period ranging more than five years, allowing us to see similar patterns even with different yearly/season behavior (one year could be different from another one for external factors like weather or building use). It is very rich (with more than 50 variables) but also sparse because the monitored variables are not the same in all the buildings. For this reason, only common variables, such as total electric and gas consumption, were used in this paper. In this way, it is possible to generalize and compare the models.

The gas consumption data is highly seasonal: daily and weekly cycles are quite perceptible, as it can be seen from fig. 3 and fig. 2. From the latter, the weekly behavior is clear: the last two days of the week (Saturday and Sunday) are completely different from the others, and Monday seems a bit different from the rest of the days. Every day, around 4:00-5:00 AM, the system seems to react by turning on the heating system, whereas in the previous hours of the night, it seems only to keep a minimum temperature. The system reveals to us that after a couple of hours, consumption decreases again. In fig. 3 the Temperature has a clear daily/hourly relation with the gas consumption while in fig. 4 the electric consumption is shown to be very smoothed and more regular than the gas one.

2) Data cleaning: The data presented some irregularities like repeated and missing data points. Although the first one may not influence the performance of the ANN, it could lead

[1] Koninklijk Nederlands Meteorologisch Instituut http://www.knmi.nl

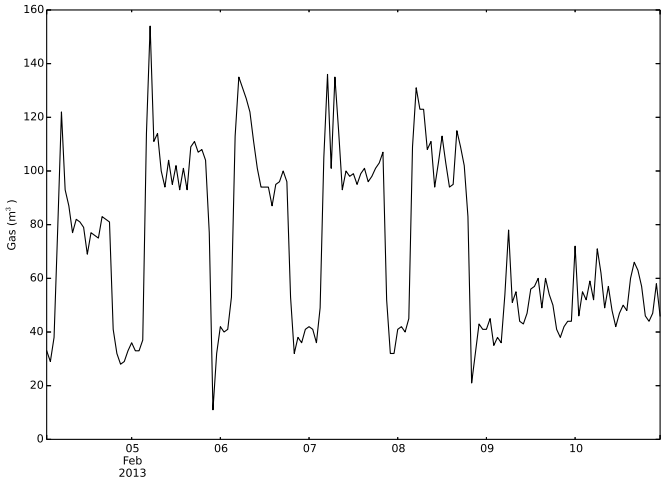


Fig. 2. Typical weekly and daily gas consumption behavior in building 740-NTH. The weekly pattern can be noticed by observing that the last two days of the week (Saturday and Sunday) have a completely different shape than the others. During the week, the daily behavior is very similar, with one peak around 4:00-5:00 AM.

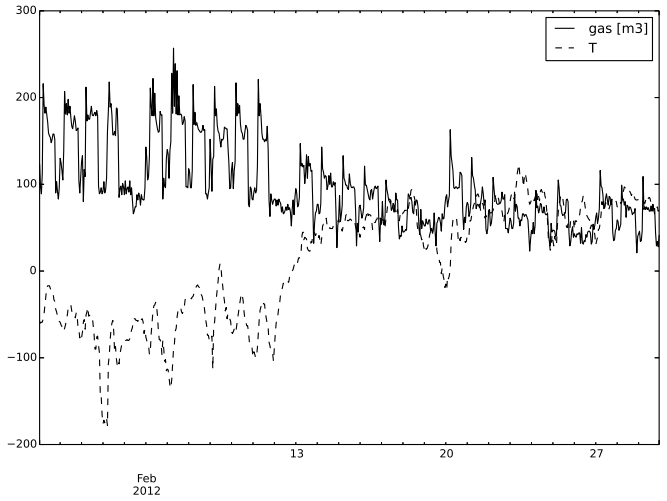


Fig. 3. Typical monthly gas consumption behavior in building 740-NTH and its relation with the temperature on building 740-NTH.

to problems when other algorithms are used (like ARIMA in this paper). Repeated data points were deleted, keeping only the last one, while missing data points were reconstructed by linear interpolation. Cubic and spline interpolation were also considered, but the performance was heavily affected by these methods.

Missing and repeated data points represent some problems in the data collection that will be further investigated.

B. Artificial neural network forecaster

More or less complex dependencies characterize time-series: Known dependencies like date-time dependencies. Hidden dependencies include the behavior of the HVAC system (when it starts, when it raises the temperature, etc.). Short/long-term dependencies between variables.

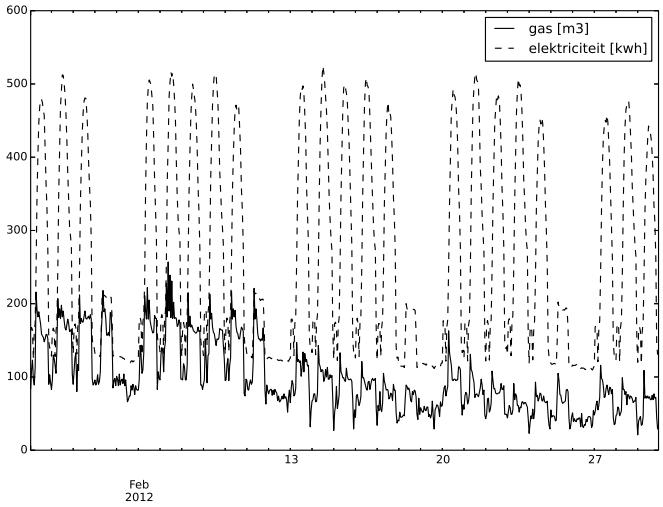


Fig. 4. Relation between electric and gas consumption in building 740-NTH

Data scientists and experts are focused on known dependencies, while the proposed ANNs will be focused on the hidden ones. The short/long-term dependency is realized by a moving window containing a "memory" of the previous states for the interesting variables, using a *Tapped delay line memory* [31]. These memories form a new set of states

$$\{\bar{x}_1(t), \bar{x}_2(t), \dots \bar{x}_n(t)\}$$

from the original states

$$\{x(1), x(2) \dots x(n)\}$$

where $\bar{x}_i(t) = x(t-i+1)$. The window types will be explained later in this section.

Since the value to predict is time-dependent, the first element to consider is adding the time feature. Energy consumption depends on the hour of the day but also on the day of the week and the seasonality of the year (month and day of the year) (as explained in section IV-A1, fig. 3 and fig. 2). The day of the week is a number from 0 to 6, where 0 is Monday and 6 is Sunday. Since the behavior of the holidays was considered similar to the weekends (particularly similar to Sundays), a function encoded all the holidays as weekend days[2]. In the future this can be improved asking a time-table list for the buildings, indicating when these are closed. The day of the year is a number between 0 and 366, and the first one is the first of January. All these date-time features by means of their sine and cosine values as usual, reported in literature [32], [33], [21]. This transforms the time component into a cyclic feature that spans a fixed length (a single day for the hour), and it is bounded in $[-1, 1]$.

Another added feature was the current system load, which is the energy consumption at the k state when the load at $k+1$

[2]Thanks to an Open source Dutch weekend list https://github.com/PanderMusubi/dutch-holidays

needs to be predicted. This was believed to be an important measure for determining building usage and holidays.

Many factors affect the energy needs of buildings. These factors can be divided into three main groups, namely the *physical environmental*, the *artificial designing parameters*, and the *human thermal discomfort*. The first is composed of weather-related parameters like outdoor temperature, wind speed, solar radiation, etc. The *artificial designing parameters* are related to the building construction: transparency ratio, orientation, etc [34], but these variables were unavailable in the dataset. The *human sensation of thermal discomfort* is correlated not only to the temperature but also to other variables such as relative humidity, irradiation, and wind speed. Even if all these data were available, the only significant weather variables found were the temperature and the wind speed.

The system consumption was believed to be related to the difference in the outdoor temperature between two instants (eq. (3)), representing a positive/negative change in the external environmental conditions.

$$\Delta T_{k+1} = T_{k+1} - T_k \tag{3}$$

where T_{k+1} is the predicted temperature for the period $k+1$ and T_k is the value measured in the instant k. It needs to be noted that the real behavior of the system was unknown, so it was not possible to know if this change would have an immediate effect on the HVAC system and its reaction time.

Gas usage has a daily cycle, but there are also secondary weekly and annual cycles that the ANN may not be able to capture. Gas usage $u(t)$ is defined as

$$u(t) = s(t) + f(t) + r(t)$$

where $s(t)$ is the seasonality at time t, $f(t)$ is the trend and $r(t)$ is called *remainder*, irregular component or difference. The time series were analyzed by the STL decomposition by LOESS [35](fig. 5), which decomposes a time series into seasonal, trend, and irregular components by an additive method. Since the ANN is interested in the *remainder* and the trend can be found from the historical part, the daily, weekly, and yearly remainders were added as features. For the same reason also, the Temperature, the wind speed, and the electric consumption were decomposed by the STL decomposition, resulting in a stationary time series added to the input.

In Zhang et al. [36], it is stated that ANN models really have advantages while dealing with a large amount of historical load data with non-linear characteristics, but the researchers neglected the linear relations, including the data. For this reason, a hybrid approach is proposed, where the ANN will be helped in linear forecasting by the popular method ARIMA (auto-regressive integrated moving averages), commonly known as the Box–Jenkins approach. In order to apply ARIMA, the time series was processed iteratively with a moving window of 21 days where the ARIMA model was fitted. After the fitting, the values of the next 24 were forecasted hours before moving the window and doing the same for the day after. The seasonal ARIMA fitting was done by the help of the Forecast R package [37] and its *auto. arima* method, which finds the

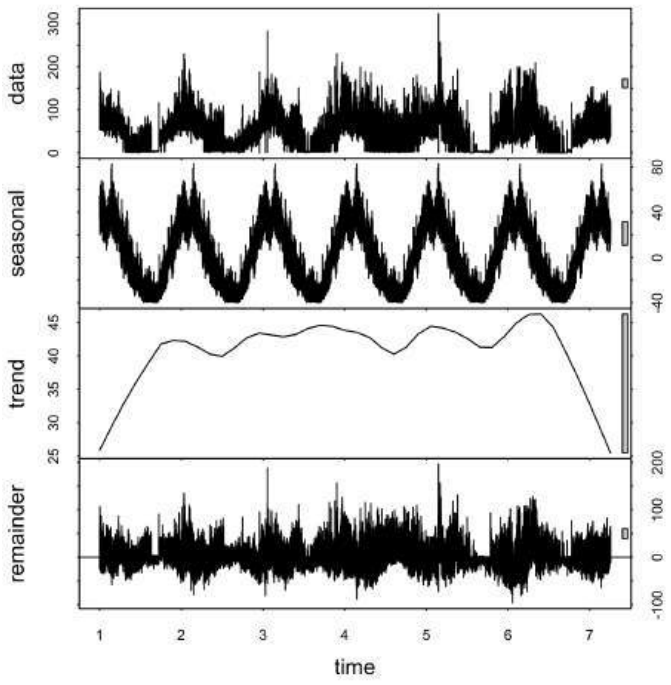


Fig. 5. Yearly STL decomposition by LOESS in building 740-NTH.

best ARIMA $(p,d,q)(P,D,Q)_m$ parameters by comparing the *Akaike information criterion* (AIC) of the tested models. Just for the sake of the reader's curiosity, the most fitted model was ARIMA $(3,0,3)(2,0,1)_{24}$. An ARIMA model with temperature dummy variables was tested, but it did not help the ANN further; the simplest models were preferred.

Taking into account the points made in this section, the ANN is predicting the gas consumption "seeing" without knowing its shape and its behavior in the previous hours/days. This limit is surpassed by some rolling windows, which will somehow simulate the Recurrent neural networks' behavior. Two rolling windows were created for the gas consumption, memorizing the sum and the peak load of the previous five hours, and two moving rolling windows were made for the STL yearly residuals, memorizing sum and peak of them.

All the features that were not between the limits $[-1,1]$ were scaled to have a faster convergence [38] of the *Stochastic Gradient Descent* (eq. (4)).

$$x_i' = \frac{x_i - \frac{max(x)+min(x)}{2}}{\frac{max(x)-min(x)}{2}} \tag{4}$$

where x_i is the original value and x_i' is the scaled one. Many practical tricks like the shuffling of the elements, the normalization and initialization were taken from [38], [39].

All the process described so far is also called *feature engineering* and was done almost iteratively, cumulatively introducing and removing features from the model and comparing the performance.

Choosing a number of hidden units for the ANN is always a tricky task. As stated by [40], [41], using *early stopping* in an

Variable	Data
Electricity load	$E(t)$
Hour	$sin(2\pi(h)/24)$; $cos(2\pi(h)/24)$
Week day	$sin(2\pi(wDay)/6)$; $cos(2\pi(wDay)/6)$
Month	$sin(2\pi(mon)/12)$; $cos(2\pi(mon)/12)$
Year day	$sin(2\pi(d)/366)$; $cos(2\pi(d)/366)$
Temperature	$T(t)$
Gas peak$'$	$\max_{1\leq k\leq 5} G(t-k)$
Gas sum$'$	$\sum_{i=1}^{5} G(t-i)$
Gas mean$'$	$\frac{1}{288}\sum_{i=1}^{288} G(t-i)$
Gas peak$''$	$\max_{1\leq k\leq 24} G(t-k)$
Gas sum$''$	$\sum_{i=1}^{24} G(t-i)$
Electricity peak$''$	$\max_{1\leq k\leq 5} E(t-k)$
Electricity sum$''$	$\sum_{i=1}^{5} E(t-i)$
Temp peak	$\max_{1\leq k\leq 5} T(t-k)$
Temp sum	$\sum_{i=1}^{5} T(t-i)$
Wind speed	$FH(t)$
ΔT_{k+1}	$T_{k+1} - T_k$
ARIMA forecast	$forecast(ARIMA(3,0,3)(2,0,1)_{24})$
STL year res.	$YearRes(t)$
STL day res.	$DayRes(t)$
STL E res.	$Res(E(t))$
ARIMA peak$'$	$\max_{1\leq k\leq 5} ARIMA(t-k)$
ARIMA sum$'$	$\sum_{i=1}^{5} ARIMA(t-i)$

TABLE II
ANN FEATURES.

oversized *Backpropagation* ANN, where the number of hidden neurons is higher than the number of the features, makes it easier to find the global optimum and avoid bad local optima. For this reason, the number of hidden units was chosen to be greater than $2 \times |features|$ and then test-driven, and the training was early stopped to prevent *overfitting* (as shown in TABLE II).

1) Architecture: ANNs are trained to minimize a cost function of the form

$$E = \frac{1}{N}\sum_{i=1}^{n} p(r_i)^2$$

where the error function p is symmetric and continuous, $r_i = Y_i - \hat{Y}_i$ is the residual between the actual value and the forecast one, and N is the number of training patterns.

Using the notations defined above, the most used cost function is based on the Mean Squared Error (MSE), commonly known in data modeling as the Least Mean Squares (LMS) method. The basic idea of LMS is to optimize the fit of a model with respect to the training data by minimizing the square of residuals

$$p(r) = \frac{1}{2}r^2$$

but it is greatly influenced by outliers [42]. In order to control the damage caused by outliers, in this paper the Least Mean Log Squares (LMLS) method (eq. (5)), presented by [42] is used. The ANN will try to minimize the Mean Log Squared Error (MLSE).

$$p(r) = \log(1 + \frac{1}{2}r^2) \tag{5}$$

The ANN is a 1-hidden-layer *Multilayer Feedfoward* ANN with a feedback structure called *Backpropagation*. This ANN is composed by *Rectifier* neurons and one output linear node.

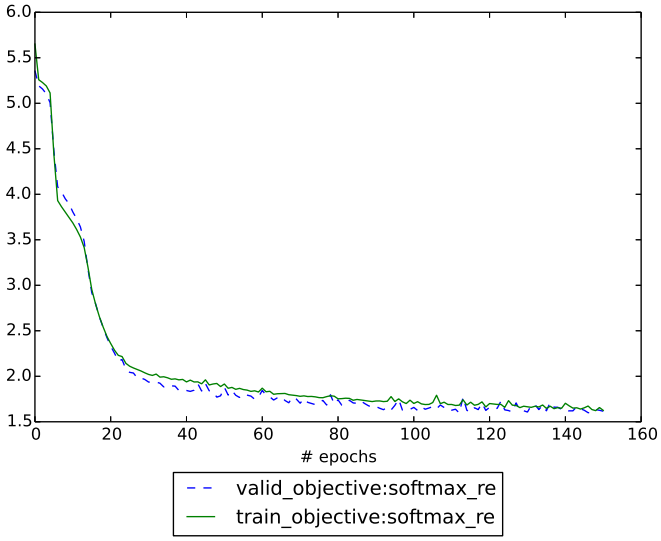


Fig. 6. Training curve of the hybrid model with 80 hidden neurons

Training is done by the *Stochastic Gradient Descent* algorithm with 10 batch size and is characterized by a learning rate of 0.003 and fixed by a Momentum of 0.05, which could help to increase the speed, avoiding local minima.

This project used Python and *pandas* for the data analysis, *Pylearn2* [43] to construct and test the ANN and the R system with the *zoo* [44] and the *Forecast* [37] packages for the ARIMA process.

C. Outlier detection

According to Chebschev's theorem [45], almost all the observations in a data set of system states fall into the interval $[\mu - 3\sigma, \mu + 3\sigma]$, where μ and σ are respectively the mean and standard deviation of the data set, and the data points outside this interval are declared outliers. In this paper the ANN is used to predict the gas consumption, for this reason a point will be considered outlier if it will fall outside the 95% confidence interval[3] expressed for the RMSE. If it is assumed that the difference between the actual values x_i and the predicted value $\hat{x}_i$ have:

$$\hat{x}_i - x_i \sim \mathcal{N}\left(0, \sigma^2\right) \tag{6}$$

- mean zero.
- follow a Normal distribution (it is assumed that it holds for the large amount of data utilized).
- and all have the same standard deviation σ.

$$\hat{x}_i - x_i \sim \mathcal{N}\left(0, \sigma^2\right) \tag{7}$$

it is possible to say that eq. (7) follows a χ^2_n distribution with n degrees of freedom. Which means: $\mathrm{P}\left(\chi^2_{\frac{\alpha}{2},n} \leq \frac{n\mathrm{RMSE}^2}{\sigma^2} \leq \chi^2_{1-\frac{\alpha}{2},n}\right) = 1-\alpha$

$$\Leftrightarrow P\left(\frac{n\mathrm{RMSE}^2}{\chi^2_{1-\frac{\alpha}{2},n}} \leq \sigma^2 \leq \frac{n\mathrm{RMSE}^2}{\chi^2_{\frac{\alpha}{2},n}}\right) = 1-\alpha$$

[3]For the following description user *fabee* of *CrossValidated* needs to be mentioned: http://tinyurl.com/l9gvz65.

$\Leftrightarrow \; P\left(\sqrt{\frac{n}{\chi^2_{1-\frac{\alpha}{2},n}}}\text{RMSE} \leq \sigma \leq \sqrt{\frac{n}{\chi^2_{\frac{\alpha}{2},n}}}\text{RMSE}\right) = 1-\alpha.$
Therefore

$$\left[\sqrt{\frac{n}{\chi^2_{1-\frac{\alpha}{2},n}}}\text{RMSE}, \sqrt{\frac{n}{\chi^2_{\frac{\alpha}{2},n}}}\text{RMSE}\right] \tag{8}$$

V. Experimental Evaluation

The ANN has been prepared to stop right on time, with a decent number of preparing ages (stages) or stop the preparation when the approval mistake rate increments. Every one of the outcomes showed are gotten from k-crease cross-approval methods, where the organization was prepared k times, each time leaving out a subset of information from preparing to test the ANN. The consequences of the k tests, were partitioned by k. The organization is constantly prepared with 70% of the information, 15% is utilized for approval, and the other 15% for testing.

Albeit a large portion of the Mean Outright Rate Mistake (MAPE) is viewed as a norm for inspecting the nature of the model expectation of energy load, it is a satisfactory blunder measure provided that the misfortune capability were straight and ongoing examinations showed it isn't [18][46]. Besides, the rate mistake is boundless in the event that there are no qualities on the series, regular in irregular information and in utilization information, and it puts a heavier punishment on certain blunders than on regrettable mistakes [47]. Due to these drawbacks, this paper will just consider the minimization of the Root Mean Square Blunder (RMSE), which punishes huge mistakes, as proposed in [48]. As recommended in [16], for each analysis likewise the Mean Outright Blunder (MAE) will be determined.

$$MAPE^4 = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{Y_i} \times 100 \tag{9}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2} \tag{10}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{Y}_i - Y_i\right| \tag{11}$$

where $\hat{Y}$ is the vector of the n predictions and Y is the vector of the true values.

A. Synthetic experiments

The strategy is tried with artificially produced information. Two days of exceptions were produced by various calculations. In the first, the genuine utilization was changed by an irregular worth, recreating the framework estimation/control breakdown, which makes the utilization bob all over (see eq. (12)). The second artificially made day was made by adding $50m^3$ of gas utilization to the genuine one, making an example that mimics a peculiar way of behaving as well as a glitch of the warming framework (see eq. (13)).

$$G(t) = G(t) + v * 30 \tag{12}$$

where v $\sim \mathcal{N}(0, \sigma^2)$

$$G'(t) = G'(t) + 50 \tag{13}$$

The two outliers were correctly detected, as it can be seen in fig. 7.

B. Measured data experiments

The method is also tested with measured data coming from a different type of day. For example, the gas consumption on a weekend was placed on a weekday, simulating a holiday. The purpose of this test was to show that an unusual pattern was detected. In fig. 8, it can be seen that the outlier mechanism works perfectly when the Sunday gas consumption is placed on a weekday.

The outlier was correctly detected, as can be seen in fig. 8. The robustness of the design was proved with different buildings, listed in section IV-A1.

Excluding this little experiment, some interesting behaviors were found through this work:

Occasions Whether or not the school was functional, the primary tests showed that the Ebatech framework was typically warming the structures (Christmas, on Tuesday, December 25th, 2012, was warmed like an ordinary Tuesday regardless of whether the structure was positively shut). This caused an avoidable waste. Utilization bobs In fig. 9, an unusual crisscross way of behaving should be visible for building 740-NTH. It appears to be that the framework is squandering energy and this shape is very surprising from the standard one (fig. 2). This goes on for quite a long time, and obviously additionally the ANN preparing is impacted by this exception like way of behaving. Tops Around the underlying long periods of September, there is a gigantic skip in utilization (up to multiple times the maximal utilization of the year). Is it a test? In building 761-KMH, unpredictable pinnacles were found consistently during April 2013, likely while the warming framework turned on. August with radiators In building 740-NTH, between August 2009 and August 2011, the warmers were dynamic despite the fact that it was not obvious that it was a virus summer. Anomalies A few different exceptions are found, yet they should be affirmed by the directors, ideally after the check of the recently referenced ways of behaving.

All these problems will be immediately reported to the company.

An ANN with the standard cost function MSE was also trained, apparently resulting in a smaller RMSE error in a faster way (section V-B). Although this can be true, the Hybrid MLSE model was more precise, better detecting possible

[4]MAPE errors will be calculated only on the non-zero values, to avoid the problems described before.

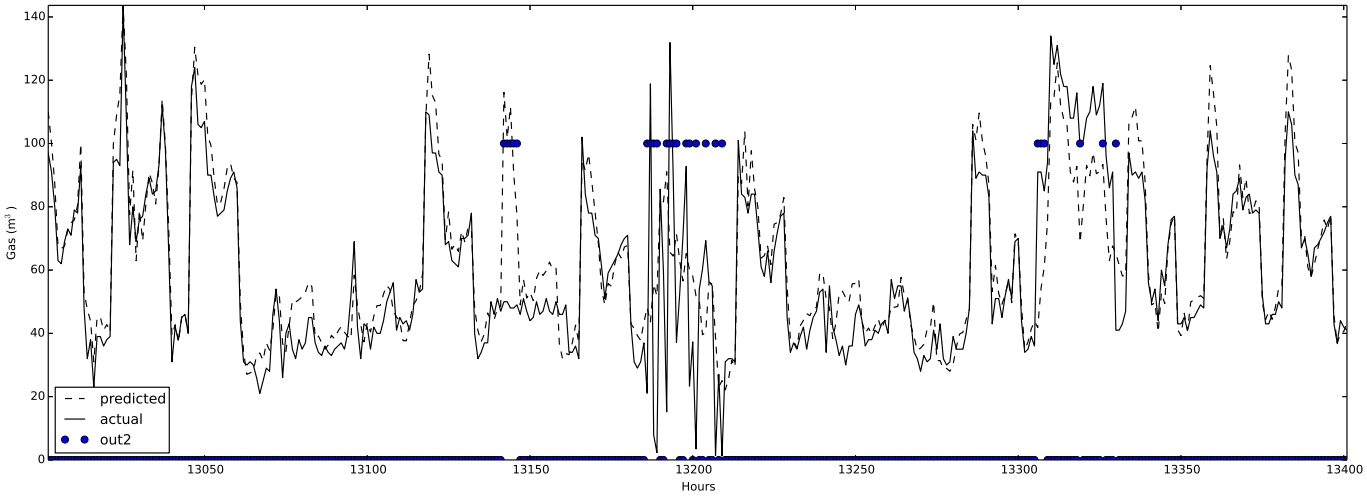


Fig. 7. Outlier detection with synthetically generated data. The circle represents the hours where an outlier is detected.

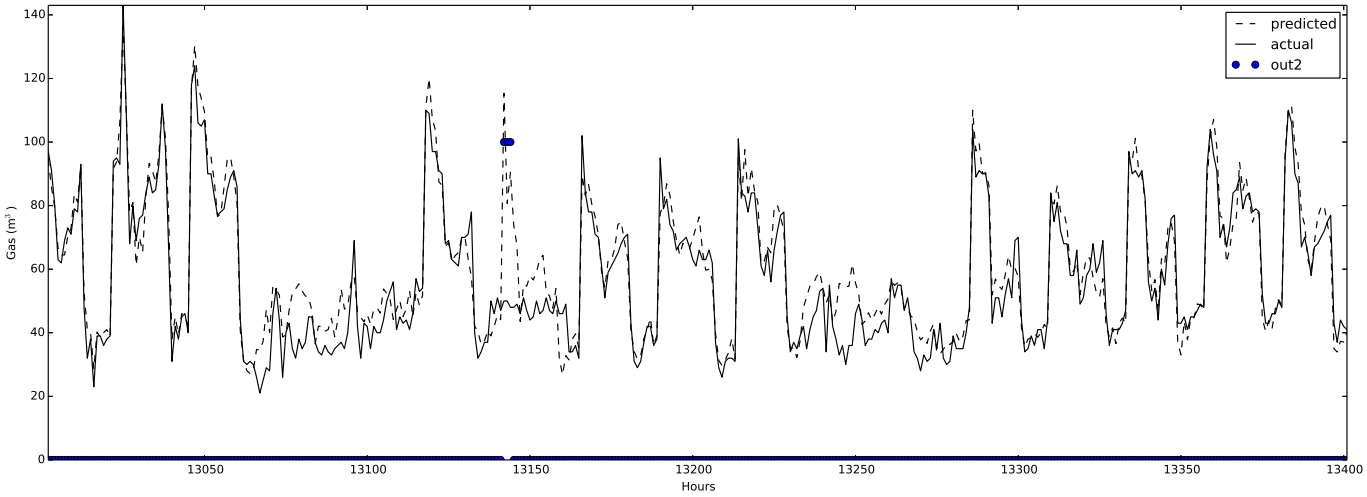


Fig. 8. Outlier detection where a Sunday one replaced the gas consumption of a weekday. The (three) circles represent the outliers detected by the system.

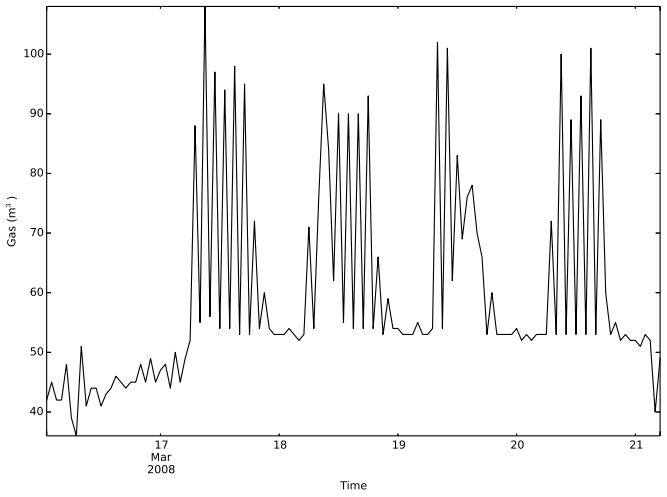


Fig. 9. Strange zig-zag behavior found by the algorithm.

Model	neurons	epochs [5]	RMSE	MAPE	MAE
ARIMA[6]	-	-	88.50	117.27	22.52
ANN	80	15	11.95	34.78	8.52
HyMSE	80	70	9.4	27.66	6.90
HyMLSE	150	140	10.02	30.05	7.26

TABLE III
BEST SELECTED RESULTS IN BUILDING 740-NTH, TO COMPARE THE ARIMA, ANN AND HYBRID MODEL. HYMSE IS THE HYBRID MODEL WITH MSE COST FUNCTION, WHILE HYMLSE IS THE SAME MODEL WITH MLSE COST FUNCTION.

outliers (as shown in TABLE III). They contributed the most to the error.

In section V-B the results in the different buildings can be read.

[5]epochs to converge
[6]Calculated iteratively as described in section IV-B
[7]epochs to converge

Model	neurons	epochs [7]	RMSE	MAPE	MAE
740-NTH	150	140	10.02	30.05	7.26
761-KMH	150	140	2.49	18.30	1.00

TABLE IV
BEST SELECTED RESULTS FOR ALL THE BUILDINGS.

VI. FUTURE WORK

ARIMA models can't detect more than one seasonality but it can be helped with Fourier terms and ARIMA *dummy variables* to produce reasonable forecasts. When multi-seasonality is present, an algorithm like TBATS can overpass the ARIMA one and detect it. This non-parametric model described in [49] could be substituted for the ARIMA one as a feature of the ANN. At the moment, it is very slow, but it is very recent, so it will probably be improved.

The daily pattern could be seen in the transformed Fourier space applying the Modified Discrete Cosine Transform (MDCT) [50]. In theory, this could help as well to understand the pattern, but it was only applied once by [51], with scarce results.

ANNs are sensitive to missing values and irregularities, but it was not possible to contact the building managers in order to confirm/identify previously known outliers. For this reason, the ANN training was done with data that was not entirely perfect, and this probably affected the performance. It is necessary to contact these building managers to further help with the training of this algorithm.

The input variables were scaled, standardizing them to a midrange 0 and range $[-1, 1]$. It is also possible to normalize them to have mean 0 and standard deviation 1. In this case, Robust estimates of location and scale are desirable if the inputs contain outliers. Some examples are [52] and the recent [53], which can be the basis of a future refinement of the ANN inputs.

In future work, it is also possible to analyze the difference between the MSE and the MLSE cost, not only in prediction but also in outlier detection.

Before 2006, ANN was almost always associated with the *Backpropagation* algorithm and with the 1-hidden layer architecture. The problem with these architectures is that they get stuck in poor local optima. In 2006, there was a huge breakthrough mainly started by [54], which is called *Deep learning*, and it represents the new fashion of ANNs based on multi-hidden layers and new algorithms.

Future improvements can be based on Recurrent Neural Networks (RNNs) and Restricted Boltzmann Machines (RBMs), which were recently proved to be interesting in time-series forecasting [51], [55], [56]. The *Pylearn2* [43] RNN framework is under development.

VII. CONCLUSION

No model can treat all circumstances precisely for a lot of verifiable burden information. The unpredictable change of the gas utilization was not really unsurprising, consequently the ANN model was assisted with powerful expense capability and with the notable ARIMA model. Albeit different papers introduced comparable models to estimate electric utilization, the crossover model introduced here is practically extraordinary in light of the fact that it centers around anticipating momentary gas utilization, which is extremely sporadic and not effectively unsurprising with exemplary strategies. Since the indicator is extremely exact (with RMSE from $8\ m^3$ in building 740-NTH to RMSE $2.5\ m^3$ in building 761-KMH as shown in TABLE IV), the exception system can undoubtedly identify weird ways of behaving characterizing an edge esteem in the certainty span without the need to have past instances of anomalies. The extent of this paper was guaging the profoundly sporadic gas utilization time series. In any case, it is accepted that comparative outcomes could likewise be gotten with the electric utilization time series. It is trusted that this could prompt another examination of the energy utilization in open structures.

REFERENCES

[1] E. C. Eurostat, *Energy balance sheets - 2010-2011 - 2013 edition*. Luxembourg: Publications Office of the European Union, 2013. [Online]. Available: http://epp.eurostat.ec.europa.eu/portal/page/portal/product_details/publication?p_product_code=KS-EN-13-001

[2] European Parliament and Council of the European Union, "Directive 2009/28/EC," Brussels, 2009.

[3] Nest, "Energy savings from nest white paper preview," https://nest.com/downloads/press/documents/efficiency-simulation-white-paper.pdf, 2014.

[4] S. Katipamula and M. R. Brambley, "Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, part i," *HVAC&R Research*, vol. 11, no. 1, pp. 3–25, 2005.

[5] S. Wu and J.-Q. Sun, "Cross-level fault detection and diagnosis of building hvac systems," *Building and Environment*, vol. 46, no. 8, pp. 1558–1566, 2011.

[6] D. Hawkins, "Identification of outliers," *London: Chap*, 1980.

[7] C. C. Aggarwal, *Outlier analysis*. Springer, 2013.

[8] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 5, pp. 631–645, 2007.

[9] C. M. Bishop *et al.*, "Neural networks for pattern recognition," 1995.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

[11] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[12] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John Wiley & Sons, 2005, vol. 589.

[13] H. Ferdowsi, S. Jagannathan, and M. Zawodniok, "A neural network based outlier identification and removal scheme," in *Prognostics and Health Management (PHM), 2013 IEEE Conference on*. IEEE, 2013, pp. 1–6.

[14] I. Khan, A. Capozzoli, S. P. Corgnati, and T. Cerquitelli, "Fault detection analysis of building energy consumption using data mining techniques," *Energy Procedia*, vol. 42, pp. 557–566, 2013.

[15] H.-x. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, 2012.

[16] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *Power Systems, IEEE Transactions on*, vol. 16, no. 1, pp. 44–55, 2001.

[17] T. Czernichow, A. Piras, K. Imhof, P. Caire, Y. Jaccard, B. Dorizzi, and A. Germond, "Short term electrical load forecasting with artificial neural networks," *Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 4, no. LRE-ARTICLE-1996-003, pp. 85–99, 1996.

[18] S. A. Kalogirou, "Artificial neural networks in energy applications in buildings," *International Journal of Low-Carbon Technologies*, vol. 1, no. 3, pp. 201–216, 2006.

[19] S. J. Nizami and A. Z. Al-Garni, "Forecasting electric energy consumption using neural networks," *Energy Policy*, vol. 23, no. 12, pp. 1097 – 1104, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0301421595001166

[20] J. W. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *Power Systems, IEEE Transactions on*, vol. 17, no. 3, pp. 626–632, 2002.

[21] P. A. González and J. M. Zamarreño, "Prediction of hourly energy consumption in buildings based on a feedback artificial neural network," *Energy and Buildings*, vol. 37, no. 6, pp. 595–601, 2005.

[22] A. H. Neto and F. A. S. Fiorelli, "Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption," *Energy and Buildings*, vol. 40, no. 12, pp. 2169–2176, 2008.

[23] E. D'Andrea, B. Lazzerini, and S. L. del Rosario, "Neural network-based forecasting of energy consumption due to electric lighting in office buildings," in *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*. IEEE, 2012, pp. 1–5.

[24] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[25] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Systems with Applications*, vol. 37, pp. 479–489, 2010.

[26] R. H. Brown and I. Matin, "Development of artificial neural network models to predict daily gas consumption," in *Industrial Electronics, Control, and Instrumentation, 1995., Proceedings of the 1995 IEEE IECON 21st International Conference on*, vol. 2. IEEE, 1995, pp. 1389–1394.

[27] A. Khotanzad, H. Elragal, and T.-L. Lu, "Combination of artificial neural-network forecasters for prediction of natural gas consumption," *Neural Networks, IEEE Transactions on*, vol. 11, no. 2, pp. 464–473, 2000.

[28] M. Adya and F. Collopy, "How e! ective are neural networks at forecasting and prediction? a review and evaluation," *J. Forecasting*, vol. 17, pp. 481–495, 1998.

[29] A. P. Douglas, A. M. Breipohl, F. N. Lee, and R. Adapa, "The impacts of temperature forecast uncertainty on bayesian load forecasting," *Power Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1507–1513, 1998.

[30] D. K. Ranaweera, G. G. Karady, and R. G. Farmer, "Effect of probabilistic inputs on neural network-based electric load forecasting," *Neural Networks, IEEE Transactions on*, vol. 7, no. 6, pp. 1528–1532, 1996.

[31] M. C. Mozer, "Neural net architectures for temporal sequence processing," 2007.

[32] M. B. Ohlsson, C. O. Peterson, H. Pi, T. S. Rognvaldsson, and B. P. Soderberg, "Predicting system loads with artificial neural networks–methods and results from" the great energy predictor shootout"," *ASHRAE Transactions-American Society of Heating Refrigerating Air-conditioning Engin*, vol. 100, no. 2, pp. 1063–1074, 1994.

[33] R. H. Dodier and G. P. Henze, "Statistical analysis of neural networks as applied to building energy prediction," *Journal of solar energy engineering*, vol. 126, no. 1, pp. 592–600, 2004.

[34] B. B. Ekici and U. T. Aksoy, "Prediction of building energy consumption by using artificial neural networks," *Advances in Engineering Software*, vol. 40, no. 5, pp. 356–362, 2009.

[35] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[36] G. Zhang, B. Eddy Patuwo, and M. Y Hu, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[37] R. J. Hyndman and Y. Khandakar, "Automatic time series for forecasting: the forecast package for r," Monash University, Department of Econometrics and Business Statistics, Tech. Rep., 2007.

[38] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[39] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 421–436.

[40] S. Lawrence, C. L. Giles, and A. C. Tsoi, "What size neural network gives optimal generalization? convergence properties of backpropagation," 1998.

[41] W. S. Sarle, "Stopped training and other remedies for overfitting," in *Proceedings of the 27th Symposium on the Interface of Computing Science and Statisfi (.'. _\'. pp. 352-360. Interface Foundation of North America, Fairfax Station. VA, USA*, 1995.

[42] K. Liano, "Robust error measure for supervised neural network learning with outliers," *Neural Networks, IEEE Transactions on*, vol. 7, no. 1, pp. 246–250, 1996.

[43] I. J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio, "Pylearn2: a machine learning research library," *arXiv preprint arXiv:1308.4214*, 2013.

[44] A. Zeileis and G. Grothendieck, "zoo: S3 infrastructure for regular and irregular time series," *arXiv preprint math/0505527*, 2005.

[45] B. G. Amidan, T. A. Ferryman, and S. K. Cooley, "Data outlier detection using the chebyshev theorem," in *Aerospace Conference, 2005 IEEE*. IEEE, 2005, pp. 3814–3819.

[46] S. Kajl, M. Roberge, L. Lamarche, and P. Malinowski, "Evaluation of building energy consumption based on fuzzy logic and neural networks applications," in *Proc of CLIMA*, 2000, p. 264.

[47] R. J. Hyndman, "Another look at forecast-accuracy metrics for intermittent demand," *Foresight: The International Journal of Applied Forecasting*, vol. 4, no. 4, pp. 43–46, 2006.

[48] R. Yao and K. Steemers, "A method of formulating energy load profile for domestic buildings in the uk," *Energy and Buildings*, vol. 37, no. 6, pp. 663–671, 2005.

[49] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, 2011.

[50] H. S. Malvar, *Signal processing with lapped transforms*. Artech House, 1992.

[51] E. Busseti, I. Osband, and S. Wong, "Deep learning for time series modeling," Technical report, Stanford University, Tech. Rep., 2012.

[52] B. Iglewicz, *Robust scale estimators and confidence intervals for location*. New York: Wiley, 1983.

[53] I. Mizera and C. H. Müller, "Location–scale depth," *Journal of the American Statistical Association*, vol. 99, no. 468, pp. 949–966, 2004.

[54] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[55] G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 1025–1032.

[56] I. Sutskever, "Training recurrent neural networks," Ph.D. dissertation, University of Toronto, 2013.

Optimized Service Function Deployment in Edge Computing Networks Using Deep Reinforcement Learning

1st Liuwei Huo
Institute of intelligent computing,
University of Electronic Science and Technology of China,
Chengdu, China
huoliuwei@163.com

2nd Bowen Zhu✉
School of Communication and Information Engineering,
University of Electronic Science and Technology of China,
Chengdu, China
acbogu@163.com

3rd Dongcheng Zhao
Tianfu Jiangxi Laboratory,
Chengdu, China
zhaodc11@gmail.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899693

***Abstract*—The deployment of service function chains (SFCs) in integrated edge Computing networks presents significant challenges, including optimizing resource utilization, minimizing latency, and reducing energy consumption. These challenges arise from the dynamic nature of workloads, varying network conditions, and complex service function dependencies.Conventional optimization approaches often struggle to handle these complexities, as they depend on static models and assumptions that may not apply in practical situations. This paper introduces an innovative solution using deep reinforcement learning (DRL) to overcome these limitations. By modeling the deployment problem as a Markov decision process, we utilize the DRL's capability to learn optimal policies through iterative trial and error interactions with the environment. Our method adaptively adjusts service function placement in real time, accounting for workload variability, service dependencies, and network conditions. Comprehensive simulations demonstrate that our proposed approach achieves significant improvements in terms of latency reduction, resource utilization, and energy consumption compared to conventional methods, and can provide a new solution for SFC deployment.**

***Index Terms*—edge computing networks, deep learning, deep reinforcement learning, service function deployment**

I. Introduction

Service Function Chaining (SFC) has emerged as a pivotal technology in modern network architectures, enabling dynamic and flexible deployment of network services. In edge computing networks, where computational resources are distributed close to the end-users, SFC plays a crucial role in enhancing service delivery, reducing latency, and improving resource utilization [1]–[3]. However, the complexity of deploying service functions optimally across distributed hosts poses significant challenges [4]. These challenges include the requirement for efficient resource management, real-time adaptation to varying workloads, and ensuring low latency and energy consumption. Conventional optimization techniques, such as First Fit, often fail to address these issues comprehensively due to their reliance on static models and assumptions that may not hold in dynamic and unpredictable network environments [5].

To overcome these limitations, this paper introduces a novel approach based on Deep Reinforcement Learning. DRL is well-suited for solving complex optimization problems by learning from interactions with the environment [6], [7]. By modeling the SFC deployment problem as a Markov Decision Process (MDP) [8], [9], we leverage DRL's ability to adaptively make decisions that optimize the placement of service functions. Our method considers real-time data on workload variability, service dependencies, and network conditions, allowing for more informed and dynamic deployment strategies.

The extensive simulations demonstrate that our proposed DRL-based method significantly outperforms traditional methods, such as the First Fit. Specifically, the simulation results show that our method can reduce latency, enhance resource utilization, lower energy consumption, and improve overall system performance. Then, we make a contribution to utilize machine learning methods to address the network optimization problems and highlight the potential of DRL in addressing the complexities of service function chaining in edge computing networks.

II. Problem Formulation and Modeling

In this section, we provide a detailed formulation and modeling of the SFC's deployment problem in edge computing networks. Then, we summarize the commonly used representation symbols in Table I.

Supported by the Science Foundation for Youths of Natural Science Foundation of Sichuan Provincial(2022NSFSC0936), the China Postdoctoral Science Foundation, No.72 General Fund(2022M720666), the Tianfu Jiangxi Laboratory (TFJX-ZD-2024-001)

TABLE I: Symbols of Optimizations

Symbol	Definition
R_c	Revenue coefficient for resource type c
η_h^c	Utilization rate of resource c on host h
r_f^c	Requested amount of resource c by service function f
x_h^f	Binary placement variable for function f in host h
ε	Energy cost coefficient
$W_h^{\max}$	Maximum power consumption of host h
$W_h^{\min}$	Minimum power consumption (idle power) of host h
d	Exponent of energy expenditure
L_f	Latency introduced by service function f
b_{fb}	Bandwidth used by service function f on host h
$B_h^{\max}$	Maximum bandwidth available on host h

A. Mathematical Formulation

In this scheme, our object is to maximize the total benefit for the system that deployed SFCs, including increasing resource benefits and reducing energy consumption. The objective function can be formally expressed as:

$$\sum_{h \in H} x_h^f \leq 1 \tag{1}$$

The constraint Eq.(1) ensures that each service function f can only be deployed on one host $h \in \mathcal{H}$. This is achieved by limiting the sum of the decision variables x_h^f (1 if service f is deployed on host $h \in \mathcal{H}$, 0 otherwise) which not exceed 1.

$$\sum_{f \in S} r_f^c \cdot x_h^f \leq R_h^c \tag{2}$$

Constraint Eq.(2) ensures that the total amount of resource requests allocated to any host $h \in \mathcal{H}$ does not exceed the maximum resource capacity R_h^c of the host. Here r_f^c represents the demand for resource c by service function f.

$$\sum_{h \in H} \sum_{f \in S} (L_f \cdot x_h^f) \leq L_{\max} \tag{3}$$

Constraint Eq.(3) controls the total delay of all service functions to not exceed the maximum allowed delay L_{max}. Here L_{max} is the maximum delay of service function f.

Then, we construct an objective function to maximize the benefits of deploying SFCs in the system, as Eq.(4):

$$\begin{aligned} \max(& \sum_{h \in \mathcal{H}} R_c \left[(1 - \eta_h^c) \cdot \sum_{f \in S} r_f^c x_h^f \right] \\ & - \sum_{h \in \mathcal{H}} \varepsilon \left[(W_h^{\max} - W_h^{\min}) \cdot \eta_h^c + W_h^{\min} \cdot x_h^f \right]^d) \end{aligned} \tag{4}$$

In the Eq.(4), R_c denotes the system's total resource benefit for resource c. Inspired by Bowen al. [10], we incorporate a service effect coefficient $(1 - \eta_h^c)$ to balance energy consumption and user experience.

$$\sum_{f \in S} b_{fb} \cdot x_h^f \leq B_{fb}^{\max} \quad \forall h \in H \tag{5}$$

Constraint Eq.(5) ensures that the total amount of bandwidth allocated to any host $h \in \mathcal{H}$ does not exceed the maximum bandwidth capacity B_{fb}^{max} of hosts. Here, b_{fb} represents the amount of bandwidth used by service function f.

III. Dynamic SFCs deployment optimization strategy based on deep reinforcement learning

In this section, we will explore how to use the DRL to solve the deployment problem of SFCs. We use the policy gradient method to optimize the constraints and use the Lagrangian relaxation method to transform the problem into an unconstrained problem. Then, we use deep reinforcement learning methods to maximize the system benefits. Specifically, the actor network uses a sequence-to-sequence (Seq2Seq) network [11], [12], and the critic network is a multi-layer deep neural network. We construct different network models consisting of RNN, LSTM, and GRU to evaluate the impact of different model architectures on the SFCs deployment performance.

A. Model Architecture

We present the architecture of our reinforcement learning agent's Seq2Seq model in Fig. 1. The Seq2Seq model comprises an encoder-decoder structure and a Bahdanau attention mechanism. When a SFC of variable length $n \in 1, ..., \mathcal{N}$ is input into the decision system, the agent determines where each VNF within the SFC should be deployed based on the network state S_t, and makes an appropriate deployment decision $\mathcal{A}_t$ (i.e., deployment location). The output of the hidden state in the decoder corresponds to the deployment decision for each VNF at the respective position in the encoder. The context vector C_t consists of the sum of the hidden states of the input sequence, weighted by the attention weights.

B. Constraint Optimization Using Policy Gradient Methods

At each cycle T, the agent observes the current state S_t, which represents the combined state of the input SFC and the physical network. The agent's policy network generates a probability distribution $\pi_\theta(a_t|s_t)$ for potential actions A_t based on the observed state. The agent samples actions a_t from this distribution, affecting the environment state transition $P(S_{t+1}|S_t, A_t)$ and obtaining an immediate reward R_t. Through repeated agent-environment interactions, the trajectories of observed states $s_0, s_1, \ldots, s_T$, actions taken $a_0, a_1, \ldots, a_{T-1}$, and rewards received $r_0, r_1, \ldots, r_{T-1}$ are accumulated to train the policy to maximize the long-term reward.

For an trajectory $\tau = (s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t)$, where $s_t \in \mathcal{S}$ is a state, $a_t \in \mathcal{A}$ is an action, and $r_t \in \mathbb{R}$ is an immediate reward at time step t, the probability of τ occurring under the agent's policy π_θ is defined as:

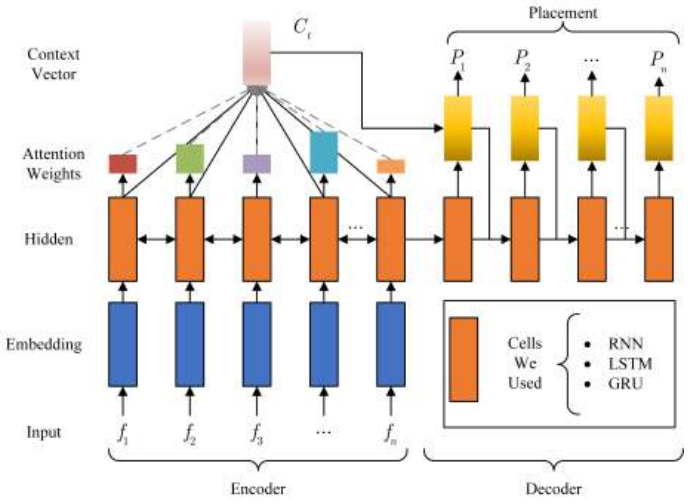


Fig. 1: Model architecture of our reinforcement learning agent

$$P(\tau|\theta) = \rho_0(s_0) \prod_{t=0}^{T-1} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t) \tag{6}$$

For the given trajectory τ, the cumulative reward accrued along τ can be expressed as:

$$R(\tau) = \sum_{t=0}^{T-1} r(s_t, a_t) \tag{7}$$

where $r(s_t, a_t)$ is the immediate reward for taking action a_t in state s_t. The final objective is to maximize the expected cumulative reward across all trajectories induced by the policy π_θ, formulated as:

$$J(\theta) = \mathbb{E}_{\tau \sim P(\tau|\theta)}[R(\tau)] \tag{8}$$

To learn the optimal policy parameters θ that maximize cumulative reward, we must calculate the gradient of the $J(\theta)$ with respect to θ, denoted $\nabla_\theta J(\theta)$ (Equation (9)). Then employs gradient ascent to update θ in the direction that increases $J(\theta)$ (Equation (10)).

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim P(\tau|\theta)}[\nabla_\theta log P(\tau|\theta) \cdot R(\tau)] \tag{9}$$

$$\theta_{t+1} = \theta_t + \alpha \cdot \nabla_\theta J(\theta) \tag{10}$$

where α represents the learning rate. According to Equation (6), the gradient of the policy function with respect to θ can be rewritten as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim P(\tau|\theta)}[\sum_{t=0}^{T-1} \nabla_\theta log \pi_\theta(a_t|s_t) \cdot R(\tau)] \tag{11}$$

However, as reward signals often entail noise and instability, directly employing cumulative rewards $R(\tau)$ induces substantial variance during training, impeding convergence. To mitigate this, we utilize advantage value function $A^\pi(s_t, a_t)$ rather than raw rewards. The advantage value function quantifies the relative benefit of taking action in state s_t compared to the average action, formulated as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim P(\tau|\theta)}[\sum_{t=0}^{T-1} \nabla_\theta log \pi_\theta(a_t|s_t) \cdot A^\pi(s_t, a_t)] \tag{12}$$

The advantage value function is formulated as:

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s) \tag{13}$$

where $Q(s, a)$ represents the state-action value, defined as the expected long-term cumulative reward obtained after taking action in state s and continuing under policy π. $V(s)$ gives the state value, indicating the expected return from state s following policy π. For each state s_t encountered, we estimate the state value $V(s_t)$ under policy π with parameters ϕ using Monte Carlo sampling, formulated as:

$$V^\pi(s_t; \phi) \approx \frac{1}{N} \sum_{i=1}^{N} R(\tau_i) \tag{14}$$

where N is the number of trajectories sampled from state s_t. Subsequently, we refines the value function parameters ϕ via gradient descent to minimize the loss between estimated state values $V(s_t)$.

$$\phi_{t+1} = \phi_t + \beta \nabla_\phi \mathcal{L}(\phi) \tag{15}$$

the loss function $\mathcal{L}$ of the value function network can be defined as:

$$\mathcal{L}(\phi) = \frac{1}{N} \sum_{i=1}^{N} (R(\tau_i) - V^\pi(s_0^i, \phi))^2 \tag{16}$$

In this paper, the training process of our proposed algorithm as Algorithm 1 shows.

IV. Experimental Evaluation

In this section, we comprehensively evaluate the performance of the proposed method in two relevant experimental scenarios. First, we analyze the model convergence behavior during training to determine learning efficiency. Second, we examine the deployment error rate of the model. Third, we evaluate the system benefits to highlight the excellent VNF deployment capabilities of the proposed algorithm. In these scenarios, we compare the proposed model using RNN, LSTM and GRU units with state-of-the-art baseline algorithms. Empirical results show that the proposed algorithm provides significant system-wide advantages, including accelerated learning, reduced deployment errors, and improved system profitability.

Algorithm 1: Training Process of proposed Algorithm

Data: parameters θ, parameters ϕ
Result: Trained policy network and value function network

for *each training epoch* **do**
  Reset environment, obtain initial state s_0;
  for $t = 0$ *to* $T-1$ **do**
    Generate action probabilities $\pi_\theta(a_t|s_t)$;
    Sample action a_t from the distribution $\pi_\theta(a_t|s_t)$;
    Execute action a_t, observe next state s_{t+1} and immediate reward r_t;
  end
  Compute cumulative rewards $R(\tau)$
  $A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$;
  Compute policy gradient $\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim P(\tau|\theta)} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot A^\pi(s_t, a_t) \right]$;
  $\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta J(\theta)$;
  for $i = 1$ *to* N **do**
    Sample a trajectory $\tau^{(i)}$ from the replay buffer;
    Compute state value estimate $V^\pi(s_0^{(i)}, \phi)$;
    Compute value function loss $L(\phi) = \frac{1}{N} \sum_{i=1}^{N} (R(\tau^{(i)}) - V^\pi(s_0^{(i)}, \phi))^2$;
    $\phi_{t+1} \leftarrow \phi_t + \beta \nabla_\phi L(\phi)$;
  end
end

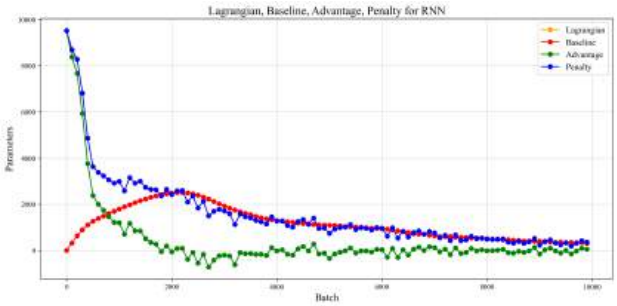
A. Model Convergence

Figures 2 and 3 illustrate the training process of the proposed algorithm. Specifically, Figures 2a, 2b, and 2c depict the training processes for RNN, LSTM, and GRU-based models, respectively. Initially, the agent generates high-penalty strategies. As training progresses, weights are updated using stochastic gradient descent, refining the strategies and reducing penalty values. In smaller problem instances, the penalty value approaches zero, and the Lagrangian value approximates the reward value, indicating the system's ability to meet most business requests when resources are sufficient. The decreasing trend in overall Lagrangian values signifies successful parameter fine-tuning and goal achievement. Near-zero penalties suggest minimal constraint violations, validating the feasibility of the migration deployment strategy.

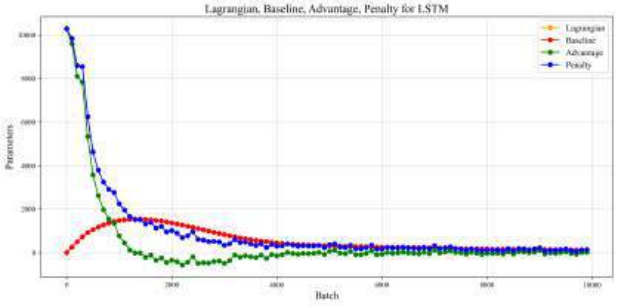
Figures 3a, 3b, and 3c show the loss values for the RNN, LSTM, and GRU-based models, respectively. These figures indicate that as training progresses, the loss values approach zero without signs of over-fitting or under-fitting, demonstrating satisfactory training outcomes.

B. Economic Gains Evaluation

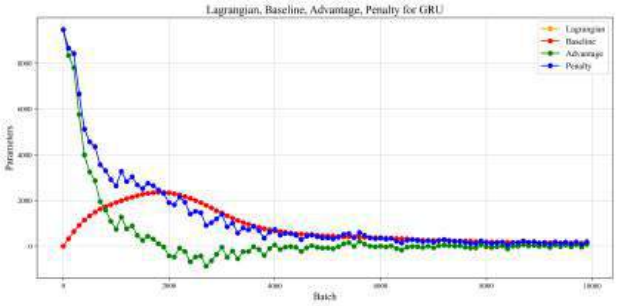
Subsequently, we undertake a rigorous comparative analysis of the economic benefits generated by the four distinct algorithms under scrutiny. To fortify the robustness and credibility of our conclusions, we meticulously designed an experimental framework comprising 128 independent trials. This extensive array of experiments was strategically devised to minimize potential biases or confounding variables that could arise from limited sampling.



(a) learning history of rnn-based model



(b) learning history of lstm-based model

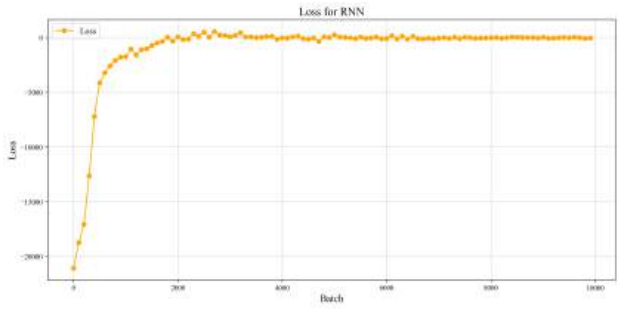


(c) learning history of gru-based model

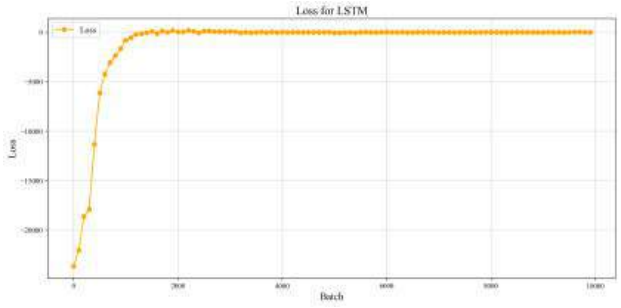
Fig. 2: learning history

Within the context of each experiment, the economic benefit is meticulously quantified as the revenue accrued from users for resource utilization, offset by the net energy consumption costs incurred under the VNF allocation strategy dictated by the respective algorithm.

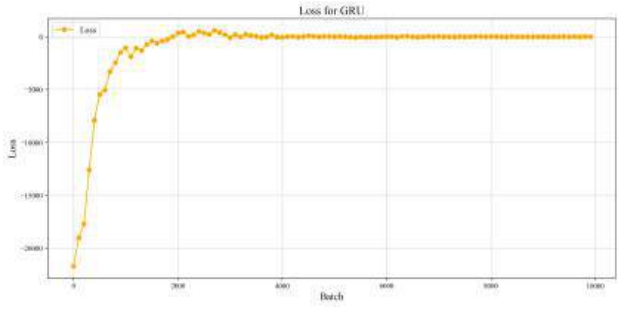
The empirical findings, derived from this exhaustive experimental setup, are visually encapsulated in Fig. 4, which provides a lucid comparison of the economic benefits yielded by each algorithm, thereby facilitating a transparent assessment of their relative performances. Notably, it is discernible from the figure that all three proposed algorithms demonstrate superior economic benefits when juxtaposed with the baseline algorithm.



(a) loss of rnn-based model



(b) loss of lstm-based model



(c) loss of gru-based model

Fig. 3: The loss of the models for training

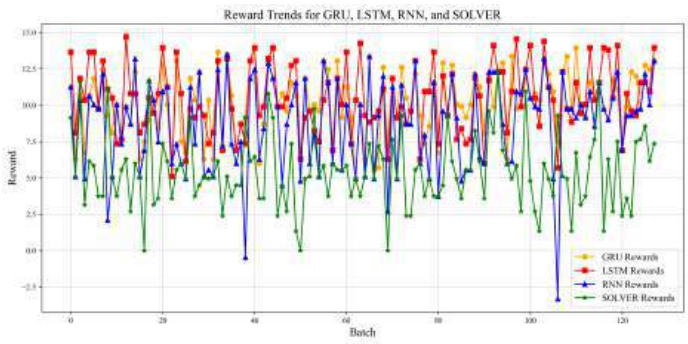


Fig. 4: Economic gains comparison

V. Conclusions

In this study, we address the limitations inherent in traditional methods for solving service function chaining problems. Our objective is to maximize the cumulative profits of cloud service providers by proposing the application of reinforcement learning to tackle these complex deployment challenges. To achieve this, we formulate the problem as a Markov Decision Process, enabling the use the method of DRL. Through rigorous testing on various experimental scenarios, our results demonstrate that the proposed algorithm exhibits robust model learning performance and significantly enhances system benefits.

References

[1] L. Kong, J. Tan, J. Huang, G. Chen, S. Wang, X. Jin, P. Zeng, M. Khan, and S. K. Das, "Edge-computing-driven internet of things: A survey," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–41, 2022.

[2] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, and J. Cao, "Edge computing with artificial intelligence: A machine learning perspective," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[3] P. McEnroe, S. Wang, and M. Liyanage, "A survey on the convergence of edge computing and ai for uavs: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 435–15 459, 2022.

[4] M. Pattaranantakul, C. Vorakulpipat, and T. Takahashi, "Service function chaining security survey: Addressing security challenges and threats," *Computer Networks*, vol. 221, p. 109484, 2023.

[5] L. Yang and A. Shami, "Iot data analytics in dynamic environments: From an automated machine learning perspective," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105366, 2022.

[6] F. Liu and X. Li, "Integrating deep reinforcement learning with evolutionary algorithms for advanced optimization in smart city energy management," *IEEE Access*, 2024.

[7] J. Liu, M. Ahmed, M. A. Mirza, W. U. Khan, D. Xu, J. Li, A. Aziz, and Z. Han, "Rl/drl meets vehicular task offloading using edge and vehicular cloudlet: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8315–8338, 2022.

[8] J. Ran, W. Wang, and H. Hu, "Dynamic service function chain deployment and readjustment method based on deep reinforcement learning," *Sensors*, vol. 23, no. 6, p. 3054, 2023.

[9] S. Guo, Y. Du, and L. Liu, "A meta reinforcement learning approach for sfc placement in dynamic iot-mec networks," *Applied Sciences*, vol. 13, no. 17, p. 9960, 2023.

[10] Z. Chen, B. Zhu, and C. Zhou, "Container cluster placement in edge computing based on reinforcement learning incorporating graph convolutional networks scheme," *Digital Communications and Networks*, 2023.

[11] Z. Chen, Z. Zhang, Z. Xiao, Z. Yang, and K.-K. Wong, "Viewing channel as sequence rather than image: A 2-d seq2seq approach for efficient mimo-ofdm csi feedback," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 7393–7407, 2023.

[12] S. Gao, S. Zhang, Y. Huang, J. Han, H. Luo, Y. Zhang, and G. Wang, "A new seq2seq architecture for hourly runoff prediction using historical rainfall and runoff as input," *Journal of Hydrology*, vol. 612, p. 128099, 2022.

Design and research of intelligent robot 3D recognition processing system

Liang Wang
Zhejiang Business Technology Institute
Ning Bo, 315100, China
ahjd09wl@126.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899485

***Abstract*—With the continuous advancement of artificial intelligence technology, intelligent robots are increasingly and extensively utilized in industrial production, daily life, as well as detection and security. This paper presents the design and research of a 3D recognition and processing system for intelligent robots, with the aim of enhancing the autonomous cognition and decision-making ability of robots in complex environments. Firstly, the system is based on advanced 3D sensing technology, through the combination of multi-line LiDAR for assisted acquisition of point cloud data and stereo vision. Subsequently, a deep learning network architecture of global features and local scale features is designed for the feature extraction and classification of point cloud data. Finally, through training and optimizing the model, accurate 3D object recognition is achieved. The experimental results demonstrate that the classification accuracy of the system on the training set and the test set has reached a high level, and it can well recognize different categories of three-dimensional objects.**

Keywords: Point cloud, Three-dimensional recognition, Computer vision, Deep learning

I. INTRODUCTION

With the swift advancement of science and technology, intelligent robots have progressively emerged as a crucial component of people's lives. In diverse application scenarios, intelligent robots are required to possess potent environmental perception and interaction capabilities, and the three-dimensional recognition system constitutes one of the key technologies for achieving this ability. The 3D recognition system can furnish the robot with 3D information regarding the surrounding environment, thereby facilitating better understanding and interaction by the robot. Nevertheless, numerous issues still persist in the current research, such as the constraints of 3D data acquisition equipment, the complexity of data processing algorithms, and the precision of feature extraction. Hence, it is imperative to conduct in-depth studies on the construction and design of the 3D recognition system for intelligent robots, with a view to providing valuable references for research in related fields.

This paper endeavors to elaborate on the construction and design process of the 3D recognition system for intelligent robots in detail and offer useful references for research in related fields. Firstly, the fundamental principle of the 3D recognition system will be introduced, encompassing 3D data acquisition, data processing, feature extraction, and recognition algorithms. On this basis, the merits and demerits of the existing technology will be analyzed, and the pressing issues in the current research will be pointed out. Secondly, the construction process of the 3D recognition system for intelligent robots will be expounded, including system architecture design, hardware selection, software design and development. The selection principles and installation methods of key equipment such as 3D cameras and depth sensors, as well as key technologies in system software development, will be presented. Thirdly, the design of the 3D recognition system for intelligent robots will be introduced, encompassing system function module design, database design, human-computer interface design, and so forth. The design principles of the system function modules, such as real-time performance, accuracy, and robustness, will be emphasized. Data storage methods and data query methods need to be considered in database design; and the design principles of the human-computer interaction interface, such as ease of use and friendliness, will be addressed. Finally, the completed 3D recognition system for intelligent robots will be tested and evaluated, including system performance tests, functional tests, and user experience tests. Through the analysis of the test results, a basis will be provided for the optimization and improvement of the system.

Through in-depth discussions on the construction and design of the 3D recognition system for intelligent robots, it is hoped to offer valuable references for research in related fields. With the continuous progress of technology, the 3D recognition system for intelligent robots will be extensively applied in an increasing number of application scenarios, bringing greater convenience to people's lives. Therefore, an in-depth understanding of the construction and design of the 3D recognition system for intelligent robots is of significant importance for promoting the development of the robot field.

II. SYSTEM ARCHITECTURE AND PRINCIPLE

The three-dimensional vision system is designed to capture and process three-dimensional information in the environment, aiming to achieve the understanding and recognition of objects and scenes. The architecture typically consists of several key components, including a sensor module, a data processing module, and an application interface. The following presents the basic architecture of the 3D vision system. Firstly, the sensor module is the core of the 3D vision system, being responsible for acquiring the 3D

979-8-3315-2931-4/24/$31.00 ©2024 IEEE

data of the scene. Common sensors encompass depth cameras, LiDAR, structured light sensors, and stereo cameras. A depth camera can obtain the depth information of each pixel point by calculating the distance between the camera and the depth map; LiDAR transmits a laser beam and measures the reflection time to obtain high-precision three-dimensional point cloud data. These sensors can effectively capture the spatial structure of the scene, providing abundant depth and geometric information. The structure principle diagram is shown in Fig 1.

Fig. 1. 3D vision light source acquisition system architecture diagram

Secondly, the acquired data is processed and analyzed by the data processing module. This module typically comprises several significant steps: data preprocessing, feature extraction, and model construction. In the data preprocessing stage, the system will undertake filtering, denoising, and data fusion of the original data acquired by the sensor to enhance the data quality. In the feature extraction stage, the geometric features and shape information of objects are mainly extracted through algorithms (such as edge detection, voxel grid method, etc.) to furnish the basis for subsequent analysis. Subsequently, a 3D model is constructed based on the extracted features, and a complete 3D scene is formed by employing point cloud registration and surface reconstruction techniques.

Finally, 3D vision systems frequently interact with other systems via application interfaces. The application interface can output the processed 3D data to different application levels, such as robot navigation, object recognition, human-computer interaction, etc. By integrating with deep learning models, the system can accomplish more complex tasks, such as real-time object detection, semantic segmentation, etc., significantly enhancing the machine's ability to perceive the environment.

In this study, the Pro S 800c Enhanced camera is utilized to collect images of target objects and obtain pose information. The camera has a binocular structure, and the optical machine projects the structural light onto the object to be photographed. The image is collected by the camera, and then the 3D image in the form of point cloud is generated through calculation. The near-end field of view of this camera is 360 × 250@0.5m, and the far-end field of view is 740 × 480@1.0m. The camera is capable of collecting images within this field space.As is shown in Fig2.

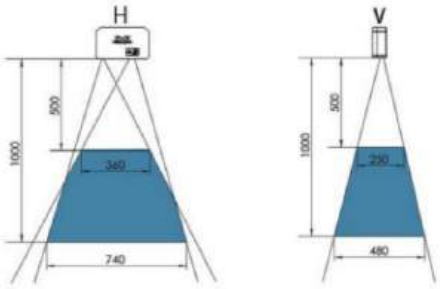


Fig. 2. The camera view

Considering that the working environment of 3D vision is more complex, with the majority being factory workshops, and the factory workshops involving oil, dust, crushing, high temperatures, humidity, and other harsh conditions depending on different work tasks and the size of the workshop, higher requirements are imposed on the adaptability of the handle at this point. Ordinary embedded control circuits are reluctant to control the entire system, unable to meet the complex environment, and are not effectively compatible with the later upgrade of the system or the loading of more sensors and different communication methods. Therefore, in combination with the actual situation, we comprehensively consider using the programmable logic controller commonly used in industrial control as the control system of this study.

III. SYSTEM MODEL ANALYSIS AND CALIBRATION

A. Camera model

3D camera model is a mathematical model that describes how an object in 3D space is projected onto a 2D image plane through the camera lens Faugeras, O. D. . (1986).[1]. It involves the transformation of multiple coordinate systems, including world coordinate system, camera coordinate system, image coordinate system and pixel coordinate system Zhang, S. . (2005).[2]. The model diagram is shown in Figure 3. Through the transformation of these coordinate systems, the mapping from three-dimensional space to two-dimensional plane can be realized, so as to obtain the three-dimensional information of the object.

Liu Xiaoli, Liu Haishan, Zhang Xiaojie, & Tang Qijian. (2022).[3] Suppose that a point is located in a three-dimensional space, its position coordinates in the world coordinate system and the camera coordinate system are given respectively, such as $X_w = (X_w, Y_w, Z_w)^T$ and $X_c = (X_c, Y_c, Z_c)^T$, and the process of conversion from the world coordinate system to the camera coordinate system is regarded as a transformation that keeps the shape and size

unchanged, then the coordinate relationship of the point in the two coordinate systems can be expressed by formula (1) :

$$X_c = RX_w + T \quad (1)$$

Here, R is an 3×3 -orthogonal matrix representing the rotation transformation from the world coordinate system to the camera coordinate system. T is a vector of 3×1 , representing the corresponding translation transformation.Together, R and T constitute the external parameters of the camera and are used to describe the rigid body transformation relationship between the world coordinate system and the camera coordinate system.

The conversion of camera coordinate system to image coordinate system is a projection transformation process from three dimensional space to two dimensional plane. In this process, the point X_c in three-dimensional space will form a projection point on the image plane. Assuming that the coordinate of the projection point is $x_c = (x_c, y_c)^T = (X_c / Z_c, Y_c / Z_c)^T$, then it can be calculated from the point coordinate formula (2) in three-dimensional space through certain mathematical relations:

$$\lambda \tilde{x}_c = [I \mid 0]\tilde{X}_c \quad (2)$$

For the scale factor of λ , I matrix for the unit.

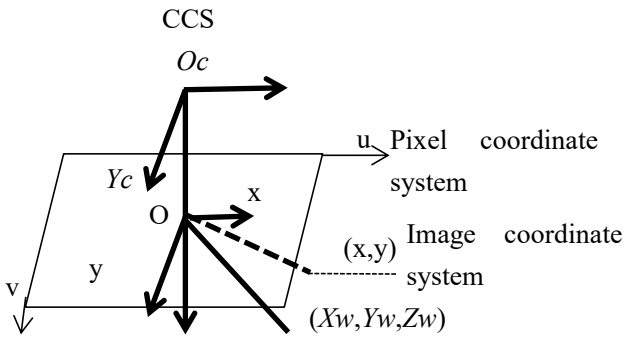


Fig.3. The camera model

With the internal parameters of the camera, we can convert the coordinate x_c in the physical coordinate system of the image to the coordinate mc in the pixel coordinate system of the image, which is expressed in the form (uc, vc) T. This process allows us to convert physical position information to pixel-level position information based on the inherent properties of the camera:

$$\tilde{m}_c = K_C \tilde{x}_c, K_C = \begin{bmatrix} f_x & a & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

B. Principle of binocular calibration

In order to realize the coordinate conversion from the target point of the image to the captured point on the actual object, it is necessary to have accurate internal and external parameter information of the camera. Internal parameters are the basic parameters inside the camera, including lens focal length, distortion and so on. Generally, the internal parameters of the camera have been calibrated and saved in the camera.

Camera external parameters represent the pose conversion relationship between robot and camera, that is, hand-eye relationship, so camera external parameters calibration is called robot hand-eye calibration. The relative pose of the robot and the camera is not fixed in different use scenarios, so the hand-eye relationship between the camera and the robot needs to be calibrated at the work site. Then, the photo sets of the calibration board with different height and Angle positions are taken as shown in Figure 4:

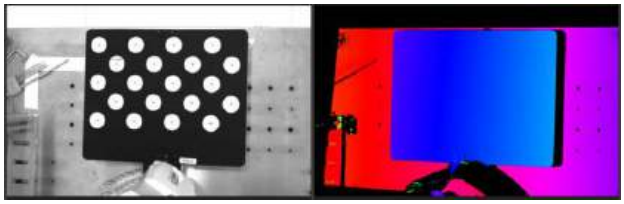


Fig.4. Calibration plate imaging photos

The end of the robot is connected with the calibration plate of known size through A flange, and the coordinate A of each mark point on the calibration plate relative to the base coordinate of the robot can be obtained. The image of each dot on the calibration board can be obtained by taking pictures of the camera, and the coordinate B of the optical center of the camera relative to each mark point on the calibration board can be obtained. The pose relation X between the optical center of the camera and the base coordinates of the robot is a substitute. A, B, and X form a closed loop, forming an equation in which the unknown X can be solved. By changing the pose of the calibration plate relative to the camera by the mobile robot, several groups of equations can be obtained, and the values of these equations can be fitted and optimized, and finally the optimal X value can be obtained. The pose relationship is shown in Figure 5 below.

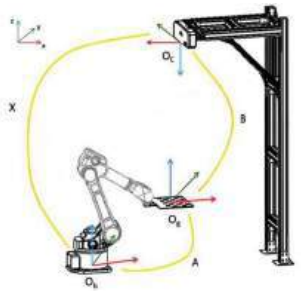


Fig.5. Robot and camera position

When the TCP touch method is used for calibration, the calibration plate is placed on the working plane, and the end of the robot is installed with A known TCP point, and the circle point of the calibration plate is touched, where A and B are known, and the value of X is solved.

In practice, coordinate A can be obtained in the following three ways:

The position relationship between the calibration plate and the end of the flange is known (the three-point method or the size of the known connector is calculated), then A can be calculated directly;

If the position relationship between the calibration plate and the end of the flange is unknown, the position relationship between the calibration plate and the end of the flange is calculated by numerical method through A series of relative movements of the calibration plate during the calibration process, and then A is calculated;

If the end of the calibration board and the robot are not fixed, the value of A can be calculated by touching the mark point of the calibration board with the known TCP coordinates. The above three methods correspond to three different methods of obtaining calibration data.

ETH method Zhao Liming, Long Dazhou, Xu Xiaodong, Zhang Yi, Feng Yang, & Li Fangfang. (2021).[4] is used to calibrate the pose relationship between the optical center of the camera and the base coordinate of the robot. If the base coordinates of the robot or the camera move, the corresponding external parameters will change accordingly, and the hand-eye relationship needs to be re-calibrated.

IV. Design for 3D Visual Recognition

In this paper, Mech-Vision industrial vision software is used to conduct three-dimensional detection of screws and nuts of different specifications. Through the three-dimensional point cloud processing mentioned above, it is verified whether ETH calibration has clear signals for robot grasping. The camera test flow chart is shown in Figure 6 below:

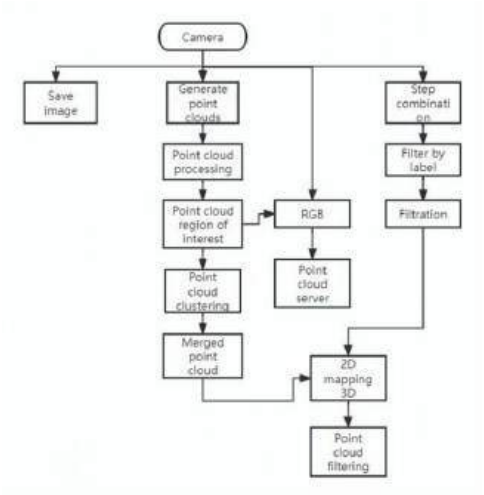
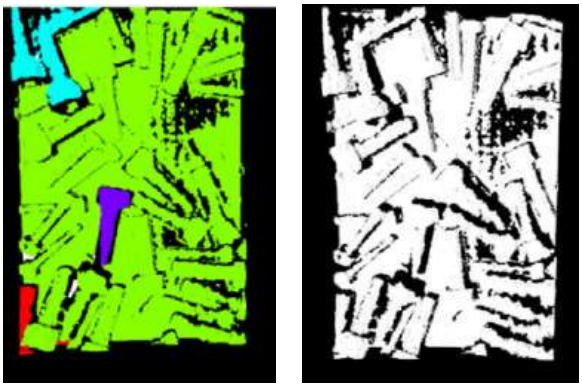


Fig. 6. Design flow chart

A. Variable domain fuzzy PID control technique

First, modify the IP address of the camera to ensure that the communication parameters are online. The parameter group name and camera parameters can be set according to the target part. If no parameter is set, the default parameter group is used. The point cloud image is generated using the generated point cloud command based on the depth image and color image obtained by the photo, and the point cloud processing (GPU) command is added from the 3D pre-processing command classification. The generated point cloud data is passed to point cloud processing (GPU) instructions. This operation is used to calculate the phase point cloud, add the point cloud clustering instruction from the 3D general processing instruction classification, and pass the data processed by the point cloud to the point cloud clustering instruction. This operation is used to distinguish the parts from the background and segment the individual parts, and then pass the point cloud data obtained from the point cloud clustering to the merged point cloud. This operation is used to merge various part point clouds obtained by clustering segmentation.shown in Figure 7:



（a）Point cloud clustering （b）Point cloud merging

Fig.7. Point cloud identification generation

B. Point cloud processing

Point cloud matching refers to comparing one set of point cloud data with another set of point cloud data to find similarities or correspondences between them. Point cloud matching has applications in many fields, such as computer vision, robot navigation, 3D modeling, and pose estimation. Point cloud matching can help us understand the relationship between two point clouds, for example, determining whether two scenes come from the same environment, or calculating the relative position and orientation between two objects.

There are many methods for point cloud matching, and the following are some common methods:

Feature-based matching: This method first extracts feature descriptors from point clouds, such as normal vectors and curvature, and then uses specific distance or similarity

measures to assess the similarity between these descriptors. Common feature descriptors include Signature of Histograms of OrienTations, Fast Point Feature Histograms, and Point Feature Histograms.

Matching based on nearest neighbor search: This method finds matches by searching for the nearest neighbor points in the point cloud Tan Junxiang, Li Shaoda, & Yang Ronghao. (2014).[5]. Common algorithms include Iterative Closest Point, Fundamental Matrix Based Registration, and Normal Distributions Transform.

Global optimization-based matching: This method transforms the point cloud matching problem into a global optimization problem, and achieves matching by minimizing an energy function. Yang, J. , Li, H. , & Jia, Y. . (2013).[6] proposed some common algorithms such as Global ICP, Trivial Line Search, and Estimation of Analytic Surface Properties.

Matching based on deep learning: In recent years, deep learning technology has also been widely applied in the field of point cloud matching. Some researchers have proposed point cloud matching algorithms based on convolutional neural networks, autoencoders, and graph neural networks.

Point cloud culling refers to the removal of noise and unwanted parts of the point cloud in order to improve the quality and availability of high point cloud data Zhao Shaoan.[7] Point cloud data often contains a lot of noise and outliers, which can lead to errors in subsequent processing and recognition algorithms. Therefore, point cloud culling is usually required before point cloud processing.

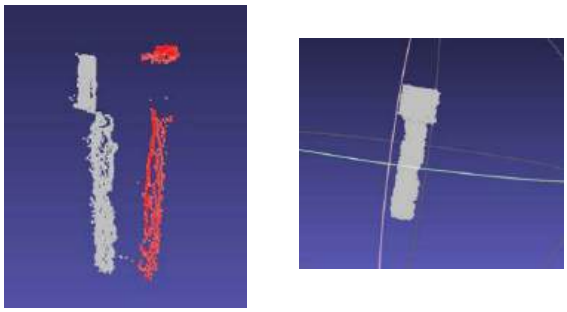
There are many methods of point cloud removal, and the following are some common methods:

Distance-based elimination: This type of method assumes that nearby points have similar propertiesXie Ze-Xiao, & Xu Shang. (2010).[8], so points that are far away can be treated as outliers and eliminated. For each point P_i, calculate its distance d to a given point P_c or region R. If d is greater than the threshold T, Pi is removed.

$$d = \left| P_c - P_i \right| \tag{4}$$

Statistics-based elimination: This type of method uses statistical analysis to identify and eliminate outliers. For each point P_i, calculate the average distance d of its k nearest neighbors. If d does not conform to the preset statistical distribution, P_i is eliminated. For example, Li Baoshun, Cen Hongyan, Bao Yaping, & Li Yifeng. (2014).[9] principal component analysis (PCA) or local neighborhood feature analysis (LOF) are used to identify outliers.

Deep learning-based culling: In recent years, deep learning technology has also been widely used in the field of point cloud culling. Some researchers, such as Wu Yi-Quan, Chen Huixian, & Zhang Yao. (2024).[10], have proposed point cloud removal algorithms based on convolutional neural networks, autoencoders and graph neural networks, etc. Based on the characteristics of structured light imaging, the point cloud of the target part usually contains a part of invalid point cloud, which is similar to the shadow part of ordinary light imaging and needs to be removed. As is shown in Figure 8:



（a）Before removal （b）After removal

Fig.8. Point cloud culling

V. CONCLUSION

The design of intelligent robot 3D recognition and processing system aims to solve the problem of 3D recognition and location of object in complex environment. This system is based on machine vision technology, combined with depth camera, laser scanner and other sensors, to achieve real-time three-dimensional data acquisition and processing of the target object.

1) In the process of system design, we first completed the selection and integration of hardware, including the selection of high-precision 3D sensors, high-performance computing platforms and stable robot arm system. Then, we design a 3D data acquisition module based on triangulation principle to ensure the accuracy and real-time data.

2) In order to verify the performance of the system, we conducted several experiments. In the experiment, the system successfully identified and positioned screw target objects in a variety of complex scenes, including with different shapes and textures. At the same time, we also test the recognition ability of the system under noise, occlusion and other conditions, and the results show that the system has strong robustness and adaptability.

3) In addition, we also apply the system to the actual industrial production line, and realize the automatic grasp and assembly of the target object. The experimental results show that the system can effectively improve production efficiency and quality, and reduce labor cost.

REFERENCES

[1] Faugeras, O. D. . (1986). The calibration problem for stereoscopic vision. Proc Cvpr Miami Beach Fl, 42(12), 195-213.

[2] Zhang, S. . (2005). High-resolution, real-time 3-d shape measurement. Optical Engineering, 45.

[3] Liu Xiaoli, Liu Haishan, Zhang Xiaojie, & Tang Qijian. (2022). A binocular 3D reconstruction method and system based on line laser galvanometer scanning. CN202210648000.7.

[4] Zhao Liming, Long Dazhou, Xu Xiaodong, Zhang Yi, Feng Yang, & Li Fangfang. (2021). Binocular 3d laser scanning imaging correction method for industrial robot processing trajectory. Journal of Intelligent Systems.

[5] Tan Junxiang, Li Shaoda, & Yang Ronghao. (2014). A comparative study of tree structure K-nearest neighbor search based on iterative nearest point matching algorithm. Science of Surveying and Mapping, 39(4), 4.

[6] Yang, J. , Li, H. , & Jia, Y. . (2013). Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. Proceedings of the 2013 IEEE International Conference on Computer Vision. IEEE.

[7] Zhao Shaoan. Research on synchronous positioning and composition algorithm of mobile robot based on three-dimensional laser point Cloud. (Doctoral dissertation, University of Electronic Science and Technology of China).

[8] Xie Ze-Xiao, & Xu Shang. (2010). icp and its improved algorithm in 3D point cloud data Mosaic. Journal of Ocean University of China (Natural Science Edition), 040(001), 99-103.

[9] Li Baoshun, Cen Hongyan, Bao Yaping, & Li Yifeng. (2014). Point cloud data segmentation algorithm based on plane extraction. Computer Applications and Software, 31(7), 5.

[10] Wu Yi-Quan, Chen Huixian, & Zhang Yao. (2024). Research progress of 3D point cloud processing based on deep learning. Chinese Journal of Lasers, 51(5), 0509001.

Comparative Analysis of Hyperparameter Tuning Methods in Classification Models For Ensemble Learning

1st Hamzah Dabool
Big Data Analytics Center
United Arab Emirates University
Alain, UAE
hamzah.dabool@uaeu.ac.ae

2nd Hany Alashwal
Big Data Analytics Center
United Arab Emirates University
Alain, UAE
halashwal@uaeu.ac.ae

3rd Hamda Alnuaimi
Big Data Analytics Center
United Arab Emirates University
Alain, UAE
202101556@uaeu.ac.ae

4th Asma Alhouqani
Big Data Analytics Center
United Arab Emirates University
Alain, UAE
201908148@uaeu.ac.ae

5th Shaikha Alkaabi
Big Data Analytics Center
United Arab Emirates University
Alain, UAE
202201302@uaeu.ac.ae

6th Amal Al Ahbabi
Big Data Analytics Center
United Arab Emirates University
Alain, UAE
201912228@uaeu.ac.ae

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899492

***Abstract*—Hyperparameter tuning plays a critical role in optimizing machine learning models, directly impacting their accuracy and generalization capabilities. In this paper, we implement and compare four prominent hyperparameter tuning algorithms: Grid Search, Random Search, Bayesian Optimization, and Genetic Algorithm. Our goal is to evaluate these methods on multiclass classification task, assessing them based on tuning time, computational complexity, accuracy score, and ease of use. Through an extensive experimental analysis, we identify the strengths and limitations of each approach, providing insights into their ideal use cases. The results reveal trade-offs between exhaustive search methods like Grid Search, which offer higher accuracy at the cost of time, and more efficient alternatives like Random Search and Bayesian Optimization, which balance exploration and exploitation. Genetic Algorithms, while less commonly used, show potential in discovering global optima.**

***Index Terms*—Hyperparameters, Hyperparameter tuning, CrossValidation, XGBoost, Grid Search, Random Search, Bayesian Search, Genetic Search**

I. Introduction

Hyperparameters are settings or configurations of a machine learning (ML) model that are chosen before training begins. Unlike normal parameters, such as the weights in a neural network, which the model learns and adjusts during training, hyperparameters remain fixed throughout the process. In other words, hyperparameters guide how the model learns and directly influence its performance and complexity [1].

Hyperparameter tuning is the process of finding the best settings for a ML model to improve its performance. Since hyperparameters are chosen before training, tuning involves testing different values to see which combination works best. This is often done using statistical and combinatorial methods, where the model is trained and evaluated multiple times with different hyperparameter settings [2].

K-fold cross-validation (K-fold CV) is a technique used to evaluate the performance of a ML model. In this method, the dataset is split into K equal parts or ”folds.” The model is trained on K-1 folds and tested on the remaining fold. This process is repeated K times, with each fold serving as the test set once. The results are averaged to give a more reliable measure of the model’s performance.K-fold CV helps reduce the risk of overfitting and provides a better understanding of how the model will perform on unseen data [3].

II. Related work

The study by [4] is the most closely related to our work, where the authors applied applied hyperparameter tuning to various machine learning models, including Decision Tree, Gaussian Naive Bayes, Random Forest, LightGBM, CatBoost, and XGBoost. They explored which hyperparameter optimization techniques were most suited for each algorithm. However, their approach is flawed, as effective hyperparameter tuning should be guided by the nature of the dataset and a clear understanding of which hyperparameters to focus on, along with their potential value ranges. Without this, the search space for optimal hyperparameter values becomes essentially infinite, irrespective of the chosen tuning technique. Moreover, following hyperparameter tuning, the model’s performance unexpectedly dropped, and their baseline Random Forest model with default settings outperformed all other tuned models. This result underscores the importance of a more data-driven and targeted hyperparameter search strategy, rather than a generic approach, which can sometimes lead to suboptimal outcomes.

III. Methodology

A. Motivation

Moreover, in existing studies published by other researchers, there is often a lack of comprehensive discussion on all possible hyperparameter tuning methods. They usually do not address the specific scenarios in which a particular tuning method is favorable. Instead, their focus has been primarily on the final accuracy of each tuning method individually. This narrow emphasis overlooks important factors such as the time required and the expected rate of improvement—critical resources needed to solve an optimization problem like hyperparameter tuning. All of these aspects contribute significantly to understanding how and when improvements can be expected.

B. Dataset

The dataset used in this research is a carefully merged compilation from three major sources: ADNI1, ADNI2, and ADNIGo. It has been specifically designed to support a comprehensive analysis across five distinct classes: 'CN' (Cognitively Normal), 'AD' (Alzheimer's Disease), 'LMCI' (Late Mild Cognitive Impairment), 'SMC' (Significant Memory Concern), and 'EMCI' (Early Mild Cognitive Impairment). The dataset contains 41 features, both categorical and numerical, with a total of 13,900 samples that provide valuable insights into these conditions. However, a key challenge we face is the presence of 250,480 missing values, which account for about 43% of the entire dataset.

C. Machine learning Algorithm

In this paper, we will focus on the XGBoost algorithm, an optimized distributed ensemble learning gradient boosting model known for its high efficiency and flexibility. XGBoost utilizes advanced regularization techniques (L1 and L2) to reduce overfitting and enhance model performance [5]. One of XGBoost's features is its ability to handle missing values directly during the training process. This is a significant advantage over many other machine learning algorithms, as XGBoost incorporates this functionality within its core algorithm, particularly in how it constructs decision trees. A detailed mathematical explanation, including LaTeX code for the equations, will be provided to demonstrate how XGBoost effectively learns from missing values [6].

XGBoost supports a variety of hyperparameters that can be tuned and tweaked in infinite ways however we will consider the hyperparameters seen in table I [7].

TABLE I
XGBoost Hyperparameters, Their Effects, and Possible Ranges Along with Default Values

Hyper parameter	Descriptions And Values		
	Effect On The Model	***Range***	***Default***
max_depth	Controls Trees max depth	[0,∞]	6
learning_rate	Controls model's learning rate	[0,1]	0.3
gamma	Partitioning of leaf nodes.	[0,∞]	0
min_child_weight	Hessian threshold	[0,∞]	1
subsample	Ratio of the training instances	(0.5,1)	1

D. Hyperparameters Tuning Algorithms

1) Grid Search: A widely used method for hyperparameter tuning. It systematically tests a predefined set of hyperparameters to identify the best combination for improving model performance. By conducting an exhaustive search across the entire parameter space [8].

2) Random Search: A random sampling approach which instead of testing all possible combinations like Grid, it randomly selects hyperparameter values from a predefined range to find the best combination for improving model performance [9].

3) Bayesian Search: advanced method for hyperparameter tuning that optimizes the search process by using past evaluation results to inform future selections. Unlike Random or Grid Search, Bayesian Search predicts the most promising hyperparameter values based on a probabilistic model, rather than testing combinations at random or exhaustively [10].

4) Evolutionary Genetic Search: method inspired by the principles of natural selection. It starts with a population of random hyperparameter combinations and iteratively improves them through processes such as selection, crossover, and mutation. The best-performing combinations are "selected" to create the next generation, while others are modified or replaced [11].

IV. Implementation and results

In this section, we will preprocess the dataset, define and train a baseline model using default hyperparameter values. Next, we will apply tuning algorithms with a universal setup of 5-fold CV over 100 iterations or generations the only exception is Grid Search as it uses a predefined set so the closest number will be 108 to keep it as comparable as possible. The tuning process for each algorithm will run in parallel to accelerate the process, and at the end, we will measure both the accuracy score and the time taken by each algorithm.

The hyperparameter search space will be defined using the values presented in Table II.

TABLE II
XGBoost Hyperparameters and Their Values

Hyperparameter	Values
max_depth	Positive Integer From (0, 50)
learning_rate	Sample uniformly from (0.1, 0.5)
gamma	Sample uniformly from (0.1, 0.5)
min_child_weight	Positive Integer From (0, 10)
subsample	Sample uniformly from (0.1, 1.0)

A. Data Preprocessing

In our study, the original dataset had 13,900 samples, but 20 were missing class values. These incomplete records were removed, reducing the dataset to 13,880 samples. Additionally, we found 2,277 duplicate samples, which were also dropped, leaving us with 11,603 samples. Due to the large number of missing values, removing features or samples with missing data would have resulted in the loss of the entire dataset.

To prepare the data for machine learning, we converted the class labels, originally strings, into numerical values (0 to 4) to represent the five classes using a label encoder. Four other features—'PTGENDER', 'PTETHCAT', 'PTRACCAT', and 'PTMARRY'—were also numerically encoded, ensuring the dataset was ready for analysis.

B. Baseline model

The XGBoost model achieved a 3-fold cross-validation score with an average testing score of 75.55% and an average training score of 95.92%. These results will serve as a baseline benchmark for comparing the scores from the tuning algorithms later on. While our primary goal is not to achieve the highest accuracy, but rather to study and compare the tuning methods, comparing the scores could still provide interesting insights.

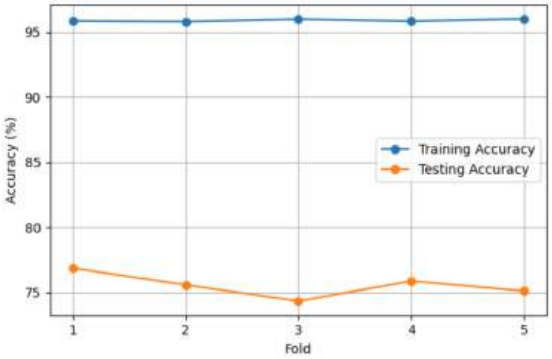


Fig. 1. Baseline model accuracy scores per fold.

C. Grid search tuning

The tuning process finished in 23.27 minutes, with the first improvement observed at the 25th iteration. Throughout the process, a total of six iterations contributed to performance improvements over the baseline model. The peak performance was achieved at the 44th iteration, with a training accuracy of 99.07% and a testing accuracy of 79.32%. The optimal hyperparameter values were: gamma = 0.1, eta = 0.5, max depth = 6, min child weight = 0, and subsample = 0.8"

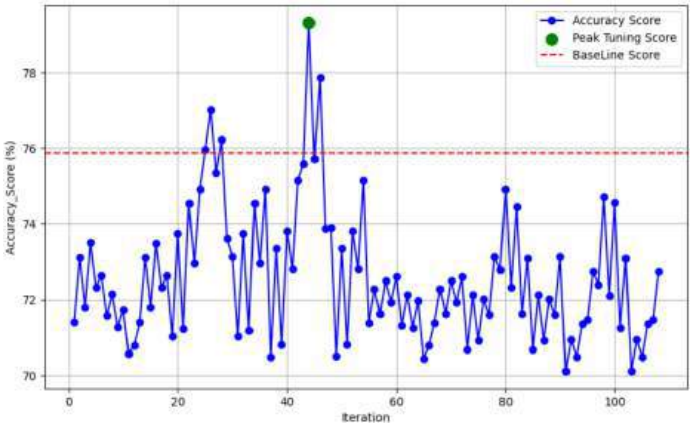


Fig. 2. Grid search tuning scores

D. Random search tuning

The tuning process finished in 10.65 minutes, only 48th and 76th iterations had improvements over the baseline model where 48th being the very first and also the peak performance of 99.25% training accuracy and a testing accuracy of 76.23%. The optimal hyperparameter values were: gamma = 0.28, eta = 0.57, max depth = 47, min child weight = 4, and subsample = 0.68."

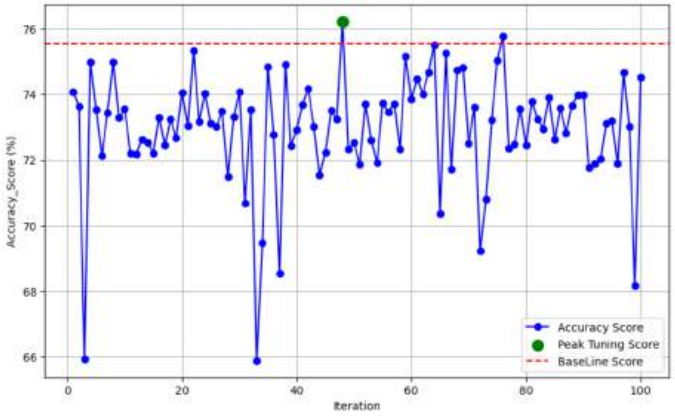


Fig. 3. Random search tuning scores

E. Bayesian search tuning

The Bayesian search process completed in 21.95 minutes, with 57 iterations showing improvements over the baseline model. The first improvement occurred at the 10th iteration, while the peak performance was reached at the 97th iteration, achieving a training accuracy of 99.60% and a testing accuracy of 79.56%. The optimal hyperparameter values at this point were: gamma = 0.1, eta = 0.56, max depth = 8, min child weight = 1, and subsample = 0.77.

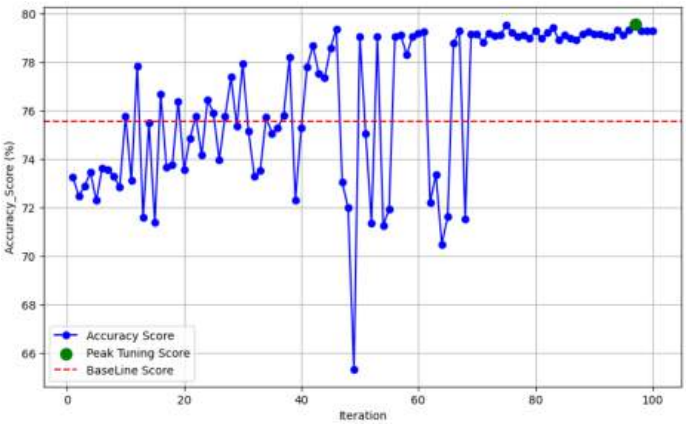


Fig. 4. Bayesian search tuning scores

F. Genetic search tuning

The tuning process spanned 1,585 minutes, approximately 26.5 hours. Over the course of 66 generations, improvements were consistently observed starting from the very first generation. Peak performance was achieved at the 22nd generation,

with a training accuracy of 99.45% and a testing accuracy of 79.29%. The optimal hyperparameters at this point were: gamma = 0.11, eta = 0.38, max depth = 10, min child weight = 2, and subsample = 0.76.

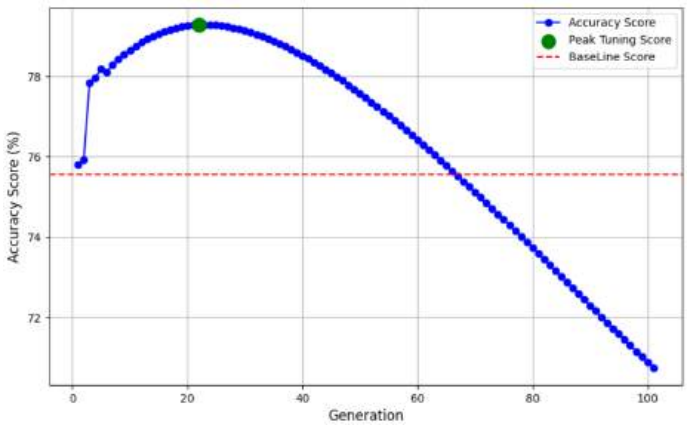


Fig. 5. Genetic search tuning scores

V. Discussion

Grid search is the most exhaustive method among the four, evaluating every possible combination of hyperparameters within a predefined grid. Its strength lies in its thoroughness and ability to ensure reliable results. This makes grid search particularly valuable when precision is critical, especially in models where every hyperparameter significantly affects performance. For instance, in scenarios such as fine-tuning critical machine learning models in the healthcare or finance domains, where accuracy is non-negotiable, grid search offers robust outcomes.However, the exhaustive nature of grid search introduces a major drawback: computational cost. When the dataset size or the number of hyperparameters increases, the number of combinations to evaluate can grow exponentially. In high-dimensional spaces, this results in excessive computation times, making grid search inefficient and impractical for larger models. Therefore, while grid search ensures comprehensive exploration, it is computationally expensive and time-consuming, making it less suitable for high-dimensional or large-scale datasets unless computational resources are abundant.

Random search addresses the limitations of grid search by randomly sampling combinations of hyperparameters. This results in a more flexible and less exhaustive exploration of the parameter space. Empirical studies have shown that random search can often outperform grid search because many hyperparameters have diminishing returns beyond certain thresholds. By not adhering to a strict grid, random search can find optimal or near-optimal solutions faster and with fewer iterations.Random search is particularly beneficial when the number of hyperparameters is large, and the objective is to obtain a good enough model in a shorter time. It works well in scenarios where approximate solutions are acceptable, such as when experimenting with different model architectures or when computational resources are limited. However, random search does not guarantee that the most optimal hyperparameters will be found, as it is inherently stochastic, which might not be suitable for scenarios requiring the highest level of precision.

Bayesian optimization offers a more sophisticated approach by using a probabilistic model to predict the most promising hyperparameter values based on past performance. Instead of evaluating combinations at random or exhaustively, Bayesian optimization intelligently explores the hyperparameter space, focusing on regions that are more likely to yield improvements. This results in a more efficient search process, as fewer iterations are needed to find optimal parameters.One of the key advantages of Bayesian optimization is its ability to find near-optimal solutions with fewer trials, making it highly efficient for computationally expensive models or large-scale datasets. It is well-suited for scenarios where computational resources are constrained, but precision is still important, such as when fine-tuning deep learning models or training complex neural networks. However, Bayesian optimization can be more complex to implement and requires careful tuning of its internal parameters (e.g., acquisition functions), which may make it less approachable for users unfamiliar with probabilistic methods.

Genetic algorithms, inspired by the principles of evolution, offer a unique approach by iterating over hyperparameters through processes such as selection, crossover, and mutation. The advantage of genetic algorithms lies in their ability to explore large and complex search spaces without relying on exhaustive evaluation. By evolving a population of hyperparameter sets over time, genetic algorithms can escape local minima and discover combinations that might be missed by grid or random search.Genetic algorithms are particularly advantageous when the hyperparameter space is highly irregular or nonlinear, and traditional methods may struggle to find optimal solutions. They excel in situations where global optimization is essential, such as in complex optimization problems with many local optima. However, genetic algorithms can be computationally expensive due to the iterative nature of the evolutionary process, and they may require significant tuning of their own parameters, such as population size and mutation rates, to achieve optimal performance.

From table III Bayesian Optimization delivered the highest improvement over the baseline, achieving a 4.01% increase with a testing accuracy of 79.56%. The tuning process was efficient, completing in approximately 22 minutes, and showed improvements over the baseline in 57 iterations, with the first improvement observed at the 10th iteration.

Grid Search also demonstrated significant improvement, with a 3.77% increase over the baseline and a testing accuracy of 79.32%. It achieved reliable results in 6 iterations, but took around 23 minutes to complete. The first improvement occurred at the 25th iteration.

Genetic Algorithms showed a notable improvement of 3.74% over the baseline, attaining a testing accuracy of 79.29%. However, it required the longest tuning time of over

Tuning Method	Training %	Testing %	Improvement			Tuning Time
			Over Baseline	First Iteration	Total Iterations	
Grid Search	99.07	79.32	3.77	25	6	00:23:16
Random Search	99.25	76.23	0.68	48	2	00:10:39
Bayes Search	99.60	79.56	4.01	10	57	00:21:57
Genetic Search	99.45	79.29	3.74	1	66	26:25:00

TABLE III
COMPARISON OF TUNING METHODS WITH IMPROVEMENT OVER BASELINE

26 hours and had the highest number of improving iterations (66), starting from the very first generation.

Random Search, while being faster with a tuning time of just 10 minutes and 39 seconds, showed minimal improvement over the baseline (0.68%) and achieved a lower testing accuracy of 76.23%. It had improvements in only 2 iterations, starting at the 48th iteration.

Bayesian Optimization is best suited for efficiency, offering the highest improvement in a relatively short time. It excels when computational resources are limited but precision is essential, particularly in complex tasks. Grid Search provides precise results but with a higher computational cost. It is appropriate when precision is critical and computational cost is less of a concern. Genetic Algorithms are effective in complex scenarios where global optimization is necessary, despite the longer time commitment. They are most effective in non-convex, irregular search spaces or when global optimization is required. Random Search is suitable for scenarios where speed is essential and approximate solutions are acceptable. It works well for large-scale models and experiments requiring quick, approximate solutions.

VI. CONCLUSION

In this study, we conducted a comprehensive comparative analysis of four prominent hyperparameter tuning methods—Grid Search, Random Search, Bayesian Optimization, and Genetic Algorithms—using the XGBoost algorithm on a multi-class classification dataset. Our aim was to evaluate these methods based on tuning time, computational complexity, accuracy, and practicality to identify their strengths, limitations, and ideal use cases.The experimental results revealed that Bayesian Optimization delivered the best balance between efficiency and performance, making it a strong choice for tuning hyperparameters when both accuracy and computational resources are important considerations. Grid Search remains a reliable method for achieving precise results, while Genetic Algorithms are preferable for complex optimization tasks despite the significant time investment. Random Search serves well in situations where rapid, approximate solutions are sufficient.Future work could explore hybrid approaches that combine the strengths of these methods or extend the comparison to include other algorithms and larger datasets. Additionally, investigating the impact of tuning more hyperparameters and the use of parallel computing could provide deeper insights into optimizing machine learning models effectively.

REFERENCES

[1] Probst, P., Boulesteix, A. L., & Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. Journal of Machine Learning Research, 20(53), 1-32.

[2] Weerts, H. J., Mueller, A. C., & Vanschoren, J. (2020). Importance of tuning hyperparameters of machine learning algorithms. arXiv preprint arXiv:2007.07588.

[3] Yadav, S., & Shukla, S. (2016, February). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In 2016 IEEE 6th International conference on advanced computing (IACC) (pp. 78-83). IEEE.

[4] Arden, F., Safitri, C. (2022, December). Hyperparameter tuning algorithm comparison with machine learning algorithms. In 2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE) (pp. 183-188). IEEE.

[5] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54, 1937-1967.

[6] Khosravi, P., Vergari, A., Choi, Y., Liang, Y., & Broeck, G. V. D. (2020). Handling missing data in decision trees: A probabilistic approach. arXiv preprint arXiv:2006.16341.

[7] XGBoost Developers. (n.d.). Parameters — XGBoost 2.0.0 documentation. https://xgboost.readthedocs.io/en/stable/parameter.html

[8] Belete, D. M., & Huchaiah, M. D. (2022). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. International Journal of Computers and Applications, 44(9), 875-886.

[9] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).

[10] Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology, 17(1), 26-40.

[11] Young, S. R., Rose, D. C., Karnowski, T. P., Lim, S. H., & Patton, R. M. (2015, November). Optimizing deep learning hyper-parameters through an evolutionary algorithm. In Proceedings of the workshop on machine learning in high-performance computing environments (pp. 1-5).

Design and Implementation of Key Word Extraction System based on Pre-training Model and Artificial Intelligence Algorithm

Junhong Chen
School of Software Engineering
South China University of Technology
Guangzhou, China;
Leihuo Studio
NetEase
Hangzhou, China
jupyterchen@163.com
*corresponding author

Kaihui Peng
Faculty of Business and Economics
University of Malaya
Kuala Lumpur, Malaysia
pengkaihui66@163.com

2024 7th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI) | 979-8-3315-2931-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACAI63924.2024.10899688

***Abstract*—With the rapid development of artificial intelligence technology in China, all walks of life have gradually integrated their advanced intelligent algorithm into it, and strive to improve the comprehensive operation efficiency. Based on the advanced technical characteristics of the pre-training model and the artificial intelligence algorithm, this study constructs the keyword extraction system model, in order to effectively improve the extraction efficiency and classification accuracy of keywords in the data resources. Combined with the empirical research and analysis, the design and implementation of the keyword extraction system based on the pre-training model and artificial intelligence algorithm effectively improves the accuracy of the keyword extraction, the error rate is further reduced, and the comprehensive operation stability of the system is significantly enhanced. Therefore, there is a positive relationship between the pre-training model, the application of the artificial intelligence algorithm and the design of the keyword extraction system. Based on the technical application of the pre-training model and the artificial intelligence algorithm, it is helpful to promote the operation efficiency and development quality of the keyword extraction system.**

Keywords—pre-training model, artificial intelligence algorithm, key word extraction system, keyword extraction, operation efficiency

I. INTRODUCTION

With the focus of countries on the use and keyword extraction of text keywords around the world, China is also gradually promoting the development of text keyword extraction and providing technical support in various fields. Vigorously develop the text keyword extraction technology, use the pre-training model and artificial intelligence algorithm, which puts forward new requirements for the relevant technical personnel of the text keyword extraction in China [1], [2]. In order to meet the application needs of text keyword extraction industry communication, the design of keyword extraction system also emerged as The Times require, to provide auxiliary reference to the industry of text keyword extraction staff. Text keyword extraction is an emerging industry developed in recent years, and the lack of professional system design has become an obstacle to the development of the industry. Keyword extraction system, pre-training model and artificial intelligence algorithm application, as the latest technical means that can effectively master the text information analysis industry, can effectively enhance the competitiveness of the executors [3]. However, the effect of the current keyword extraction system is poor and cannot meet the design goal of innovative system of keyword extraction system. Therefore, this paper studies and analyzes the practical method of text keyword extraction based on pre-training model and artificial intelligence algorithm.

II. RELATED WORKS

At present, some experts have studied the professional extraction of text keyword extraction, and put forward some research results. The application analysis method of keyword extraction system design based on pre-training model and artificial intelligence algorithm is proposed. By the scientific extraction concept and extraction principle, the keyword extraction system design evaluation and analysis index, a diversified keyword extraction system evaluation and analysis index system is constructed, and a random algorithm is used to evaluate the importance of the extraction evaluation and analysis index, so as to achieve the main purpose of system design feedback and data dimension reduction [4]. Using training model and artificial intelligence algorithm to perform evaluation analysis for input indicators, the keyword extraction system design evaluation analysis model applied to system design practice, experimental results show that based on the training model and artificial intelligence algorithm can effectively evaluate the keyword extraction system design analysis, has a certain application value [5].

In order to solve the shortcomings of the current design practice of keyword extraction system and to improve the effect of keyword extraction system design, the pre-training model of keyword extraction system based on artificial intelligence algorithm is designed. First of all, the current keyword extraction system design, evaluation analysis, build keyword extraction system design application analysis influencing factors, the factors affecting the quality of the keyword extraction system, build keyword extraction system application samples, finally using artificial intelligence algorithm for sample extraction, keyword extraction system design training model [6]. Through examples based on the training model and artificial intelligence algorithm of system design efficiency analysis, the

analysis results show that the artificial intelligence algorithm can obtain high precision system design evaluation results, and keyword extraction system design application error is lower than other contrast algorithm, effectiveness is good, but the method process is too cumbersome problems [7], [8].

In the practical analysis of keyword extraction system design based on pre-training model and artificial intelligence algorithm, the design and implementation of keyword extraction system design based on the pre-training model and artificial intelligence algorithm are proposed [9]. First of all, the current situation of the keyword extraction system design practice is analyzed, the application index of the keyword extraction system design based on the pre-training model and artificial intelligence algorithm is established, the weights of the keyword extraction system design evaluation and analysis index are defined through correlation analysis, and the keyword extraction system design is constructed [10]. The test results show that the accuracy of the keyword extraction system design evaluation based on the pre-training model and artificial intelligence algorithm is high, which can effectively improve the practical effect of the keyword extraction system design.

III. METHODS

A. Pre-training Model and AI Algorithm Analysis

Establish the sampling model of the pre-training model and the artificial intelligence algorithm, and carry out the design and analysis of the keyword extraction system. Establish the design parameter model of the keyword extraction system, as shown in (1):

$$S_n = D\left[F(G_0 + H\Delta G)\right] + J_n \tag{1}$$

In (1), D represents the design value function, F represents training model and artificial intelligence algorithm evaluation error, G represents training model and artificial intelligence algorithm design index correlation, H represents training model and artificial intelligence algorithm design features extraction set, J represents keyword extraction system design effect evaluation statistical model index, S represents training model and artificial intelligence algorithm initial characteristics, n represents the pre-trained model and the AI algorithm. As shown in (2):

$$\sum = R(r_1, r_2, \cdots, r_T), r_i = \sqrt{G_i}, \forall i \neq j \tag{2}$$

In (2), R_n represents the keyword extraction system design, and G represents the system design model of pre-training model and artificial intelligence algorithm, from which the results are obtained, as shown in (3):

$$\bigcup_{i=1}^{L} H_i = J - K_s \tag{3}$$

In (3), H represents the pre-training model and the artificial intelligence algorithm, the initial feature conditions, and J and K represent the information flow of the keyword extraction system practice. The specific model design is shown in (4):

$$Z_{2x}(X) = C\{V(B)M(N+B)\} = W(Q) \tag{4}$$

In (4), C represents on the basis of training model and artificial intelligence algorithm keyword extraction system practice information flow, $V(B)$ on behalf of training model and artificial intelligence algorithm design index, $M(N+B)$ on behalf of training model and artificial intelligence algorithm design resources index, $W(Q)$ on behalf of keyword extraction system design evaluation value, derivative model as shown in (5):

$$M_x(N) = -\frac{1}{2} D^2 F^2 \tag{5}$$

In (5), $M(N)$ represents the design effect model of the keyword extraction system based on the pre-training model and artificial intelligence algorithm; D and F represent the data distribution model of the pre-training model and artificial intelligence algorithm, which provides the data input basis for the design and implementation of the keyword extraction system.

B. Design and Analysis of Keyword Extraction System Practice

By analyzing the data model of the keyword extraction system design, the control target model of the system design prediction is established, as shown in (6):

$$DOC \sum_{a \in J} X_d C_p \tag{6}$$

In (6), X represents the measurement function of the design error of the keyword extraction system, and C represents the initial feature conditions of the keyword extraction system design. The analysis results of keyword extraction system design practice, as shown in (7):

$$YU. \sum_{a \in A} I_p^{bw} \leq O_b^{bw}(P), Q \in q \tag{7}$$

In (7), Y representative of keyword extraction system design evaluation level, U represents the keyword extraction system data distribution model, I represents the keyword extraction system evaluation error, O represents the keyword extraction system practice information flow, P represents the keyword extraction system design level, Q represents the keyword extraction system design resource distribution. Therefore, the prediction probability of the keyword extraction system design is obtained, as shown in (8):

$$T_e(Y) = U_e(G) \tag{8}$$

In (8), $T(Y)$ represents the establishment of the keyword extraction system distribution model, and $U(G)$ represents the similarity analysis of the output index of the keyword extraction system design.

IV. RESULTS AND DISCUSSION

Combined with the above design of the keyword extraction system, the data analysis is conducted, and the system design

practice is studied based on artificial intelligence algorithm. The implementation based on artificial intelligence algorithm can not only retain the effective characteristics of the extraction data to the maximum extent, avoid the elimination of the extraction data features, but also extract the extraction data features, so as to improve the effect of the keyword extraction system design. The implementation structure based on the AI algorithm is shown in Fig. 1.

	learners	business	language	profess...	can	tea
learners	8	3	3	3	2	2
business	3	6	3	2	1	1
language	3	3	5	3	1	1
professio...	3	2	3	4	2	1
can	2	1	1	2	3	1
teaching	2	1	1	1	1	3
skills	2	0	0	1	1	1
grammar	0	1	2	1	1	1
expressions	1	1	1	1	2	0
context	1	2	2	2	1	0
environm...	1	1	1	1	1	0
communi...	1	1	1	2	2	1
fosters	0	1	1	1	1	0
often	1	2	1	1	1	1
documents	1	2	2	1	1	0
seeking	1	2	1	1	1	1
specific	2	1	0	0	1	1
needs	2	1	0	0	0	1
helps	2	1	2	2	0	0
course	2	0	1	1	0	0

Fig. 1. The extraction process implemented based on the artificial intelligence algorithm

Through artificial intelligence algorithm of keyword extraction system design of training model construction, combined with the above content, the keyword extraction system design process is completed, based on artificial intelligence algorithm to keyword extraction system design data feature extraction, at the same time through the training model analysis, achieve higher keyword extraction system design performance.

In order to analyze the effectiveness of the design and implementation of the keyword extraction system based on the pre-training model and the artificial intelligence algorithm, the pre-training model environment in Table I is used to conduct the experimental comparison, as shown in Table I.

TABLE I. DESIGN TEST DATA OF KEYWORD EXTRACTION SYSTEM BASED ON PRE-TRAINING MODEL AND ARTIFICIAL INTELLIGENCE ALGORITHM

Pre-trained Model Indicators	Key Word Extraction System Design Parameters
Pre-trained model I subset	12.465.87
Subset of the pretrained models	25.195.264
Subset of the pretrained models	31.045.98
Subset of the pretrained models	168.106.12

In order to reflect the scientific nature of the design effect of the keyword extraction system based on the pre-training model and artificial intelligence algorithm, the subset of the above keyword extraction system was selected as the test object, and the number of experimental samples is shown in Table I. Thus, the design results are further realized based on the pre-training model and the artificial intelligence algorithm. The details are shown in Fig. 2.

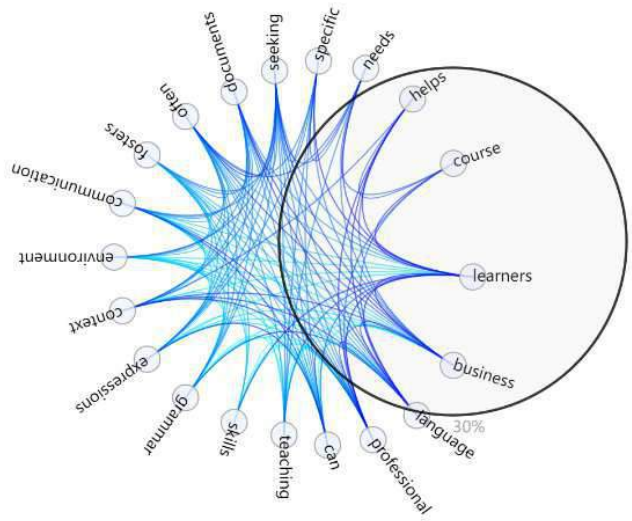


Fig. 2. Comparison of the design results of the keyword extraction system based on the pre-training model and the artificial intelligence algorithm

The analysis of Fig. 2 can see that the practical design effect of the keyword extraction system is realized based on the pre-training model and artificial intelligence algorithm. It can be seen that this paper compares the design efficiency of the keyword extraction system based on the optimization of the pre-training model and artificial intelligence algorithm, and the comparison results are shown in Fig. 3.

..	fosters	often	docum...	seeking	specific	needs	helps	cou
	0	1	1	1	2	2	2	2
	1	2	2	2	1	1	1	0
	1	1	2	1	0	0	2	1
	1	1	1	1	0	0	2	1
	1	1	1	1	1	0	0	0
	0	1	0	1	1	1	0	0
	1	0	0	0	0	0	0	0
	1	1	1	1	0	0	0	0
	1	1	1	1	1	0	0	0
	1	1	1	1	0	0	1	0
	1	1	1	1	0	0	0	0
	1	1	1	1	0	0	0	0
	2	1	1	1	0	0	0	0
	1	2	1	2	1	1	0	0
	1	1	2	1	0	0	0	0
	1	2	1	2	1	1	0	0
	0	1	0	1	2	1	0	0
	0	1	0	1	1	2	0	1
	0	0	0	0	0	0	2	1
	0	0	0	0	0	1	1	2

Fig. 3. Comparison of the design efficiency of the keyword extraction system based on the pre-training model and the artificial intelligence algorithm

According to the analysis of Fig. 3, it can be seen that the design efficiency of the keyword extraction system based on the pre-training model and artificial intelligence algorithm is the highest. It can be seen from the comparison results that the design time of the keyword extraction system is obviously short, which effectively improves the efficiency of the design practice of the keyword extraction system.

Therefore, we can see that the design efficiency of the keyword extraction system based on the pre-training model and

artificial intelligence algorithm is relatively better than that of other models, which shows that the keyword extraction system based on the pre-training model and artificial intelligence algorithm can effectively improve the design efficiency of the keyword extraction system, indicating that the system design practice of this algorithm has the best effect. In order to verify the effectiveness of this paper based on pre-training model and AI algorithm, the pre-training model and AI algorithm and other methods, as shown in Table II.

TABLE II. PERFORMANCE COMPARISON OF THE KEYWORD EXTRACTION SYSTEM DESIGN BASED ON THE PRE-TRAINED MODEL AND THE ARTIFICIAL INTELLIGENCE ALGORITHM

Model Indicators	Accuracy Analysis	Operation Effect Analysis
Pre-training model	25.69	153.567.52
Artificial intelligence algorithm	68.31	35.862.197
Key word extraction system design	67.88	356.145.75

As can be seen from Table II, the design and implementation of the keyword extraction system based on AI algorithm has better accuracy and evaluation and analysis efficiency, because there are many data indicators when using the AI algorithm is used to analyze the system design practice, which affects the overall accuracy. This shows that based on the pre-training model and artificial intelligence algorithm can effectively improve the practical effect of the keyword extraction system design, and can further improve the actual and implementation efficiency of the keyword extraction system.

V. CONCLUSION

To sum up, based on the training model and artificial intelligence algorithm of keyword extraction system design practice mode application, the system needs to constantly master and advanced text data processing concept, depth analysis, the executor using advanced keyword extraction system for professional processing and solve problems, to strengthen the comprehensive processing ability of keyword extraction system. So, based on the pre-training model and artificial intelligence algorithm keyword extraction system design and implementation, need to explore multiple influencing factors, realize the common goal of keyword extraction system application, so this paper based on artificial intelligence algorithm of keyword extraction system design and implementation, and by the experimental training model and effectiveness of artificial intelligence algorithm, thus, based on the training model and artificial intelligence algorithm, help to improve the operation efficiency of keyword extraction system and practical value.

REFERENCES

[1] X. Ding, G. Shi, Z. Liu, and H. Hu, "Risk chain mining of hazard sources in metro operation system safety: A new method to mine and control risk for safety management," *Urban Rail Transit*, vol. 9, no. 2, pp. 147–178, May 2023, doi: 10.1007/s40864-023-00192-3.

[2] L. Gao, Y. Liu, J. Zhu, and Z. Yu, "A cognitively inspired multi-granularity model incorporating label information for complex long text classification," *Cognit. Comput.*, vol. 16, no. 2, pp. 740–755, Dec. 2023, doi: 10.1007/s12559-023-10237-1.

[3] J. P, P. R. S. Bhama, and M. B, "Sign language recognition using deep CNN with normalised keyframe extraction and prediction using LSTM," *J. Sci. Ind. Res.*, vol. 82, no. 07, pp. 745–755, Jul. 2023, doi: 10.56042/jsir.v82i07.2375.

[4] K. Ma, M. Tian, Y. Tan, Q. Qiu, Z. Xie, and R. Huang, "Ontology-based BERT model for automated information extraction from geological hazard reports," *J. Earth Sci.*, vol. 34, no. 5, pp. 1390–1405, Oct. 2023, doi: 10.1007/s12583-022-1724-z.

[5] Q. Qiu *et al*., "Extracting named entity using entity labeling in geological text using deep learning approach," *J. Earth Sci.*, vol. 34, no. 5, pp. 1406–1417, Oct. 2023, doi: 10.1007/s12583-022-1789-8.

[6] D. R. CH and S. K. Saha, "Generation of multiple-choice questions from textbook contents of school-level subjects," *IEEE Trans. Learn. Technol.*, vol. 16, no. 1, pp. 40–52, Feb. 2023, doi: 10.1109/tlt.2022.3224232.

[7] H. Shang, G. Zhao, J. Shi, and X. Qian, "A multiview text imagination network based on latent alignment for image-text matching," *IEEE Intell. Syst.*, vol. 38, no. 3, pp. 41–50, May 2023, doi: 10.1109/mis.2023.3265176.

[8] Z. Wang *et al*., "Auto-extraction of building outline with multi-band polarimetric SAR," *J. Electron. Inf. Technol.*, vol. 45, no. 7, pp. 2511–2518, Jul. 2023, doi: 10.11999/JEIT220776.

[9] S. Yang *et al*., "Extracting pulmonary nodules and nodule characteristics from radiology reports of lung cancer screening patients using transformer models," *J. Healthcare Inf. Res.*, vol. 8, no. 3, pp. 463–477, May 2024, doi: 10.1007/s41666-024-00166-5.

[10] S. Zhang, F. Su, Y. Wang, S. Mai, K. P. Pun, and X. Tang, "A low-power keyword spotting system with high-order passive switched-capacitor bandpass filters for analog-MFCC feature extraction," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 70, no. 11, pp. 4235–4248, Nov. 2023, doi: 10.1109/tcsi.2023.3299855.