CISDS

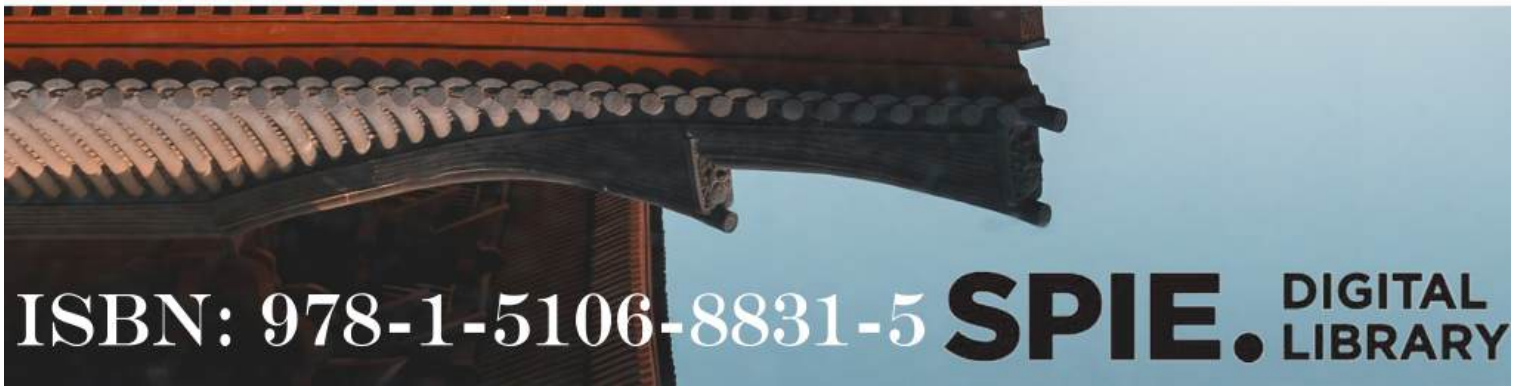
November 22-24
2024

Nanjing
China

2024 3rd International Conference on Communications, Information System and Data Science

CONFERENCE

PROCEEDING



ISBN: 978-1-5106-8831-5

PROCEEDINGS OF SPIE

Third International Conference on Communications, Information System, and Data Science (CISDS 2024)

Cheng Siong Chin
Shensheng Tang
Daniele Giusto
Yonghui Li
Editors

22–24 November 2024
Nanjing, China

Organized by
International Computing and Engineering Association (Hong Kong, China)

Sponsored by
Nanjing Tech University (China)

Published by
SPIE

Volume 13519

Proceedings of SPIE 0277-786X, V. 13519

SPIE is an international society advancing an interdisciplinary approach to the science and application of light.

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE Vol. 13519, 1351901 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3059864

The papers in this volume were part of the technical conference cited on the cover and title page. Papers were selected and subject to review by the editors and conference program committee. Some conference presentations may not be available for publication. Additional papers and presentation recordings may be available online in the SPIE Digital Library at SPIEDigitalLibrary.org.

The papers reflect the work and thoughts of the authors and are published herein as submitted. The publisher is not responsible for the validity of the information or for any outcomes resulting from reliance thereon.

Please use the following format to cite material from these proceedings:
Author(s), "Title of Paper," in *Third International Conference on Communications, Information System, and Data Science (CISDS 2024)*, edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE 13519, Seven-digit Article CID Number (DD/MM/YYYY); (DOI URL).

ISSN: 0277-786X
ISSN: 1996-756X (electronic)

ISBN: 9781510688315
ISBN: 9781510688322 (electronic)

Published by
SPIE
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone +1 360 676 3290 (Pacific Time)
SPIE.org

Printed in the United States of America by Curran Associates, Inc., under license from SPIE.

Publication of record for individual papers is online in the SPIE Digital Library.

SPIE. DIGITAL LIBRARY
SPIEDigitalLibrary.org

Paper Numbering: A unique citation identifier (CID) number is assigned to each article in the Proceedings of SPIE at the time of publication. Utilization of CIDs allows articles to be fully citable as soon as they are published online, and connects the same identifier to all online and print versions of the publication. SPIE uses a seven-digit CID article numbering system structured as follows:

- The first five digits correspond to the SPIE volume number.
- The last two digits indicate publication order within the volume using a Base 36 numbering system employing both numerals and letters. These two-number sets start with 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B … 0Z, followed by 10-1Z, 20-2Z, etc. The CID Number appears on each page of the manuscript.

Contents

COMPUTER VISION AND DATA MINING

Conference Committee

Conference Chairs

Chong-Yung Chi, National Tsing Hua University (Taiwan)
Ljiljana Trajkovic, Simon Fraser University (Canada)
Mouquan Shen, Nanjing Tech University (China)

Program Chairs

Yonghui Li, The University of Sydney (Australia)
Wanyang Dai, Nanjing University (China)
Hai Ning Liang, Xi'an Jiaotong-Liverpool University (China)

Steering Chairs

Wen-Jer Chang, Taiwan Ocean University (Taiwan)
Hong Lin, University of Houston-Downtown (United States)
Qiang (Shawn) Cheng, University of Kentucky (United States)

Publication Chairs

Cheng Siong Chin, Newcastle University in Singapore (Singapore)
Shensheng Tang, Bethel University (United States)
Daniele Giusto, University of Cagliari (Italy)

Publicity Chairs

Alexandre Lobo, University of Saint Joseph, Macao (Macao, China)
Huiru (Jane) Zheng, Ulster University (United Kingdom)
Tien-Ying Kuo, National Taipei University of Technology (Taiwan)

International Technical Program Committee

Arti Arya, PES University (India)
Chin-Shiuh Shieh, National Kaohsiung University of Science and Technology (Taiwan)
Nu Nu War, National Research University (Russian Federation)
Paulo Gil, University of Coimbra (Portugal)
Goi Bok Min, Universiti Tunku Abdul Rahman (Malaysia)
Javier Gozalvez, Universidad Miguel Hernández de Elche (Spain)
José Manuel Molina López, Universidad Carlos III de Madrid (Spain)
Anand Nayyar, Duy Tan University (Vietnam)
Dmitri Kvasov, University of Calabria (Italy)
Jingshan Huang, The University of South Alabama (United States)

Ong Pauline, Universiti Tun Hussein Onn Malaysia (Malaysia)
Abdul Ghani Albaa, Princess Sumaya University for Technology (Jordan)
Zayar Aung, National Research University (Russian Federation)
Leoneed Mihaylov Kirilov, Bulgarian Academy of Sciences (Bulgaria)
Thomas Lee, University of Illinois at Chicago (United States)
Tri Kuntoro Priyambodo, Universitas Gadjah Mada (Indonesia)
Számel László Budapest, University of Technology and Economics (Hungary)
Sathish Kumar Selvaperumal, Asia Pacific University of Technology and Innovation (Malaysia)
Syed Farooq Ali, University of Management and Technology (Pakistan)
Weitian Tong, Georgia Southern University (United States)
Rushit Dave, Minnesota State University (United States)
M. Marukkannan, Institute of Road and Transport Technology (India)
Hemn Barzan Abdalla, Wenzhou-Kean University (China)
Iouliia Skliarova, Universidade de Aveiro (Portugal)
Shiwen Ni, Shenzhen Institute of Advanced Technology (China)
Paulo Batista, Universidade de Évora (Portugal)
Xiaoye Liu, University of Southern Queensland (Australia)
Jaime A. Martins, Universidade do Algarve (Portugal)
Nurazean Maarop, Universiti Teknologi Malaysia Kuala Lumpur (Malaysia)
Noorlin Mohd Ali, Universiti Malaysia Pahang (Malaysia)
T. Arudchelvam, Wayamba University of Sri Lanka (Sri Lanka)
Takfarinas Saber, University College Dublin (United Kingdom)
Tan Yi Fei, Multimedia University (Malaysia)
Man Fung LO, The University of Hong Kong (Hong Kong, China)
Xianzhi Wang, University of Technology Sydney (Australia)
Michael Opoku Agyeman, University of Northampton (United Kingdom)
Ali Yavari, Swinburne University of Technology (Australia)
Faridah Binti Yahya, Malaysian Institute of Information Technology (Malaysia)
Maumita Bhattacharya, Charles Sturt University (Australia)

Low-Altitude Target Detection Algorithm for Intelligent Scenic Areas Based on Improved YOLOv10

Xiao Li[a], Ji Sun[b], Pei Li[b], Ye Tao[a], and Hui Li[*a]
[a]Qingdao University of Science and Technology, Songling Road, Qingdao, China
[b]Tsingtao Beer Museum, Dengzhou Road, Qingdao, China

ABSTRACT

With the development of drone technology, its application in intelligent scenic areas provides a new solution for tourist flow monitoring. To enhance detection accuracy and satisfy real-time demands, this study proposed a low-altitude target detection algorithm of intelligent scenic areas based on improved YOLOv10, and developed an intelligence scenic areas tourist flow monitoring and statistic system accordingly. By introducing the Large Separable Kernel Attention (LSKA) mechanism, the algorithm optimizes the Spatial Pyramid Pooling Fast (SPPF) module and effectively capturing long-range dependencies in images. In addition, we added a Small Target Detection Layer(STDL) to the YOLOv10 network structure to retain more location information and detailed features about small targets. Results from experiments conducted on the VisDrone2019 dataset show that, compared to the original YOLOv10 model, the enhanced version demonstrates an improvement in Recall by 2.0% and an increase in *mAP*@0.5 by 1.7%. Compared with other mainstream models, our proposed algorithm has improved on many evaluation metrics, and fulfills the requirements for real-time detection. It has been successfully applied to Tsingtao Beer Museum and has achieved good results. The results of the experiments indicate that the algorithm performs well in detecting low-altitude aerial photography images of drones, and provides effective technical assistance for the safety management of intelligent scenic areas.

Keywords: Intelligent scenic areas, Large separable kernel attention, Drone, Small target detection

1. INTRODUCTION

Rapid advances in drone technology and dramatic cost reductions have led to the expansion of drones from the military to the civilian sector and into many aspects of our daily lives. In aerial photography, the flexibility and ease of handling of drones makes them an indispensable tool for obtaining aerial views. With the rise of the concept of intelligent scenic areas, the application of drones in tourism monitoring and management has become particularly crucial, which is of great significance to enhance tourist experience and ensure the safety of scenic areas.

However, the drone aerial photography target detection algorithm faces many challenges in practical application. The background of drone aerial photography is complicated, the target is blocked frequently, and the wide field of view brought by high-altitude operation, the interference in the image increases, which creates challenges for the precise identification and categorization of the target. In addition, targets in drone aerial images are often presented as small targets due to the distance, and these small targets are small in size and low in resolution in the image, resulting in limited key feature information and easy to be lost during the downsampling process, which further increases the accuracy requirements of the detection algorithm. Concurrently, the drones' lightweight construction and the constraints on computational resources, as well as the demand for real-time performance, also raise more stringent requirements for the development of target detection algorithms.

Traditional object detection algorithms, such as Faster R-CNN[1], have excellent detection accuracy, but their complex workflow and high computational requirements limit their application in real-time scenarios. On the other hand, single-stage algorithms such as YOLO[2] are better suited to meet the needs of real-time applications because they simplify the detection process and improve speed. Recent research[3-6] combines the advantages of deep and shallow networks in order to maintain sufficient detail information while also obtaining rich semantic context, thus significantly improving the detection accuracy of small targets. Although the method of combining shallow and deep features in theory can enhance the model's ability to detect small targets, this fusion strategy inevitably introduces complexity into the network structure. In addition, researchers have explored video object detection methods. For example, FastVOD-Net[7] effectively utilizes the temporal dependence between frames and introduces an attention-guided semantic refinement module to achieve a suitable balance between real-time performance and accuracy. CDANet[8] effectively utilizes the semantic information of

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 1351902 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058111

specific categories and the temporal correlation between frames to enhance the representation of object appearance, thereby promoting the classification of detected objects. However, this complex network structure may result in higher computational costs, especially when processing high-resolution videos. In the past few years, many of the most advanced image processing methods[9,10] have achieved outstanding performance in this field, and their designs or modules can be integrated into the YOLO series to achieve better performance.

Aiming at the challenge of drone target detection, this study improved the YOLOv10[11] network. As a novel model for real-time end-to-end target detection, YOLOv10 effectively reduces its dependence on non-maximum suppression (NMS) by adopting a consistent dual assignment strategy, reducing the inference delay while maintaining excellent detection performance. The model design of YOLOv10 is comprehensively optimized, and the model components are fine-tuned from the perspective of efficiency and accuracy, which significantly improves the detection capability. In light of the aforementioned factors, this study presents an enhanced version of low-altitude target detection algorithm for YOLOv10, which encompasses several key improvements:

(1) Due to the considerable distance involved in drone imaging, targets occupy fewer pixels within the captured images, leading to less distinct target features. To address this challenge, we significantly enhance our capability to extract small target features by incorporating a dedicated small target detection head, thereby improving detection accuracy.

(2) Considering the significant scale variation of the targets to be detected, which leads to reduced detection accuracy, the LSKA mechanism[12] is employed to enhance the SPPF module and boost the network's ability for multi-scale feature extraction.

(3) We developed an intelligence scenic areas tourist flow monitoring and statistic system, and tested it on a low-altitude video shot at the Tsingtao Beer Museum, further verifying the effectiveness and practicality of the algorithm.

Through the research of this paper, we hope to provide real-time and accurate tourist flow data for intelligent scenic areas through low-altitude target detection technology of drones, optimize tourist experience and improve safety management efficiency.

2. METHODS

The core of our method lies in fusing shallow and deep features by adding a small object detection head, thereby improving the model's perception of low-resolution targets. Additionally, we innovatively integrate LSKA into the SPPF structure, by using a separable large kernel convolution operation to capture more rich spatial context information. The specific architecture is shown in Figure 1.

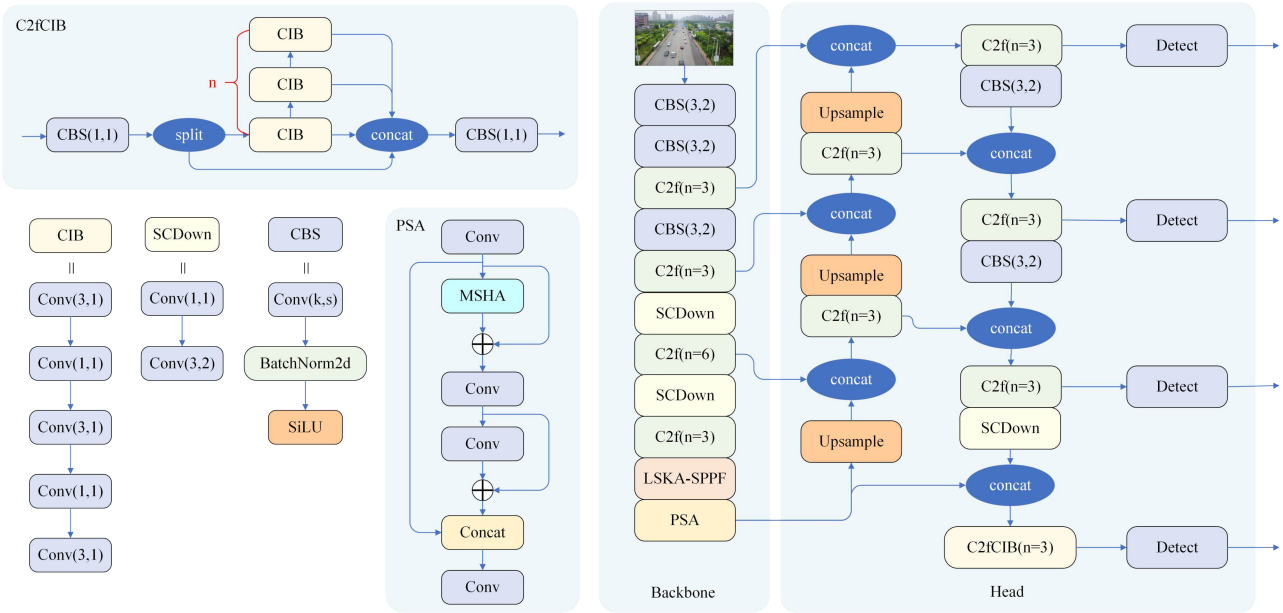


Figure 1. Technical Roadmap.

2.1 LSKA-SPPF Module

In the drone target detection scene, the size of the detected object varies greatly due to the change in the drone's flight altitude, and the size of the same object in different images is also different. The original SPPF module employs three consecutive pooling layers to integrate outputs from each layer for effective multi-scale fusion, which significantly reduces the computational complexity. However, the multi-layer pooling operation of SPPF module makes it simple to ignore the feature information of small targets, and the effect is not good in the drone target detection task. Consequently, our study incorporates the LSKA attention mechanism to enhance the SPPF module and boost the network's ability for multi-scale feature extraction.

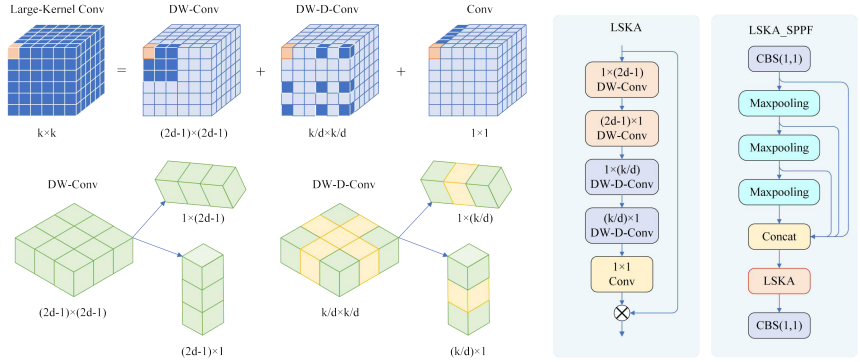


Figure 2. LSKA-SPPF.

LSKA employs deepwise convolution (DW-Conv) and deepwise dilated convolution (DW-D-Conv) to effectively model large convolution kernels. This approach enhances spatial awareness while simultaneously reducing the computational load and parameter count associated with large convolution kernels. For a large k × k convolution kernel, LSKA decompositions it into three parts: Part 1. one 1×(2d-1) and one (2d-1)×1 depthwise convolution, where d is the size of the expansion coefficient; Part 2. one $1 \times \lfloor \frac{k}{d} \rfloor$ and one $\lfloor \frac{k}{d} \rfloor \times 1$ depthwise dilated convolution; Part 3. one 1×1 convolution. The LKSA structure is shown in Figure 2. Given the input feature map $F^C \in R^{C \times H \times W}$, LSKA uses deepwise convolution to perform separate convolution operations for each channel of the input, with each convolution kernel processing information from one input channel; It then uses deepwise dilated convolution to share parameters between the input channels, thereby reducing the number of parameters; And finally a 1×1 convolution is applied to integrate the information and produce the final feature map. The output of the LSKA can be obtained using Eqs. (1) , (2) , (3) and (4), where * and $\otimes$。represent the convolution and Hadamard product, respectively.

$$\overline{Z}^C = \sum_{H,W} W^C_{(2d-1)\times 1} * \left(\sum_{H,W} W^C_{1\times(2d-1)} * F^C \right), \tag{1}$$

$$Z^C = \sum_{H,W} W^C_{\lfloor \frac{k}{d} \rfloor \times 1} * \left(\sum_{H,W} W^C_{1\times \lfloor \frac{k}{d} \rfloor} * \overline{Z}^C \right), \tag{2}$$

$$A^C = W_{1\times 1} * Z^C, \tag{3}$$

$$\overline{F}^C = A^C \otimes F^C, \tag{4}$$

2.2 Small Target Detection Layer

Small object detection is an important step in the task of object detection, especially in scenarios such as remote sensing images and surveillance monitoring, where small object detection has wide application value. However, due to the limited pixel area and limited feature information of small objects in images, detecting small objects is quite challenging. Therefore, improving the detection effect of small objects is of great significance for improving the overall performance of the algorithm.

In addressing the challenging task of detecting small targets in drone imagery, this study has conducted thorough optimizations and improvements on the YOLOv10 algorithm. The original YOLOv10 framework, while capable of

recognizing objects of varying sizes through its multi-level detection structure, specifically features three detection layers tailored to effectively identify targets larger than 8×8, 16×16, and 32×32 pixels, respectively. However, its detection efficacy for even smaller targets, particularly those 4×4 pixels and above, still leaves room for enhancement. In response, this study innovatively integrates a novel detection branch into the YOLOv10 network architecture, designed exclusively for capturing small targets within images. An additional detection layer with 128 channels is added on top of the original detection layer, focusing on improving the detection accuracy of small targets with a size of 4x4 pixels or more. Furthermore, to further strengthen the adaptability of this new branch, we have made meticulous adjustments to the size of the feature maps, aiming to enable the network to capture intricate details within the images more acutely, which is crucial for the precise identification of small targets.

2.3 Intelligence Scenic Areas Tourist Flow Monitoring and Statistic System

We have developed an intelligent scenic area tourist flow monitoring and statistics system using PyQt5, which aims to provide real-time monitoring and analysis tools for scenic area management. It combines the advanced target detection capability of YOLOv10 model and the data association capability of ByteTrack algorithm to track multiple targets in the video. The improved YOLOv10 algorithm we proposed can also be well adapted to this system. The system first uses YOLOv10 to process the input video frames and detect the locations and classifications of all objects in every frame. Then ByteTrack algorithm intervenes and divides them into two groups of high confidence and low confidence according to the detection confidence. Then, the algorithm uses the Kalman filter to predict the trajectory of each target and tries to match these predictions with the detection results in the current frame. The innovation of ByteTrack is that it retains almost all detection boxes, not just the high-confidence ones, and in doing so can better handle occlusion and target re-emergence. For tracks that don't match the high-confidence check box, ByteTrack makes use of the low-confidence check box for a secondary match to recover the obscured target and eliminate background interference. With this strategy, ByteTrack is able to effectively maintain the trajectory of the target even when it is obscured or reappears, and maintains a low ID switching rate, thus achieving accurate and robust multi-target tracking. The system visualization is shown in Figure 3.

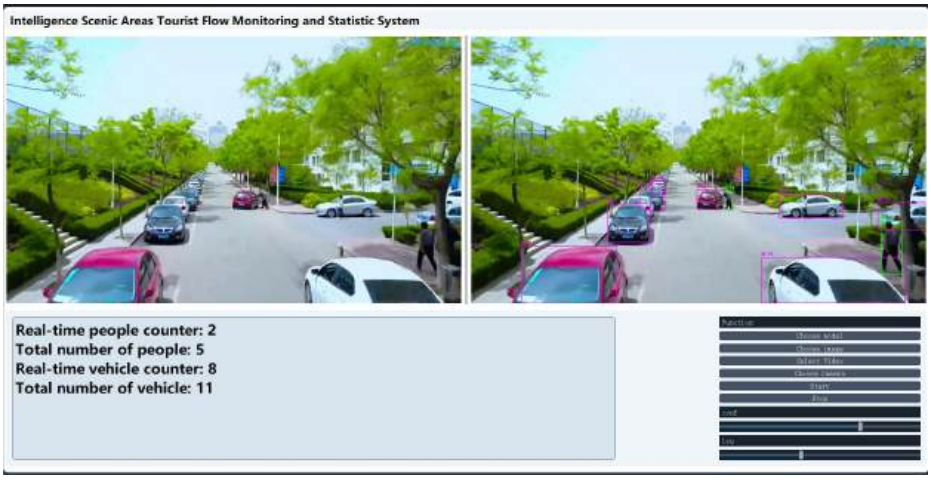


Figure 3. System Visualization.

The system allows users to choose live surveillance cameras or pre-recorded video files as video sources through a user-friendly interface. The core functions of the system include real-time video frame processing, target tracking line drawing, detection start and stop control, as well as real-time display of the current frame and the cumulative number of pedestrians and vehicles. In addition, the system provides an intuitive user interface, which is convenient for operators to select video sources and models, and displays statistics in real time to assist managers to quickly obtain key information. The design of the system focuses on user-friendly and real-time performance to meet the monitoring needs of people and traffic flow during peak hours. Relying on the powerful computing power of the deep learning model, the system can maintain high accuracy detection and tracking performance under changing environmental conditions, so as to provide strong technical support for the operation and management of the scenic area. Through accurate data analysis and real-time feedback, the system not only enhances the security monitoring capability of the scenic area, but also provides decision support for traffic planning and tourist experience optimization.

3. EXPERIMENT AND RESULT ANALYSIS

3.1 Experimental Setup and Details

In this experiment, we used the following hardware and software configurations: Ubuntu 20.04.3 LTS operating system, Intel(R) Xeon(R) Silver 4316 CPU @ 2.30GHz, and NVIDIA GeForce RTX 3090 GPU. On the software side, we used Python 3.9, Pytorch 2.0.1, Torchvision 0.15.2, CUDA 11.7, and CUDNN 8.5. To ensure a fair comparison of the algorithm's performance, consistent hyperparameters were applied across all experimental runs. Specifically, we set the initial learning rate to 0.01, a batch size of 8, and conducted training for 300 epochs.

3.2 Datasets and Evaluation Metrics

The VisDrone2019[13] dataset contains 8599 images taken by drones in different scene weather, viewing height, and lighting conditions, annotated over 540k object bounding boxes, covering 10 predefined categories, diverse and wide-ranging. The locations of small targets in these subsets are different but the environment is similar. The dataset can provide real scenes, meet the experimental requirements of small target detection, and has important guiding significance for evaluating algorithm performance and promoting technology development. In this paper, the training dataset of the improved algorithm adopts VisDrone2019-DET-train, and the validation dataset adopts VisDrone2019-DET-val.

In order to show the detection effect of the improved model on the original model, the experiment in this paper evaluates the detection performance and model parameters. The experimental indexes include Precision (*P*), Recall (*R*), mean Average Precision(*mAP*) and Frames Per Second (*FPS*). These metrics provide a comprehensive overview of a model's accuracy, robustness, and real-time capability.

3.3 State-of-the-art Comparison

In order to validate the efficacy of our proposed module, we performed experiments using the VisDrone2019 validation dataset, achieving optimal detection results across various evaluation metrics. Among them, black bold ranks first.

In this section, we will conduct a thorough analysis of the performance of our proposed algorithm on the VisDrone2019 dataset and directly compare it with other YOLO series models, aiming to highlight its unique performance advantages. To ensure fairness and objectivity in the comparison, all models involved in the comparison were rigorously and consistently evaluated on the same benchmark—the VisDrone2019 validation dataset.

Table 1. VisDrone2019 Val Dataset.

Method	*P*(%)	*R*(%)	*mAP@*0.5(%)	*mAP@*0.5:0.95(%)	*FPS*
YOLOv5n	45.5	33.6	33.9	19.8	31.30
YOLOv6n	42.8	30.9	31.4	18.3	31.11
YOLOv8n	45.4	34.0	34.3	20.0	**32.14**
YOLOv10n	46.0	34.3	34.4	20.2	27.62
Ours	**46.1**	**36.3**	**36.1**	**21.2**	26.51

Table 1 summarizes the experimental results of various detection models evaluated on the VisDrone2019 validation dataset. The proposed method, referred to as “ours”, has demonstrated exceptional performance, surpassing existing YOLO variants in key metrics such as Precision, Recall, and mean Average Precision (*mAP*). This highlights its superior accuracy and reliability in detection capabilities. Our model attains a Precision of up to 46.1% and surpasses YOLOv10n by 2% in terms of Recall, significantly reducing missed detections while robustly enhancing target detection capabilities.

Furthermore, with an *mAP@*0.5 of 36.1% and an *mAP@*0.5:0.95 of 21.2%, our model effectively showcases its outstanding performance in detecting multi-scale targets, rendering it highly valuable for practical applications. Additionally, our model maintains a reasonable frame rate of 26.51 *FPS* while ensuring high detection accuracy; this indicates that our model possesses commendable real-time performance suitable for practical use cases.

3.4 Visualization

We conducted a series of visualization experiments to prove the effectiveness of our improved YOLOv10 algorithm in low altitude target detection in intelligent scenic areas.

Figure 4. VisDrone2019 Visualization.

As shown in Figure 4, we selected four scenes from the VisDrone dataset for testing. Scene 1 features natural vegetation and leisure activities, aiming to evaluate the algorithm's ability to detect objects of different sizes under potential occlusion conditions. Scene 2 showcases a typical low-light nighttime shopping center environment, aiming to explore the algorithm's reliability in complex and low visibility conditions. Scene 3 showcases a basketball court environment, aiming to explore the algorithm's practicality in fast-moving scenes. Scene 4 showcases a complex traffic environment, with dense and frequent occlusions on city streets, providing a challenging environment for target detection. These visual experiments collectively prove the algorithm's versatility and effectiveness in complex environments. Through the above visual experiments, it can be proved that the system can adapt to various real environments, thereby demonstrating its ability to enhance the safety and management of smart tourist attractions.

Figure 5. Tsingtao Beer Museum Visualization.

We used drones to capture images of the crowd outside the Tsingtao Beer Museum and used the proposed algorithm to detect and count the people in real-time. As shown in Figure 5, through visual analysis, we found that our algorithm could maintain high accuracy and stability even under challenging conditions such as high pedestrian flow and small target size, accurately identifying and counting the crowd. Through the on-site test at the Tsingtao Beer Museum, our method effectively improved the intelligent level of scenic area safety management and crowd monitoring. Our low-altitude detection algorithm can meet the high requirements in actual applications such as smart scenic areas, providing high precision and reliability detection services, and providing more comprehensive and accurate data support for scenic area managers, thus enhancing the tourism experience.

3.5 Ablation Study

To evaluate the performance of our algorithm in low-altitude target detection, we conducted an ablation experiment. The results of the experiment are listed in Table 2, which includes the benchmark performance of the YOLOv10 model and the changes in model performance after adding various modules.

Table 2. VisDrone2019 Ablation Study.

Method	*P*(%)	*R*(%)	*mAP*@0.5(%)	*mAP*@0.5:0.95(%)
YOLOv10n	46.0	34.3	34.4	20.2
YOLOv10n+STDL	45.9	35.7	35.9	21.1
YOLOv10n+STDL+LSKA	46.1	36.3	36.1	21.2

Baseline model (YOLOv10n): YOLOv10n is our algorithm's baseline model. Its performance in Precision, Recall and mean Average Precision under different *IoU* thresholds provides references for subsequent experiments.

YOLOv10n+STDL: Based on the benchmark model, we have incorporated STDL with the aim of capturing features of small targets to enhance the accuracy in identifying small objects. The results indicated that, when compared to the baseline model, the Recall improved by 1.4% following the incorporation of STDL. This enhancement signifies a substantial improvement in the model's generalization performance. Additionally, both *mAP*@0.5 and *mAP*@0.5:0.95 demonstrated improvements, further demonstrating that overall performance was enhanced across various *IoU* thresholds.

YOLOv10n+STDL+LSKA: Further, we introduce the LSKA based on the previous model. LSKA captures richer spatial features by means of large-size separable convolution kernel, thus enhancing the model's ability to detect objects. Experimental results show that the addition of LSKA improves the Recall of the model from 34.3% to 36.3%, and the *map*@0.5 from 34.4% to 36.1%. In addition, over the more challenging *IoU* threshold range of 0.5 to 0.95, the model's *mAP* increased from 20.2% to 21.2%. These results show that LSKA enhances the model's ability to detect targets by capturing richer spatial features, especially when dealing with small targets and complex scenes, and can enhance the accuracy and robustness of detection more effectively.

In summary, the gradual incorporation of the STDL and LSKA-SPPF modules has resulted in a notable improvement across different performance metrics of the algorithm. This demonstrates that these modules contribute positively to improving both the accuracy and reliability of the algorithm.

4. CONCLUSIONS

We proposed a low-altitude target detection algorithm for intelligent scenic areas, which is based on the improved YOLOv10 and enhances the network's feature processing ability by adding STDL and introducing LSKA. Experimental results show that compared with the original YOLOv10 model, the improved model has a 2% increase in Recall and a 1.7% increase in *mAP*@0.5 on the VisDrone2019 dataset. Moreover, our model outperforms mainstream models in multiple evaluation indicators and can meet the real-time detection requirements.

In addition, the intelligence scenic areas tourist flow monitoring and statistic system developed in this research combines the target detection capability of YOLOv10 model and the data association capability of ByteTrack algorithm to achieve accurate tracking and traffic statistics of multiple targets in the video.

In the future, we will explore the algorithm's potential applications across various detection scenarios and optimize its effectiveness in practical settings to further bolster the system's robustness and precision.

ACKNOWLEDGMENTS

This work was supported by the Key Technology Research and Industrial Demonstration Projects in Qingdao City (23-7-2-qljh-4-gx).

REFERENCES

[1] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with regionproposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), 1137–1149(2017).

[2] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 779–788 (2016).

[3] Zhang, Z., "Drone-yolo: An efficient neural network method for target detection in drone images,"Drones 7(8) (2023).

[4] Yang, C., She, L., and Yang, L., "Improved yolov5 remote sensing image target detection algorithm," Computer Engineering and Applications 59(15), 76–86 (2023).

[5] Han, J., Yuan, X., Wang, Z., and Chen, Y., "Uav dense small target detection algorithm based on yolov5s," Journal of ZheJiang University (Engineering Science) 57(6), 1224–1233 (2023).

[6] Yang, L., Lianquan, W., Haitao, Y., Jinlin, N., Xianteng, C., Huapeng, W., and Qinglong, Z., "A small target detection algorithm from uav perspective," Infrared Technology 45(9), 925–931 (2023).

[7] Qi, Q., Wang, X., Hou, T., Yan, Y., and Wang, H., "Fastvod-net: A real-time and high-accuracy video object detector," IEEE Transactions on Intelligent Transportation Systems 23(11), 20926–20942 (2022).

[8] Qi, Q., Yan, Y., and Wang, H., "Class-aware dual-supervised aggregation network for video object detection," IEEE Transactions on Multimedia 26, 2109–2123 (2024).

[9] Chen, H., Wang, Y., Guo, J., and Tao, D., "Vanillanet: the power of minimalism in deep learning," Advances in Neural Information Processing Systems 36 (2023).

[10] Wang, C., He, W., Nie, Y., Guo, J., Liu, C., Han, K., and Wang, Y., "Gold-yolo: efficient object detector via gather-and-distribute mechanism," Proceedings of the 37th International Conference on Neural Information Processing Systems (2024).

[11] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., and Ding, G., "Yolov10: Real-time end-to-end object detection," (2024).

[12] Lau, K. W., Po, L.-M., and Rehman, Y. A. U., "Large separable kernel attention: Rethinking the large kernel attention design in cnn," Expert Systems with Applications 236, 121352 (2023).

[13] Du, D., Zhu, P., and et al, "Visdrone-det2019: The vision meets drone object detection in image challenge results," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW) , 213–226(2019).

An Improved Dual-Threshold Generalized Likelihood Ratio Test Range-Spread Target Detector

Xinrong Xu[a], Weixing Sheng*[b], Renli Zhang[c], and Xiaoyu Cong[d]

[a]Nanjing University of Science and Technology, 210094 Nanjing. China
[b]Nanjing University of Science and Technology, 210094 Nanjing. China
[c]Nanjing University of Science and Technology, 210094 Nanjing. China
[d]Nanjing University of Science and Technology, 210094 Nanjing. China

ABSTRACT

Currently, high-resolution radar is increasingly widely used for target detection. However, the higher range resolution causes the target energy to be dispersed over multiple range cells, creating the range-spread targets rather than the point targets, which leads to a degradation of the detection capability of traditional point-target detection methods. In this paper, an improved dual-threshold generalized likelihood ratio test detector is proposed for high-resolution radar. The concept of the "effective scatterer" is introduced to extract the strong scatterer cells of the targets. Then the Lilliefors test is adopted for pre-judgement and based on the pre-judgement result, the calculation of the first threshold and the second threshold is optimized so that the proposed detector can improve the detection performance at a constant false alarm rate. The simulation results show that the proposed detector outperforms traditional detection methods for range-spread targets.

Keywords: Range-spread target, double threshold detector, generalized likelihood ratio test (GLRT), Lilliefors Test

1. INTRODUCTION

High-resolution radar can be widely used due to its ability to more accurately portray the structure of the target, which is conducive to improving the performance of radar target detection.[1–4] For high-resolution radars, high resolution range profile (HRRP) is defined as the projection vector sum of the target's scattered point sub-echo in the radar's line-of-sight direction. When the range cell is much smaller than the target, the target energy is dispersed into multiple range cells in HRRP, becoming a range-spread target.[5] Although it is possible to characterize the structure of the target more accurately, the dispersion of the target energy leads to a degradation of the detection capability of the traditional constant false alarm detection methods for point targets.[6] Therefore, the algorithms that can detect range-spread targets is necessary to be studied.

Range-spread target detection methods typically utilize the total target echo energy. However, in practice, the number and location of target scattering points distributed over the HRRP can hardly be predicted. P.K. Hughes first introduced the energy-integrating detector[7] (Integrator), which accumulated all the range cells in the to-be-detected area to perform target detection. Obviously, a large number of noise cells are accumulated in this method, resulting in degraded detection performance.

A dual-threshold based high-resolution radar generalized likelihood ratio test detector (GLRT-DT) was proposed in Refs. 8. The detector utilizes the first threshold to extract strong scatterers and derives the expression of the second threshold, which avoids the quantization loss caused by energy integration and provides better robustness and detection performance. In addition, the concept of "effective scatterers" is introduced in Refs. 9 to extract possible scatterers in HRRP, which improves the computation of the first threshold of the algorithm.

Further author information: (Send correspondence to Xinrong Xu)
Xinrong Xu: E-mail: xuxinrong0704@163.com
Weixing Sheng: E-mail: shengwx@njust.edu.cn
Renli Zhang: E-mail: zhangrenli_nust@163.com
Xiaoyu Cong: E-mail: cxiaoyu1942@126.com

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 1351903 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057716

A dual-threshold range-spread target detection method (OESS-RSTD) based on online estimation of strong scattering points was proposed in Refs. 10, which estimates the number of strong scattering points by means of the K-means clustering algorithm. A weighted dual-threshold GLRT detector was proposed in Refs. 11. A weighting method was employed to optimize the calculation of the second threshold. However, during the calculation of the second threshold, it is not guaranteed that the optimal weighting factor can always be obtained, which results in only a limited improvement in detection performance.

In this paper, an improved dual-threshold GLRT range-spread target detector is proposed. The concept of the "effective scatterer" is introduced to extract the strong scatterer cells of the targets. Based on the GLRT-DTW detector, the Lilliefors test is introduced for pre-judgement in the proposed detector to improve the weighting method to calculate the second threshold based on the pre-judgement result. Simulation results show that the proposed detector has better detection performance than the traditional detectors.

2. DUAL THRESHOLD GLRT DETECTOR

2.1 mathematical model

In the target detection problem of high-resolution radar, the radar echo signal is matched filtered with the transmitted signal to obtain HRRP data. Fig. 1 illustrates the schematic of the to-be-detected HRRP data for the range-spread target detector.

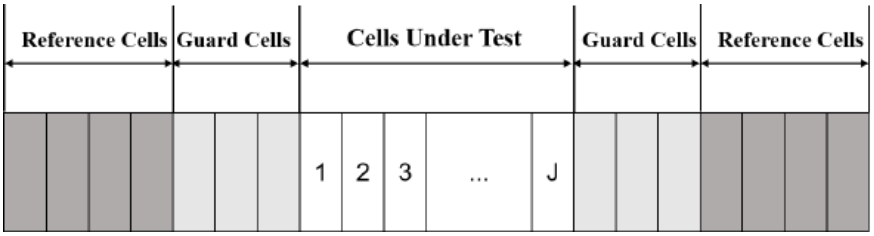


Figure 1. the schematic of the to-be-detected HRRP data for the range-spread target detector.

For the dual-threshold target detection algorithm, J-length vector $\boldsymbol{x} = \{x_1, x_2, ..., x_J\}$ is defined as the HRRP data to-be-detected. The target signal is $\boldsymbol{s} = \{s_1, s_2, ..., s_J\}$, and the output of complex Gaussian white noise with variance σ^2 is $\boldsymbol{n} = \{n_1, n_2, ..., n_J\}$. Thus, the problem can be formulated as a binary detection: $H_0 : x_i = n_i$ indicates that the target does not exist and contains only complex Gaussian white noise; $H_0 : x_i = s_i + n_i$ indicates that the target exists and the echo signal contains both the target signal and the noise signal. At the same time, it is convenient to obtain the output of the square-law detection $\boldsymbol{y} = \{y_1, y_2, ..., y_J\} = \{|x_1|^2, |x_2|^2, ..., |x_J|^2\}$.

Under the H_0 assumption, the echo signal after square-rate detection obeys an exponential distribution with parameter σ^2 and the probability density function (PDF) is:

$$p(y_i; H_0) = \frac{1}{\sigma^2}\exp(-\frac{y_i}{\sigma^2}) \tag{1}$$

The cumulative distribution probability is:

$$P_Y(y; H_0) = 1 - \exp(-\frac{y}{\sigma^2}) \tag{2}$$

For the dual-threshold target detection algorithm, we estimate the number of "possible scatterers" K by setting a first threshold $Th1$. At this point, we assume that the target consists of K scatterers and define the index position of the range cell that contains the scatterers as $i_1, i_2, ..., i_K$. Thus, we can get $|s_{i_1}|, |s_{i_2}|, ..., |s_{i_K}| > 0$. Then the likelihood function can be expressed as:

$$p(\boldsymbol{x}; H_0) = \prod_{m=1}^{J} p(x_{i_m}|H_0) = \left(\frac{1}{\pi\sigma^2}\right)^J \exp\left(-\sum_{m=1}^{J}\frac{|x_{i_m}|^2}{\sigma^2}\right) \tag{3}$$

$$
\begin{aligned}
p(\boldsymbol{x};H_1) &= p(\boldsymbol{x}|s_{i_1},s_{i_2},...,s_{i_K};H_1) \\
&= \left(\frac{1}{\pi\sigma^2}\right)^J \exp\left(-\sum_{m=1}^{J}\frac{|x_{i_m}-s_{i_m}|^2}{\sigma^2}\right)
\end{aligned} \tag{4}
$$

The likelihood ratio is expressed as:

$$
\begin{aligned}
\Lambda(s_{i_1},s_{i_2},...,s_{i_K}) &= \frac{p(\boldsymbol{x};H_1)}{p(\boldsymbol{x};H_0)} = \frac{p(\boldsymbol{x}|s_{i_1},s_{i_2},...,s_{i_K};H_1)}{p(\boldsymbol{x};H_0)} \\
&= \exp\left[\frac{1}{\sigma^2}\left(\sum_{m=1}^{K}|x_{i_m}|^2 - \sum_{m=1}^{K}|x_{i_m}-s_{i_m}|^2\right)\right]
\end{aligned} \tag{5}
$$

When $x_{i_m}=s_{i_m}$, $|x_{i_m}-s_{i_m}|^2$ is minimized and $\Lambda(s_{i_1},s_{i_2},...,s_{i_K})$ is maximized. Therefore, the process of generalized likelihood ratio test (GLRT) can be expressed as follows:

$$
\Lambda_K = \sum_{m=1}^{K} y_{i_m} = \sum_{m=1}^{K} y^{(m)} \underset{H_0}{\overset{H_1}{\gtrless}} Th2, \tag{6}
$$

where variable $y^{(m)}$ is used to represent the m-th strongest element of $\boldsymbol{y}$. $\boldsymbol{y}=[y_1,y_2,\ldots,y_J]$ in descending order can be expressed as:$\{y^{(1)},y^{(2)},...,y^{(J)}\}$.

2.2 Threshold setting

For a dual-threshold target detector, the first threshold is usually utilized to extract the "possible target scatterers", and then the energy accumulation of the "possible target scatterers" is compared with the second threshold to obtain the detection result. In GLRT-DT, the first threshold of the detector is determined by the Akaike Information Criterion (AIC)[12] to obtain $Th1=\sigma^2$.

The total false alarm probability of the detector can be expressed as:

$$
P_{fa} = \sum_{K=1}^{J} P(K;H_0)P_{fa_K}, \tag{7}
$$

where $P(K;H_0)$ denotes the probability that the estimated number of strong scattering cells is K in the case of H_0; P_{fa_K} denotes the false alarm probability of the second threshold, and the expression is:

$$
P_{fa_K} = \int_{Th2}^{\infty} p_{\Lambda_K}(\Lambda|K;H_0)d\Lambda \tag{8}
$$

where $p_{\Lambda_K}(\Lambda|K;H_0)$ denotes the conditional probability density function of Λ_K in the case of different K under the H_0 assumption. And The expressions of P_{fa_K} and $p_{\Lambda_K}(\Lambda|K;H_0)$ are given in Refs. 8:

$$
P(K;H_0) = C_J^K\left[1-\exp\left(-\frac{Th1}{\sigma^2}\right)\right]^{J-K}\exp\left(-\frac{K\cdot Th1}{\sigma^2}\right) \tag{9}
$$

$$
p_{\Lambda_K}(\Lambda|K;H_0) = \left(\frac{1}{\sigma^2}\right)^K \cdot \frac{(\Lambda-K\cdot Th1)^{K-1}}{(K-1)!}\exp\left(-\frac{\Lambda-K\cdot Th1}{\sigma^2}\right) \tag{10}
$$

Hence, the value of the second threshold $Th2$ is related to P_{fa_K} according to Eq. (8). Then, combining Eq. (10) with the cumulative distribution function of the gamma distribution, the expression for the second threshold can be given by Refs. 8:

$$
Th2 = G^{-1}(1-P_{fa_K};K,\sigma^2) + K\cdot Th1, \tag{11}
$$

where G^{-1} denotes the inverse of the cumulative distribution function of the gamma distribution.

3. THE PROPOSED IMPROVED RANGE-SPREAD TARGET DETECTOR

3.1 Problem analysis

In GLRT-DT,[8] in order to simplify the calculation, it is assumed that P_{fa_K} in case of different values of K are equal, that is:

$$P_{fa_1} = P_{fa_2} = ... = P_{fa_J}, K = 1, 2, ..., J \tag{12}$$

However, in practice, P_{fa_K} should be different under different values of K, and accordingly the second threshold $Th2$ is changed. The GLRT-DTW[11] detector is an extension of the GLRT-DT detector by introducing weighting factor to improve the calculation of P_{fa_K}:

$$w_K P_{fa_K} = \gamma, \tag{13}$$

where γ denotes a constant and an expression for the weighting factor w_K is given:

$$w_K = \lambda \cdot sigmoid(|K - \alpha|) + \epsilon, \tag{14}$$

where λ and ϵ are two constants to adjust the range of values of w_K; α denotes the desired number of strong scatterers determined by the concept of effective scatterers.[13] Therefore, when $K = \alpha$, the weighting factor w_K btains the minimum value and P_{fa_K} achieves the maximum value. According to Eq. (8), at this point $Th2$ is reduced accordingly, which improves the detection probability of the algorithm.

The above method partly improves the detection performance of the detector. However, to obtain the optimal weighting factor, it is necessary to make the number of scatterers K passing the first threshold equal to the desired number of "effective scatterers" α. But it is unable to measure the probability that the algorithm achieves the optimal weight factor. During the actual detection process, the following two cases may occur:

(1) The occurrence of $K \neq \alpha$ in the H_1 case leads to the inability to obtain the optimal weighting factor and reduces the algorithm's detection performance;

(2) The occurrence of $K = \alpha$ in the H_0 case leads to an increase in the false alarm probability of the algorithm.

Therefore, further optimization of the calculation of the second threshold can be considered to enable the detector to increase the detection probability more accurately and ensure the false alarm probability.

3.2 Improved dual-threshold GLRT detector based on Lilliefors test pre-judgement

In order to solve the above problems, in this paper, the Lilliefors test pre-judgement is introduced to optimize the calculation of the detector threshold, which can not only effectively improve the detection probability, but also ensure the stability of the overall false alarm probability.[14] Firstly, the binary hypothesis is established based on the Lilliefors test: null hypothesis $H_{LT} = 0$: the data obey the exponential distribution; alternative hypothesis $H_{LT} = 1$: the data disobey the exponential distribution.

In the proposed detector, the first threshold is determined by the "effective scatterer" introduced in ESS-GLRT.[9] The concept of effective scatterers α was introduced in Refs. 13. The number of effective scatterers is determined by the fact that the energy from the α-th strong scatterer in the HRRP is more than half of the average energy of the previous $\alpha - 1$ strong scatterers.[9] In ESS-GLRT, the first threshold is set as $Th1 = y^{(\alpha)}$. However, when the value of Th1 is in the range of $(y^{(\alpha+1)}, y^{(\alpha)}]$, the number of scatterers passing the first threshold is α. According to Eq. (11), the setting of the second threshold is positively correlated with the first threshold. Therefore, when $H_{LT} = 1$, we can appropriately lower the value of Th1 to approach $y^{(\alpha)}$ as much as possible to improve the detection probability:

$$Th1 = \begin{cases} y^{(\alpha+1)} + \frac{y^{(\alpha)} - y^{(\alpha+1)}}{10} & , H_{LT} = 1 \\ y^{(\alpha)} & , H_{LT} = 0 \end{cases} \tag{15}$$

In addition, the significance level ρ in the Lilliefors test can be used to control the accuracy of the pre-judgement. In this paper, the significance level ρ can be expressed as the false alarm probability of the Lilliefors test. Thus, the total false alarm probability can be expressed as:

$$\begin{aligned} P_{fa} = \sum_{K=1}^{J} P(K;H_0)P_{fa_K} &= (1-\rho)P_{fa_K}^{(0)} + \rho \cdot P_{fa_K}^{(1)} \\ &= (1-\rho)\sum_{K=1}^{J} P(K;H_0)P_{fa_K}^{(0)} + \rho \sum_{K=1}^{J} P(K;H_0)P_{fa_K}^{(1)} \end{aligned} \tag{16}$$

where $P_{fa_K}^{(0)}$ and $P_{fa_K}^{(1)}$ are applied to $H_{LT}=0$ and $H_{LT}=1$, respectively. Thus, we can set:

$$\begin{cases} \omega_1 \cdot P_{fa_1}^{(1)} = \omega_2 \cdot P_{fa_2}^{(1)} = ... = \omega_J \cdot P_{fa_J}^{(1)} & , H_{LT}=1 \\ \frac{P_{fa_1}^{(1)}}{\omega_1} = \frac{P_{fa_2}^{(1)}}{\omega_2} = ... = \frac{P_{fa_J}^{(1)}}{\omega_J} & , H_{LT}=0 \end{cases}, \tag{17}$$

where $\omega_j, j=1,2,\ldots,J$ denotes the weighting factor and β is a constant. The weighting factor is set to:

$$\omega_j = sigmoid(|j-K|), \tag{18}$$

where the nonlinear transformation function $sigmoid(x)$ is $sigmoid(x)=\frac{1}{1+e^{-x}}$ and the optimal value of ω_j is obtained at $j=K$. The weighting factor ω_j can be always kept at the optimal value for different under different values of K. According to Eq. (17), when the pre-judgement result is $H_{LT}=1$, $P_{fa_K}^{(1)}$ achieves the maximum value, and accordingly the value of the second threshold $Th2$ is reduced, which improves the detection probability of the algorithm; when the pre-judgement result is $H_{LT}=0$, $P_{fa_K}^{(0)}$ achieves the minimum value, and $Th2$ is increased accordingly ensuring the total false alarm probability of the algorithm.

By substituting Eq. (17) into Eq. (16) and making P_{fa} equal to the desired false alarm probability p_{fa}, the equation can be expressed as:

$$\begin{aligned} P_{fa} &= (1-\rho)\sum_{K=1}^{J} P(K;H_0)P_{fa_K}^{(0)} + \rho \sum_{K=1}^{J} P(K;H_0)P_{fa_K}^{(1)} \\ &= (1-\rho)\sum_{K=1}^{J} P(K;H_0)\omega_K \cdot \beta + \rho \sum_{K=1}^{J} \frac{P(K;H_0)}{\omega_K}\beta \\ &= p_{fa} \end{aligned} \tag{19}$$

The expression for β is derived as:

$$\beta = \frac{p_{fa}}{(1-\rho)\sum_{K=1}^{J} P(K;H_0)\omega_K + \rho \sum_{K=1}^{J} \frac{P(K;H_0)}{\omega_K}} \tag{20}$$

Thus, the expression for P_{fa_K} is:

$$P_{fa_K} = \begin{cases} P_{fa_K}^{(1)} = \frac{\beta}{\omega_K} = \frac{p_{fa}}{\omega_K \cdot [(1-\rho)\sum_{j=1}^{J} P(j;H_0)\omega_j + \rho \sum_{j=1}^{J} \frac{P(j;H_0)}{\omega_j}]} & , H_{LT}=1 \\ P_{fa_K}^{(0)} = \omega_K \cdot \beta = \frac{\omega_K \cdot p_{fa}}{(1-\rho)\sum_{j=1}^{J} P(j;H_0)\omega_j + \rho \sum_{j=1}^{J} \frac{P(j;H_0)}{\omega_j}} & , H_{LT}=0 \end{cases} \tag{21}$$

The overall block diagram of the proposed dual threshold detector is shown in Fig. 2:

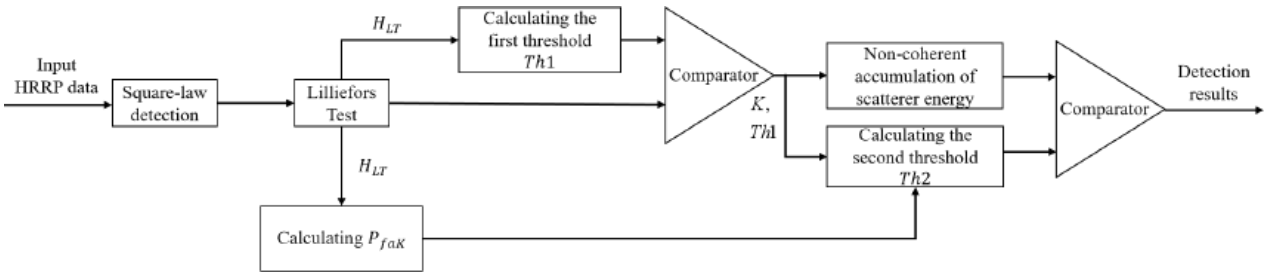


Figure 2. Block diagram of the proposed detector

The overall detection steps are as follows:

(1) The Lilliefors test is utilized to obtain the pre-judgement result H_{LT} for the HRRP data to-be-detected;

(2) The first threshold $Th1$ is obtained by combining the pre-judgement result H_{LT} with Eq. (15). The HRRP data are then compared with $Th1$ to obtain the number of scatterers K passing the first threshold;

(3) Combined with the pre-judgement result H_{LT}, the second threshold $Th2$ is calculated. Then non-coherent accumulation is used for K strong scatterer cells to compute statistics $\Lambda_K = \sum_{m=1}^{K} y^{(m)}$. The statistic Λ_K is compared with the second threshold $Th2$ to the target detection result.

4. SIMULATION RESULTS

The proposed algorithm is compared with Integrator,[7] GLRT-DT,[8] ESS-GLRT,[9] OESS-RSTD[10] and GLRT-DTW.[11] In the simulation, the signal-to-noise ratio (SNR) is calculated as the ratio of the total signal of all scattering points to the noise power:

$$SNR = 10lg(\sum_{k=1}^{K} \frac{A_k^2}{\sigma^2}), \tag{22}$$

where A_k is the amplitude of the target strong scatterer echo.

Simulation is performed under three different target strong scatterer distribution models, as shown in Tab. 1:

Table 1. Typical Scattering Models

Model name	Distribution characteristics of scatterers
sparse uniform	4 scatterers, each accounted for 25% of the total energy
sparse nonuniform	6 scatterers, one accounted for 30%, one accounted for 20%, and the rest account for 10% respectively
dense nonuniform	22 scatterers, one accounted for 40%, two accounted for 20%, and the rest account for 1% respectively

In the simulation, the length of the to-be-detected area is set to $J = 64$. The expected total false alarm probability p_{fa} is 10^{-4}. The number of Monte Carlo trials is set to 10^6. Then the simulation results are shown below:

(1) Fig. 3(a) shows the simulation results for the sparse uniform distribution model. The performance of the individual detectors from excellent to poor is as follows: Proposed, ESS-GLRT,[9] GLRT-DTW,[11] GLRT-DT,[8] OESS-RSTD,[10] Integrator.[7]

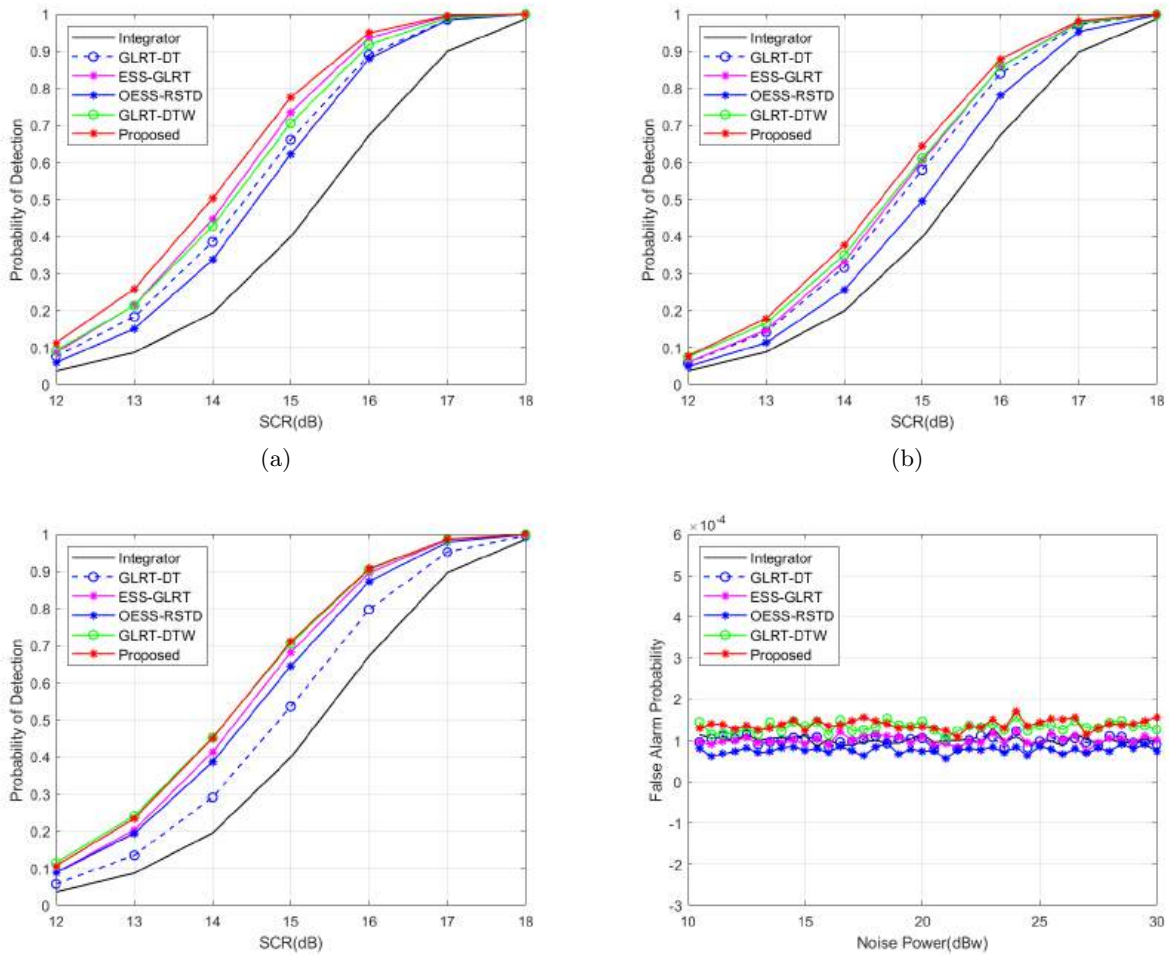


Figure 3. Comparison of detector performance: (a)Under sparse uniform model; (b)Under sparse non-uniform model; (c)Under dense non-uniform model; (d)The false alarm probability curves versus noise power for detectors.

(2) Fig. 3(b) shows the simulation results for the sparse nonuniform distribution model. The performance of the individual detectors from excellent to poor is as follows: Proposed, GLRT-DTW,[11] ESS-GLRT,[9] GLRT-DT,[8] OESS-RSTD,[10] Integrator.[7]

(3) Fig. 3(c) shows the simulation results for the dense nonuniform distribution model. The performance of the individual detectors from excellent to poor is as follows: GLRT-DTW,[11] Proposed, ESS-GLRT,[9] OESS-RSTD,[10] GLRT-DT,[8] Integrator.[7]

From the simulation results of the above three different target strong scatterer distribution models, the proposed detector always has better detection performance regardless of the uniform or non-uniform, sparse or dense distribution conditions of strong scatterers. Hence, the proposed detection method effectively improves the detection performance. Moreover, according to Fig. 3(d), the proposed detector is able to maintain a stable false alarm probability under different noise power.

5. CONCLUSION

An improved dual threshold GLRT detector is proposed in this paper. In this detector, Lilliefors test is introduced for pre-judgment. Then based on the pre-judgment result, the calculation of the first and second thresholds are improved. Simulation results show that the proposed detector outperforms the ESS-GLRT, GLRT-DTW, GLRT-DT, OESS-RSTD, and Integrator detectors in the detection performance.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61971224.

REFERENCES

[1] T. Long, Z. L. and Liu, Q., "Advanced technology of high-resolution radar: target detection, tracking, imaging, and recognition," *Sci. China Inf. Sci.* **62**(no.040301), 1–26 (2019).

[2] Z. Ren, W. Yi, W. Z. and Kong, L., "Range-spread target detection based on adaptive scattering centers estimation," *IEEE Trans. Geosci. Remote Sens.* **61**(no.5100414), 1–14 (2023).

[3] S. Xu, J. Xue, P. S., "Adaptive detection of range-spread targets in compound gaussian clutter with the square root of inverse gaussian texture," *Digit. Signal Process.* **56**, 132–139 (2016).

[4] B. Xiong, Z. Wang, Q. H. and He, Z., "Model-based adaptive detector of range-spread targets with secondary data support," *Proc. IGARSS* , 2789–2792 (2022).

[5] Z. Wang, G. Li, M. L., "Adaptive detection of distributed target in the presence of signal mismatch in compound gaussian clutter," *Digit. Signal Process.* **102**(no.102755) (2020).

[6] T. Jian, Z. Xie, H. W. G. W. J. H., "Persymmetric subspace glrt-based detector for range-spread targets," *Digit. Signal Process.* **129**(no.103658) (2022).

[7] Hughes, P. K., "A high-resolution radar detection strategy," *IEEE Trans. Aerosp. Electron. Syst.* **AES-19**(no.5), 663–667 (1983).

[8] T. Long, L. Zheng, Y. L. X. Y., "Improved double threshold detector for spatially distributed target," *IEICE Trans. Commun.* **95**(no.4), 1475–1478 (2012).

[9] K. Qu, X. Yang, Q. L. Z. L. and Huang, G., "Improved generalized likelihood ratio detector for range-spread target," *Proc. ICSIDP* , 1–5 (2019).

[10] P. Guo, Z. Liu, D. L., "Range spread target detection based on online estimation of strong scattering points," *J. Electron. Inf. Techn.* **42**(no.4), 910–916 (2020).

[11] B. Liu, X. Xu, W. S. X. C., "A robust weighted glrt-dt range spread target detector without priori information," *Signal Process.* **225**(no.109622) (2024).

[12] Akaike, H., "A new look at the statistical model identification," *Signal Process.* **19**(no.6), 716–723 (1974).

[13] Q. Fu, e. a., "Analysis of radar detection performance for low altitude small target," *Proc. ICCSP* (2016).

[14] Lilliefors, H. W., "On the kolmogorov-smirnov test for normality with mean and variance unknown," *J. Am. Stat. Assoc.* **62**(no.318), 399–402 (1967).

A Hybrid Prediction Method for Lithium-Ion Battery Degradation: SMA-ARIMA-LSTM Integration

Haoran Li *[a], Yanhong Bai [a,b], Yongchao Sun [a], Huixue Zhi [a]
[a]Department of Electronic Information Engineering Taiyuan University of Science and Technology,Taiyuan,China;[b]Department of Intelligent Manufacturing and Industry, Shanxi University of Electronic Technology,Linfen,China

ABSTRACT

Accurately predicting the lifespan of lithium-ion batteries is crucial for reducing maintenance costs and advancing clean energy technologies. Traditional prediction methods often to accurately estimate the lifespan due to the diverse volatility characteristics of lithium-ion battery degradation. This paper proposes a Kurtosis-driven SMA-ARIMA-LSTM method to the lifespan of lithium-ion batteries. First, the data is decomposed into low and high volatility components using a moving average (SMA). Then, the I model is applied to the low volatility part, while the LSTM network handles the high volatility part. Finally, the parallel prediction results of these two parts are combined to the remaining lifespan of the lithium-ion battery. The model is validated using four sets of CS2 series lithium-ion battery degradation data provided by the CALCE center at University of Maryland. The results show that the hybrid model significantly improves prediction accuracy compared to standalone LSTM or ARIMA models, achieving near-perfect determination coefficients while significantly mean and root mean square errors. This effectively captures the overall degradation trend of the battery and the capacity regeneration phenomenon. The experimental results demonstrate that the proposed SMA-IMA-LSTM method achieves a fitting rate of over 95%, with MAE kept within 0.9% and RMSE kept within 1.4%, thus realizing precise prediction of the remaining lifespan of lithium-ion batteries.

Keywords: lithium-ion battery, battery capacity prediction, kurtosis optimized moving average, autoregressive integrated moving average, long short-term memory network.

1. INTRODUCTION

To address global warming and the energy crisis, China has adopted a national strategy to peak carbon emissions by 2030 and achieve carbon neutrality by 2060 [1]. Lithium-ion batteries, known for their high energy density, efficiency, environmental benefits, and long lifespan, are widely used in electric vehicles and other applications [2]. However, battery performance declines with repeated charge cycles due to irreversible electrochemical reactions, causing electrode material loss and reduced capacity [3]. When capacity falls to 70%-80% of the rated level, the battery reaches a failure threshold and is no longer suitable for use. Continued operation beyond this point can lead to rapid performance drops and even safety risks [4]. Accurate prediction of battery capacity and Remaining Useful Life (RUL) is crucial for advancing battery technology, ensuring safety, lowering costs, and extending battery life in practical applications.

*1300015104@qq.com ; phone 1 327 572-3829

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE Vol. 13519, 1351904 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057881

Current methods for predicting the remaining useful life (RUL) of lithium-ion batteries fall into two main categories: model-driven and data-driven approaches [5]. Model-driven methods use the battery's chemical and physical properties to create mathematical models. For instance, the Thevenin equivalent model characterizes and predicts battery capacity by analyzing internal electrochemical and physical properties [6]. Electrochemical models consider degradation mechanisms, such as material loss and lithium-ion storage changes, while time and frequency domain equivalent circuit models enhance prediction accuracy and reveal battery behavior. However, these approaches require a deep understanding of battery degradation mechanisms and depend heavily on specialized knowledge, making online prediction challenging [7].

On the other hand, data-driven methods use statistical and machine learning techniques to directly predict battery life from historical data, bypassing the need theoretical expertise and allowing for the consideration of complex variations. For instance, Robert R. Richardson and his team adopted a data-driven predictive approach using Gaussian Process Regression leveraging machine learning algorithms to analyze historical operation data to predict the health state of batteries [8]. Vo Thanh Ha and his team adopted a triple regression model experimental method in the study of remaining life prediction of lithium-ion batteries electric vehicles [9]. They compared linear regression, bagging regressor, and random forest regressor to model the battery cells based on voltage-related parameters, thereby achieving accurate of lithium-ion battery capacity. ZHANG Y Z et al. used the long short-term memory (LSTM) model to predict the remaining useful life of. LSTM can effectively handle time series data, automatically capturing the dynamic patterns of battery capacity decay, thereby improving the accuracy and robustness of the predictions [10]. However, due to the characteristics of lithium-ion batteries, such as capacity regeneration during charging and external noise, precise become difficult [11]. The aging of batteries is influenced by various factors, such as temperature, discharge rate, cycle number, and manufacturing process, making it challenging for a model to consider all these factors simultaneously, which can lead to inaccuracies in predictions.

To address these challenges, this study employs a multi-model integration approach to enhance the accuracy and robustness of lithium battery life prediction. Battery capacity is used as a key indicator, and a method combining Simple Moving Average (SMA), LSTM, and ARIMA models is proposed. First, SMA is applied for initial decomposition, using kurtosis to optimize the moving average period for better data smoothing and adaptability. ARIMA then captures the linear, low-fluctuation trends, while LSTM handles the high-fluctuation, complex patterns. The results from these models are integrated to compensate for the limitations of single-model predictions, providing a comprehensive time series forecast. The method was validated using four battery degradation datasets from the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland. Compared to traditional LSTM and ARIMA models, this integrated approach demonstrates greater robustness, better nonlinear tracking, and improved prediction accuracy for lithium-ion battery aging.

2. THEORETICAL BASIS AND METHOD

2.1 Simple moving average technique

Simple moving average (SMA) is a commonly used time series smoothing technique, which is used for the preliminary decomposition of time series in this study [12]. The basic principle of SMA is to smooth time series data by calculating the arithmetic average of data points within a fixed time window. The mathematical formula of SMA can be expressed as:

$$SMA_t = \frac{X_t + X_{t-1} + \cdots + X_{t-n+1}}{n} \tag{1}$$

Where SMA_t is the simple moving average at time t, X_t , X_{t-1} , ⋯, X_{t-n+1} is the n consecutive data points in the time series, and n is the selected time window size.

In this study, SMA was used to decompose the original time series into two parts, low and high volatility. The low volatility is calculated directly by SMA and reflects the overall trend and cyclical changes in the time series. The high-volatility part is obtained by subtracting SMA from the raw data, which includes short-term fluctuations and noise. The advantage of this decomposition method is that it can effectively separate long-term trends and short-term fluctuations in time series.

2.2 Autoregressive integral moving average model

Autoregressive integral moving average model (ARIMA) is a statistical model widely used in time series analysis and prediction. The ARIMA model combines autoregressive (AR), differential (I), and moving average (MA) components to effectively capture trend, seasonal, and cyclical patterns in time series data.

The ARIMA model is expressed as ARIMA(p,d,q), where p is the order of the autoregressive term, d is the order of the difference term and q is the order of the moving average term. The basic formula of the ARIMA model is as follows:

$$\Phi_p(B)(1-B)^d X_t = \theta_q(B)\varepsilon_t \tag{2}$$

Where B is the lag operator, $\Phi_p(B)$ is the autoregressive polynomial, representing the d order difference, $\theta_q(B)$ is the moving average polynomial, ε_t is the white noise process.

In this study, In this study, the ARIMA model is primarily used to predict the low-fluctuation segment obtained through SMA decomposition. The strength of the ARIMA model lies in its ability to handle non-stationary time series and effectively capture linear relationships and short-term dependencies in the data. By applying the ARIMA model to the low-fluctuation segment, it can effectively simulate and predict long-term trends and cyclical patterns in lithium battery life data.

2.3 Long short-term memory network

A Long short-term memory network (LSTM) is an advanced form of recurrent neural network (RNN) that is particularly suitable for solving problems that require understanding long time series data. The LSTM unit consists of three main components: the forget gate, the input gate and the output gate, whose structure is shown in Figure.1. These components work together to optimize the management of information flow, especially in processing complex battery performance data, in the specific application of battery life prediction, the LSTM model can be described by the following mathematical expression:

Forget gate: Decides which prior information is irrelevant and removes it from the cell state.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{3}$$

Among them, f_t is the output of the forget gate, W_f is the weight of the forget gate, b_f is the bias item, σ is the sigmoid activation function, h_{t-1} is the hidden state of the previous time step, x_t is the input of the current time step.

Input gate: Updates the battery status to add new and valuable information.

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{4}$$

Where i_t is the output of the input gate, W_i is the weight of the input gate, b_i is the bias term.

Output gate: Based on the current cell state, determines what information to output and affects the next prediction of the network.

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

Where o_t is the output of the output gate, W_o is the weight of the output gate, b_o is the bias item.

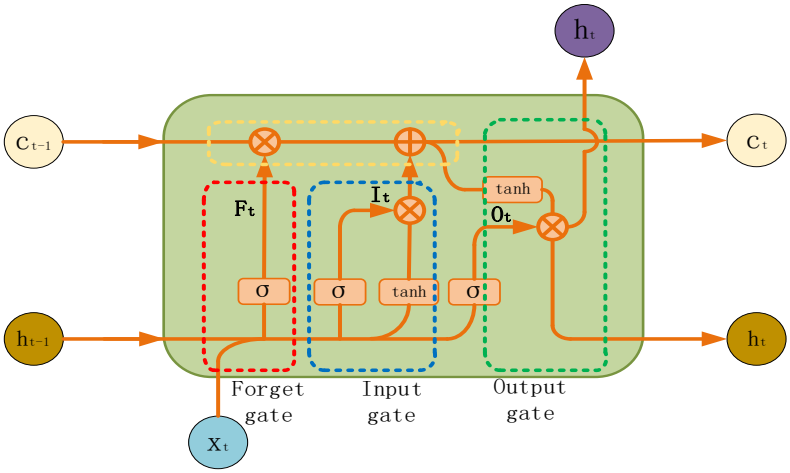


Figure 1: Structure of LSTM.

3. SUMMARY OF EXPERIMENTAL DATA AND CONSTRUCTION OF PREDICTION MODEL

3.1 Degenerate data set

Using a data set provided by the Advanced Life Cycle Engineering (CALCE) Center at the University of Maryland, the research team used four CS2 series lithium-ion batteries, numbered CS2_35, CS2_36, CS2_37, and CS2_38. These batteries are tested under a standard constant-current/constant-voltage charging protocol, with an initial charging current of 0.5C until the voltage reaches 4.2V, and then maintain that voltage until the charging current drops below 0.05A. In the discharge test, the battery runs at a constant current of 1C until the voltage drops to 2.7V. The standard of battery life is to reduce the capacity from the initial 1100mAh to 70%, or 770mAh. The degradation capacity change curve of lithium battery is shown in Figure.2.

3.2 Construction of prediction model

This method uses the capacity of lithium battery as the prediction target, analyzes the data characteristics according to the different volatility characteristics in the degradation trend, and combines the advantages of different models to predict, so as to improve the prediction accuracy. The specific steps are as follows:

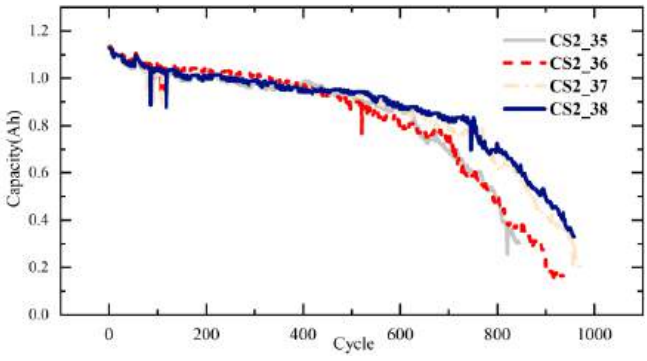


Figure.2: Battery capacity degradation curve.

Data preprocessing and optimization of moving average cycles: Historical capacity data is smoothed using a simple moving average (SMA), and kurtosis is calculated across periods (4 to 99). The period with kurtosis closest to 3 (±5%) is chosen as optimal.Time series decomposition: The optimal SMA is used to split data into low- and high-volatility parts.Data partitioning: Low- and high-volatility data are divided into training and test sets, with the first 20% as training data and the remaining 80% as test data.

Model building and prediction: The ARIMA model is applied to the low-fluctuation part, and the LSTM model to the high-fluctuation part. Prediction results merging and model evaluation: ARIMA and LSTM test predictions are combined to form the final forecast. Performance metrics, including MSE, RMSE, MAPE, R^2, and MAE, are calculated from the test set.

This framework, shown in Figure 3, integrates key steps—data preprocessing, time series decomposition, multi-model prediction, and result evaluation—to enhance accuracy by capturing distinct characteristics in battery data. Specifically, SMA decomposes the time series into high- and low-volatility segments, followed by ARIMA modeling for low volatility and LSTM for high volatility. Combining the two models' predictions enables more accurate forecasting of lithium-ion battery capacity trends and remaining servicelife.

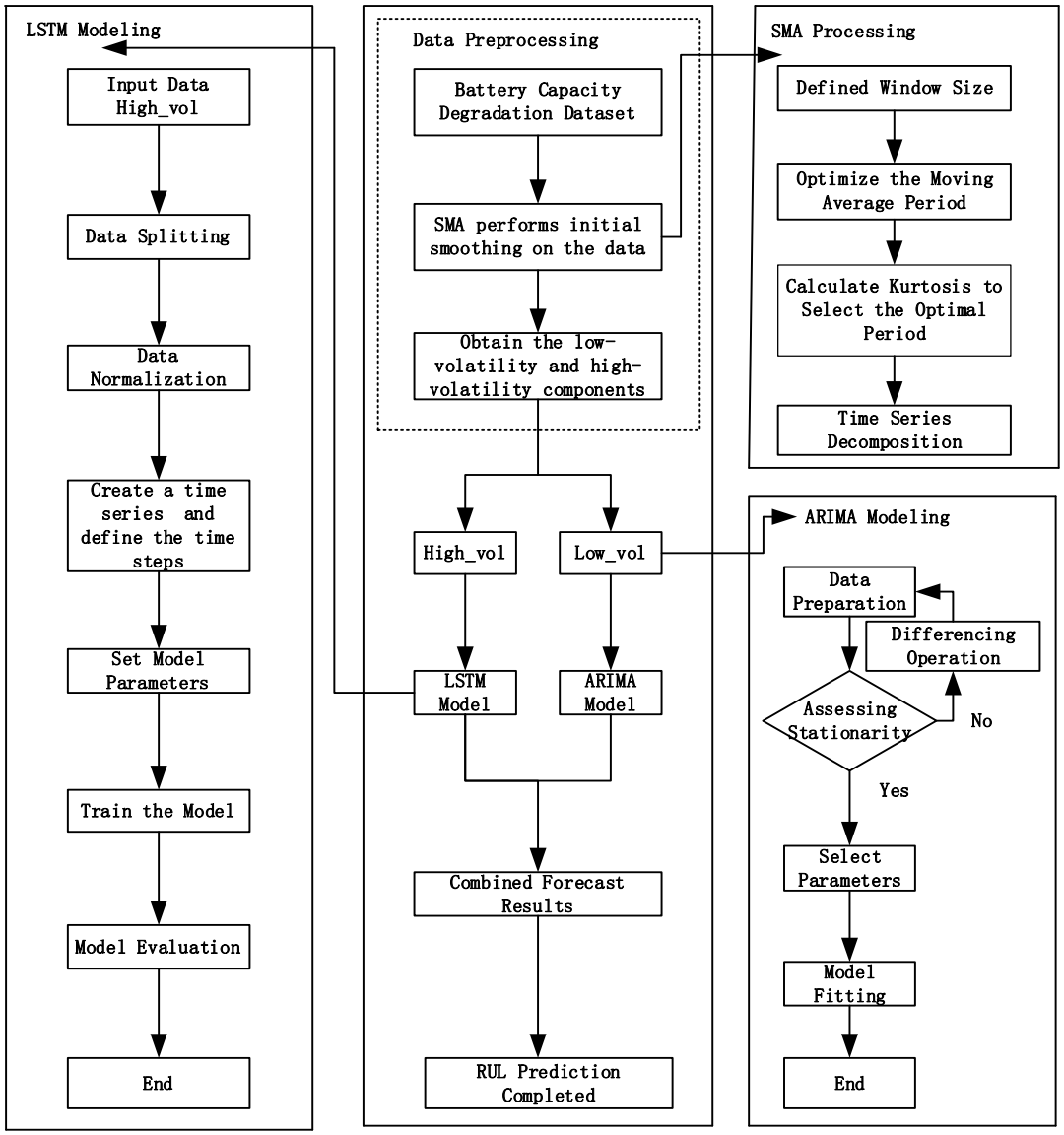


Figure 3: The flow of the prediction model.

4. EXPERIMENTAL RESULTS AND ANALYSIS

This paper predicts the capacity of lithium-ion batteries (CS2_35, CS2_36, CS2_37, and CS2_38)using three methods: the LSTM neural network, the ARIMA model, and a proposed SMA-ARIMA fusion framework for comparison. The dataset is split, with 20% used for training and 80% for testing. Prediction results for each method are shown in Figure 4. Performance is evaluated using mean absolute error (MAE), root mean square error (RMSE), and fit coefficient (R^2), as defined in equations (6)–(10).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{8}$$

$$SS_{res} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{9}$$

$$SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \tag{10}$$

Where n is the sample size, y_i is the actual value of the i observation, $\bar{y}_i$ is the average value of the i observation, $\hat{y}_i$ is the predicted value of the i observation. The results of comparing the proposed method with a single traditional model are shown in Table 1.

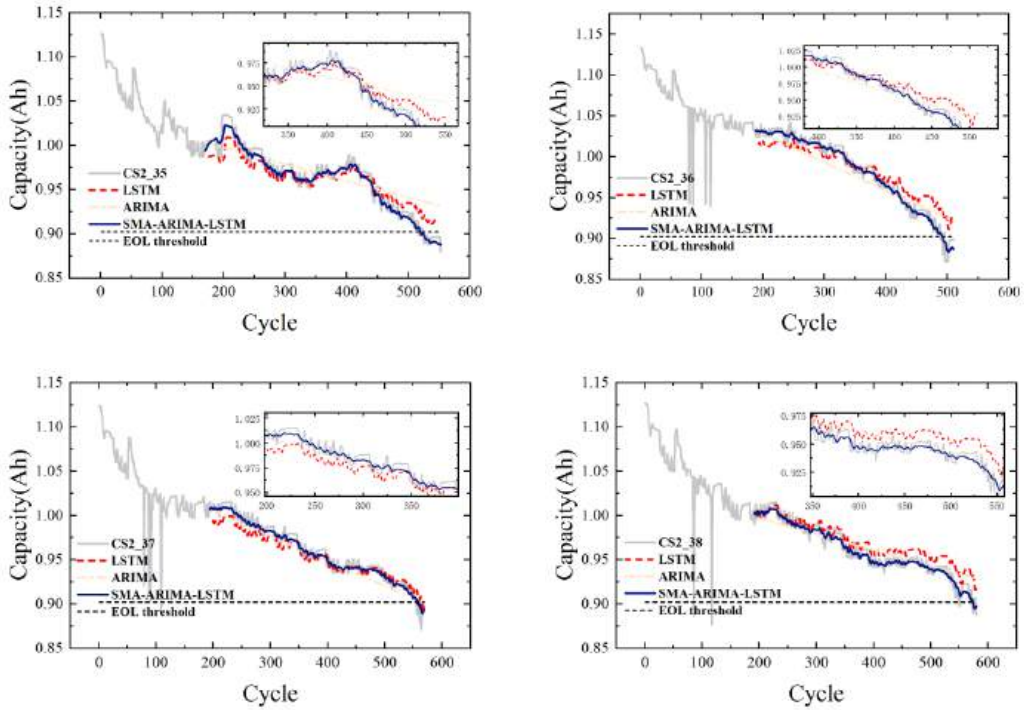


Figure 4: 4 groups of battery prediction results.

As can be intuitively seen from Figure 4, a single traditional model can only roughly predict the degradation trend of the battery due to its inherent limitations, and it does not perform well in capturing nonlinear or complex changes in the battery life (such as capacity regeneration phenomenon), which leads to reduced prediction accuracy. The method proposed in this paper can better address these challenges in battery life prediction. It can more accurately capture the dynamic change of battery capacity, especially in the regeneration stage of the late battery life, thus significantly improving the accuracy and reliability of the prediction, and accurately predicting the remaining service life of the lithium battery.

From Table 1, we can see the key performance indicators of each model, and we can see that the prediction accuracy of a single model is low. On the one hand, this is related to the cause of the data itself, that is, the change range of the data is small and there is a complex nonlinear relationship; on the other hand, the single model is too simple to fully explain the variation in the data. The prediction results of the proposed method show the lowest mean absolute error and root mean square error, and the fitting degree of the predicted curve and the actual curve is higher than 0.95. It can be seen that the proposed fusion method enhances the adaptability and generalization ability of the model to complex data patterns, and improves the prediction accuracy.

Table 1: Evaluation index.

Battery	Evaluation Index	LSTM	ARIMA	SMA-ARIMA-LSTM
CS2_35	MAE	0.0335	0.0154	0.0087
	RMSE	0.0562	0.0194	0.0134
	R^2	0.9096	0.6988	0.9523
CS2_36	MAE	0.0867	0.0141	0.0073
	RMSE	0.1305	0.0172	0.0121
	R^2	0.7963	0.8476	0.9584
CS2_37	MAE	0.0499	0.0123	0.0071
	RMSE	0.0818	0.0126	0.0103
	R^2	0.8435	0.8414	0.9667
CS2_38	MAE	0.0433	0.0089	0.0076
	RMSE	0.0691	0.0118	0.0127
	R^2	0.8448	0.8273	0.9568

5. CONCLUSION

In this study, a novel hybrid method combining kurtosis optimization moving average, ARIMA and LSTM is proposed for the prediction of remaining useful life (RUL) of lithium batteries. By analyzing degradation data from the CALCE Center at the University of Maryland, we verify the effectiveness of this method. Key findings include:

1.The MA-ARMI-LSTM framework is significantly better than a single model in forecasting accuracy, and can accurately capture the degradation trend and regeneration phenomenon of battery capacity.

2.Kurtosis optimization is used to determine the optimal SMA window size, improve the effectiveness of time sequence decomposition, and lay a foundation for subsequent modeling.

3.After breaking down the data into low-volatility and high-volatility segments, ARIMA and LSTM capture linear trends and nonlinear patterns, respectively, improving overall forecasting performance.

This method has important application value to the battery management system, and can improve the efficiency and life of the battery. Despite the remarkable results of this study, there are still some limitations.

First, our model is mainly based on data under laboratory conditions, and future studies can consider verifying the effectiveness of the method in more diverse real-world use scenarios. Second, the integration of other advanced machine learning techniques, such as attention mechanisms or graph neural networks, into the current framework could be explored to further improve predictive performance. In addition, given the differences in the characteristics of different types of

lithium-ion batteries, future studies can also be extended to more types of batteries to verify the generality of the method. Future studies can verify the effectiveness of the method in more real-world scenarios and integrate more advanced techniques to further improve performance.

ACKNOWLEDGMENT

This research was supported by the Research Start-Up Fund for Talent Introduction of Shanxi University of Electronic Technology, Grant No.: 2023RKJ018.

REFERENCES

[1] ZHANG J S, SHEN J L, XU L S, et al. The CO2 emission reduction path towards carbon neutrality in the Chinese steel industry: a review [J].Environ Impact Assess.Rev.2023,99:107017.

[2] Song K, Hu D, Tong Y, et al. Remaining life prediction of lithium-ion batteries based on health management: A review [J]. J. Energy Storage,2023,57:106193.

[3] HUANG J, LI J L, LI Z. A state of health rapid assessment method for decommissioned lithium-ion batteries [J]. Power System Protection and Control,2021,49(12):25-32.

[4] WU Q, XU R L, YANG Q X, et al. Lithium battery capacity estimation method based on PCA and GA-BP neural network [J]. Electronic Measurement Technology,2022,45(6):66-71.

[5] ZHAO, S S, ZHANG C L, WANG, Y Z. Lithium-ion battery capacity and remaining useful life prediction using board learning system and long short-term memory neural network[J]. Energy Storage,2022,52(Part B):104901.

[6] ZHOU Y Z, WANG S L, XIE Y X, et al. A new SOC estimation method for lithium battery based on extended Kalman and Thevenin [J]. Chinese Battery Industry,2019,23(01):39-43.

[7] ZHENG X Y, DENG X G, CHAO Y P. State of health prediction of lithium-ion batteries based on energy-weighted Gaussian process regression[J].Journal of Electronic Measurement and Instrumentation,2020,34(6):63-69.

[8] Robert R. Richardson, Michael A. Osborne, David A. Howey,Gaussian process regression for forecasting battery state of health, Journal of Power Sources, Volume 357, 2017, Pages 209-219, ISSN 0378-7753.

[9] Ha, V.T.; Giang, P.T. Experimental Study on Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Three Regression Models for Electric Vehicle Application. Appl. Sci. 2023, 13, 7660.

[10] ZHANG Y Z, XIONG R, HE H W, et al. Long Short-Term Memory Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-Ion Batteries[J]. IEEE Trans Veh Technol. 2018,67(7):5695-5705.

[11] YE X, WANG H R, LI B Y. Remaining useful life prediction method of lithium-ion battery based on variational mode decomposition and optimized LSTM [J]. electronic measurement technology,2022,45(23):153-158.

[12] IVANOVSKI Z, MILENKOVSKI A, NARASANOV Z. Time Series Forecasting Using a Moving Average Model for Extrapolation of Number of Tourist[J]. UTMS Journal of Economics,2018,9(2):121-132.

An Improved BiTCN Model Merging Multi-Head Attention-BiGRU for Photovoltaic Power Generation Prediction Based on Meteorological Data

Jianhong Gan[a], Xi Lin[a], Changyuan Fan[a], Youming Qu[b], Peiyang Wei[a], Yaoran Huo[c], and Zhibin Li[d]

[a]Chengdu University of Information Technology, Chengdu, China
[b]Hunan Meteorological Bureau Information Center, Changsha, China
[c]Information & Communication Company, State Grid Sichuan Electric Power Company, Chengdu, China
[d]Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi, Xinjiang, China

ABSTRACT

The widespread application of photovoltaic power generation in smart grids makes accurate generation forecasting essential for grid management and planning. Aiming at the randomness and unpredictability of solar energy in PV power generation prediction, this paper proposes a short-term PV power generation prediction model with higher accuracy based on BiTCN, multi-focus mechanism, and BiGRU model. In addition, DRSN is introduced to improve the residual block of BiTCN so as to extract the important features of PV power generation and reduce the redundant features. In addition, the combination of multiple attention mechanisms enables the model to analyze all aspects of the data in parallel, ensuring a comprehensive examination of the information. The BiGRU model accurately captures the long-term dependencies and inherent characteristics of time series data. In order to further improve the prediction accuracy, the AR model was used to optimize the linear extraction ability of the model. The experimental results show that the MAE, RMSE, and R^2 of the proposed model are superior to the traditional model in complex data sets, including weather forecast data, station weather data, and power data. MAE, RMSE and R^2 evaluation indexes show that the model has good prediction accuracy.

Keywords: photovoltaic power generation prediction, BiTCN, multi-head attention mechanism, BiGRU

1. INTRODUCTION

As technology advances, the use of solar PV in smart grids is becoming increasingly widespread. Precise and dependable PV forecasting can offer significant advantages to contemporary smart grid management. However, the PV system is influenced by the environment, weather, and solar radiation, and its output power is unstable and fluctuating.[1] Hence, to minimize the randomness and intermittency of solar power generation. Enhancing the accuracy of PV power prediction is crucial for power system dispatching, safety, and economic efficiency.

In recent years, many researchers have done extensive research on PV power generation forecasting, using a wide range of methods to build models. PV power prediction is a branch of time series forecasting. At present, the proposed time series prediction methods mainly include the physical method, statistical method and probability method, artificial method, and mixed method.

Artificial Intelligence methods have emerged as key components in the current PV power prediction framework. Their robust nonlinear mapping and feature extraction capabilities and rapid progress in computer and data mining technologies have made them essential.[2] Conventional neural network prediction methods surpass non-AI methods in accuracy and better accommodate the nonlinear characteristics of PV power generation. Bai

Further author information:
X.L.: E-mail: 1398038843@qq.com
C.F.: E-mail: 318019168@qq.com

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE Vol. 13519, 1351905 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057853

et al.[3] introduced the TCN. TCN model can handle lengthy input sequences in their entirety, allowing for faster data processing and robust parallel computing capabilities. Through the use of residual blocks and extended causal convolution, TCN maintains strong predictive performance for long-time series, surpassing that of LSTM in some cases.[4] Wang et al.[5] proposed an efficient contract TCN model, which used the improved DRSN to replace the original TCN residuals to improve the accuracy of PV power prediction.

However, due to the random, volatile, and unstable nature of PV electricity, an individual forecast model often fails to meet the standards of engineering practice.[6,7] Limouni et al.[8] developed a hybrid PV prediction model combining weather factors and a TCN-LSTM network, demonstrating superior performance over TCN or LSTM alone. Pu et al.[9] introduced an interactive behavior-learning method based on the TCN-GRU model, which exhibited excellent coupling ability in scenarios with incomplete information, providing an effective solution.

Despite the broad application of TCN in time series analysis, prior research predominantly utilized one-way TCN, which only extracts forward information and overlooks the impact of backward information on predictions.[10] TCN also has the following shortcomings: Secondary redundant feature information can interfere with TCN's feature extraction, affecting final predictions

In summary, to solve the limitation and uni-directivity of TCN, this paper proposes a combination model based on BiTCN and BiGRU. Additionally, DRSN is used to improve the residual block of the BiTCN model, introduce a multi-head attention mechanism, and add a linear AR model to improve the prediction accuracy and optimize the linear extraction ability of the model.

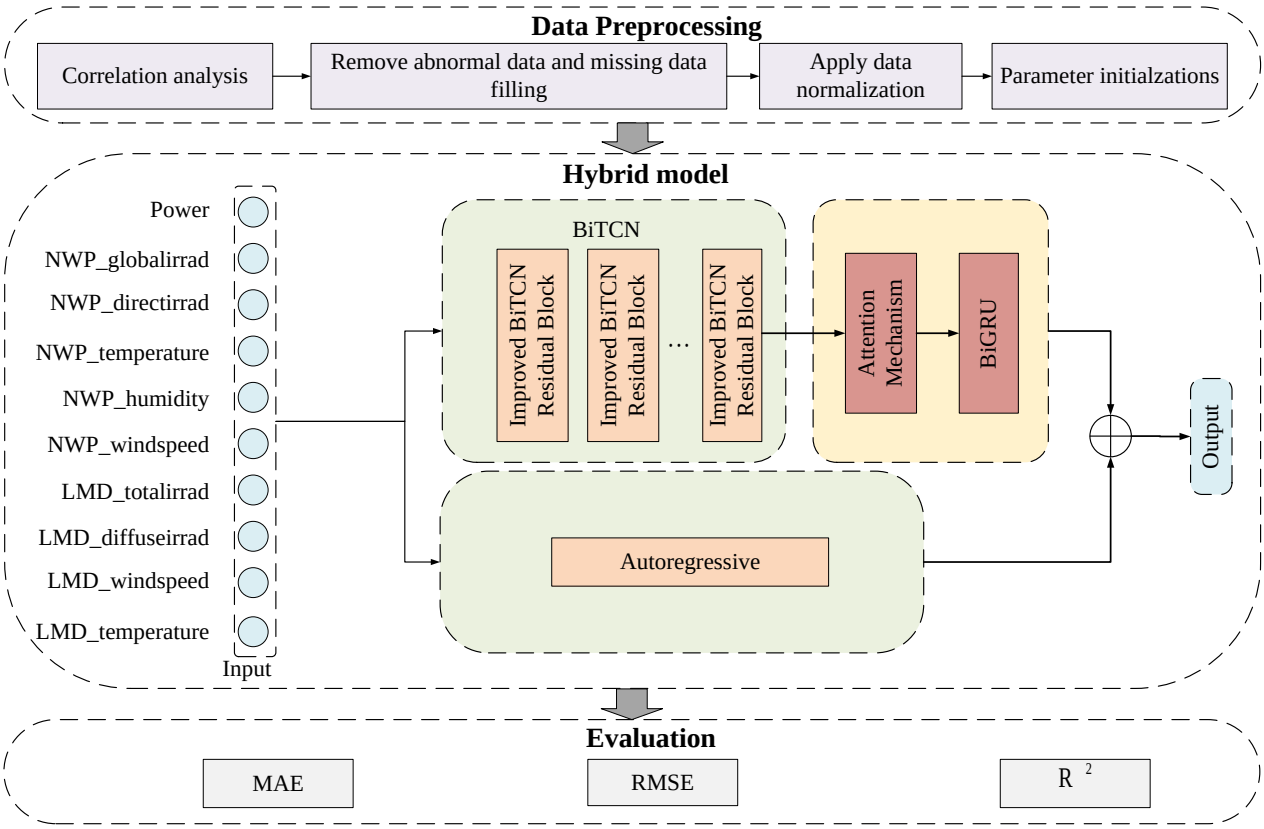


Figure 1: The structure of the bidirectional dilated causal convolutional network.

2. METHODOLOGY

2.1 Proposed method

The combined deep learning model is composed of three main modules: BiTCN, BiGRU, and Multi-Head Attention. The BiTCN takes into account both past and future information of the sequence, capturing bidirectional dependencies more comprehensively than traditional TCN. At the same time, the multi-layer time convolutional network implemented by BiTCN with an increasing expansion rate can adapt to feature learning at different time scales. As a highly efficient bidirectional RNN, BiGRU can effectively capture long-duration dependencies in sequences while minimizing both the model parameters and training duration. It allows the model to focus on different aspects of sequence information in parallel and comprehensively consider the feature representation of multiple subspaces, thereby improving the efficiency of important feature extraction in the sequence. By enhancing the model's focus on key information, the multi-head attention mechanism improves both performance and interpretability in dealing with data that has complex internal structures. In addition, the AR model and the output-weighted combination of the BiTCN-MA-BiGRU model were utilized to achieve the final prediction results. The flowchart of the model structure is shown in Fig.1.

2.2 BiTCN

TCN network is mainly composed of three core modules: causal convolution, extended convolution, and residual connection, which combine the advantages of CNN and RNN.[11] It effectively avoids the problem of gradient hours or gradient explosions that often occur in recurrent neural networks and has the advantages of parallel computation, low memory, improved network performance, and capturing long and short-term time features in input sequences. Time causality in time series prediction is emphasized in the TCN network. For the value of t time of the previous layer, only the value of t time of the next layer and the value before it are dependent. That is to say, given the output $X^{T+1} = x_0, x_1, \ldots, x_T$, and the corresponding predictive output sequence $Y^{T+1} = y_0, y_1, \ldots, y_T$, it is specified that the predictive output of y^t can only use the sequence before t time, the expression is:

$$\hat{y}(t) = F_\theta(x_0, x_1, \ldots, x_t)(\forall)_t \in [0, T-1] \tag{1}$$

Where F_θ represents the forward propagation process in the neural network, and θ represents the parameters in the network.

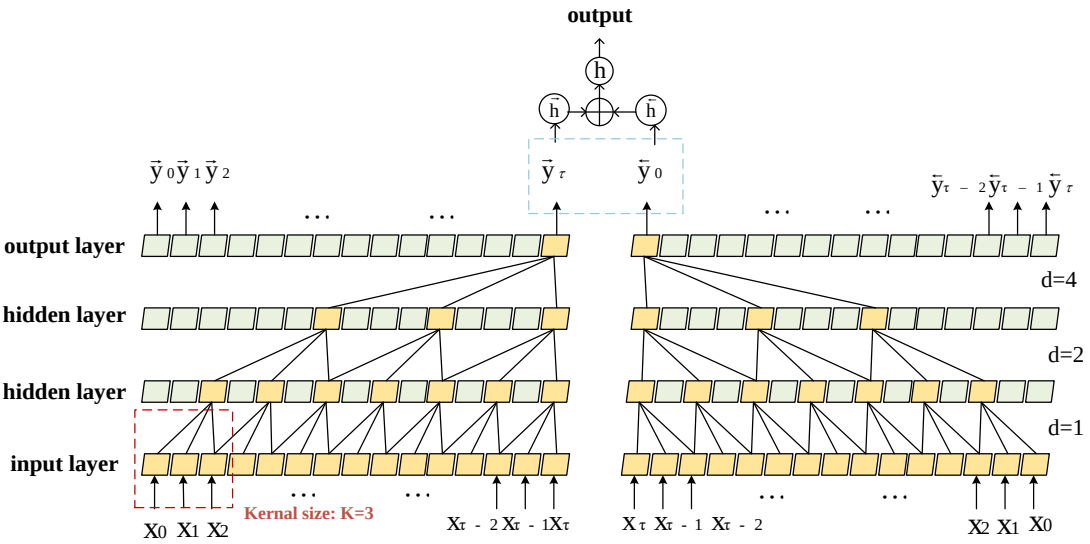


Figure 2: The structure of the bidirectional dilated causal convolutional network.

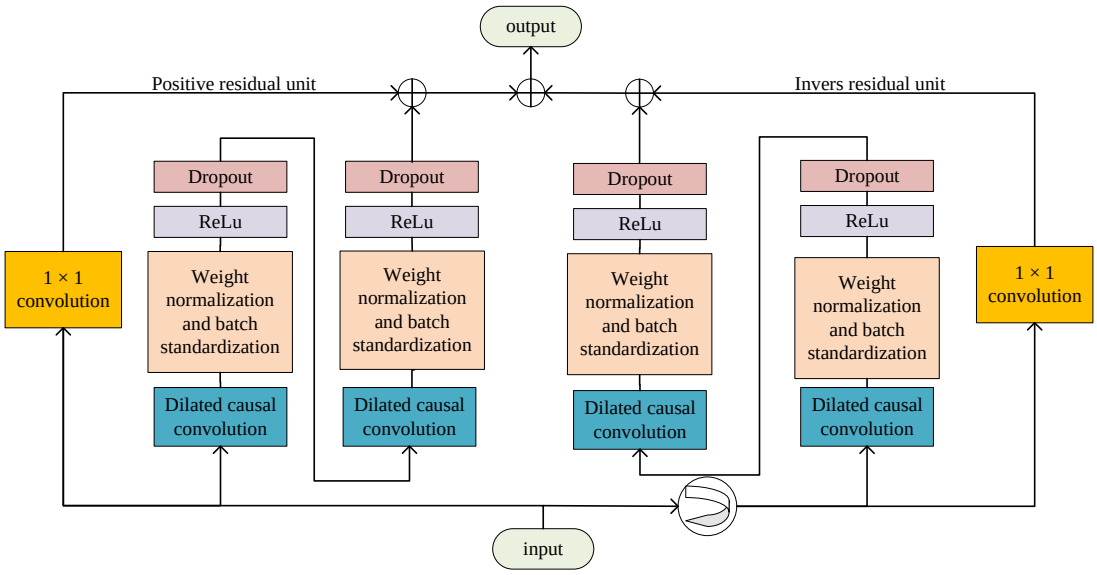


Figure 3: The structure of residual block in BiTCN.

The structure shown on the left in Fig.2 ensures that future data does not leak into the past. Unlike traditional convolution, the TCN convolution layer skips the specified step size by sampling at intervals during convolution so that a larger receptive field and longer time series dependence can be obtained at the same size output. However, since the traditional TCN network only extracts the forward information and ignores the backward information. The TCN network has been modified into a bidirectional version to expand the receptive field with fewer layers while maintaining the feature mapping dimension. Given a 1D sequence of inputs $x \in R^n$ and a convolution filter mapping $0, \cdots, k-1 \in R$, the dilated convolution for the components s in the sequence is defined as follows:

$$F(s) = \sum_{j=0}^{k-1} f(j) \cdot x_{s-d \cdot j} \tag{2}$$

Where k represents the size of the convolution kernel and $s-d \cdot j$ captures past information. The dilation factor d determines the number of zero vectors placed between adjacent convolution kernels. With each application of a convolution layer to the input sequence, the dilation factor d grows exponentially. However, as the number of network layers grows substantially, problems such as gradient attenuation or even the vanishing gradient problem can occur. A residual block is incorporated into BiTCN to mitigate these problems and enable high-efficiency feature extraction from the sequence. The structure of the residual block is illustrated in Fig.3.

2.3 Improved TCN residuals block

The DRSN is an enhanced algorithm for residual networks that integrates attention mechanisms and soft thresholding methods to support autonomous filter learning. Compared with the traditional wavelet threshold, it is more efficient and accurate, which can avoid the inconvenience and blindness of artificial threshold settings and achieve the purpose of reducing the influence of secondary redundancy features on the network.[12] DRSN enhances the original residual network by incorporating a soft thresholding mechanism and an attention mechanism. The soft thresholding mechanism is used for signal noise reduction by setting a threshold value. Features with absolute values beneath this threshold are assigned a value of zero, while other features are shrunk towards 0. The output of the soft thresholding mechanism and its derivatives are as follows:

$$f(x) = \begin{cases} x - \tau, & x > \tau \\ 0, & -\tau \le x \le \tau \\ x + \tau, & x < -\tau \end{cases} \tag{3}$$

$$\frac{d_{f(x)}}{d_x} = \begin{cases} 1, & x > \tau \\ 0, & -\tau \le x \le \tau \\ 1, & x < -\tau \end{cases} \tag{4}$$

Where x is the input value, $f(x)$ is the output after the soft threshold, and τ is the threshold. This approach efficiently lessens the model's load in handling redundant information while enhancing its emphasis on significant features. Consequently, it boosts overall prediction accuracy and enhances the model's robustness. Through this method, DRSN not only optimizes the feature extraction process but also enhances the adaptability and interpretation ability of the model to complex time series data. Additionally, the ReLu activation function was replaced with the GeLu activation function, resulting in less information loss and a smoother model, thereby improving the model's generalization capability.

2.4 BiGRU

GRU networks perform well when processing time series tasks. Compared to LSTM, GRU has a more streamlined design and demonstrates superior performance in convergence speed, parameter updates, and generalization. It effectively captures the dependency relationships within time series data. BiGRU splits the traditional GRU neural unit into forward and reverse transmission states, each corresponding to updating the hidden state based on historical and future data, respectively.

$$R_t = \sigma\left(W_r \cdot [\, h_{t-1}, x_t \,] \right) \tag{5}$$

$$Z_t = \sigma\left(W_z \cdot [\, h_{t-1}, x_t \,] \right) \tag{6}$$

$$\widetilde{h_t} = tanh\left(W_h \cdot [\, R_t * h_{t-1}, x_t \,] \right) \tag{7}$$

$$h_t = \left(1 - Z_t \right) * h_{t-1} + Z_t * \widetilde{h_t} \tag{8}$$

Where Z_t, R_t, $\widetilde{h_t}$ and h_t represent the update gate, reset gate, candidate hidden state, and final hidden state, respectively; W_r, W_z and W_h denote the weight matrices; σ is the sigmoid activation function; $tanh$ is the hyperbolic tangent function; h_{t-1} is the hidden state from the previous time step; and x_t is the input at the current time step.

2.5 Autoregressive model

Autoregressive model (AR) is the process of data autoregressive operation, using x_1 to x_{t-i} to predict the value of x_t time, x_t expression is:

$$x_t = \sum_{i=1}^{p} \varepsilon_i x_{t-i} + \beta_i \tag{9}$$

Where ε_i is a constant coefficient and β_i is a random error. The prediction results in x_i of the AR model, and the output $yt - 1 + \Delta$ value of the BiTCN-MA-BiGRU model are combined by linear weighting to form the final prediction result in y^* expression:

$$y^* = \alpha y_\alpha + \beta y_{t-1+\Delta} \tag{10}$$

In this context, α and β are the weight coefficients, where $\alpha + \beta = 1$.

3. CASE STUDY

3.1 Analyze dataset

The data set used in this paper was recorded every 15 minutes from August 2018 to September 2019 at a power station in Hebei, China. The dataset include weather forecast data, locally recorded weather data, and power generation data. The data set is divided into training set, verification set, and test set in a ratio of 7:1.5:1.5. Since PV power generation mainly relies on solar irradiance, the power generation at night is zero. Therefore, a total of 13 hours of data from 6 AM to 7 PM is selected, and a total of 52 data points are collected. The evaluation indexes used in this paper are RMSE, MAE and R^2.

3.2 Feature analysis method

To minimize the computational complexity of the prediction model, SCC and PCC were employed to assess the relationships among input variables. The Correlation coefficients between meteorological variables and power generation in the data set by using this method are shown in Table 1.

Table 1: Correlation coefficient analysis.

Variable	SCC	PCC
Nwp_globalirrad(W/m^2)	0.906	0.886
Nwp_directirrad(W/m^2)	0.891	0.880
Nwp_temperature(℃)	0.462	0.451
Nwp_humidity(%)	-0.362	-0.365
Nwp_windspeed(m/s)	0.182	0.184
Lmd_totalirrad(W/m^2)	0.968	0.966
Lmd_diffuseirrad(W/m^2)	0.869	0.773
Lmd_temperature(℃)	0.448	0.448
Lmd_windspeed(m/s)	0.381	0.354

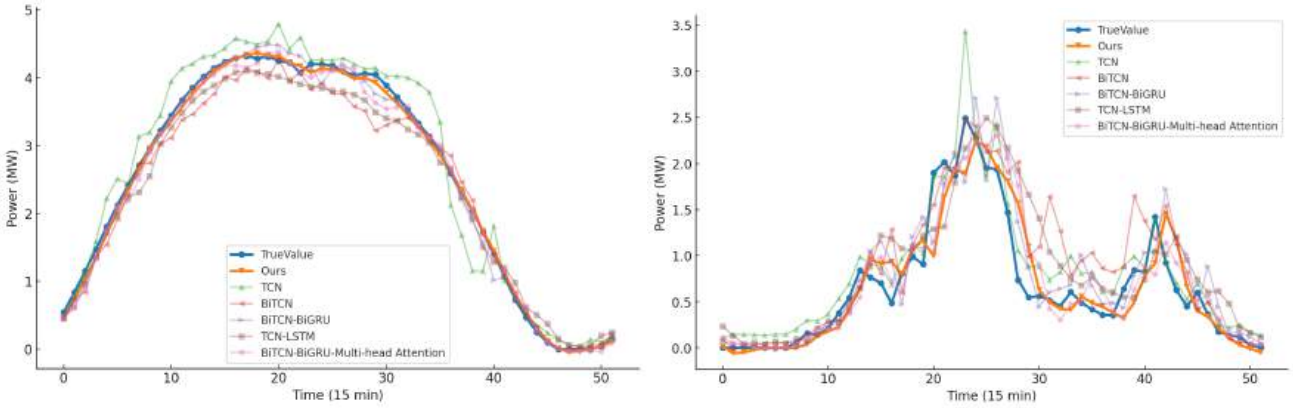
3.3 Performance comparison of other models

This paper conducted a single-step prediction test to verify the performance of the model, which involved predicting a future data point 15 minutes in advance. Table 2 summarizes the evaluation indices of the six models on the prediction outcomes of the single-step. The proposed model is compared and evaluated against TCN, TCN-LSTM, BiTCN, BiTCN-BiGRU, and BiTCN-BiGRU-MA based on MAE, RMSE, and R^2. The model presented in this paper is referred to by Ours in the following. In Table 2, it is evident that in single-step forecasting, the MAE, RMSE, and R^2 of the proposed model are 0.193, 0.325, and 0.946, respectively. From the results of the single TCN and BiTCN, it can be proved that the two-way TCN is better than the one-way TCN, indicating that this method is improved effectively. From the perspective of the hybrid model, the R^2 of TCN-LSTM and BiTCN-BiGRU is also significantly improved. In terms of the evaluation indices, compared to the BiTCN-BiGRU-MA model, the proposed model reduces MAE by 3.2%, decreases RMSE by 5.1%, and R^2 is increased by 1.3%. Therefore, it is demonstrated that the improved method proposed in this paper outperforms the unimproved method.

Table 2: Prediction model evaluation index table of station 0

Model	MAE	RMSE	R^2
TCN	0.314	0.486	0.884
TCN-LSTM	0.329	0.465	0.894
BiTCN	0.270	0.437	0.906
BiTCN-BiGRU	0.243	0.393	0.924
BiTCN-BiGRU-MA	0.225	0.376	0.933
Ours	0.193	0.325	0.946

To further verify the effectiveness of the proposed hybrid photovoltaic power generation prediction model, the aforementioned six models are applied to forecast the data of the region containing the power station in the



(a) Comparison of PV power prediction on clear days. (b) Comparison of PV power prediction in cloudy weather.

Figure 4: Photovoltaic power generation forecast for clear and cloudy weather.

test set, covering the period of 6:00 to 19:00 on both clear and cloudy days. The outcomes of the experiment are shown in Fig.4a and Fig.4b. Specifically, Fig.4a presents the forecasting outcomes for clear days, while Fig.4b illustrates the prediction outcomes for overcast conditions. In the single-step prediction, the model introduced in this study has more advantages in fitting the true value than other models and has a better power prediction trend in the position with large fluctuation of power generation.

4. CONCLUSION

This paper introduces a PV short-term power prediction model based on an improved BiTCN, a multi-focus mechanism, and BiGRU. Experimental verification shows that this model outperforms traditional TCN, TCN-LSTM, BiTCN-BiGRU, and BiTCN-MA-BiGRU models on complex datasets, particularly in MAE, RMSE, and R^2 evaluation indicators. Using different data sets has proven that the model possesses strong robustness and generalization ability. The proposed model offers a new forecasting method for smart grid management and an effective solution for photovoltaic power management in smart grids, demonstrating its potential and value in practical applications. However, delays in forecasting during severe weather changes still affect the usability of forecast results and need to be addressed in future work.

ACKNOWLEDGMENTS

This work is supported by the following funds: Sichuan Provincial Science and Technology Program: Research and Application of Key Technologies for Hail Prevention Command and Equipment in Xinjiang (2024YFHZ0151). Innovation and Development Special Fund of Hunan Meteorological Bureau (CXFZ2024-ZDZX03). Fund Project: The Second Comprehensive Scientific Investigation Project of the Ministry of Science and Technology on Extreme Weather and Climate Events and Disaster Risk on the Tibetan Plateau (2019QZKK0104).

REFERENCES

[1] Li, Z., Rahman, S. M., Vega, R., and Dong, B., "A hierarchical approach using machine learning methods in solar photovoltaic energy production forecasting," *Energies* **9**(1), 55 (2016).

[2] Wang, H., Liu, Y., Zhou, B., Li, C., Cao, G., Voropai, N., and Barakhtenko, E., "Taxonomy research of artificial intelligence for deterministic solar power forecasting," *Energy Conversion and Management* **214**, 112909 (2020).

[3] Bai, S., Kolter, J. Z., and Koltun, V., "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271* (2018).

[4] Xiang, L., Liu, J., Yang, X., Hu, A., and Su, H., "Ultra-short term wind power prediction applying a novel model named satcn-lstm," *Energy Conversion and Management* **252**, 115036 (2022).

[5] Wang, M., Rao, C., Xiao, X., Hu, Z., and Goh, M., "Efficient shrinkage temporal convolutional network model for photovoltaic power prediction," *Energy* **297**, 131295 (2024).
[6] Heo, J., Song, K., Han, S., and Lee, D.-E., "Multi-channel convolutional neural network for integration of meteorological and geographical features in solar power forecasting," *Applied Energy* **295**, 117083 (2021).
[7] Netsanet, S., Zheng, D., Zhang, W., and Teshager, G., "Short-term pv power forecasting using variational mode decomposition integrated with ant colony optimization and neural network," *Energy Reports* **8** (2022).
[8] Limouni, T., Yaagoubi, R., Bouziane, K., Guissi, K., and Baali, E. H., "Accurate one step and multistep forecasting of very short-term pv power using lstm-tcn model," *Renewable Energy* **205**, 1010–1024 (2023).
[9] Pu, X., Hao, X., Jiarui, W., Pei, W., Yang, J., and Zhang, J., "A novel gru-tcn network based interactive behavior learning of multi-energy microgrid under incomplete information," *Energy Reports* **9**, 608–616 (2023).
[10] Zhang, D., Chen, B., Zhu, H., Goh, H. H., Dong, Y., and Wu, T., "Short-term wind power prediction based on two-layer decomposition and bitcn-bilstm-attention model," *Energy* **285**, 128762 (2023).
[11] Samal, K. K. R., Panda, A. K., Babu, K. S., and Das, S. K., "Multi-output tcn autoencoder for long-term pollution forecasting for multiple sites," *Urban Climate* **39**, 100943 (2021).
[12] Zhao, M., Zhong, S., Fu, X., Tang, B., and Pecht, M., "Deep residual shrinkage networks for fault diagnosis," *IEEE Transactions on Industrial Informatics* **16**(7), 4681–4690 (2019).

An iterated greedy algorithm based on NSGA-II for distributed hybrid flow shop scheduling problem

Dandan Liu[a], Xu Liang*[a], Zhiyuan Zou[a]
[a]School of Computing, Beijing Information Science and Technology University, Beijing, China

ABSTRACT

The distributed hybrid flow shop scheduling problems (DHFSP) widely exist in various industrial production processes, and thus have received widespread attention. However, studies on HFSP considering green objective in distributed production environment are quite limited. Therefore, this paper investigated a distributed hybrid flow shop scheduling problem with objectives of minimization the makespan and total energy consumption (TEC). To solve it, an iterated greedy algorithm based on NSGA-II (NSGAIG) is developed. In the proposed algorithm, a random initialization strategy is used to generate the initial solution. Then, starting from the population initialization, a multi-objective local search is carried out for the current optimal solution in the population to obtain the global optimal solution. Next, random variation method is used to increase the exploration space of the algorithm. Finally, the proposed NSGAIG algorithm is compared with other multi-objective optimization algorithms. Experimental results indicate that the proposed NSGAIG outperforms its compared algorithms in solving this problem.

Keywords: Distributed shop, Iterated greedy, Multi-objective optimization, Hybrid flow shop

1. INTRODUCTION

With economic globalization, closer collaboration among enterprises, and the advancement of intelligent manufacturing, a new distributed manufacturing model is gradually replacing the traditional centralized single-factory processing approach [1]. The distributed hybrid flow shop scheduling problem (DHFSP) is commonly seen across various industrial processes, typically involving multiple factories or production sites, with each factory exhibiting the characteristics of the hybrid flow shop scheduling problem (HFSP). Clearly, DHFSP is an extension of the traditional HFSP, which has been proven to be NP-hard [2], making DHFSP NP-hard as well. Facing such a trend, it is of great academic value and practical significance to develop effective optimization algorithms to solve DHFSP problems by using emerging technologies.

In recent years, environmental issues have become a major global focus, with green manufacturing drawing significant attention from both industry and academia. The manufacturing sector is a primary source of energy consumption, responsible for 35% of global carbon dioxide emissions and contributing substantially to environmental pollution [3]. Therefore, manufacturing enterprises have an important responsibility to deal with the problem of environmental degradation. How to balance production efficiency and energy consumption through scientific and rational scheduling optimization has become an important topic in the current research of green manufacturing.

At present, most of the research on DHFSP still focuses on the optimization of production efficiency, ignoring the optimization of energy consumption [4]. The existing research usually adopts the single-objective optimization method, which is difficult to take into account multiple performance indicators at the same time. Although there have been some efforts in recent years to incorporate energy consumption into scheduling optimization, they often do not fully consider the actual distributed production environment, so their applicability and effectiveness in industrial applications need to be further improved.

To address the above shortcomings, this paper investigates a distributed hybrid flow shop scheduling problem aimed at minimizing both maximum makespan and total energy consumption (TEC), and proposes an improved greedy algorithm (NSGAIG) based on NSGA-II. A random initialization strategy is introduced to enhance the initial solution based on the problem's nature, and multi-objective local search is applied to refine the current optimal solutions within the population, Finally, random mutation is incorporated to expand the algorithm's exploration space.

The rest of this article is organized as follows. Section 2 is literature review. Section 3 introduces the specific problems and the mathematical model of DHFSP. An iterated greedy algorithm based on NSGA-II for DHFSP is presented in Section 4. Section 5 shows the experimental results. Finally, Section 6 concludes this article and the prospect of future work.

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 1351906 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057725

2. LITERATURE REVIEW

In existing research on solving the DHFSP, optimization objectives are typically divided into single-objective and multi-objective categories. For single objective optimization problems, Wang et al. [5]introduced a bi-population cooperative memetic algorithm to minimize the makespan. Sun et al. [6] addressed a distributed hybrid blocking flow shop scheduling problem with makespan criterion and developed a hybrid genetic algorithm. Zhang et al. [7] examined a practical distributed hybrid differentiation flow shop scheduling problem and presented a general EA framework called distributed coevolutionary memetic algorithm to minimize the makespan. Qin et al. [8] presented a collaborative iterative greedy algorithm to deal with the blocked DHFSP with the objective of minimal makespan.

For multi-objective optimization problems, Li et al. [9] formulated a mathematical model for DHFSP with machine velocity and resource constraints, and developed a collaboration-based multi-objective algorithm. Lei et al. [10] studied DHFSP with sequence-dependent setup times and proposed a multi-class teaching–learning-based optimization to minimize the makespan and maximum tardiness simultaneously. Zhang et al. [11] focused on energy-efficient heterogeneous DHFSP and designed a multi-objective memetic algorithm with particle swarm optimization and Q-learning-based local search. Shao et al. [12] considered the a DHFSP under nonidentical time-of-use electricity tariffs and proposed an ant colony optimization behavior-based multi-objective evolutionary algorithm based on decomposition.

While numerous studies have addressed solutions for the multi-objective DHFSP, significant potential for improvement remains. Thus, developing a more efficient algorithm for green, and energy-efficient DHFSP is of great importance.

3. PROBLEM DESCRIPTION AND MODEL FORMULATION

DHFSP consists of three intercoupled sub-problems, that is factory allocation, machine selection, and job sequencing. Based on hybrid flow shop scheduling, it introduces multi-shop distributed scheduling, which can be defined as the following. There are F identical factories, each with the same number of processes, and N jobs will be allocated to F factories. In a factory, each stage s ($s = 1,2,\dots,S$) contains at least one machine, and at least one stage has two parallel machines, that is, $M_{f,s} \geq 2$, where $M_{f,s}$ represents the number of machines in stage s of factory f ($f = 1,2,\dots,F$). For each job j ($j = 1,2,\dots,N$) needs to complete a series of processes on one or more machines in s stages. The speed of each machine can vary. For a job j in the stage s, the standard processing time is denoted as $P_{s,j}$. If the speed of machine m ($m = 1,2,\dots,M_{f,s}$) at sth stage is $V_{s,m}$, then the actual processing time of the job j on machine m will be $P_{s,j}/V_{s,m}$. Assume that every machine is working and there is no malfunction. The indices, constant symbol and variable symbol of the DHFSP are summarized in Table 1.

$$\min H = [C_{max}, TEC] \quad (1) \qquad \sum_{f=1}^{F} X_{j,f} = 1, j \in \boldsymbol{J} \quad (2)$$

$$X_{j,f} = \sum_{m=1}^{M_{f,s}} \sum_{p=1}^{N} Y_{j,p,m,s,f}, s \in \boldsymbol{I}, j \in \boldsymbol{J}, f \in \boldsymbol{K} \quad (3) \qquad Z_{j,m,s,f} = \sum_{p=1}^{N} Y_{j,p,m,s,f}, s \in \boldsymbol{I}, j \in \boldsymbol{J}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (4)$$

$$\sum_{j=1}^{N} Y_{j,p,m,s,f} \leq 1, s \in \boldsymbol{I}, p \in \boldsymbol{P}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (5)$$

$$\sum_{j=1}^{N} Y_{j,p,m,s,f} \geq \sum_{j=1}^{N} Y_{j,p+1,m,s,f}, s \in \boldsymbol{I}, p \in \{1,2,\dots,N-1\}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (6)$$

$$MS_{f,m,p+1} \geq MC_{f,m,p}, s \in \boldsymbol{I}, p \in \{1,2,\dots,N-1\}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (7)$$

$$MC_{f,m,p} = MS_{f,m,p} + \sum_{j=1}^{N} \left(Y_{j,p+1,m,s,f} * \frac{P_{s,j}}{V_{s,m}}\right), \ s \in \boldsymbol{I}, p \in \{1,2,\dots,N-1\}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (8)$$

$$MS_{f,m,p} \leq S_{s,j} + R\left(1 - Y_{j,p,m,s,f}\right), \ s \in \boldsymbol{I}, j \in \boldsymbol{J}, p \in \boldsymbol{P}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (9)$$

$$MS_{f,m,p} \geq S_{s,j} - R\left(1 - Y_{j,p,m,s,f}\right), \ s \in \boldsymbol{I}, j \in \boldsymbol{J}, p \in \boldsymbol{P}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (10)$$

$$C_{s,j} = S_{s,j} + \sum_{j=1}^{M_{f,s}} \left(Z_{j,m,s,f} * \frac{P_{s,j}}{V_{s,m}}\right), s \in \boldsymbol{I}, j \in \boldsymbol{J}, f \in \boldsymbol{K} \quad (11) \qquad S_{s+1,j} \geq C_{s,j}, s \in \{1,2,\dots,S-1\}, j \in \boldsymbol{J} \quad (12)$$

$$S_{s,j} \geq 0, s \in \boldsymbol{I}, j \in \boldsymbol{J} \quad (13) \qquad MS_{f,m,p} \geq 0, s \in \boldsymbol{I}, p \in \boldsymbol{P}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (14)$$

$$C_{max} \geq C_{s,j}, s \in \boldsymbol{I}, j \in \boldsymbol{J} \quad (15) \qquad EC_w = \sum_{s=1}^{S} \sum_{m=1}^{M_{f,s}} \sum_{j=1}^{N} \left(\left(C_{s,j} - S_{s,j}\right) * PW_{s,m}\right) \quad (16)$$

$$EC_i = \sum_{s=1}^{S} \sum_{f=1}^{F} \sum_{m=1}^{M_f} \sum_{p=1}^{N-1} \left(\left(MS_{f,m,p+1} - MC_{f,m,p}\right) * PI_{s,m}\right) + \sum_{s=1}^{S} \sum_{f=1}^{F} \sum_{m=1}^{M_f} \left(MS_{f,m,1} * PI_{s,m}\right) \quad (17)$$

$$TEC = EC_w + EC_i \quad (18) \qquad X_{j,f} \in \{0,1\}, j \in \boldsymbol{J}, f \in \boldsymbol{K} \quad (19)$$

$$Y_{j,p,m,s,f} \in \{0,1\}, s \in \boldsymbol{I}, j \in \boldsymbol{J}, p \in \boldsymbol{P}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (20) \qquad Z_{j,m,s,f} \in \{0,1\}, s \in \boldsymbol{I}, j \in \boldsymbol{J}, f \in \boldsymbol{K}, m \in H_{f,s} \quad (21)$$

Table 1. Description of indices, constant symbol and variable symbol.

Symbol	Description	Symbol	Description
s	index of stages, s∈ {1,2, … , S}.	$MS_{f,m,p}$	start time of the mth machine in the pth position in factory f.
j	index of jobs, j∈ {1,2, … , N}.	$MC_{f,m,p}$	completion time of the mth machine in the pth position in factory f.
f	index of factories, f∈ {1,2, … , F}.	EC_w	energy consumption during processing period.
m	index of machines, m∈ {1,2, … , $M_{f,s}$}.	EC_i	energy consumption during idle period.
p	index of positions, p∈ {1,2, … , N}.	$PW_{s,m}$	processing energy consumption per unit time (processing power) of job on machine m in stage s.
$\boldsymbol{I}$	set of stages, $\boldsymbol{I}$ ∈ {1,2, … , S}.	$PI_{s,m}$	idle energy consumption per unit time (idle power) of job on machine m in stage s.
$\boldsymbol{J}$	index of jobs, $\boldsymbol{J}$ ∈ {1,2, … , N}.	$V_{s,m}$	processing speed on the mth machine in stage s.
$\boldsymbol{K}$	set of factories, $\boldsymbol{K}$ ∈ {1,2, … , F}.	$s_{s,j}$	start time of job j in stage s.
$H_{f,s}$	index of machines, $H_{f,s}$ ∈ {1,2, … , $M_{f,s}$}.	$C_{s,j}$	completion time of job j in stage s.
$\boldsymbol{P}$	set of job positions, $\boldsymbol{P}$ ∈ {1,2, … , N}.	C_{max}	maximum completion time of all factories, namely makespan.
N	number of jobs.	TEC	total energy consumption.
S	number of stages.	$X_{j,f}$	a binary number, set 1 if job j is assigned to factory f; Otherwise, set 0.
F	number of factories.	$Z_{j,m,s,f}$	a binary number, the value is 1 if the job j is processed on the mth machine in stage s of factory f; otherwise, it is 0.
$M_{f,s}$	number of machines in stage s of factory f.	$Y_{j,p,m,s,f}$	a binary number, set 1 if the job j is on the mth machine in the pth position in stage s of factory f; otherwise, set 0.
M_f	total number of machines in each factory.	R	an infinitely positive number.
$P_{s,j}$	the standard processing time of job j in stage s.		

The objective function (1) aims to minimize both makespan and total energy consumption (TEC) simultaneously. Constraint (2) ensures that each job is assigned to exactly one factory. Constraint (3) specifies that each job can only be processed on one machine at each stage within a factory. Constraint (4) ensures that each job assigned to a factory is accurately placed on one machine at each stage. Constraint (5) ensures that each machine can process only one job at a time. Constraint (6) states that a job can only be processed on a machine after the immediately preceding position on the machine is occupied. Eq. (7) represents that the start time of the next machine position cannot be earlier than the completion time of the previous one. Eq. (8) defines the completion time constraint for each machine position. Constraints (9) and (10) establish the relationship between the machine position and the job sequence, linking the start time of the machine position to the job's start time. Constraint (11) ensures that the makespan of an operation is the sum of the start time and actual processing time for that operation. Constraint (12) guarantees that the current operation can only begin after the completion of the previous operation. Constraint (13) ensures that the start time of each job is non-negative. Constraint (14) guarantees that the start time of each machine is non-negative. Eq. (15) defines the maximum makespan. Eq. (16) defines the processing energy consumption for each job. Formula (17) defines the idle energy consumption for each machine. Formula (18) defines the TEC objective. Constraints (19)-(21) define the valid range for the decision variables.

4. PROPOSED NSGAIG ALGORITHM

This section introduces the proposed NSGAIG algorithm for solving the DHFSP. First, the overall framework of the NSGAIG algorithm is outlined. Then, the individual components of the algorithm are explained, including the encoding and decoding process, initialization strategy, selection and crossover methods, mutation operation, destruction and reconstruction phase, and the local search procedure.

4.1 Framework of the Proposed NSGAIG

In this section, the NSGAIG method algorithm to handle DHFSP will be introduced in detail. The algorithm is mainly composed of genetic operators and iterated greedy heuristics. Firstly, based on the Nawaz-Enscore-Ham (NEH) heuristic algorithm, a random initialization strategy is proposed to generate high quality initial solutions. The NSGAIG algorithm begins with the initialization of the population. Then, a local search is conducted on the current optimal solution within the population to identify the global best solution Next, individuals are randomly selected to stay in the offspring according to the selection probability. And crossover and mutation are also used to produce new individuals. Destruction and construction operation and local search respectively calculate the optimal of individuals in the new population and obtain

the local optimal. Finally, if the local best solution is better than the global best solution, the global best is updated with the local best. Otherwise, the current global best solution is randomly used to replace an individual in the offspring population.

4.2 Encoding and decoding

A solution is represented by $\pi = \{\pi^1, \pi^2, \ldots, \pi^f, \ldots, \pi^F\}$, where each list π^f contains jobs allocated to factory f, with the job sequence in π^f indicating the processing sequence in the first stage. The decoding mechanism is as follows. In the first stage, jobs in each π^f are assigned to the first available machine using the first available machine (FAM) rule. In the next stage, jobs are reordered based on the completion time of the previous phase. Additionally, if multiple jobs have the same start time in the current phase, they are arranged according to their order from the previous phase.

4.3 Initialization Strategy

A good initial solution should cover more promising solution space. Therefore, the quality of the initial solution has a great impact on subsequent iterative searches. As a simple and powerful initialization strategy, NEH heuristic algorithm is often integrated into intelligent optimization algorithms. According to the features of DHFSP, a random initialization strategy is proposed in this paper, which includes the following three different initialization strategies. (1) MakespanInit: An initial solution is generated using the NEH2 rule to minimize the maximum makespan. Job j is inserted into all possible locations among all factories, and the location that results in the smallest makespan is selected. The job is inserted into that location, and the corresponding factory f is updated to record the insertion. (2) TECInit: For each job j, it is placed in all available positions across factories, and the one with the lowest TEC is chosen. The job is inserted into the selected location, and the factory f is updated accordingly. (3) MakespanTECInit: For each job j, it is tested in all possible positions in all factories. The location that minimizes both TEC and makespan is selected. The remaining solutions are randomly generated to maintain diversity in the population.

4.4 Selection

In the Selection strategy of this study, Binary Tournament Selection method is adopted. By comparing the fitness of two individuals randomly selected in the current population, the individual with better fitness is selected to enter the next generation population. Binary tournament selection has high flexibility and adaptability, and its randomness can effectively avoid premature convergence to the local optimal solution while maintaining population diversity. The selection process is carried out in the following steps. 1) Random selection: two individuals are randomly chosen from the population. 2) Fitness comparison: the fitness values of the two individuals are compared. 3) Winner selection: the individual with the better fitness (higher fitness for maximization problems, lower for minimization problems) is selected as the winner. 4) Repeat: steps 1-3 are repeated until the required number of individuals for the next generation is selected. This process ensures that better-performing individuals are more likely to be selected, while still allowing diversity by giving lower-performing individuals a chance to compete.

4.5 Crossover

This study adopts Multi-Point Crossover as a core genetic operator to promote diversity and preserve advantageous gene segments from both parent solutions. The process for Multi-Point Crossover involves the following steps. 1) Select crossover points: Randomly select multiple crossover points along the parent chromosomes. 2) Divide parent chromosomes: split each parent's chromosome at the chosen crossover points to create segments. This segmentation allows for the exchange of larger portions of genetic information, increasing the chance of retaining beneficial gene combinations. 3) Exchange segments: alternate segments between the two parent chromosomes to form the offspring. 4) Form offspring: combine the selected segments to create a new chromosome for each offspring. These offspring inherit mixed characteristics from both parents, preserving strong genetic traits while introducing diversity.

4.6 Mutation

The Mutation strategy adopts the random mutation method to enhance the exploration ability of the solution space. Specifically, two different mutation operators are selected for random mutation: swap mutation and insertion mutation. (1) Swap mutation operates by randomly choosing two positions within the solution and swapping their values, thereby altering the sequence of the solution. (2) Insertion mutation randomly selects an element and inserts it into another position within the solution, thus changing the local structure. In practice, the mutation operator is applied to the current solution in a random manner. This random mutation approach aids in balancing global and local search abilities, thereby improving the algorithm's robustness and search efficiency.

4.7 Destruction and reconstruction

The algorithm presented in this paper includes a destruction and reconstruction (DR) operation, which reorganizes and optimizes the optimal solution within the population. The DR operation consists of two phases: destruction and reconstruction, with parameter d controlling the destruction size. The steps of this operator are as follows. 1) Randomly select d jobs from the current job sequence in a factory. 2) Insert the first extracted job into all positions in the remaining job sequence. 3) Choose the sequence with the minimum makespan as the current remaining sequence. 4) Repeat steps 2-3 until all d jobs are inserted.

4.8 Local search

The local search is performed on the candidate solution found by the DR operator, and a new solution is obtained. For the factory with the maximum makespan and the factory with the maximum TEC, that is, the two key factories are recorded as $F1$ and $F2$ respectively. For $F1$ and $F2$, two different search operators are used to optimize the existing solutions. They are the exchange operator and the insertion operator. Randomly choose a job from the key factories $F1$ or $F2$ and reinsert it into all available positions across all factories to generate a new adjacent solution. Alternatively, randomly select a job from the key factories $F1$ or $F2$ and swap it with another job to create a new adjacent solution.

5. PROPOSED NSGAIG ALGORITHM

5.1 Experimental Setting

To fully account for DHFSP, we consider a different number of combinations between jobs, stages, and factories, where $N \in \{50,100,150,200\}$, $S \in \{2,4,6,8,10\}$, and $F \in \{2,3,4,5,6\}$. Therefore, $4 \times 4 \times 5 = 100$ instances are generated, and each algorithm is run independently on the instance 10 times. The processing speed for each machine is denoted as $V_{s,m} \in \{0.5,1.0,1,5\}$. The processing power $PW_{s,m}$ is calculated as $4 \times V_{s,m} \times V_{s,m}$. The idle power $PI_{s,m}$ is set to $0.25 \times PW_{s,m}$. In this paper, three performance metrics are used to evaluate the effectiveness of multi-objective optimization algorithms in solving these instances: inverted generation distance (IGD), hypervolume (HV), and Spread.

5.2 Parameter Calibration

The performance of the algorithm is influenced by parameter settings, including population size ps, crossover probability pc, mutation probability pm, and destruction scale d. To identify the optimal combination of these parameters, the Taguchi design of experiments method is employed. The levels for each parameter are as follows: $ps = \{20,40,60,80\}, pc = \{0.6,0.7,0.8,0.9\}, pm = \{0.2,0.4,0.6,0.8\}$ and $d = \{2,3,4,5\}$. An orthogonal array $L_{16}(4^4)$ is used, consisting of 16 different combinations consisting of these parameters. The values for the four key parameters are presented in Table 2. The proposed NSGAIG algorithm is executed 10 times for each combination on each instance. The performance metric IGD is employed to assess the main effects of these parameters, with smaller IGD values being preferable. The main effect plot for the four parameters is shown in Fig. 1. Based on the observation, the optimal combination of parameter settings is: $ps = 60, pc = 0.7, pm = 0.6, d = 3$.

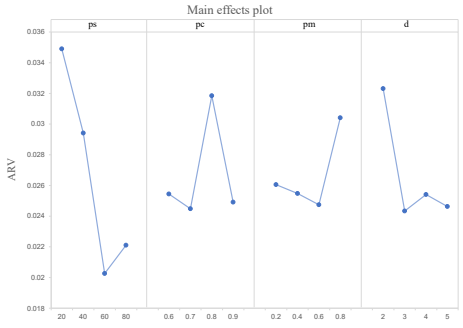


Figure 1. Main effects plot of NSGAIG.

5.3 Comparison of NSGAIG and other algorithms

To evaluate the performance of the proposed NSGAIG, comparison experiments are conducted in this section. NSGAIG is compared with well-established multi-objective metaheuristics, including MOHIG [13], NSGA-II [14] and SPEA2 [15]. In order to ensure fairness, all algorithms use the same termination criteria. Additionally, they adopt the same coding and

decoding mechanisms, as well as the crossover and mutation operations (if applicable). Based on calibration experiments, the parameter settings for the competing algorithms are as follows. For NSGAII, pc and pm are set to 0.9 and 0.2, respectively. For SPEA2, pc and pm are set to 0.9 and 0.3. For MOHIG, pc and pm are equal to 0.8 and 0.4, with d set to 4. The archive size for SPEA2 is 60. The ps is 80 for all algorithms. Each algorithm is executed 10 times on each instance for performance evaluation.

Table 2. Parameter values at each factor level.

Parameter	Factor level			
	1	2	3	4
ps	20	40	60	80
pc	0.6	0.7	0.8	0.9
pm	0.2	0.4	0.6	0.8
d	2	3	4	5

Tables 3, 4, and 5 present the average values of three performance indicators. Table 3 shows the comparison of IGD values for the given instances, where a lower IGD indicates better performance. The findings from Table 3 are as follows: 1) The NSGAIG algorithm outperforms the other algorithms by obtaining 10 better values, significantly surpassing the other methods. 2) NSGA-II achieved four better solutions, while MOHIG and SPEA2 failed to obtain the optimal solutions. 3) The average values in the final row further support the superior efficiency of NSGAIG. Table 4, presenting the HV values, shows that NSGAIG achieved 14 better results across the 14 instances. Finally, Table 5 reveals that although the first two algorithms obtained the same number of optimal values, the average value in the last row further demonstrates the advantage of NSGAIG.

Table 3. Statistics values of IGD metric under different algorithms.

Parameter / Algorithm	NSGAIG	MOHIG	NSGA-II	SPEA2
F=2	0.028385179	0.033661596	**0.02834724**	0.167169606
F=3	**0.022381887**	0.030222052	0.022915631	0.157480089
F=4	0.041307233	0.03005707	**0.024400707**	0.134425441
F=5	**0.017752806**	0.022594292	0.028171432	0.149439358
F=6	**0.012465414**	0.020857419	0.021516755	0.184425785
S=2	**0.015903519**	0.019972323	0.027580836	0.139084269
S=4	**0.020482655**	0.026108917	0.024529626	0.161634751
S=6	**0.020546914**	0.029032837	0.021442807	0.158127673
S=8	**0.020766548**	0.030735152	0.025169943	0.169912759
S=10	0.044592884	0.0315432	**0.024920182**	0.164180828
N=50	0.040938534	0.030648949	**0.030624935**	0.14816343
N=100	**0.02033131**	0.029247955	0.030010617	0.179208481
N=150	**0.01943928**	0.024350491	0.019597727	0.150407779
N=200	**0.017124892**	0.025666548	0.018681436	0.156572534
Mean	**0.024458504**	0.027478486	0.024850705	0.158588056

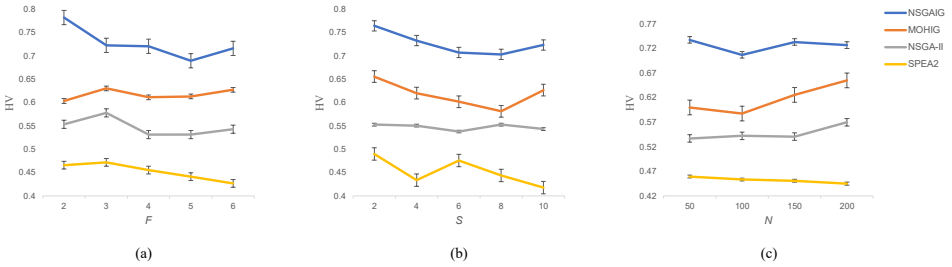


Figure 2. Means plot with 95% confidence level HSD interval for the interaction between algorithms.

In this section, we focus on the interactive effects of parameters on the HV metric. Figure 2 presents the 95% confidence level HSD interval mean plot, illustrating the interaction between algorithms and parameters such as the number of factories, jobs, and stages under the HV metric. It is evident that the average value achieved by NSGAIG consistently outperforms those of the other algorithms.

Table 4. Statistics values of HV metric under different algorithms.

Parameter / Algorithm	NSGAIG	MOHIG	NSGA-II	SPEA2
F=2	**0.782187705**	0.603462027	0.553549233	0.465638435
F=3	**0.722707556**	0.630238937	0.578085542	0.471776113
F=4	**0.720604015**	0.611244521	0.53139971	0.455424588
F=5	**0.689633511**	0.613141658	0.531189294	0.441129069
F=6	**0.716040303**	0.62733	0.542889838	0.426732151
S=2	**0.764533818**	0.655675727	0.552842456	0.489778549
S=4	**0.732736339**	0.62012776	0.550475758	0.433699044
S=6	**0.707194743**	0.601824098	0.537895331	0.47564017
S=8	**0.703301811**	0.58118096	0.552702537	0.443838996
S=10	**0.723406377**	0.626608599	0.543197535	0.417743596
N=50	**0.737603532**	0.60009878	0.536895233	0.459426263
N=100	**0.707061244**	0.587732631	0.542246596	0.453325054
N=150	**0.733262139**	0.625504663	0.540674935	0.450836336
N=200	**0.727011555**	0.65499764	0.56987413	0.444972631
Mean	**0.726234618**	0.617083429	0.547422723	0.452140071

Table 5. Statistics values of Spread metric under different algorithms.

Parameter / Algorithm	NSGAIG	MOHIG	NSGA-II	SPEA2
F=2	**1.310578158**	1.339179947	1.406213212	1.757558323
F=3	1.366544738	**1.330541221**	1.479888234	1.687776143
F=4	1.337597572	**1.336713801**	1.411708544	1.688380756
F=5	1.375334697	**1.354534822**	1.421054456	1.729555232
F=6	**1.356243301**	1.382247332	1.446724454	1.744986773
S=2	**1.330941834**	1.355647723	1.403857523	1.726297146
S=4	**1.353726362**	1.384049788	1.416721956	1.712928493
S=6	1.357905273	**1.320770077**	1.445167592	1.751852194
S=8	1.332526038	**1.332425661**	1.472823945	1.712506231
S=10	1.35119896	**1.340323873**	1.427017884	1.704673164
N=50	**1.347623603**	1.357728959	1.367386142	1.717150656
N=100	**1.346192149**	1.356932472	1.40010389	1.692604154
N=150	1.350266817	**1.332164716**	1.461168007	1.743878814
N=200	**1.338956205**	1.341747551	1.503813081	1.732972158
Mean	**1.346831122**	1.347500567	1.43311778	1.721651446

6. CONCLUSION AND FUTURE EXPECTATION

In this paper, the distributed hybrid flow shop scheduling problem is studied. First, a mathematical model of the problem is established to solve the maximum makespan and total energy consumption. A novel iterated greedy algorithm based on NSGA-II is developed to effectively address this problem. The algorithm begins with a random initialization strategy to generate a diverse set of initial solutions, ensuring broad coverage of the solution space. A multi-objective local search mechanism is then applied to fine-tune the solutions, pushing them toward global optimality by exploiting the local regions

of the search space. Additionally, a random local search method was applied to expand the algorithm's exploration capabilities, enabling it to avoid premature convergence. Finally, experiments are conducted to compare the performance of various multi-objective optimization algorithms across 100 instances. The comparison, based on IGD, HV, and Spread metrics, demonstrates that the NSGAIG algorithm outperforms the other algorithms in solving this problem.

Future research could further enhance this work by exploring more advanced local search strategies or hybridizing the algorithm with deep learning techniques to improve solution quality and computational efficiency. Additionally, extending the DHFSP model to include more complex constraints, such as machine reliability and carbon emissions, would broaden its applicability in sustainable manufacturing.

ACKNOWLEDGEMENTS

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] Toptal A, Sabuncuoglu I. Distributed scheduling: a review of concepts and applications[J]. International Journal of Production Research, 2010, 48(18): 5235-5262.
[2] Lu C, Zheng J, Yin L, et al. An improved iterated greedy algorithm for the distributed hybrid flowshop scheduling problem[J]. Engineering Optimization, 2024, 56(5): 792-810.
[3] Xin X, Jiang Q, Li S, et al. Energy-efficient scheduling for a permutation flow shop with variable transportation time using an improved discrete whale swarm optimization[J]. Journal of Cleaner Production, 2021, 293: 126121.
[4] Shao Z, Shao W, Chen J, et al. A feedback learning-based selection hyper-heuristic for distributed heterogeneous hybrid blocking flow-shop scheduling problem with flexible assembly and setup time[J]. Engineering Applications of Artificial Intelligence, 2024, 131: 107818.
[5] Wang J, Wang L. A bi-population cooperative memetic algorithm for distributed hybrid flow-shop scheduling[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020, 5(6): 947-961.
[6] Sun X, Shen W, Vogel-Heuser B. A hybrid genetic algorithm for distributed hybrid blocking flowshop scheduling problem[J]. Journal of Manufacturing Systems, 2023, 71: 390-405.
[7] Zhang G, Liu B, Wang L, et al. Distributed co-evolutionary memetic algorithm for distributed hybrid differentiation flowshop scheduling problem[J]. IEEE Transactions on Evolutionary Computation, 2022, 26(5): 1043-1057.
[8] Qin H X, Han Y Y, Liu Y P, et al. A collaborative iterative greedy algorithm for the scheduling of distributed heterogeneous hybrid flow shop with blocking constraints[J]. Expert Systems with Applications, 2022, 201: 117256.
[9] Li R, Li J, Li J, et al. A collaboration-based multi-objective algorithm for distributed hybrid flowshop scheduling with resource constraints[J]. Swarm and Evolutionary Computation, 2023, 83: 101409.
[10] Lei D, Su B. A multi-class teaching–learning-based optimization for multi-objective distributed hybrid flow shop scheduling[J]. Knowledge-Based Systems, 2023, 263: 110252.
[11] Zhang W, Li C, Gen M, et al. A multiobjective memetic algorithm with particle swarm optimization and Q-learning-based local search for energy-efficient distributed heterogeneous hybrid flow-shop scheduling problem[J]. Expert Systems with Applications, 2024, 237: 121570.
[12] Shao W, Shao Z, Pi D. An ant colony optimization behavior-based MOEA/D for distributed heterogeneous hybrid flow shop scheduling problem under nonidentical time-of-use electricity tariffs[J]. IEEE Transactions on Automation Science and Engineering, 2021, 19(4): 3379-3394.
[13] Lu C, Liu Q, Zhang B, et al. A Pareto-based hybrid iterated greedy algorithm for energy-efficient scheduling of distributed hybrid flowshop[J]. Expert Systems with Applications, 2022, 204: 117555.
[14] Ma H, Zhang Y, Sun S, et al. A comprehensive survey on NSGA-II for multi-objective optimization and applications[J]. Artificial Intelligence Review, 2023, 56(12): 15217-15270.
[15] Maurya V K, Nanda S J. Time-varying multi-objective smart home appliances scheduling using fuzzy adaptive dynamic SPEA2 algorithm[J]. Engineering Applications of Artificial Intelligence, 2023, 121: 105944.

Acquisition of adaptive knowledge in case-based reasoning for the online set-point control of industrial process

Kwang Rim Song[1*a], Song Ho Kim[a], Chol Guk Han[a], Un Sim Ri[a]
[a]Faculty of Automatics, Kim Chaek University of Technology, Pyongyang, Democratic People's Republic of Korea

ABSTRACT

In CBR for industrial process control, it is an important issue for improving the efficiency of reasoning to adapt the case base using the online knowledge acquired during operation. In this paper, a new method for online updating and addition of case-base is proposed and applied to set-point control of the rolled cake production process. First, we present an approach for quantitatively evaluating the rolled cake based on sensory quality. Second, we propose an interactive method to acquire knowledge from the operator's experience and adapt the case base according to the behavior of the process and the quantitatively evaluated quality level. The comparative experiments for the validation show that the proposed method results in the 4.1% improvement in the ratio of good products.

Keywords: case-based reasoning, rolled cake, set point control, industrial process

1. INTRODUCTION

The scientific operation of the industrial process with the quantitative evaluation of sensory characteristics is a major concern to improving the quality in the food industry. In order to achieve the target quality in the industrial process with the controllable parameters, set point would be adjusted to control the process under some disturbance [1]. In particular, for food production process where the quality would be sometimes represented as the sensory characteristics, it is important to evaluate the quality of the product quantitatively. In general, as human thinking and reasoning have ambiguities, Reference [2] proposes an approach of evaluating the cake with visual quality characteristics using the expert system based on fuzzy logic. In [3], a fuzzy set and neural network technique is applied to quantitatively evaluate the visual quality level of the biscuit products and to determine the appropriate set point.

There are several approaches to set-point control of industrial process, including intelligent supervisory control [4], and case-based reasoning [5]. In the hybrid intelligent control method [6] for optimal operation of roasting furnace and the hierarchical intelligent supervisory control method [7] based on reinforcement learning, the supervisory controller is composed to control the set-point so that it can replace the operator. For the same purpose, approaches based on the operator's experience have been widely used. In [8], a methodology to deal with the expert's knowledge through steps such as collection of sensory indices, measurement, and acquisition of control rules is proposed and applied to the food production process. A methodology for determining the optimal operation policy with rule base derived from human experience is introduced in [9] where iterative nature of the batch production process is considered.

In addition to rule based reasoning, case-based reasoning that utilizes the operation expertise and case to infer similar solutions corresponding to similar conditions has been widely applied to the optimal operation of industrial process. In [10], a case-based reasoning is used to obtain the optimal set-point of ore feed rate and water injection rate in the milling process, and Reference [11] realizes the decision making and control of coal combustion process using CBR and rule-based reasoning. One of the major challenges facing to application of case-based reasoning to the set-point control of industrial process is to constantly adapt the case-base to the changing characteristics of the process, thus increasing the accuracy of the reasoning. Reference [12] proposed a methodology for updating and reusing the case base by acquiring online adaptation knowledge during the process design phase, but the method for updating the case base with operational experience and new knowledge obtained during the online operation of the industrial process was not explicitly proposed.

The aim of this paper is to implement the set-point control of the industrial process with sensory quality characteristics, utilizing an improved case-based reasoning method that allows online updating and addition of the case-base. First, we

* skl8861@star-co.net.kp; phone +85023811811; fax +85023814410

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE Vol. 13519, 1351907 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057800

propose an approach of quantitatively evaluating the sensory quality of product, using the knowledge obtained from experts. Next, a new methodology for adapting the case base by acquiring new knowledge during the operation is proposed. Finally, the comprehensive CBR algorithm for the optimal operation of the rolled cake process is presented.

The rest of the paper is organized as follows: the next section presents the quality evaluation method. Section 3 shows the application of case-based reasoning. Section 4 shows the effectiveness of the proposed method through comparative experiments on the cake production process. Finally, we conclude.

2. QUALITY EVALUATION

2.1 Sensory characteristics of cake

Cake is a kind of food that has been kneaded in wheat flour, corn flour or rice flour with various kinds of sugars and oils, cow's milk and eggs, edible spices, etc. and baked in a certain shape. The process of cake production consists of raw material preparation, kneading, rolling, shaping, baking and packing. In the preparation of raw materials, it should be prepared and quantified the powdered raw materials and liquid according to the mixing ratio. The kneading process has a great effect on the quality of the product, where the main raw materials are mixed with an emulsion composed of sugar, corn syrup and oil. The main characteristics of cake dough are moisture content of dough, the kneading temperature and time. The evaluation indexes of the rolled cake include the quantitative ones such as moisture content, sugar content, oil content, ash, alkalinity, and salt content etc., and the sensory ones such as appearance, taste and odor. The quantitative characteristics of food product are usually analyzed off-line in the laboratory and these values are used in the quality design stage including the initial determination of raw material mix ratio and process parameters.

2.2 Quality evaluation by fuzzy inference

A fuzzy inference system for quality evaluation of rolled cake is one that quantitatively evaluates the quality status of a product from sensory evaluation indicators for the product.

2.2.1 Fuzzy rule base

To construct the fuzzy rule base, three experts evaluated 500 test samples respectively. Three sensory evaluation characteristics - appearance, taste, and color, were selected as the antecedent variables of the fuzzy rule, each of which is divided into five fuzzy sets - "Very Poor (VP), Poor (P), Medium (M), Good (G), and Excellent (E)" as shown in Figure 1.

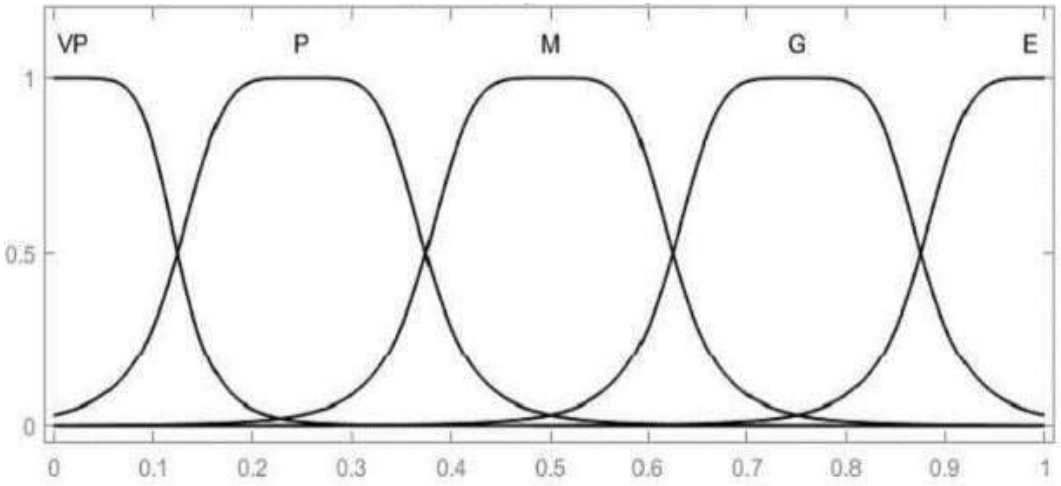


Figure 1: Fuzzy partitioning of input variables

The output variable of the fuzzy rule is also divided into five set to represent the evaluation results of the experts. The result of evaluating the test samples by the first expert are classified into clusters, each of which has the identical value in terms of antecedent variables and consequent variable. The first row in Table 1 represents that the number of rules, of which antecedent variables have the value of "Medium", "Good" and "Poor" respectively and consequent variable has the value of "Poor (P)", is three.

Table 1: The classification of the evaluation result (expert 1)

Cluster Number	Antecedent			Consequent				
	Appear	*Taste*	*Color*	VP	P	M	G	E
1	Medium	Good	Poor		3	16	1	
2	Poor	Good	Medium		4	10	2	
…	…	…	…					
26	Medium	Good	Good			5	14	

We generate the rule base from clusters given in Table 2, where each cluster produces rules that have the identical antecedent value and the different consequent value. Thereafter, the weight of each rule is calculated as the proportion of the corresponding output characteristic value in the total number of test samples for each cluster. For instance, cluster 1 is separated into three rules, whose consequent variables have the value of "P", "M", and "G" respectively. And the weights of each rule are equal to 3/(3 + 16 + 1), 16/(3 + 16 + 1), and 1/(3 + 16 + 1), respectively.

Table 2: Rule base for quality evaluation (Expert 1)

Rule Number	Antecedent			Consequent	Weight
	Appear	*Taste*	*Color*		
1	Medium	Good	Poor	M	0.8
…	…	…	…		
79	Medium	Good	Good	G	0.73

In the same way, we obtain the clustered evaluation results obtained from the expert 2 and 3 to generate the rule base and calculate its weights. The number of generated rules is 82 and 78, respectively.

2.2.2 Fuzzy inference system

The membership function of the output variable is designed to represent the rank between 1 and 5 according to the fuzzy sets. The inference algorithm is as follows:

Step 1: Calculate the fitness E_i of the rule from membership degree for input variables by (1).

$$E_i = \mu_{i1} \times \mu_{i2} \times \mu_{i3} \tag{1}$$

where μ_{ij} is the membership degree of the jth input variable in the ith rule.

Step 2: Perform defuzzification using (2).

$$\mathrm{Q} = \frac{\sum_i E_i \times (w_i \times y_i)}{\sum_i E_i} \tag{2}$$

where Q is the quality score and w_i is the weight of the ith rule. y_i is the central value of the membership function corresponding to the characteristic value of the consequent of the ith rule.

Step 3: Using (3), average the output of each rule base to calculate the final quality score.

$$\overline{\mathrm{Q}} = (\mathrm{Q}_1 + \mathrm{Q}_2 + \mathrm{Q}_3)/3 \tag{3}$$

where Q_i is the output of the rule base for the ith expert, and $\overline{\mathrm{Q}}$ is the final quality level.

3. CASE BASED REASONING FOR THE SET-POINT CONTROL

3.1 Case representation

Confirming that the process is out of steady state through the sensory evaluation, the operator adjusts the set-point of the process parameters such as the first-section upper temperature (SP_1), the first-section lower temperature (SP_2), the second-

section upper temperature (SP_3), the second-section lower temperature (SP_4), and the feeding speed (SP_5) to return it into the steady state. Case-based reasoning is utilized to calculate the correction amount $\Delta\widehat{SP}_i(t)$ for the set point $\widehat{SP}_i(t)$ for the ith parameter of the industrial process. Given the adjusted temperature set-point $\widehat{SP}_i^{new}(t) = \widehat{SP}_i(t) + \Delta\widehat{SP}_i(t)$, the controller controls the process according to the new set point so that process transits to a new state. We build the case base with the sensory characteristics, observations and set points, which have been used for adjusting the set point of process parameters. Case base is represented as (4):

$$\boldsymbol{RB}=\{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_k, \ldots, \boldsymbol{R}_K\} \tag{4}$$

where $\boldsymbol{R}_k$ stands for the kth case and K is the number of case. The structure of the case is shown in Table 3.

Table 3: Structure of the case

Condition Characteristics									**Problem Solution**
SP_1	SP_2	SP_3	SP_4	SP_5	$\widehat{SP}_i$	Appearance	Taste	Color	$\Delta\widehat{SP}_i$
r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	sp_i

In Table 3, r_1~r_5 are the current observations of the process and r_6 is the set point of ith variables of interest. r_7, r_8, and r_9 are the variables representing the grades of appearance, taste, and color, respectively and have the value set {1, 2, 3, 4, 5} for the sensory indicators, with 1 for VP and 5 for E.

3.2 Case Retrieval & Reuse

In the retrieval step of CBR, we search for cases that match the current quality state and the operation state of the rolled cake production process. The characteristics representing the current state R is $S=\{r_j\}$, j=1,...,9 and the corresponding solution is $sp_i=\Delta\widehat{SP}_i(t)$. The kth case, R_k in the case base consists of the condition characteristics $S_k=\{r_{j,k}\}$ and the corresponding solution $\Delta\widehat{SP}_{i,k} = sp_{i,k}$, where k=1, ..., K.

The similarities between the current state and the case-base are individually calculated in terms of the process state and the quality state. The similarity between r_1, ..., r_6 and $r_{j,k}$ of the case base is calculated as (5), which corresponds to the process state.

$$sim(r_j, r_{j,k}) = 1 - \frac{|r_j - r_{j,k}|}{max(r_j, r_{j,k})} \quad j=1, \ldots, 6 \tag{5}$$

The similarity between r_7 of the current state R and $r_{7,k}$ of the kth case, which indicates the similarity of the appearance, is calculated as follows.

$$sim(r_7, r_{j,k}) = \begin{cases} 1, & r_7 = r_{7,k} \\ 0.6, & |r_7 - r_{7,k}| = 1 \\ 0.5, & |r_7 - r_{7,k}| = 2 \\ 0.4, & |r_7 - r_{7,k}| = 3 \\ 0.3, & |r_7 - r_{7,k}| = 4 \end{cases} \tag{6}$$

The similarity between r_8 of the current state R and $r_{8,k}$ of the kth case, which indicates the similarity of the taste, is calculated as follows.

$$sim(r_8, r_{8,k}) = \begin{cases} 1, & r_8 = r_{8,k} \\ 0.6, & |r_8 - r_{8,k}| = 1 \\ 0.45, & |r_8 - r_{8,k}| = 2 \\ 0.3, & |r_8 - r_{8,k}| = 3 \\ 0.2, & |r_8 - r_{8,k}| = 4 \end{cases} \tag{7}$$

The similarity between r_9 of the current state R and $r_{9,k}$ of the kth case, which indicates the similarity of the color, is calculated as follows.

$$sim(r_9, r_{9,k}) = \begin{cases} 1, & r_9 = r_{9,k} \\ 0.6, & |r_9 - r_{9,k}| = 1 \\ 0.4, & |r_9 - r_{9,k}| = 2 \\ 0.2, & |r_9 - r_{9,k}| = 3 \\ 0.1, & |r_9 - r_{9,k}| = 4 \end{cases} \quad (8)$$

The similarity between the current state R and the kth case R_k is calculated by classifying the similarity between the quality characteristics and the similarity between the process parameters. The similarity between quality characteristics is calculated as follows.

$$SIM_{quality}(R, R_k) = \frac{\sum_{j=7}^{9} \rho_j \times sim(r_j, r_{j,k})}{\sum_{j=7}^{9} \rho_j} \quad (9)$$

The similarity between process parameters is calculated as follows.

$$SIM_{process}(R, R_k) = \frac{\sum_{j=1}^{6} \rho_j \times sim(r_j, r_{j,k})}{\sum_{j=1}^{6} \rho_j} \quad (10)$$

The overall similarity is calculated by considering the weight w.

$$SIM(R, R_k) = w_1 \times SIM_{quality}(R, R_k) + w_2 \times SIM_{process}(R, R_k) \quad (11)$$

where the weights w_1 and w_2 are set to be 0.7 and 0.3, respectively, and the weighting factors between the individual parameters are set to be $\rho_j = \{1/6,1/6,1/6,1/6,1/6,1/6,1/3,1/3,1/3\}$. Since the case-based reasoning is aimed at computing the correction of the process parameters suitable for the quality state, in order to improve the quality of the solution, we extract subset of cases from the case base so that their condition characteristics represent the similar state to the current quality. To do this, for each case of the case base, we calculate the quality level, $\bar{Q}(\text{appearance, taste, color})$ as shown in Section 2 and calculate the following evaluation index.

$$\triangle Q_k = \left|\bar{Q}(r_7, r_8, r_9) - \bar{Q}(r_{7,k}, r_{8,k}, r_{9,k})\right| \quad (12)$$

By extracting the cases with $\triangle Q_k < 0.5$ from case base, we obtain the subset of case base, $\boldsymbol{RB}^1 = \{\boldsymbol{R}_1^1, \boldsymbol{R}_2^1, \ldots, \boldsymbol{R}_M^1\}$. The solution result corresponding to the current state R is computed using the similarity with respect to the extracted case-base subset $\boldsymbol{RB}^1$ as follows.

$$sp_i = \frac{\sum_{m=1}^{M} SIM(R, R_m^1) \times sp_{i,k}}{\sum_{m=1}^{M} SIM(R, R_m^1)} \quad R_m^1 \in \boldsymbol{RB}^1 \quad (13)$$

3.3 Adaptation of Case Base

Applying the new set point $\widehat{SP}_i^{new}(t) = \widehat{SP}_i(t) + \Delta\widehat{SP}_i(t)$ calculated from $\Delta\widehat{SP}_{i,k}(t) = sp_{i,k}$ to the process the controller would control the process according to the new set point so that the process state would transit to the new set point and the quality state change. In order to determine if the state of the product is improved as intended, it is first necessary to check if the state of the process has entered a stable state around the new set point. To do this, we obtain the deviation $\Delta SP_i(t) = \left|SP_i(t) - \widehat{SP}_i^{new}(t)\right|$ between the measured value of the process and the set point, based on which the test statistics T^2 and its limit are calculated as follows [13].

$$T^2 = \mathbf{x}'\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}'\mathbf{x} \quad (14)$$

$$T^2_{\text{limit}} = \frac{h(n^2-h)}{n(n-h)} F_{\alpha,h,n-h} \quad (15)$$

where $F_{\alpha, h, n-h}$ is the significance level of the F-distribution with degrees of freedom h, n - h, and $\mathbf{x} = [\Delta SP_1(t), \Delta SP_2(t), \Delta SP_3(t), \Delta SP_4(t), \Delta SP_5(t)]'$. $\mathbf{P}$ is the principal component matrix, Λ is the covariance matrix, and h is the number of principal components determined using the method of cumulative percentage of variance. n is the number of the observations.

The state queue $\mathbf{AR} = \{AR^u\}$ represents the sequence of states that have been registered since the beginning of the parameter adjustment. If the test statistics lies under the limit, we conclude that the process operates in the steady state

around the new set point and push the current state ***R*** in the state queue **AR**. That is, $u=u+1$, $AR^u=R$. The push-in process of the state queue is shown in Figure 2.

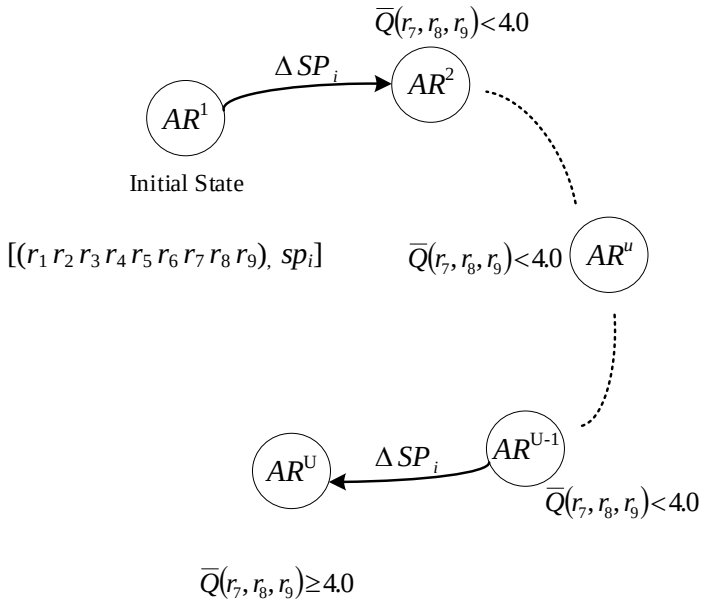


Figure 2: Registration of the state into the state sequence

We then determine the quality level for the state and if not passed, calculate the new set-point using (5)~(12). If the quality level for a given state R is $\bar{Q}(r_7, r_8, r_9) > 4.0$, the parameter adjustment is considered to be successful and perform the updating and adding the case base considering the states in the state sequence AR. For the state AR^u u=1 ,,, U in the state queue **AR**, we compute the maximum of similarity values with case in the case base, $SIM^{max}_{quality}(R, R_k)$, $SIM^{max}_{process}(R, R_k)$.

With respect to the maximum similarity, we perform the updating and addition of the case base in terms of two conditions as follows.

- Condition #1: $\left(SIM^{max}_{quality}(AR^u, R_k) \geq 0.9\right) \wedge \left(SIM^{max}_{process}(AR^u, R_k) \geq 0.7\right)$

As this means that there exists more than one case in case base whose condition characteristics are identical to the quality characteristics and process parameters of a given state, the solution value of the case is updated as follows:

$$\Delta\widehat{SP}_{i,mk} = \widehat{SP}_i - \widehat{SP}_{i,mk} \tag{16}$$

where *mk* is the index of the case whose similarity value satisfies Condition #1. $\widehat{SP}_i$ is the set point of the *i*th process parameter in the final steady state AR^U in the state queue **AR**.

- Condition #2: $\left(SIM^{max}_{quality}(AR^u, R_k) < 0.9\right) \vee \left(SIM^{max}_{process}(AR^u, R_k) < 0.7\right)$

Even though the maximal similarity of the process parameters is $SIM^{max}_{process}(AR^u, R_k) > 0.7$, if the quality state is slightly different, i.e., $SIM^{max}_{quality}(AR^u, R_k) < 0.9$, it is considered as a new rule and added to the case base. The solution value of the added case is calculated using (16). In the same way as above, for all the states in the state queue AR, the update and addition of case base is performed and the state queue is initialized as an empty set.

4. EXPERIMENT

For the purpose of validation, the proposed method is applied to the rolled cake process, in which the length of the kiln is 16 m, the number of temperature control stages is 2, the feed rate range is 0.05-0.5 m/s, the supply power is 380 V ± 10% in three phases and 50-60 Hz. The main elements for control: the programmable logic controller (PLC) and the intelligent controller, receive signals from sensors equipped in the process to control the actuator according to the requirements of the process operation. HMI allows the operator to turn on/off the equipment, set up the process parameters and record the

situations for the operation of the process. In the field layer, the situation information such as temperature, speed, etc. for supervisory and control is measured and transmitted to the control layer.

In order to demonstrate the performance of the inference mechanism to overcome the variation, two types of raw materials with slightly different water-absorbing properties are used for the production of dough to generate the quality variation. Figure 3 shows the temperature behavior when the experienced operator adjusted the set point by his visual inspection of the quality condition. As shown in the figure, the set point adjustment was performed several times to reach a steady state, elapsing about 1380s (23 steps) for the first material and 1560s (26 steps) for the second material. The set-point adjustment and the process behavior by the proposed method are shown in Figure 4.

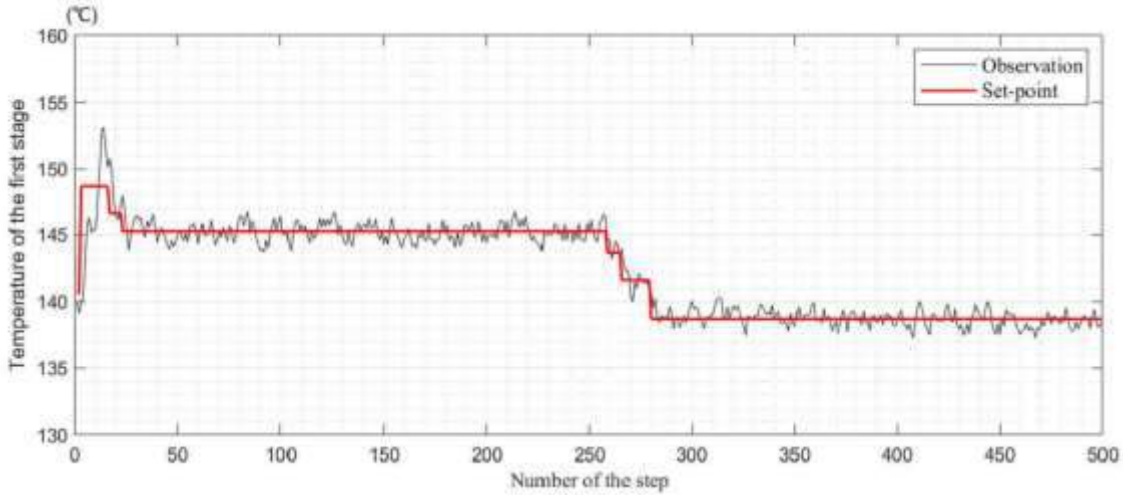


Figure 3: Behavior of the process in the case of manually controlling the set point.

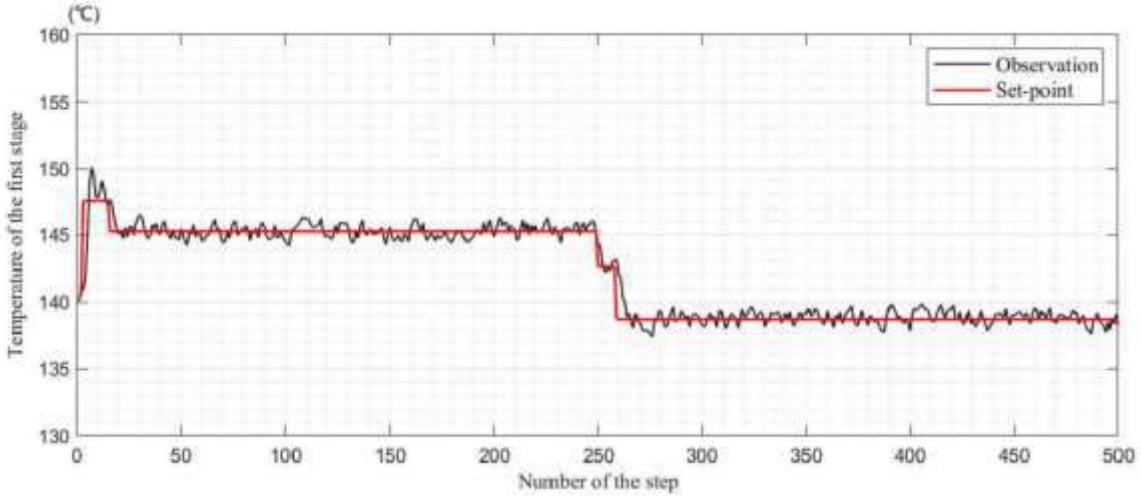


Figure 4: Behavior of the process in the case of adjusting the set point by the proposed method.

For the first material, the adjustments of the set-point were performed three times and for the second, two times. 16 and 14 steps are elapsed respectively to stabilize the process, corresponding to an average of 900 s (15 steps) in time. In order to perform the comparative analysis of the quality of the product, the quality level of the samples collected at 15 intervals is shown in Figure 5. The proposed method improved the average quality level from 4.39 to 4.51 and increased the acceptance rate by 4.1% compared with the manual operation.

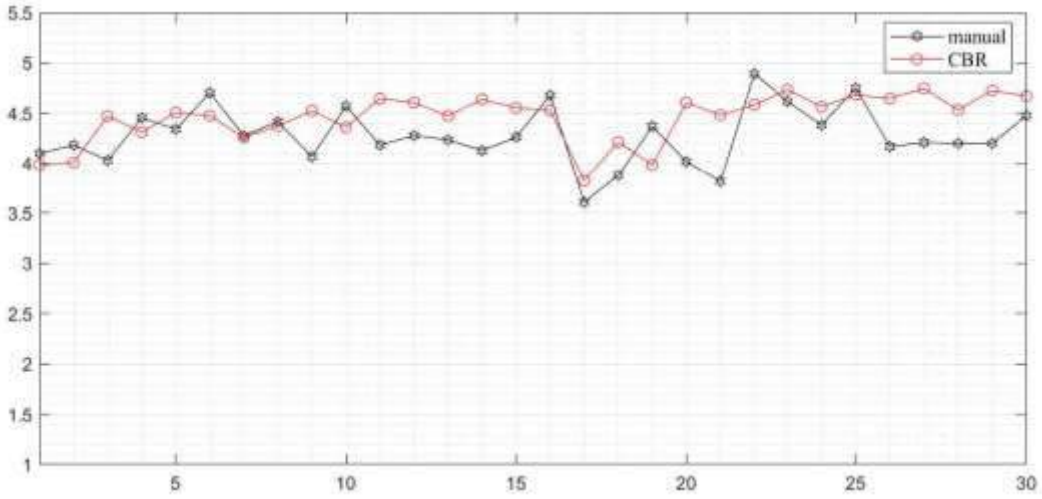


Figure 5: Comparative analysis of the quality level.

5. CONCLUSION

This paper presents a methodology to quantitatively evaluate the quality of the rolled cake and to support the decision-making of the operator to stabilize it. An inference base for quantitatively evaluating the quality of the product from the sensory characteristics of the product was constructed, using the knowledge collected from several experts. Decision support system based on CBR was established to support the operator to reasonably adjust the set-points under the operating conditions when quality anomalies occur. In view of the set-point tracking characteristics of the automatically controlled process, we determine when to adjust the set-point using multivariate statistical process monitoring. Then a batch method to update the case base in CBR for setup adjustment is proposed, which introduces the state queue for online adaptation of case-base during the operation. The comparative experiment demonstrates that proposed method has the ability to overcome the abnormal condition and significantly improves the quality of the product. Future work will concern with the function of automatically determining the weight coefficients in CBR to expand the range of applications for industrial processes with sensory quality characteristics.

ACKNOWLEDGMENTS

The authors would like to thank the editors and reviewers for their interest in completing this paper well.

REFERENCES

[1] Sarra ADJIM, Rachida GHOUL HADIBY, and Alli CHERMITTI Abdelkader SLIMANE, “Fuzzy Control of Product Quality in a Manufacturing System Modeled with Interval Constrained Petri Net,” International Journal of Engineering Research in Africa 40, 151-161 (2018).

[2] HJALTE TRNKA, JIAN X. WU, MARCO VAN DE WEERT, HOLGER GROHGANZ, and JUKKA RANTANEN, “Fuzzy Logic-Based Expert System for Evaluating Cake Quality of Freeze-Dried Formulations,” Journal of Pharmaceutical Sciences 102, 4364–4374 (2013).

[3] S. Kupongsak, and J. Tan, “Application of fuzzy set and neural network techniques in determining food process control set points,” Fuzzy Sets and Systems 157, 1169–1178 (2006).

[4] Yao, L., Postlethwaite, I., Browne, W., Gu, D., Mar, M., and Lowes, S., “Design, implementation and testing of an intelligent knowledge-based system for the supervisory control of a hot rolling mill,” Journal of Process Control 15, 615–628 (2005).

[5] S.I. Lao, K.L. Choy, G.T.S. Ho, Richard C.M. Yam, Y.C. Tsim, and T.C. Poon, "Achieving quality assurance functionality in the food industry using a hybrid case-based reasoning and fuzzy logic approach," Expert Systems with Applications 39, 5251–5261 (2012).
[6] Chai, T. Y., Ding, J. L., and Wu, F., "Hybrid intelligent control for optimal operation of shaft furnace roasting process," Control Engineering Practice 19, 264–275(2011).
[7] Song Ho Kim, Kwang Rim Song, Il Yong Kang, and Chung Il Hyon, "On-line set-point optimization for Intelligent Supervisory Control and Improvement of Q-learning Convergence," Control Engineering Practice 114 (2021). DOI: 10.1016/j.conengprac.2021.104859.
[8] I. Allais, N. Perrot, C. Curt a, and G. Trystram, "Modelling the operator know-how to control sensory quality in traditional processes," Journal of Food Engineering 83, 156–166(2007).
[9] C. Curt, J. Hossenlopp, and G. Trystram, "Control of food batch processes based on human knowledge," Journal of Food Engineering, 79, 1221–1232(2007).
[10] Jie-sheng Wang, Na-na Shen, and Shi-feng Sun, "Integrated Modeling and Intelligent Control Methods of Grinding Process," Mathematical Problems in Engineering, Article ID 456873(2013). http://dx.doi.org/10.1155/2013/456873.
[11] Kosta Boshnakov, Venko Petkov, and Metodi Nikolov "DECISION MAKING FOR CONTROL OF COMBUSTION PROCESS OF PULVERIZED COAL," Journal of Chemical Technology and Metallurgy 50, 183-192(2015).
[12] E. Roldan Reyes, S. Negny, G. Cortes Robles, and J.M. Le Lann, "Improvement of online adaptation knowledge acquisition and reuse in case-based reasoning: Application to process engineering design," Engineering Applications of Artificial Intelligence 41,1–16(2015).
[13] Kwang Rim Song, Song Ho Kim, Chol Jun Han, and Il Yong Kang, "Monitoring Industrial Processes with Multiple Operation Modes: a Transition-Identification Approach Based on Process Variability," Ind. Eng. Chem. Res 62, 3358–3370(2023).

LLM-Based Method for Generating Vulnerable Code Equivalents

Jianfei Chen[a], Jing Liu[b], Jiao Wang[c], and Dongchang Li[c]

[a]State Grid Shandong Electric Power Company, Jinan, China
[b]State Grid Shandong Electric Power Research Institute, Jinan, China
[c]Beijing KeDong Electric Power Control System Co.Ltd., Beijing, China

ABSTRACT

This paper introduces an LLM-based method for generating vulnerable code equivalents to enhance software system security. With the rapid growth in software development, code security has become a significant concern. Traditional defense mechanisms are inadequate against unknown threats stemming from unidentified vulnerabilities. The proposed method leverages Large Language Models (LLMs) to generate executables that are functionally equivalent but structurally diverse, allowing for prompt replacement of vulnerable code and ensuring system stability. By integrating fuzz testing, the approach validates the functional equivalence of generated code through code coverage tracking, reducing the number of input sets needed while increasing coverage. The method aims to address three key questions: ensuring software system operation under attack, generating executables efficiently, and testing functional equivalence effectively. The study demonstrates that LLMs can improve executable generation efficiency, combine with fuzz testing for thorough validation, and maintain code correctness. The experiments show the method's effectiveness in patching vulnerabilities and producing functionally equivalent executables, offering a potential defense against new threats.

Keywords: Software Security, Vulnerable Code, Large Language Models, Fuzz Testing, Equivalent Executables

1. INTRODUCTION

With the rapid development of information technology, the demand for software development has seen an explosive growth. Concurrently, the quality and efficiency of software development have become pressing issues within the field. The proliferation of agile development and DevOps practices has enhanced the speed and efficiency of software development. However, this has also increased the risk of code security. Development practices such as rapid iteration and frequent deployment may lead to security issues being overlooked or not addressed in a timely manner, resulting in software systems that fail to operate properly or even crash.

The inability to eliminate, control, or thoroughly investigate software vulnerabilities and backdoors has led to unknown threats based on unknown vulnerabilities and backdoors becoming a significant security challenge for software. Traditional defense mechanisms based on firewalls and intrusion detection rely on prior knowledge of attacks, focusing on external security reinforcement of target systems and the detection of known threats. These are "after-the-fact" add-on security defense mechanisms and are incapable of addressing unknown threats triggered by unknown vulnerabilities and backdoors.

In the face of unknown threats, to ensure the normal operation of software systems, this paper proposes the equivalent generation of vulnerable code. By generating equivalent executables, problematic code can be promptly replaced to ensure the stable operation of software systems. The emergence of Large Language Models

Further author information:
Jianfei Chen: E-mail: chenjianfei@sgcc.com.cn
Jing Liu: E-mail: liujing@sgcc.com.cn
Jiao Wang: E-mail: maywangjiao@163.com
Dongchang Li: E-mail: ldc9211@sina.com

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 1351908 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058052

(LLMs)[1–3] has accelerated the process of generating equivalent executables, and by combining this with fuzz testing methods, functional equivalence testing of executables generated or transformed by LLMs can be conducted through tracking code coverage.[4,5]

Utilizing LLMs to generate equivalent executables can improve efficiency and speed,[6] enabling systems to generate a diverse range of code variants more rapidly. By combining this with fuzz testing methods, the executables can be tested by tracking code coverage. Through fuzz testing,[7,8] the number of input set traversals can be reduced while continuously increasing code coverage, thus achieving comprehensive testing of various paths within the executable.

In terms of functional equivalence testing, the functionality of the executable can be determined by comparing output sets.[9,10] By analyzing the output sets obtained from LLM-generated executables and comparing them with the original code's output sets, the functional equivalence of the executables can be assessed, thereby determining whether they conform to the expected code behavior.

Therefore, the scientific questions this paper aims to address are as follows:

Q1: How can the normal operation of a software system be ensured when the vulnerable code executables within the system are under attack?

Q2: Although the concept of executables can effectively solve system security issues, how can code executables be generated more efficiently?

Q3: How can the functional equivalence of executables be tested more efficiently?

2. BACKGROUND INFORMATION

2.1 Large Language Models

Large Language Models (LLMs) are powerful text processing frameworks developed based on deep learning techniques.[11–13] By learning the linguistic patterns and semantic relationships from vast amounts of textual data, models such as the Codex behind GitHub Copilot,[14] Microsoft's CodeBERT,[14] and Salesforce's CodeT5[15] have demonstrated the ability to understand the structure and logic of programming languages. They can generate code that adheres to programming standards based on natural language descriptions or partial code snippets, greatly enhancing the efficiency of software development. Based on this acquired knowledge, models can generate new code snippets or transform input code into target languages, providing robust support for tasks like automated code writing and language translation. Thus, LLMs play a significant role in code generation and transformation tasks, offering powerful text processing and generation capabilities by analyzing the underlying structure and patterns of input sequences. When a user submits a code generation or transformation task,[16–18] the input text is first preprocessed into a numerical representation that the model can understand, such as word vectors or character encodings. Then, by inputting these numerical representations into the LLM, the model can generate and transform code based on its learned knowledge and contextual information.[19–21]

In terms of generating executables, leveraging LLMs brings significant advantages. LLMs can learn a vast array of code samples and program structures, as well as coding standards and best practices, enabling the model to generate code snippets or structures with a certain level of quality and rationality, thereby more efficiently generating executables. By incorporating LLMs, developers can reduce the workload of manual coding and improve the speed and efficiency of code generation. Moreover, by applying fuzz testing to code generated by LLMs and using the test feedback as input, through multiple iterations of analyzing abnormal information and error reports, the model can identify vulnerable code snippets[1,22] and quickly locate and fix issues in the code based on contextual information, enhancing code quality and security.

2.2 Fuzz Testing

Fuzz testing involves providing random inputs to a test program to detect its behavior and output under various input conditions. When applied to code generated by LLMs, fuzz testing can be used to verify whether the generated code meets expectations, thereby validating the accuracy and reliability of the code. One representative gray-box testing method is AFL (American Fuzzy Lop).[23,24] As shown in fig1, the uniqueness of this method lies in its generation of new test cases through mutation based on genetic algorithms, coordinating

execution processes, analyzing execution information, and testing the program. AFL utilizes dynamic binary instrumentation to monitor the execution path and code coverage of the program, tracking the code coverage during fuzz testing.[25,26] This can effectively detect code, determine which code paths are executed and which are not triggered, thereby enhancing the security and stability of the code.

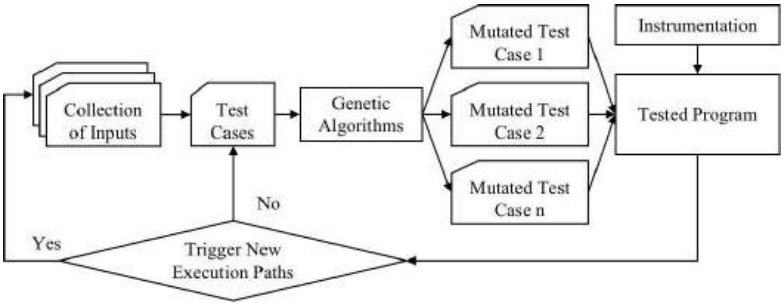


Figure 1. AFL Algorithm Flowchart

In the context of code generated by LLMs, the goal of fuzz testing is to verify the functional equivalence of the generated code. Fuzz testing generates a series of random inputs as test samples and provides these samples to the code generated by the LLMs for execution. Subsequently, by detecting and validating the output of the generated code, it can be determined whether the generated code conforms to the expected functionality and behavior, improving the efficiency of detecting functionally equivalent executables.

3. GENERATE FUNCTIONALLY EQUIVALENT CODE EXECUTABLES

3.1 Method Design

To enhance the efficiency of executable generation and ensure the normal operation of software systems, this paper proposes an innovative method. The core idea of this method is to leverage the capabilities of LLMs to generate a variety of executables with heterogeneous features. LLMs, with their understanding of language structure and context, can produce executables that are syntactically correct and semantically diverse. In the process of executable generation, fuzz testing is integrated to ensure the functional equivalence of the executables. By tracking code coverage, it is ensured that the heterogeneous executables cover different execution paths and boundary conditions within the software system, thereby guaranteeing their functional equivalence.Through this method, the efficiency and quality of executable generation can be effectively improved, and the functional equivalence of the executables is ensured, enhancing the robustness and security of the system. The specific implementation process of this method is shown in fig2.

Step 1: Model Selection. Initially, select closed-source LLMs to accomplish code generation or transformation tasks. Closed-source LLMs utilize advanced generation algorithms and techniques that have been researched and optimized over time to more efficiently complete code generation or transformation tasks. Additionally, closed-source LLMs are trained and inferred on high-performance hardware, providing faster computational speeds and higher parallelism, thereby accelerating task completion.

Step 2: Task Design. Whether it is a code generation or transformation task, ensure the accuracy and consistency of the natural language description when designing the task. That is, the input text must remain consistent across different executables to ensure that the model's understanding of the task does not deviate, which is a prerequisite for the functional equivalence of the executables.

Step 3: Program Correctness Verification. Detect the correctness of the obtained executables; if the generated code results in errors during compilation or execution, feedback the compiler's return messages to the LLM. The model uses this information to adjust the generation or transformation process, re-generating code that meets expectations and ensuring the correctness of the code.

Step 4: Utilize fuzz testing techniques to generate test cases and continuously improve code coverage, obtaining the output sets of various executables. By generating diverse test cases, potential errors, vulnerabilities, or inconsistencies are more efficiently discovered, and the behavior of the code under different input conditions

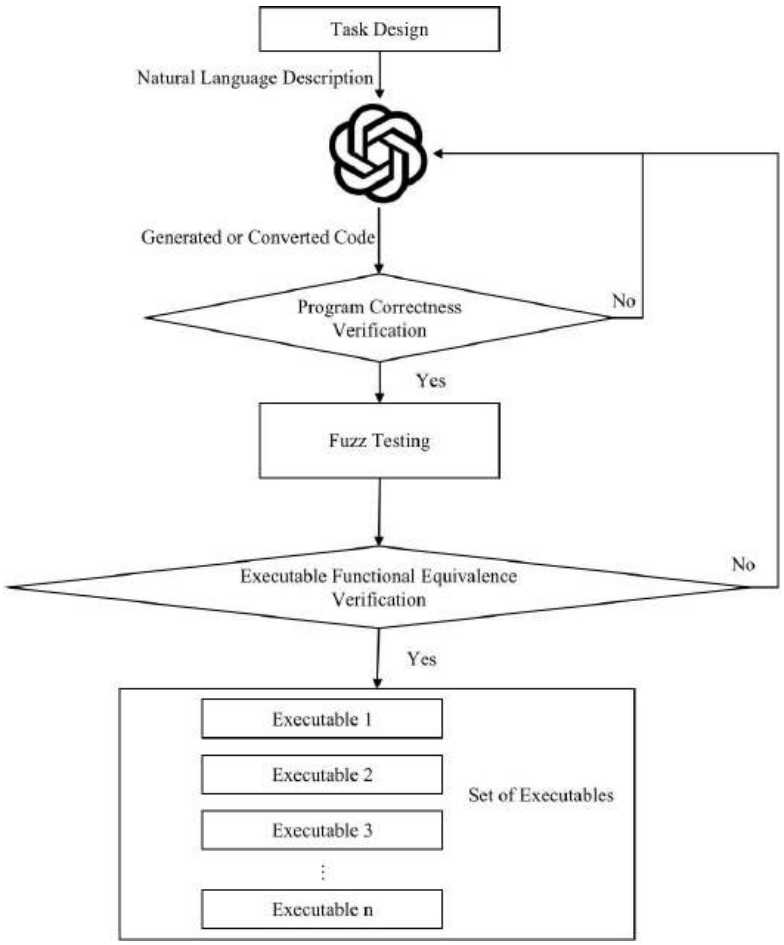


Figure 2. LLM-Based Method for Generating Vulnerable Code Equivalents Flowchart

is verified without traversing the entire input set. Introducing various boundary values, random data, and exceptional situations, fuzz testing triggers different branches and logical paths in the code, thereby increasing code coverage. The continuous improvement of coverage helps identify functional equivalence issues in the code, where the same input may lead to different outputs.

Step 5: Continue feedback for functional non-equivalence while ensuring Step 3. As code coverage continues to improve, re-feed the discovered non-equivalence situations back to the model. At the same time, continue to ensure the correctness detection of code in Step 3 to maintain the quality and correctness of the generated code. Through continuous feedback and detection, gradually improve the generation capabilities of the LLMs and ultimately obtain functionally equivalent executables.

3.2 LLMs Repair Vulnerabilities in Executables

When vulnerable code in a software system is attacked and generates an error, the method proposed in this paper aims to ensure the normal operation of the software system by using executables to promptly replace the vulnerable code. However, repairing the vulnerabilities in the fragile code and ensuring that the executables do not contain the same vulnerabilities is an important scientific question. To address the question, as shown in fig3, the paper designs a vulnerability repair framework for executables based on LLMs.

1. To construct a robust code repair framework, this paper collects information related to executable errors through three approaches, which will serve as the "Prompt" input for the Large Language Model.

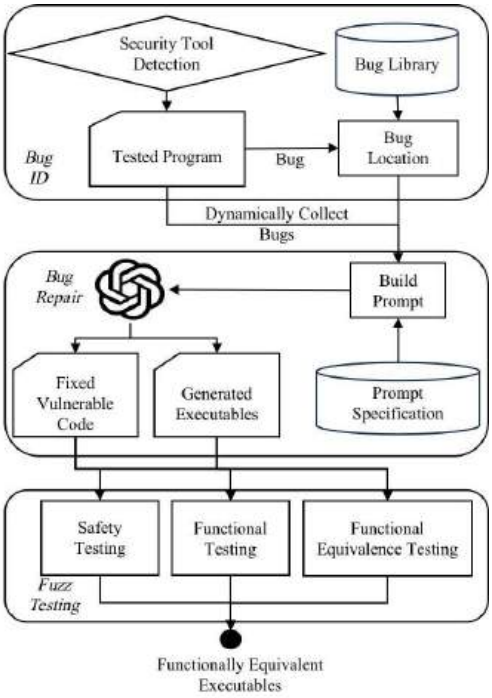


Figure 3. Framework for LLMs Repair Vulnerabilities in Executables

(a) Through manual efforts, we have curated a repository of common program errors and their respective fixes, establishing a vulnerability database that includes a wide range of error types along with their corresponding solutions.

(b) We have utilized existing security tools to detect faults within the software system, encompassing both vulnerabilities and anomalies. These tools employ a suite of methods, including static analysis and dynamic monitoring, to identify potential code issues. By executing these tools, we are able to collect comprehensive details regarding code errors and vulnerabilities, complete with recommended fixes.

(c) Our framework is equipped with the ability to dynamically capture error messages that arise during the software system's runtime. In the event of an error during execution, the system can immediately record the error's nature, location, and surrounding context. This real-time error capture provides vital insights into the problems occurring throughout the code's execution.

2. With a substantial collection of error messages from Vulnerable code, these messages are crafted into an apt "Prompt" input designed to direct the Large Language Model towards generating potential corrective code. Beyond mere code correction, the Large Language Model further translates the fixed code into executables across various programming languages. Through logical reconstruction of the original code, this process eliminates underlying vulnerabilities and errors, ensuring that the newly created executables entities are free from the same flaws.

3. Fuzz testing is employed on the corrected code and resulting executables to verify their functional integrity under diverse input scenarios and to evaluate their security against potential threats. Concurrently, code coverage tools monitor the execution paths and branches traversed during testing. Analysis of these coverage reports reveals whether the functionalities of the different executables are equivalent. Should any discrepancies arise, the unusual inputs are reintroduced into the LLMs to refine the code transformation process.

4. The corrected code and resultant executables are amalgamated into a comprehensive set. In the event of

an attack on vulnerable code, these executables are deployed as replacements, ensuring the uninterrupted operation of the software system.

3.3 Functional Equivalence Testing of Executables

Following the generation of executables by the LLM, it is essential to perform functional equivalence testing among them. This study employs fuzz testing techniques to craft a diverse array of test cases, systematically enhancing the code coverage for each executable to collect their output sets. This process also verifies the behavior of the code under various input scenarios, revealing any instances of functional inequality among the executables. With each iteration that increases code coverage, the identified disparities are re-ingested into the model. This cycle of persistent feedback and evaluation refines the LLM's generative capabilities, leading to the development of functionally equivalent executables.

However, while the AFL algorithm augments code coverage by tracking the execution paths of test cases and generating new ones, it is not without its shortcomings. The algorithm's approach to test case creation is somewhat indiscriminate, with a coarse level of analysis and a lack of specificity, resulting in suboptimal testing efficiency. Drawing on previous research, to enhance the efficiency of functional equivalence testing for executables, this paper introduces an improvement to the AFL algorithm. The proposed method conducts fuzz testing based on seed distance, prioritizing key functions that are pivotal for increasing execution coverage. By boosting the probability of execution transitioning from the program's current state to the targeted key functions, coverage is significantly expanded. This refined approach provides direction to fuzz testing, enabling the generation of more purposeful test cases and, consequently, improving the detection efficiency of functional equivalence in executables. The detailed improvement process is depicted in Figure4.

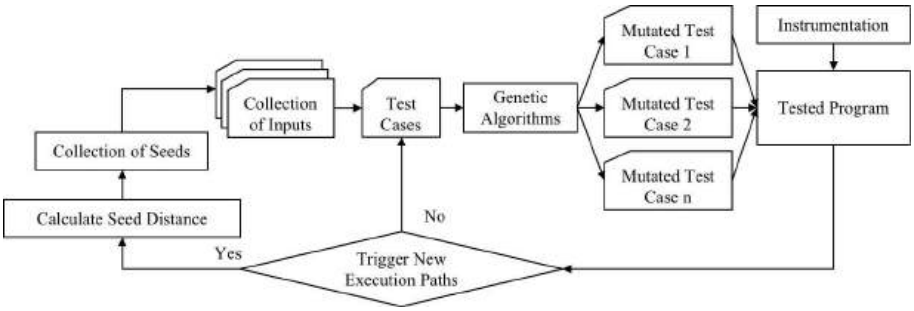


Figure 4. Fuzzy Testing Method Based on Seed Distance

Firstly, preprocess the target program to obtain the call graph $G(V, E)$, where V represents the nodes in the graph G, indicating each function, and E represents the edges in the graph G, indicating the calling relationships between functions. The out-degree O and in-degree I of each node correspond to the number of times a function is called and is called by another function, respectively. Analyze the call graph and calculate the weight of each node as shown in Formula (1).

$$W(v_i) = \frac{O(v_i)}{I(v_1) + I(v_2) + ... + I(v_{i-1}) + I(v_{i+1}) + ... + I(v_n)}, \tag{1}$$

Where $v_i \in V$, the node with the highest weight is used as the anchor node r for fuzz testing, which serves as the key function starting point in the fuzz testing process.

Then, the distance $Dis(r, v_i)$ from the seed node to the anchor node is determined through instrumentation tools. The seed node serves as the initial test case input, and the instrumentation tool measures the distance between the seed node and the anchor node, which is the length of the execution path.

Ultimately, a sorting mechanism is employed based on the proximity of seed nodes to the anchor node, with those nearest to the anchor node being selected first for fuzz testing inputs. This strategy prioritizes seeds that are in closer proximity to the target function, which not only boosts the coverage but also enhances the overall efficacy of the fuzz testing process.

This proximity-based selection offers the dual benefit of efficiency and focus; seeds that are closer to the target function are more likely to efficiently navigate and encompass critical pathways within the target application. By concentrating on these seeds for testing, the generation of test cases becomes more refined, leading to higher code coverage. Additionally, this approach expedites the fuzz testing process by prioritizing tests on seeds that are closest to the target function, thus streamlining the detection of functional equivalence among various executables.

4. EXPERIMENTS AND ANALYSIS

4.1 Experiment 1

In order to substantiate the effectiveness of the LLM-driven approach for the remediation and generation of executables as presented in this paper, a comprehensive scan of existing C++ code was conducted using the security utilities of our framework. This process revealed an executable segment prone to overflow vulnerabilities, which is depicted in Listing 1.

Listing 1. C++ Code with Bugs

```
int countTrees(int n){
    std::vector<int> dp(n+1, 0);
    dp[0] = 1;
    for(int i = 1; i <= n; i++){
        for(int j = 0; j <= i; j++){
            dp[i] += dp[j] * dp[i - j - 1];
        }
    }
    return dp[n];
}
```

The code utilizes the principles of dynamic programming to determine the quantity of binary tree structures possible with a specified number of nodes, n. Nonetheless, the scanning process has uncovered a latent issue: there is a risk of overflow when processing a substantial count of nodes. Such overflow vulnerabilities have the potential to push computation results beyond the expressible limits of the data type, resulting in unpredictable behavior and incorrect results. Subsequently, this paper will assess the detection framework's ability to eradicate these latent vulnerabilities and guarantee that the newly generated executable is devoid of similar defects.

4.1.1 Bugs Fixing

By correlating the vulnerability information identified by security tools with the entries in the vulnerability database and seeking corresponding remediation advice, we can translate both the vulnerability details and the proposed fixes into natural language inputs conforming to the predefined Prompt protocol for the Large Language Model. Consequently, the framework delivers the rectified code executable, as illustrated in Listing 2.

Listing 2. Fixed C++ Code

```
std::uint64_t countTrees(int n){
    std::vector< std::uint64_t > dp(n+1, 0);
    dp[0] = 1;
    for(int i = 1; i <= n; i++){
        for(int j = 0; j <= i; j++){
            dp[i] += dp[j]* dp[i - j-1];
        }
    }
    return dp[n];
}
```

Within the fixed code, the data type of the *dp* array has been upgraded from int to $std :: uint64\_t$, an unsigned 64-bit integer, effectively expanding the representable value range. However, the LLM advises that this alteration alone might not suffice for all potential overflow concerns. For scenarios demanding an even broader data range, it is imperative to employ a more extensive integer type, such as $boost :: multiprecision :: uint128\_t$ or alternative high-precision libraries, to accurately manage computations involving large data sets. Upon re-scanning the amended executable with security tools, it was confirmed that the overflow error had been successfully rectified. This outcome underscores the efficacy of the framework detailed in this paper in addressing vulnerabilities within delicate code segments.

4.1.2 Executables Generation

To ascertain if the executables generated by our framework have mitigated the initial security vulnerabilities, an experiment was conducted. The experiment involved inputting C++ code prone to overflows, along with formalized vulnerability data, into the framework. Consequently, this process yielded executables in both Rust and Java. The execution outcome for the Rust language is depicted in Listing 3.

Listing 3. A Bug-Free Rust Language Executable

```
fn count_trees(n: u32)-> u64{
    let mut dp: Vec<u64> = vec![0;(n + 1) as usize];
    dp[0] = 1;
    for i in 1..n{
        for j in 0..i{
            dp[i as usize] += dp[j as usize]* dp[(i - j-1) as usize];
        }
    }
    dp[n as usize]
}
```

In the Rust code that has been transformed, the framework opts for $u32$ and $u64$ types to supplant the initial integer types, thereby ensuring that overflow is averted within the integer range. $Vec < u64 >$ takes the place of the previous dynamic programming array, and Rust's iterator and loop syntax are used to substitute for the loop constructs present in C++. By harnessing Rust's inherent types, safety features, and robust error handling, the executable can more effectively handle input/output operations and numerical overflows, ensuring its security and accuracy, and thereby resolving the initial overflow concerns.

Similarly, in the Java language executable produced by the framework, as demonstrated in Listing 4, the overflow issue is fixed. The long type replaces the original integer type to prevent overflow within the positive range, and in accordance with Java's language specifications, the *long*[] array supersedes the former dynamic programming array.

Listing 4. A Bug-Free Java Language Executable

```
public class BinaryTreeCount{
    public static long countTrees(int n){
    long[] = dp new long[n + 1];
        dp[0] = 1;
        for (int i = 1; i <= n; i++){
            for(int j = 0; j < i; j++){
            dp[i] += dp[j]* dp[i - j-1];
        }
    }
    return dp[n];
}
```

The framework put forth in this paper has shown remarkable efficacy in eradicating vulnerabilities within delicate code. Utilizing an array of security tools and methodologies, a thorough inspection and verification of the generated executables are performed. The findings from the experiments demonstrate that the newly

minted executables are devoid of the vulnerabilities present in their original counterparts. Beyond this, rigorous experimental validation has established that the framework can adeptly detect and remediate an array of vulnerability types within executables, encompassing buffer overflows, code injection, authentication bypass, and more. Importantly, the framework transcends a mere one-off fix for vulnerabilities; it is equipped with capabilities for ongoing monitoring and automated remediation. With the discovery of new vulnerabilities, the framework can swiftly implement corrections, ensuring the ongoing security and stability of the system.

4.2 Experiment 2

To ascertain the comparative efficacy of the seed-distance-based fuzz testing approach versus the AFL algorithm with respect to code coverage, the subsequent experiment was devised. Three program codes of differing complexities were selected, and a trio of comparative experiments was executed. These experiments were performed on a host machine configured with a 64-bit Ubuntu 16.04 operating system, powered by a 3.4 GHz Intel i7 processor (featuring 4 cores and 8 threads), and endowed with 16 GB of RAM. The study compared the code coverage achieved by both the AFL algorithm and an enhanced variant thereof when allotted the same duration on a variety of programs. This comparison aimed to evaluate the extent of efficiency enhancement afforded by the optimization method. The experimental outcomes are detailed in Table 1."

Table 1. Code Coverage Comparison Between AFL and the Proposed Method

Program	Time Complexity	Method	Time (h)	Coverage Rate
Program 1	O(nlogn)	AFL	1	58.34%
			3	73.23%
			6	89.45%
		Proposed Method	1	52.86%
			3	69.92%
			6	91.56%
Program 2	$O(n^2)$	AFL	5	46.74%
			10	60.23%
			15	71.49%
		Proposed Method	5	49.01%
			10	65.59%
			15	77.03%
Program 3	$O(n^3)$	AFL	12	39.58%
			18	49.89%
			24	57.42%
		Proposed Method	12	43.57%
			18	55.28%
			24	65.24%

The experimental findings reveal that for the relatively straightforward Program 1, the AFL algorithm initially achieved a notably high code coverage rate. This is largely attributed to the AFL algorithm's capability to swiftly generate a plethora of test cases that encompass the majority of execution paths in low-complexity code. Conversely, the optimized algorithm's performance at the outset was not as impressive as the AFL algorithm's, owing to the extra time needed for distance-based seed sorting. While this step might seem to hinder short-term efficiency, it is crucial for bolstering test coverage in subsequent stages. As the experiment progressed, there was a notable shift; after about 6 hours, the optimized algorithm began to outperform the AFL algorithm in terms of coverage rate for the same amount of time spent. This improvement is credited to the optimized algorithm's intelligent scheduling for seed selection and path exploration, enabling it to more efficiently utilize generated test cases to uncover new code paths.

When confronted with the more intricate Programs 2 and 3, the advantages of the optimized algorithm became even more pronounced. Within the identical testing duration, the optimized algorithm consistently

delivered higher branch and statement coverage rates compared to the AFL algorithm. This suggests that the optimized algorithm is more adept at directing the generation and mutation of test cases for complex programs, leading to a more efficient verification of the functional equivalence across different executables.

Synthesizing all the experimental outcomes, we can conclude that the optimized AFL algorithm enhanced the code coverage rate by an average of approximately 26.32% over the original AFL algorithm within the same testing timeframe. This marked increase validates the optimized algorithm's efficiency and its potency in assessing the functional equivalence of executables. Furthermore, while the optimized algorithm's initial focus on seed distance sorting temporarily compromises speed, the long-term benefits become increasingly evident as testing intensifies, particularly with programs of higher complexity. The optimized algorithm's performance is particularly exceptional, significantly amplifying the efficiency of detecting functional equivalence in executables.

5. CONCLUSION

Amidst the rapid evolution of software systems, this paper introduces an LLM-based method for generating vulnerable code equivalents to mitigate associated security challenges. By developing executables that are functionally equivalent but implemented differently, this approach bolsters the system's elasticity and defensive capabilities against attacks. Through harnessing the generative power of LLMs and employing rigorous fuzz testing for validation, this study significantly enhances the security and robustness of software systems. The experimental outcomes indicate that our method not only effectively patches code vulnerabilities but also swiftly produces functionally equivalent executables, demonstrating its potential to guard against novel threats.

As artificial intelligence and machine learning technologies progress, the role of large language models in the realm of software security is set to become more pivotal. Future endeavors may delve into optimizing the training regimen of these models to yield higher caliber executables. Additionally, advancements in fuzz testing, focusing on increased automation and efficiency, will be pivotal research avenues. The integration of cross-disciplinary approaches, such as merging big data analytics with cyber security techniques, could pave the way for innovative breakthroughs in software security. Collectively, these endeavors aim to establish a more intelligent and secure software ecosystem, fortifying our defenses against the escalating sophistication of cyber threats.

ACKNOWLEDGMENTS

This work is supported by the science and technology project of State Grid Corporation of China: "Research and Application of Fuzzy Testing Technology for Power System Terminals" (Grand No. 5700-202316312A-1-1-ZN).

REFERENCES

[1] Ahmad, W. U., Chakraborty, S., Ray, B., and Chang, K.-W., "Unified pre-training for program understanding and generation," *arXiv preprint arXiv:2103.06333* (2021).

[2] Athiwaratkun, B., Gouda, S. K., Wang, Z., Li, X., Tian, Y., Tan, M., Ahmad, W. U., Wang, S., Sun, Q., Shang, M., et al., "Multi-lingual evaluation of code generation models," *arXiv preprint arXiv:2210.14868* (2022).

[3] Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al., "Program synthesis with large language models," *arXiv preprint arXiv:2108.07732* (2021).

[4] Wondracek, G., Comparetti, P. M., Kruegel, C., Kirda, E., and Anna, S. S. S., "Automatic network protocol analysis.," in [*NDSS*], **8**, 1–14, Citeseer (2008).

[5] Bavarian, M., Jun, H., Tezak, N., Schulman, J., McLeavey, C., Tworek, J., and Chen, M., "Efficient training of language models to fill in the middle," *arXiv preprint arXiv:2207.14255* (2022).

[6] Fuzzing, I. W., "Sage: whitebox fuzzing for security testing," *SAGE* **10**(1) (2012).

[7] Cha, S. K., Avgerinos, T., Rebert, A., and Brumley, D., "Unleashing mayhem on binary code," in [*2012 IEEE Symposium on Security and Privacy*], 380–394, IEEE (2012).

[8] Gorbunov, S. and Rosenbloom, A., "Autofuzz: Automated network protocol fuzzing framework," *Ijcsns* **10**(8), 239 (2010).

[9] Godboley, S., Dutta, A., Pisipati, R. K., and Mohapatra, D. P., "Ssg-afl: Vulnerability detection for reactive systems using static seed generator based afl," in [*2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*], 1728–1733, IEEE (2022).

[10] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al., "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology* **15**(3), 1–45 (2024).

[11] Chernyavskiy, A., Ilvovsky, D., and Nakov, P., "Transformers:"the end of history" for natural language processing?," in [*Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*], 677–693, Springer (2021).

[12] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al., "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374* (2021).

[13] Chen, X., Liu, C., and Song, D., "Execution-guided neural program synthesis," in [*International Conference on Learning Representations*], (2018).

[14] Chen, X., Song, D., and Tian, Y., "Latent execution for neural program synthesis beyond domain-specific languages," *Advances in Neural Information Processing Systems* **34**, 22196–22208 (2021).

[15] Clark, K., "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555* (2020).

[16] Kenton, J. D. M.-W. C. and Toutanova, L. K., "Bert: Pre-training of deep bidirectional transformers for language understanding," in [*Proceedings of naacL-HLT*], **1**, 2, Minneapolis, Minnesota (2019).

[17] Ellis, K., Nye, M., Pu, Y., Sosa, F., Tenenbaum, J., and Solar-Lezama, A., "Write, execute, assess: Program synthesis with a repl," *Advances in Neural Information Processing Systems* **32** (2019).

[18] Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., et al., "Codebert: A pre-trained model for programming and natural languages," *arXiv preprint arXiv:2002.08155* (2020).

[19] Fried, D., Aghajanyan, A., Lin, J., Wang, S., Wallace, E., Shi, F., Zhong, R., Yih, W.-t., Zettlemoyer, L., and Lewis, M., "Incoder: A generative model for code infilling and synthesis," *arXiv preprint arXiv:2204.05999* (2022).

[20] Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al., "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068* (2022).

[21] Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., Bahri, D., Schuster, T., et al., "Ul2: Unifying language learning paradigms," *arXiv preprint arXiv:2205.05131* (2022).

[22] Guo, D., Lu, S., Duan, N., Wang, Y., Zhou, M., and Yin, J., "Unixcoder: Unified cross-modal pre-training for code representation," *arXiv preprint arXiv:2203.03850* (2022).

[23] Böhme, M., Pham, V.-T., and Roychoudhury, A., "Coverage-based greybox fuzzing as markov chain," in [*Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*], 1032–1043 (2016).

[24] Böhme, M., Pham, V.-T., Nguyen, M.-D., and Roychoudhury, A., "Directed greybox fuzzing," in [*Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*], 2329–2344 (2017).

[25] Pak, B. S., "Hybrid fuzz testing: Discovering software bugs via fuzzing and symbolic execution," *School of Computer Science Carnegie Mellon University* (2012).

[26] Haller, I., Slowinska, A., Neugschwandtner, M., and Bos, H., "Dowsing for {Overflows}: A guided fuzzer to find buffer boundary violations," in [*22nd USENIX Security Symposium (USENIX Security 13)*], 49–64 (2013).

Boosting Static Bug Detection via Demand-Driven Points-to Analysis

Xuqing Yang[a]

[a]National University of Defense Technology, 109 Deya Road, Changsha, China

ABSTRACT

Static bug detection techniques have advanced significantly in identifying issues such as null pointer dereferences, memory leaks, and use-after-free vulnerabilities. However, existing methods that rely on pre-computed points-to analysis often struggle with scalability and precision, especially when handling complex pointer manipulations and deep call contexts. To address the scalability challenges of precise points-to analysis, we propose a fused approach for bug detection. Initially, we utilize an inexpensive Andersen points-to analysis to construct a sparse yet coarse program memory model. High-precision analysis is then applied selectively, only when necessary, reducing redundant computations and enhancing accuracy. This combination of coarse modeling and on-demand precision enables efficient and scalable bug detection. Experimental results across five real-world benchmarks show that our demand-driven flow-, context- and path-sensitive approach achieves up to a 4.55x speedup in analysis time compared to traditional eager flow-sensitive analysis. Notably, our approach successfully completes the analysis of large-scale programs such as sqlite3, which time out under traditional approaches. Additionally, our approach reduces false positives by over 70%, maintaining the detection of all true positive bugs. These results demonstrate the effectiveness of our approach in improving the efficiency and precision of static bug detection.

Keywords: static bug detection, demand-driven pointer analysis

1. INTRODUCTION

Recent advancements in the analysis of value flows have significantly contributed to static bug detection methods, including identifying issues such as null pointer dereference.[1,2] Despite these advancements, challenges persist in scaling value-flow analysis to industrial settings. Detecting bugs that involve intricate pointer manipulations, deep call hierarchies while maintaining low false positive rates, remains difficult—especially when analyzing large codebase within constrained timeframes.

Traditional techniques such as data-flow analysis and symbolic execution, exemplified by frameworks likeC-alysto[1] IFDS,[3] and Saturn,[4] propagate data-flow facts along control-flow paths to all program points. These dense analyses are known to suffer from performance bottlenecks.[5–7] For instance, recent studies[1] report that Saturn[4] and Calysto[1] require between 6 to 11 hours to verify a single property, such as null pointer dereference, in programs containing 685 KLoC (thousand lines of code).

Sparse value-flow analysis (SVFA)[5,7–9] addresses these performance issues by using sparse value-flow graphs (SVFGs) to track value flow through data dependencies, avoiding unnecessary propagation. These are termed "layered" approaches because they rely on an independent points-to analysis to first determine data dependencies. However, since highly precise points-to analyses struggle to scale to millions of lines of code,[10] layered SVFA techniques often sacrifice flow or context sensitivity in points-to analysis and avoid using SMT solvers to check path feasibility, as seen in Fastcheck[5] and Saber.[7]

In static bug detection scenarios, bugs are often localized to specific regions of the code, diminishing the necessity for a global analysis. Demand-driven program analyses concentrate on examining only the code sections relevant to a specific query, which aligns well with our requirements.

Further author information: (Send correspondence to Xuqing Yang)
Xuqing Yang: E-mail: xuqingyang22@nudt.edu.cn, Telephone: +86 19239628802

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 1351909 · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057629

In this work, we employ a fused approach to bug detection. First, we use an inexpensive Andersen points-to analysis to build a sparse but coarse program memory model. Then, we refine this memory model on-the-fly using a demand-driven points-to analysis.

Like many bug-finding techniques,[1,5,7,8] our approach is soundy.[11] However, it offers significantly greater scalability without sacrificing much precision or recall. Our evaluation demonstrates that our method is up to 4.55 times faster in detecting null pointer dereferences and reduces false positives by over 70% compared to traditional approaches.

In summary, this paper makes the following contributions:

- A demand-driven approach that significantly reduces the overhead of building a fully precise SVFG by refining only the necessary parts during analysis.
- A targeted refinement approach that applies high-precision points-to analysis selectively, focusing on critical points to accurately distinguish between true and false value flows.
- An experiment that evaluates our approach's scalability, precision, and recall, demonstrating its effectiveness in detecting real-world bugs.

2. OVERVIEW

The Static Value-Flow Graph (SVFG) functions as a crucial program representation in our demand-driven analysis. It is constructed based on the interprocedural memory SSA form, which captures the def-use relations for both top-level and address-taken variables. On the memory SSA, the def-use chains (value-flows) for address-taken variables are initially over-approximated using Andersen's fast yet coarse analysis. Our demand-driven pointer analysis then refines the imprecise points-to information and value-flows derived in this manner.

To illustrate the key ideas in our approach, we present a motivating example that contrasts traditional Andersen Points-to Analysis with a context-sensitive, demand-driven points-to analysis.

In the example code in Figure 1a, two pointers p and q are allocated by separate calls to the function *wrapper*. Each call allocates memory and returns a pointer to a newly allocated object. However, due to the context-insensitivity of Andersen Points-to Analysis, both calls to *wrapper* are modeled as allocating a single object o, causing p and q to be treated as aliases. This imprecision leads to an incorrect assumption that both p and q point to the same memory object, resulting in an inaccurate static value flow graph (SVFG) as shown in Figure 1b. This imprecision causes the edge between *N2* and *N3* to be incorrectly added, leading to false positives.

In contrast, a precise context-sensitive points-to analysis models each call separately, resulting in object o being associated with distinct contexts cs_1 and cs_2, respectively. This allows the SVFG to correctly represent the memory objects allocated by each call and prevents the creation of an incorrect edge between *N2* and *N3*, as shown in Figure 1c.

Our approach, shown in Figure 1d, takes a demand-driven way. Rather than constructing a fully precise SVFG from the outset, we only refine the value flow graph when necessary. For example, when analyzing *N2* as a source node in the SVFG, we encounter an erroneous edge leading to *N3* due to the context-insensitive assumption. By leveraging a high-precision, demand-driven points-to analysis at this point, we can dynamically verify whether p and q are truly aliases. Context-sensitive demand-driven analysis reveals that $pts(p) = \{o : cs_1\}$ and $pts(q) = \{o : cs_2\}$, indicating that p and q point to different objects. Consequently, we can invalidate the erroneous edge between *N2* and *N3*, thereby avoiding false positives.

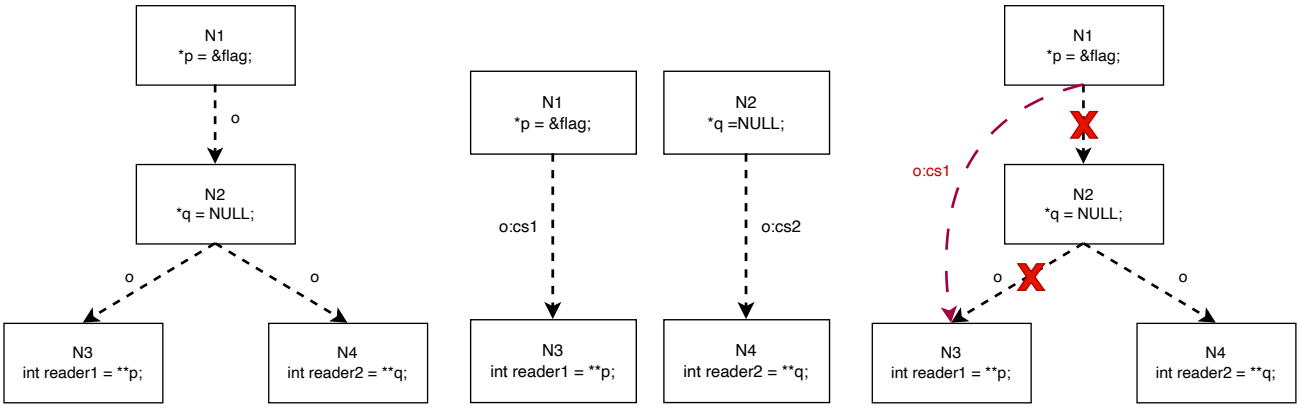
Additionally, when analyzing *N1* as a source node, our approach identifies that the path $N1 \rightarrow N2 \rightarrow N3$ is incorrect because *N2* is not related to *N1* or *N3*. In fact, *N1* and *N2* represent distinct and independent flows. Consequently, the correct path should directly connect *N1* to *N3*. By refining the SVFG on-demand, our approach reconstructs the direct edge $N1 \rightarrow N3$, facilitating accurate analysis for subsequent steps.

```c
int flag;
int **wrapper(int sz)
{
    return malloc(sz);  //o
}
void vulnerable()
{
    // Andersen PTA: pts(p)={o}, pts(q)={o}
    // CS PTA: pts(p)={o:cs1}, pts(q)={o:cs2}
    int **p = wrapper(8);   //cs1
    int **q = wrapper(8);   //cs2

    *p = &flag;
    **p = 1;
    *q = NULL;

    int reader1 = **p;
    int reader2 = **q;
}
```

(a) example code



(b) Coarse value-flow graph (c) Precise value-flow graph (d) Refined value-flow graph

Figure 1: Comparing the coarse value-flow graph, precise value-flow graph and our refined value-flow graph for bug detection. In (b)(c)(d), the label on an edge represents a memory object.

3. VALUE-FLOW REFINEMENT

As shown in Figure 2, the analysis begins with a fast but approximate preliminary step, which leverages an Andersen-style pointer analysis[12] to construct a sparse yet coarse Static Value-Flow Graph (SVFG). This initial phase efficiently provides a high-level representation of the program, serving as the basis for the subsequent steps.

In the next phase, a demand-driven pointer analysis is employed to refine the SVFG. By starting with the imprecise value-flow information, this analysis progressively enhances the accuracy of pointer relations and value flows, ultimately enabling precise bug detection.

The core of our algorithm involves iteratively processing a worklist that holds edges and associated context information as shown in Alg. 1. Our goal is to refine the static value-flow graph (SVFG) by addressing spurious aliases and erroneous value flows that arise due to imprecision in pointer analysis. The main focus is on store, load, and callsite nodes, where aliasing issues and incorrect value flows occur. The pseudocode for our algorithm is shown below.

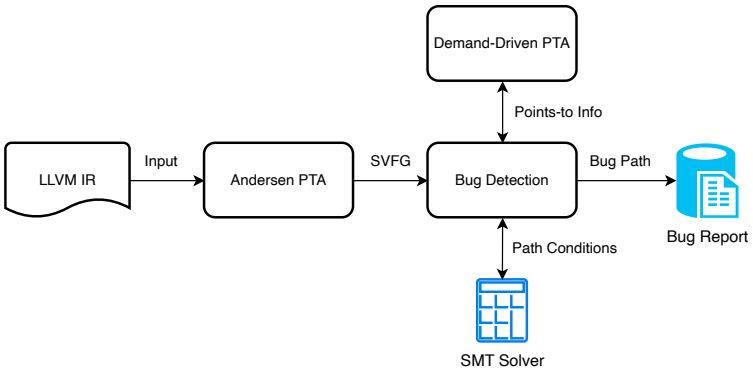


Figure 2: Workflow of our approach

Algorithm 1 Main body of analysis

```
1: while Worklist is not empty do
2:     k, objCxt, indPath ← SELECT(Worklist)
3:     if typeof(k) is STORE then
4:         processStore(k, objCxt, indPath)
5:     else if typeof(k) is LOAD then
6:         processLoad(k, objCxt, indPath)
7:     else if typeof(k) is CALL then
8:         processCall(k, objCxt, indPath)
9:     else
10:        Worklist.push(k.nextEdge, calleeObjCxt, {})
11:    end if
12: end while
```

We consider three key scenarios in our algorithm:

1. Store → ... → Load: This is the simplest case. As shown in Alg. 2, We use a high-precision, demand-driven points-to analysis to verify whether the pointers dereferenced by the store and load are truly aliases. If not, we invalidate the value flow between the store and load to eliminate false positives.

Algorithm 2 processLoad(k, $objCxt$, $indPath$)

```
1: loadPts ← pts(k)
2: objCxt ← objCxt & loadPts
3: if objCxt is not empty then
4:     Worklist.push(k.next, objCxt)
5: end if
```

2. Store → ... → Store: When encountering a second store along the value flow path, we check if a strong update is applicable. As shown in Alg. 3, Using the demand-driven points-to analysis, we analyze the points-to sets of both store instructions. If each store points to a single object and the object is a singleton, then the value at the first store must be overwritten at the second store. This prevents the propagation of incorrect value flows. If a strong update is not possible, we bypass the current store by constructing a *ReconnectEdge* between the successors of the two stores, effectively summarizing the path.

Algorithm 3 processStore(k, $objCxt$, $indPath$)

```
storePts ← pts(k)
if not strongUpdate(k, objCxt) then
    newEdge ← buildReconnectEdge(k.nextIntraEdge, indPath)
    indPath.add(newEdge)
end if
if objCxt is not empty then
    Worklist.push(k.next, objCxt, indPath)
end if
```

3. Store → ... → Call: For interprocedural flows, imprecision in traditional Andersen PTA can lead to incorrect points-to sets for function pointer parameters. This results in erroneous value flows entering the callee function. To detect such erroneous flows early and prevent redundant searches, As shown in Alg. 4, we compute precise points-to sets for the parameter pointers. The value flow of objects pointed to by parameters is considered to correctly propagate into the callee. For objects not pointed to by parameters, the erroneous edges due to imprecise pointer analysis are invalidated by establishing a *ReconnectEdge* between the store and the callsite return point.

Algorithm 4 processCall(k, $objCxt$, $indPath$)

```
CSCHIPts ← pts(k)
calleeObjCxt ← objCxt & CSCHIPts
callerObjCxt ← objCxt − CSCHIPts
if calleeObjCxt is not empty then
    Worklist.push(k.nextInterEdge, calleeObjCxt)
end if
if callerObjCxt is not empty then
    buildReconnectEdge(k.nextIntraEdge, indPath)
    Worklist.push(k.nextIntraEdge, calleeObjCxt)
end if
```

It is important to note that top-level variables in LLVM IR are already precisely modeled, so there is no need for further refinement in these cases.

In summary, our algorithm combines a lightweight but imprecise points-to analysis for efficiency with a precise demand-driven points-to analysis for targeted refinement. This enables us to handle complex value flows while avoiding the overhead of full-scale analysis.

4. EVALUATION

In this section, we evaluate our proposed approach, Demand-Driven Flow-, Context- and Path-Sensitive Analysis (DD FS-CS-PS), against a traditional Eager Flow-Sensitive Analysis (Eager FS) approach. Our evaluation focuses on two key aspects: performance and bug detection accuracy for null pointer dereference issues.

4.1 Experimental Setup

The evaluation compares two approaches:

- DD FS-CS-PS: Starts with a coarse SVFG and refines analysis using a demand-driven Flow-, Context- and Path-Sensitive (FS-CS-PS) points-to analysis.
- Eager FS: Constructs a relatively precise SVFG globally using a flow-sensitive points-to analysis and performs direct analysis.

For our comparison, we use five real-world benchmarks: lua(22 kloc), tig(36 kloc), coturn(40 kloc), nginx(165 kloc), and sqlite3(508 kloc).

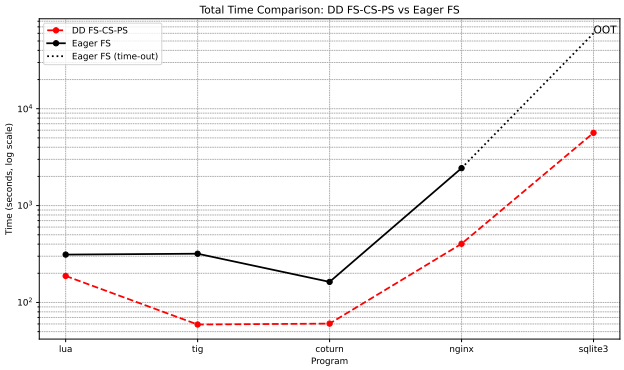
Environment. All experiments are conducted on a 64-bit machine with a 3.7GHz Intel Xeon 8-core CPU and 32 GB memory. The reported data represents the medians of three runs.

4.2 Performance Analysis

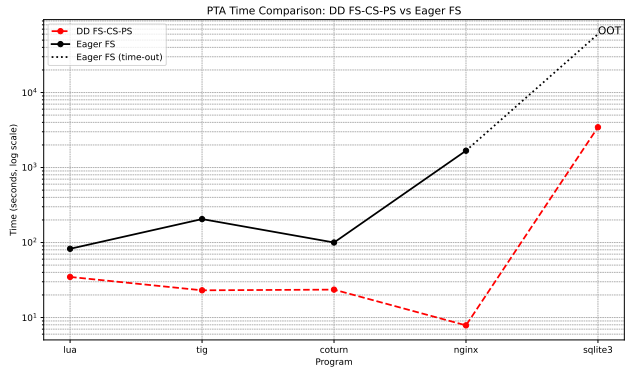
The performance comparison between DD FS-CS-PS and Eager FS is shown in Table 1. The total analysis time and points-to analysis (PTA) time are measured in seconds. We set the timeout to 7200 seconds. Additionally, we provide a visual comparison of the results in Figure 3, illustrating the total analysis time and PTA time for null pointer dereference detection across the five benchmark programs.

Table 1: Comparison of DD FS-CS-PS and Eager FS

Program	DD FS-CS-PS		Eager FS	
	total time	pta time	total time	pta time
lua	187.814	34.775	311.985	82.477
tig	59.1881	23.067	318.562	205.492
coturn	60.4932	23.558	163.078	100.069
nginx	403.388	7.863	2434.256	1675.802
sqlite3	5636.192	3447.915	OOT	OOT



(a) Total time comparison



(b) PTA time comparison

Figure 3: Comparing the time of DD FS-CS-PS and Eager FS for NPD detection. We present the results of the 5 programs analyzed within the time budget.

The results demonstrate that our demand-driven approach achieves significant speedup compared to the traditional Eager FS approach. For smaller benchmarks like lua, tig and nginx, DD FS-CS-PS achieves up to

5x speedup in total analysis time compared to Eager FS approach. For larger benchmarks like sqlite3, DD FS-CS-PS can complete the analysis while Eager FS times out due to excessive computation time.

The speedup in DD FS-CS-PS is attributed to two factors:

- Demand-Driven Refinement: By refining the SVFG only when necessary, we avoid the overhead of constructing and analyzing a fully precise SVFG upfront.
- Precise FS-CS-PS Analysis: Our FS-CS-PS points-to analysis achieves higher precision than the traditional flow-sensitive analysis. The increased precision leads to more strong updates, thereby avoiding redundant analysis and enhancing efficiency.

4.3 Bug Detection Accuracy

The comparison of bug reports is shown in Table 2, which lists the number of false positives (FP) and the total number of bug reports generated by each approach.

Table 2: Bug Report Comparison between DD FS-CS-PS and Eager FS

Program	DD FS-CS-PS		Eager FS	
	#FP	#Rep	#FP	#Rep
lua	0	1	6	7
tig	2	5	9	12
coturn	0	3	12	15
nginx	3	4	33	34
sqlite3	4	8	\	\

Our approach significantly reduces the number of false positives compared to the Eager FS. For example, in the lua benchmark, DD FS-CS-PS reports only 2 false positives, while Eager FS reports 19. The reduction in false positives is due to the increased precision of our context-sensitive analysis, which more accurately captures the points-to relationships. Additionally, DD FS-CS-PS successfully identifies all the true positives reported by Eager FS, demonstrating the reliability of our approach.

The precision of the demand-driven FS-CS-PS points-to analysis ensures that:

- False Positives are Minimized: By precisely capturing context-sensitive points-to relationships, our approach eliminates spurious value flows that would otherwise lead to false reports.
- Reliability is Maintained: Our approach detects all true positive null pointer dereference bugs identified by the traditional Eager FS analysis, proving that our higher precision does not compromise accuracy.

5. RELATED WORK

Existing methods for static bug detection can generally be divided into two primary categories based on how they track the flow of values: those that utilize data dependencies and those that focus on control flows. Notably, all current techniques that leverage data dependencies depend on a pre-computed points-to analysis to establish these dependencies, following what is known as a layered design. Due to the high computational cost of precise points-to analysis,[10] these techniques commonly use flow-insensitive analyses to prevent excessive overhead during pre-computation.[5–9,13,14] In contrast, our proposed "on-the-fly refine" strategy employs a lightweight pre-computed points-to analysis and only enhances precision when it is necessary.

On the other hand, approaches that rely on abstraction, such as SLAM,[15] BLAST,[16] and SATABS,[17] use abstraction refinement to enhance scalability. However, as the level of abstraction is refined, scalability often suffers. Likewise, CBMC[18] faces challenges with scalability because it supplies constraints to an SAT solver indiscriminately, irrespective of their relevance. Cheetah,[19] which builds on the IFDS framework,[3] bears some similarity to our approach by performing local analysis first before progressively broadening the scope to the entire codebase. Compositional techniques like Magic,[20] Saturn,[1,1,4] Compass,[21] and Blitz[22] also share similarities with

our method in terms of modular analysis. Nonetheless, these methods have proven to be inefficient in detecting bugs that can be characterized by value-flow paths, as they redundantly track data-flow facts through control flow paths.[5,7]

6. CONCLUSION

In this paper, we presented a fused approach for static bug detection that addresses the scalability challenges of precise points-to analysis. By employing an inexpensive Andersen points-to analysis to build a sparse yet coarse program memory model, and selectively refining the analysis with high-precision techniques only when necessary, our approach effectively reduces redundant computations while maintaining accuracy. This strategy of combining coarse modeling with on-demand precision allows for both efficient and scalable bug detection.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] Babic, D. and Hu, A. J., "Calysto: scalable and precise extended static checking," in [*Proceedings of the 30th International Conference on Software Engineering*], *ICSE '08*, 211–220, Association for Computing Machinery, New York, NY, USA (2008).

[2] Hovemeyer, D. and Pugh, W. W., "Finding more null pointer bugs, but not too many," in [*Proceedings of the 7th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering, PASTE'07, San Diego, California, USA, June 13-14, 2007*], Das, M. and Grossman, D., eds., 9–14, ACM (2007).

[3] Reps, T., Horwitz, S., and Sagiv, M., "Precise interprocedural dataflow analysis via graph reachability," in [*Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*], *POPL '95*, 49–61, Association for Computing Machinery, New York, NY, USA (1995).

[4] Xie, Y. and Aiken, A., "Scalable error detection using boolean satisfiability," in [*Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*], 351–363 (2005).

[5] Cherem, S., Princehouse, L., and Rugina, R., "Practical memory leak detection using guarded value-flow analysis," in [*Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation*], 480–491 (2007).

[6] Oh, H., Heo, K., Lee, W., Lee, W., and Yi, K., "Design and implementation of sparse global analyses for c-like languages," in [*Proceedings of the 33rd ACM SIGPLAN conference on Programming Language Design and Implementation*], 229–238 (2012).

[7] Sui, Y., Ye, D., and Xue, J., "Detecting memory leaks statically with full-sparse value-flow analysis," *IEEE Transactions on Software Engineering* **40**(2), 107–122 (2014).

[8] Livshits, V. B. and Lam, M. S., "Tracking pointers with path and context sensitivity for bug detection in c programs," in [*Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*], 317–326 (2003).

[9] Snelting, G., Robschink, T., and Krinke, J., "Efficient path conditions in dependence graphs for software safety analysis," *ACM Transactions on Software Engineering and Methodology (TOSEM)* **15**(4), 410–457 (2006).

[10] Hind, M., "Pointer analysis: Haven't we solved this problem yet?," in [*Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering*], 54–61 (2001).

[11] Livshits, B., Sridharan, M., Smaragdakis, Y., Lhoták, O., Amaral, J. N., Chang, B.-Y. E., Guyer, S. Z., Khedker, U. P., Møller, A., and Vardoulakis, D., "In defense of soundiness: A manifesto," *Communications of the ACM* **58**(2), 44–46 (2015).

[12] Andersen, L. O., "Program analysis and specialization for the c programming language," (1994).

[13] Das, M., Lerner, S., and Seigle, M., "Esp: Path-sensitive program verification in polynomial time," in [*Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation*], 57–68 (2002).

[14] Dor, N., Adams, S., Das, M., and Yang, Z., "Software validation via scalable path-sensitive value flow analysis," *ACM SIGSOFT Software Engineering Notes* **29**(4), 12–22 (2004).

[15] Ball, T. and Rajamani, S. K., "The slam project: Debugging system software via static analysis," in [*Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*], 1–3 (2002).

[16] Henzinger, T. A., Jhala, R., Majumdar, R., and Sutre, G., "Lazy abstraction," in [*Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*], 58–70 (2002).

[17] Clarke, E., Kroening, D., Sharygina, N., and Yorav, K., "Predicate abstraction of ansi-c programs using sat," *Formal Methods in System Design* **25**, 105–127 (2004).

[18] Clarke, E., Kroening, D., and Yorav, K., "Behavioral consistency of c and verilog programs using bounded model checking," in [*Proceedings of the 40th annual Design Automation Conference*], 368–371 (2003).

[19] Do, L. N. Q., Ali, K., Livshits, B., Bodden, E., Smith, J., and Murphy-Hill, E., "Just-in-time static analysis," in [*Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*], 307–317 (2017).

[20] Chaki, S., Clarke, E. M., Groce, A., Jha, S., and Veith, H., "Modular verification of software components in c," *IEEE Transactions on Software Engineering* **30**(6), 388–402 (2004).

[21] Dillig, I., Dillig, T., Aiken, A., and Sagiv, M., "Precise and compact modular procedure summaries for heap manipulating programs," *ACM SIGPLAN Notices* **46**(6), 567–577 (2011).

[22] Cho, C. Y., D'Silva, V., and Song, D., "Blitz: Compositional bounded model checking for real-world programs," in [*2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*], 136–146, IEEE (2013).

Client dependability evaluation in Federated Learning framework

Kexiu Han[a], Guoyue Zhang[a], Liangbin Yang*[a], Jing Bai[a]
[a]University of International Relations, HaiDian, BeiJing, CHN 100080

ABSTRACT

Federated learning (FL) is a widely adopted distributed machine learning paradigm, individual clients train local models by using their private datasets and then send model updates to a central server. While its decentralized training process can protect data privacy, it is vulnerable to attacks such as model poisoning attack and backdoor attack. The effect of malicious clients can be mitigated by applying robust FL methods. However, most existing solutions ignored the client dependability. This paper explores a method for quantitatively assessing the client dependability in FL framework. Firstly, based on semi-Markov process (SMP), we build a multi-dimensional evaluation model for understanding how the client's behaviors under attack and its recovery behaviors affect the client dependability. Then, we deduce the formulas of calculating the availability, security risk and reliability in order to analyze the quantitative relationship between different factors and the client dependability from these three perspectives. Furthermore, we perform numerical analysis to investigate how different system parameters impact the client dependability.

Keywords: Availability, Federated Learning, Reliability, Semi-Markov process, Security Risk

1. INTRODUCTION

Federated learning (FL) is a widely adopted distributed machine learning paradigm that emphasizes enhanced privacy protections [1][2][3]. In FL framework, individual clients train local models by using their private datasets. Clients do not share raw data and only send model updates to a central server, where these updates are aggregated to improve the global model. This approach not only protects data privacy, but also enables collaborative learning across diverse client environments [4][5].

Despite the rapid growth of FL, the paradigm is not without its challenges, particularly in terms of security. Due to its decentralized architecture, it is vulnerable to attacks. For instance, malicious clients can corrupt the global model by changing local data or model parameters. These attacks usually poison multiple clients within the framework. If no measures are taken, the system will become increasingly inefficient [6][7].

More and more researchers are focusing on the defense schemes to effectively mitigate these threats. Most existing studies focused on identifying the differences between the training results of malicious clients and those of benign clients. By finding these differences, they sought to implement corrective measures that enhance the robustness of the FL framework. While many of these studies have demonstrated promising results in mitigating the impacts of attacks, they ignored the impact of client dependability on defense schemes, which is crucial to improving the trustworthiness of FL framework. Therefore, there is an urgent need to develop solutions that can quantitatively analyze the client dependability in FL framework [8].

In this paper, we propose a novel method to evaluate the client dependability in FL framework based on semi-Markov process (SMP). To address the above issue, a multi-dimensional SMP model is conducted to describe the behaviors of clients from being attacked to recovering. To the best of our knowledge, it is the first time to evaluate the client dependability in FL framework under the condition that the compromised time, failure time and rejuvenation time follow general distributions. Our contributions can be summarized as follows:

- We build a multi-dimensional SMP model to quantitatively analyze the client dependability. Specifically, the proposed model can capture the dynamic interaction of the client's behaviors under attack and its recovery behaviors.

- We propose an approach to evaluate the client dependability in terms of availability, security risk and reliability. Specifically, the proposed formulas can analyze the impact of attack time, failure time and rejuvenation time on client dependability.

* Corresponding Author; ylb@uir.cn; phone +86 13693622645

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190A · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058096

• We perform extensive experiments to thoroughly verify the effectiveness and practicality of the proposed method under different system parameters. The experimental results provide insights to enhance the trustworthiness of FL framework.

The reminder of this paper is organized as follows. In Section II, we review relevant representative papers from three aspects: FL framework, defense schemes and dependability evaluation. Section III describes the proposed SMP model for the client dependability evaluation. In Section IV, we give the experiment settings and results. Finally, Section V concludes this paper and discusses future work.

2. RELATED WORKS

This section reviews relevant representative papers from three aspects: FL framework, defense schemes, dependability evaluation.

2.1 Federated Learning Framework

In recent years, FL has attracted growing attention. Researchers have proposed many novel FL frameworks. For example, Chai *et al*. [9] proposed a Tier-based FL framework named *TiFL*. The tiered design and adaptive client selection algorithm can mitigate data heterogeneity, and control the training throughput and accuracy. This framework is suitable for smart city, medical analysis systems, etc. Li *et al*. [10] presented a privacy-preserving FL framework called *Chain-PPFL*. For each iteration, clients form a chain with a flexible and distributed way. Due to its simple architecture, it is suitable for practical scenarios with good communication environment, such as intelligent Internet of Things (IoT). In addition, Zaccone *et al*. [11] presented a FL framework via sequential *superclient* training called *FedSeq*. It provides a solution to lighten the affect of non-independent and identically distributed data by grouping clients with different data distributions to form *superclients*.

2.2 Defense Schemes

Cao *et al*. [12] proposed a Byzantine-robust FL method named *FLTrust*. It utilizes the root dataset to train a model on the server, and compares this model with the client model to detect and delete malicious clients. Zhang *et al*. [13] presented a defense method called *FLDetector* which detects and deletes malicious clients by comparing actual gradient with predicted gradients. Sun *et al*. [14] developed a method of calculating the contribution of clients, and adjusted the weights of clients according to their contributions when aggregating the global model. Thein *et al*. [15] designed a personalized federated learning solution named *PFL*, which analyzes the cosine similarity between the global model and local updates to detect poisoned clients. This mechanism can relieve the impact of IoT data heterogeneity and handle poisoning attacks. However, these methods all ignored the impact of the client dependability, which is crucial to improving the trustworthiness of FL framework.

2.3 Dependability Evaluation

Model-based dependability analysis has been applied in many fields. For example, Yang *et al*. [16] proposed the continuous time Markov chain (CTMC) model to analyze the availability and reliability of a standby repairable system. Araujo *et al*. [17] proposed the availability and reliability evaluation model for mobile cloud architectures based on the CTMC. Those models are utilized to compare distinct strategies. Oliveira *et al*. [18] analyzed the system dependability by combining the CTMC model and reliability block diagram (RBD). They used the CTMC model and RBD to capture the component behaviors and the relationships between these behaviors, respectively. This study estimates poultry weight with hierarchical models such as Markov chain and equation. Gao *et al*. [19] conducted reliability analysis for the repairable redundant system based CTMC. Oszczypała *et al*. [20] developed the CTMC model that is used for evaluating and increasing the availability of parallel k-out-of-n system. However, these studies assumed that all event occurrence times follow exponential distribution. In contrast to the existing studies, our model relaxes this limitation, which can be better applied to practical systems. In addition, we describe the behaviors of the client being attacked in detail.

3. SYSTEM DESCRIPTION AND DEPENDABILITY EVALUATION MODEL

In this section, we first introduce the FL framework considered in this paper, then propose the dependability evaluation model and finally derive the formulas of calculating the dependability measures.

3.1 System Description

In the FedSeq framework, there are *n* clusters, each consisting of *m* clients. In each iteration, the clients in a cluster are trained sequentially. Specifically, the first client first trains the model based on the original global model sent by the server and its local data. Then, the trained model will be sent to the next client to continue training. Each iteration ends until the last client finishes its training and updates the model to the server.

Initially, the client works robustly. However, the client faces various security vulnerabilities. If the monitoring system can detect the occurrence of any kind of penetration into the system defense mechanisms, the necessary measures are taken to bring the client back to robustness. However, if no measures are taken, the client is successfully exploited by the attackers. If the intrusion detection system is able to successfully identify the exploitation of the client by the attacker, the appropriate recovery measures will be taken. Otherwise, the client fails. After the client completes the repair and restarting, it returns to a perfect working state. When a client is exposed to security vulnerabilities or exploited by an attacker, other clients in the cluster may also be exposed to security vulnerabilities. Once more than two clients in a cluster are exposed to security vulnerabilities, all clients in a cluster will be restarted. We assume that the time at which the client is exposed to security vulnerabilities follows exponential distribution and the remaining event occurrence times follow general distribution.

3.2 SMP Model

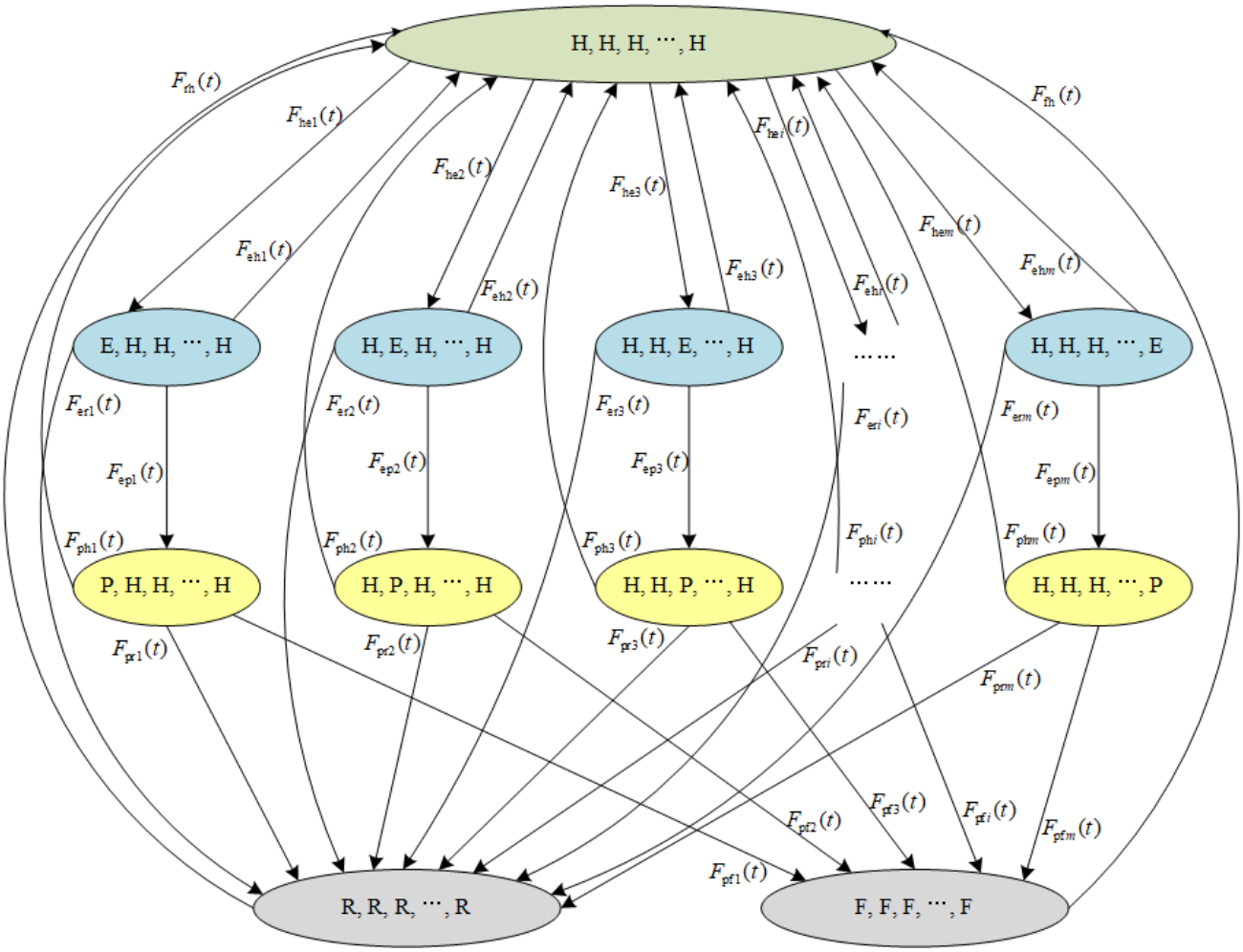


Fig. 1 SMP model

Based on the system description, the client state can be denoted by a m-tuple index ($S_1,S_2,S_3,S_4,...,S_m$). Here, S_m is the state of the m^{th} client in a cluster, which has five types: Healthy, Exposed, Exploited Recovery and Failed.

- State H (Healthy): This is the stage where the client is working robustly. Recovery and repair measures bring the client back to this state.
- State E (Exposed): In this state, the client is exposed to security vulnerabilities, but the client can still continue to work.
- State P (Exploited): The client enters this state after it has been successfully exploited by an attacker. The client are still available in this state.
- State R (Rejuvenation): In this state, the client is restarted and cannot work properly.
- State F (Failed): In this state, the client fails because appropriate recovery measures are not taken in time.

There are a total of 5^m system states, 5^m-2m-3 of which are meaningless system states. For example, $(E_1,E_2,E_3,E_4,...,E_m)$ denotes that all clients in a cluster are exposed to security vulnerabilities. However, once more than two clients in a cluster are exposed to security vulnerabilities, all clients in a cluster will be restarted. Therefore, this state is meaningless.

Fig. 1 shows the SMP model for dependability, which can describe the behaviors of all clients in a cluster. In this figure, the states marked in blue indicate that the system is at low security risk, the states marked yellow indicate that the system is at high security risk, the states marked grey indicate that the system is unavailable and the state marked green indicates that the system can operate robustly. denotes the distribution function followed by the time it takes a client to move from one state to another.

3.3 Dependability Analysis

In order to evaluate the client dependability, we first solve the one-step transition probability matrix $\mathbf{P}$ of the embedded discrete time Markov chain (EDTMC) of SMP by constructing a kernel matrix $\mathbf{K}(t)$ as shown in Equation (1).

$$\begin{bmatrix}
0 & k_{\text{f0f1}} & 0 & k_{\text{f0f3}} & \cdots & k_{\text{f0f}i} & 0 & \cdots & k_{\text{f0f}(2m-1)} & 0 & 0 & 0 \\
k_{\text{f1f0}} & 0 & k_{\text{f1f2}} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & k_{\text{f1f}(2m+1)} & 0 \\
k_{\text{f2f0}} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & k_{\text{f2f}(2m+1)} & k_{\text{f2f}(2m+2)} \\
k_{\text{f3f0}} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & k_{\text{f3f}(2m+1)} & 0 \\
\cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \cdots \\
k_{\text{fif0}} & 0 & 0 & 0 & \cdots & 0 & k_{\text{f}(i+1)\text{f}0} & \cdots & 0 & 0 & k_{\text{fif}(2m+1)} & 0 \\
k_{\text{f}(i+1)\text{f0}} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & k_{\text{f}(i+1)\text{f}(2m+1)} & k_{\text{f}(i+1)\text{f}(2m+2)} \\
k_{\text{f}(i+2)\text{f0}} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & k_{\text{f}(i+2)\text{f}(2m+1)} & 0 \\
\cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \cdots \\
k_{\text{f2}m\text{f0}} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & k_{\text{f2}m\text{f}(2m+1)} & k_{\text{f2}m\text{f}(2m+2)} \\
k_{\text{f}(2m+1)\text{f0}} & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \cdots \\
k_{\text{f}(2m+2)\text{f0}} & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0
\end{bmatrix} \tag{1}$$

where the non-null elements in $\mathbf{K}(t)$ can be calculated by using Equations (2)-(9).

$$k_{0i}(t)=\int_0^t \sum_1^{\lfloor i/2 \rfloor}(1-F_{e(2j-1)}(t))\sum_{\lfloor i/2 \rfloor+2}^{\lceil (2m+3)/2 \rceil-2}(1-F_{e(2j-1)}(t))dF_{ei}(t) \tag{2}$$

$$k_{i0}(t)=\int_0^t (1-F_{\text{epi}}(t))(1-F_{\text{eri}}(t))dF_{\text{ehi}}(t) \tag{3}$$

$$k_{i(i+1)}(t)=\int_0^t (1-F_{\text{ehi}}(t))(1-F_{\text{eri}}(t))dF_{\text{epi}}(t) \tag{4}$$

$$k_{i(2m+1)}(t)=\int_0^t(1-F_{ehi}(t))(1-F_{epi}(t))dF_{eri}(t) \quad (5)$$

$$k_{(i+1)0}(t)=\int_0^t(1-F_{pri}(t))(1-F_{pfi}(t))dF_{phi}(t) \quad (6)$$

$$k_{(i+1)(2m+1)}(t)=\int_0^t(1-F_{phi}(t))(1-F_{pfi}(t))dF_{pri}(t) \quad (7)$$

$$k_{(i+1)(2m+2)}(t)=\int_0^t(1-F_{phi}(t))(1-F_{pri}(t))dF_{pfi}(t) \quad (8)$$

$$k_{(2m+0)1}(t)=\int_0^t(1-F_{phi}(t))(1-F_{pri}(t))dF_{pfi}(t) \quad (9)$$

By solving $V=VK(\infty)$, we can obtain the steady-state probabilities of EDTMC. We then use Equations (10)-(14) to calculate the mean sojourn time for each system state.

$$h_0=\int_0^\infty\sum_1^{\lceil(2m+3)/2\rceil-2}(1-F_{e(2j-1)}(t))dt \quad (10)$$

$$h_i=\int_0^\infty(1-F_{epi}(t))(1-F_{eri}(t))(1-F_{ehi}(t))dt \quad (11)$$

$$h_{i+1}=\int_0^t(1-F_{ehi}(t))(1-F_{eri}(t))(1-F_{efi}(t))dt \quad (12)$$

$$h_{2m+1}=\int_0^\infty(1-F_{rhi}(t))dt \quad (13)$$

$$h_{2m+2}=\int_0^\infty(1-F_{fhi}(t))dt \quad (14)$$

By solving $\pi_{FLp}=v_ph_p/\sum_0^{2m+2}v_ih_i$, the steady-state probability of system state *p* can be obtained.

(1) Availability Analysis

In our model, states 0, 2m+1 and 2m+2 are all available states. Therefore, the steady-state availability can be derived by using Equation (15).

$$A=\pi_{FL0}+\pi_{FL(2m+1)}+\pi_{FL(2m+2)} \quad (15)$$

(2) Security Risk Analysis

Low security risk indicates the probability that the clients in a cluster are exposed to security vulnerabilities, which can be calculated by Equation (16). High security risk indicates the probability that the clients in a cluster are exploited by an attacker, which can be calculated by Equation (17).

$$S=\sum_1^{\lceil(2m+3)/2\rceil-2}\pi_{FL(2i-1)} \quad (16)$$

$$S=\sum_1^{\lceil(2m+3)/2\rceil-2}\pi_{FL(2i)} \quad (17)$$

(3) Reliability Analysis

We assess the reliability by calculating the mean time to failure (MTTF). In order to obtain MTTF, we consider that no recovery measures are applied when the client is in an unavailable state. Therefore, MTTF can be derived by using Equation (18).

$$\text{MTTF}=\sum_0^{2m}V_i^*h_i \quad (18)$$

where V_i^* is the expected number of visits to *i* until the client is unavailability and can be obtained by Equation (19). h_i can be obtained by Equations (10)-(14)

$$V_i^*=\alpha_i+\sum_0^{2m}V_j^*p_{ji} \quad (18)$$

where α denotes the initial probability of each state.

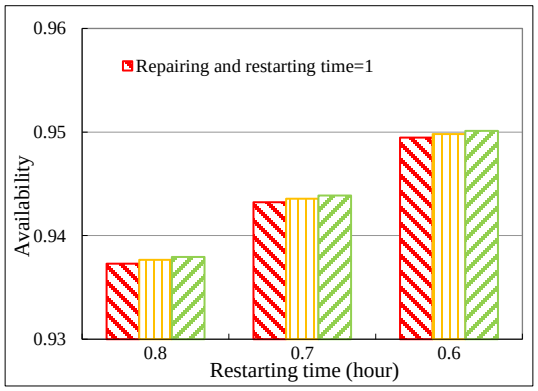
4. EXPERIMENT RESULTS

In this section, we perform extensive experiments to thoroughly verify the effectiveness and practicality of the proposed method under different system parameters.

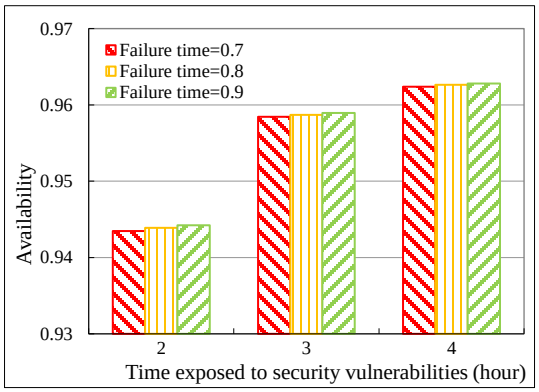
4.1 Experimental Configuration

In experiments, the time exposed to security vulnerabilities is assumed to follow exponential distribution with a mean of 2 to 4 hours. The failure time is assumed to follow Hypoexponential distribution with a mean of 0.5 to 4 hours. The time at which the client is exploited by an attacker is assumed to follow Weibull distribution with a mean of 2 to 4 hours. The rejuvenation time is assumed to follow exponential distribution with a mean of 0.1 to 1 hours. Numerical experiments are conducted on MAPLE.

4.2 Experimental Results



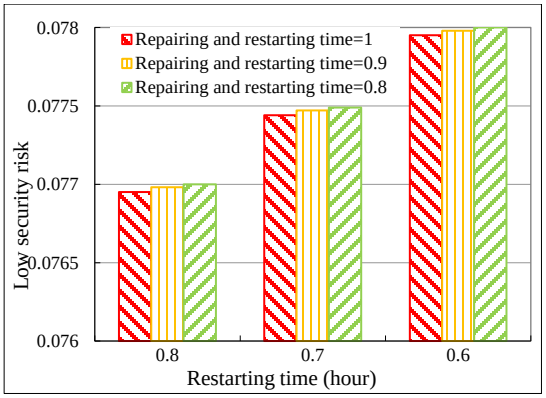
(a) Restarting time and repairing and restarting time



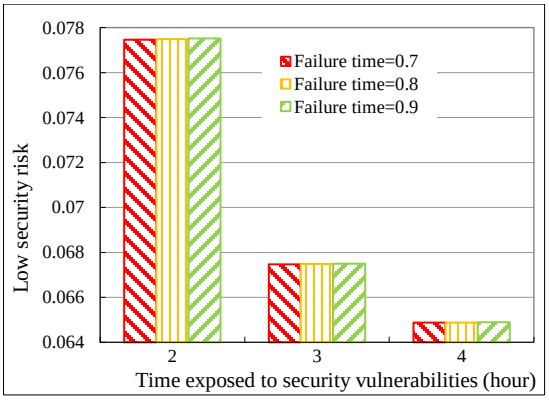
(b) Time exposed to security vulnerabilities and failure time

Fig.2. The availability under different parameters

Fig.2 illustrates the availability under different parameters. We can obtain that the availability increases as the repairing and restarting time and restarting time decreases. We also can see that the longer the failure time and time exposed to security vulnerabilities, the higher the availability. This is because as these parameters increase, the longer the system is in available states.



(a) Restarting time and repairing and restarting time

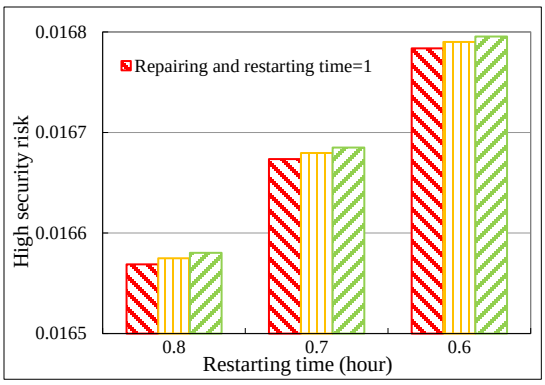


(b) Time exposed to security vulnerabilities and failure time

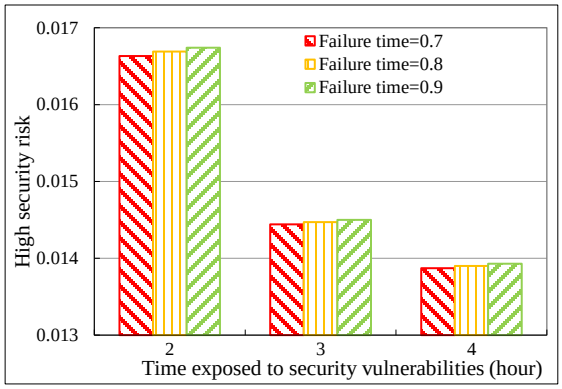
Fig.3. The low security risk under different parameters

Fig.3 illustrates the low security risk under different parameters. It can be seen that with the increase of repairing and restarting time, restarting time, and time exposed to security vulnerabilities, the low security risk decreases. In addition, the increase of failure time leads to an increase of low security risk.

Fig.4 illustrates the high security risk under different parameters. As the repairing and restarting time, restarting time, and time exposed to security vulnerabilities decrease, the high security risk increases. In addition, the increase of failure time leads to an increase of high security risk.

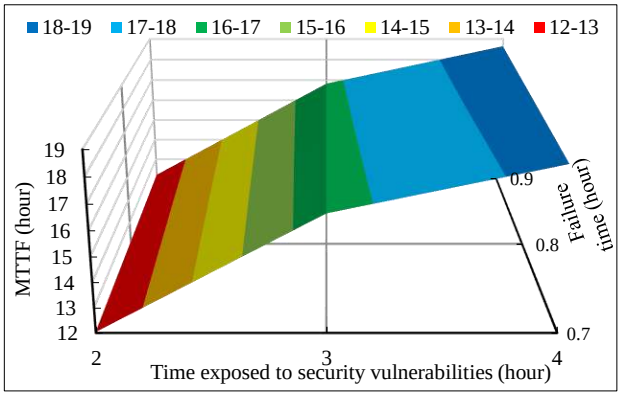


(a) Restarting time and repairing and restarting time

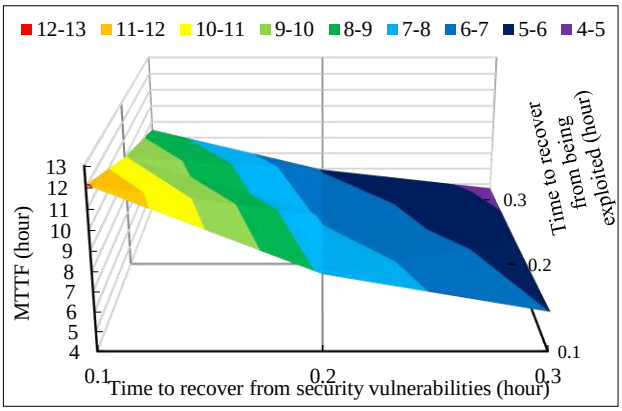


(b) Time exposed to security vulnerabilities and failure time

Fig.4. The high security risk under different parameters



(a) Time exposed to security vulnerabilities and failure time



(b) Time to recover from security vulnerabilities and Time to recover from being exploited

Fig.5. MTTF under different parameters

Fig.5 illustrates MTTF under different parameters. As the time exposed to security vulnerabilities and failure time, MTTF increases. As the time to recover from security vulnerabilities and time to recover from being exploited decrease, MTTF increases.

5. CONCLUSION

This paper proposes a multi-dimensional model for the client dependability evaluation in FL framework. Then, the formulas of calculating the availability, security risk and reliability are deduced. Furthermore, numerical analysis are conducted under different system parameters. The model presented in this paper not only deepens the understanding of the dynamic behaviors of clients in FL framework, but also provides insights to improve the trustworthiness of FL framework. In the future, we will study more potential behaviors of clients to help design better defense schemes.

ACKNOWLEDGEMENTS

This work was supported by Fundamental Research Funds for the Central Universities, University of International Relations (3262024T01).

REFERENCES

[1] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, Yuan Gao, "A survey on federated learning," Knowl. Based Syst. 216, 106775 (15 March 2021). https://doi.org/10.1016/j.knosys.2021.106775

[2] Abbas Yazdinejad, Ali Dehghantanha, Hadis Karimipour, Gautam Srivastava, Reza M. Parizi, "A Robust Privacy-Preserving Federated Learning Model Against Model Poisoning Attacks," IEEE Trans. Inf. Forensics Secur. 19, 6693-6708 (27 June 2024). https://doi.org/10.1109/TIFS.2024.3420126

[3] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, Wensheng Zhang, "A survey on federated learning: challenges and applications," Int. J. Mach. Learn. Cybern. 14(2), 513-535 (11 November 2022). https://doi.org/10.1007/s13042-022-01647-y

[4] Pian Qi, Diletta Chiaro, Antonella Guzzo, Michele Ianni, Giancarlo Fortino, Francesco Piccialli, "Model aggregation techniques in federated learning: A comprehensive survey," Future Gener. Comput. Syst. 150, 272-293 (January 2024). https://doi.org/10.1016/j.future.2023.09.008

[5] Bingyan Liu, Nuoyan Lv, Yuanchun Guo, Yawen Li, "Recent advances on federated learning: A systematic survey," Neurocomputing 597, 128019 (7 September 2024). https://doi.org/10.1016/j.neucom.2024.128019

[6] Pengrui Liu, Xiangrui Xu, Wei Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," Cybersecur. 5(1), 4 (02 February 2022). https://doi.org/10.1186/s42400-021-00105-6

[7] Nuria Rodríguez Barroso, Daniel Jiménez-López, María Victoria Luzón, Francisco Herrera, Eugenio Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," Inf. Fusion 90(C), 148-173 (01 February 2023). https://doi.org/10.1016/j.inffus.2022.09.011

[8] Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, Ning Xie, Gérôme Bovet, Gregorio Martínez Pérez, Burkhard Stiller, "FederatedTrust: A solution for trustworthy federated learning," Future Gener. Comput. Syst. 152(C), 83-98 (04 March 2024). https://doi.org/10.1016/j.future.2023.10.013

[9] Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, Yue Cheng, "TiFL: A Tier-based Federated Learning System," HPDC 2020, 125-136(23 June 2020). https://doi.org/10.1145/3369583.3392686

[10] Yong Li, Yipeng Zhou, Alireza Jolfaei, Dongjin Yu, Gaochao Xu, Xi Zheng, "Privacy-Preserving Federated Learning Framework Based on Chained Secure Multi-party Computing," IEEE Internet of Things Journal. 8(8), 6178-6186(15 April 2021). https://doi.org/10.1109/JIOT.2020.3022911

[11] Riccardo Zaccone, Andrea Rizzardi, Debora Caldarola, Marco Ciccone, Barbara Caputo, "Speeding up Heterogeneous Federated Learning with Sequentially Trained Superclients," ICPR 2022, 3376-3382(2022). https://doi.org/10.1109/ICPR56361.2022.9956084

[12] Xiaoyu Cao, Minghong Fang, Jia Liu, Neil Zhenqiang Gong, "FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping," NDSS 2021. https://doi.org/10.48550/arXiv.2012.13995

[13] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, Neil Zhenqiang Gong, "FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients," KDD 2022, 2545-2555(14 August 2022). https://doi.org/10.1145/3534678.3539231

[14] Qiheng Sun, Xiang Li, Jiayao Zhang, Li Xiong, Weiran Liu, Jinfei Liu, Zhan Qin, Kui Ren, "ShapleyFL: Robust Federated Learning Based on Shapley Value," KDD 2023, 2096-2108(04 August 2023). https://doi.org/10.1145/3580305.3599500

[15] Thin Tharaphe Thein, Yoshiaki Shiraishi, Masakatu Morii, "Personalized federated learning-based intrusion detection system: Poisoning attack and defense," Future Gener. Comput. Syst. 153, 182-192 (April 2024). https://doi.org/10.1016/j.future.2023.10.005

[16] Dong-Yuh Yang, Chia-Huang Wu, "Evaluation of the availability and reliability of a standby repairable system incorporating imperfect switchovers and working breakdowns," Reliab. Eng. Syst. Saf. 207, 107366 (March 2021). https://doi.org/10.1016/j.ress.2020.107366

[17] Jean Araujo, Danilo Oliveira, Rubens Matos, Gabriel Alves, Paulo Maciel, "Availability and Reliability Modeling of Mobile Cloud Architectures," IEEE Trans. Ind. Informatic. 1-13(10 March 2023). https://doi.org/10.1109/TII.2023.3254547

[18] Felipe Oliveira, Paulo Pereira, Jamilson Dantas, Jean Araujo, Paulo Maciel, "Dependability Evaluation of a Smart Poultry House: Addressing Availability Issues Through the Edge, Fog, and Cloud Computing," IEEE Trans. Ind. Informatics 20(2), 1304-1312 (12 May 2023). https://doi.org/10.1109/TII.2023.3275656

[19] Shan Gao, Jinting Wang, Jie Zhang, “Reliability analysis of a redundant series system with common cause failures and delayed vacation,” Reliab. Eng. Syst. Saf. 239, 109467 (November 2023). https://doi.org/10.1016/j.ress.2023.109467
[20] Mateusz Oszczypala, Jakub Konwerski, Jaroslaw Ziolkowski, Jerzy Malachowsk, “Reliability analysis and redundancy optimization of k-out-of-n systems with random variable k using continuous time Markov chain and Monte Carlo simulation,” Reliab. Eng. Syst. Saf. 242, 109780 (February 2024). https://doi.org/10.1016/j.ress.2023.109780

FPGA-Based Hardware Optimization and Implementation of YOLOv4-tiny

Kai Wang*[a], Yanhong Bai[a,b], Xiaosong Li[a]
[a] Department of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan China 030024; [b] Department of Intelligent Manufacturing Industry, Shanxi University of Electronic Science and Technology, Linfen China 041000

ABSTRACT

To address the challenges associated with the YOLOv4-Tiny algorithm's complex structure, high computational resource requirements, and extensive parameter count—which collectively impede efficient FPGA deployment—we propose a hardware-software co-optimization strategy. This approach replaces the YOLOv4-Tiny backbone with the MobileNetv1 network and integrates the Convolutional Block Attention Module (CBAM) into the enhanced feature extraction network. Channel pruning is applied to streamline the network structure, and weights and biases are quantized to 16-bit fixed-point representations. Compared to the original YOLOv4-Tiny, this optimized network reduces parameters by 40% while retaining nearly identical recognition accuracy.Using high-level synthesis tools, we generate FPGA IP cores, design a parallel pipelined convolutional architecture, and implement inter-layer blocking between convolutional layers to enhance computational efficiency. This improved algorithm is deployed on a Zynq-7020 FPGA chip. Experimental results show that the optimized algorithm achieves a computational performance of 43.4 GOP/s, offering a speedup of 1.6 to 4.1 times compared to existing studies, with an energy efficiency ratio 4.8 to 10.7 times greater than current implementations. These findings indicate that the proposed strategy significantly improves algorithm deployment efficiency on resource-limited FPGA platforms.

Keywords: YOLOv4-Tiny; algorithm Pruning; algorithm quantization; FPGA; parallel pipeline structure

1. INTRODUCTION

In recent years, object detection [1] technology has become a key research area in computer vision due to its broad applications. Among various object detection algorithms, the YOLO [2] (You Only Look Once) series stands out for its notable advantages in speed and accuracy. However, these advanced object detection models typically require substantial computational resources and extensive storage, which sharply contrasts with the limited computational power, restricted storage, and stringent power requirements of FPGA devices.

To address these challenges, researchers have explored various model compression and lightweighting techniques [3]. For instance, Reference [4] proposed replacing standard convolutions with depthwise separable convolutions to reduce network parameters and computational load. Additionally, Reference [5] introduced knowledge distillation to decouple and transfer knowledge from complex models, enabling the training of a lightweight BearingPGA-Net network suitable for FPGA-based bearing fault diagnosis. Conversely, researchers have developed specialized FPGA hardware architectures to deploy neural networks, aiming to further improve computational and energy efficiency. Main research directions focus on designing custom hardware accelerators, such as systolic arrays and system-level pipelines [6]. Given that convolution operations are the most time-consuming component, Reference [7] designed a dedicated convolution accelerator, leveraging FPGA's high parallelism to optimize convolutional networks. By employing low-precision fixed-point representations for weights and activation parameters, experimental results demonstrated a 32% reduction in weight storage.

This paper primarily addresses the following innovations in deploying deep learning algorithms on resource-limited FPGAs: (1) Backbone Network Replacement and Model Compression: We replace the backbone network of the target algorithm and apply pruning and quantization to weights and parameters, maximizing FPGA resource utilization.(2) Design of a Highly Parallel and Deeply Pipelined Hardware System: By designing modules for on-chip storage, convolutional computation, and task scheduling, we implement a highly parallel, deeply pipelined hardware system for

2889931846@qq.com; phone 19326472537

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE Vol. 13519, 135190B · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057771

object detection.(3) Deployment on the ZYNQ-7020 FPGA Board: Deploying the optimized YOLOv4-Tiny model on the ZYNQ-7020 FPGA board enables high-speed processing and low-power operation for the object detection system.

Experimental results demonstrate that, compared to running the YOLOv4-Tiny algorithm on a Cortex-A9 processor, the optimized model reduces single-image processing time from 6.5 minutes to 390 milliseconds while maintaining high detection accuracy. Section 2 of the article discusses the algorithm improvement methods, Section 3 presents the hardware design framework and implementation scheme, Section 4 reports the experimental hardware results, and Section 5 provides the conclusions.

2. YOLOV4-TINY NETWORK STRUCTURE IMPROVED

The improved YOLOv4-Tiny network structure is shown in Figure 1, consisting of three main components: the backbone feature extraction network, the enhanced feature extraction network, and the prediction output. The CBL module, as defined in the figure, comprises a convolutional layer, batch normalization, and a Leaky-ReLU activation function. This module serves to enhance the non-linear representation capacity of features while ensuring the stability of network training.To adapt the network for resource-constrained FPGA devices and achieve model lightweighting with minimal accuracy loss, the YOLOv4-Tiny architecture has been modified. In the backbone feature extraction network, MobileNet replaces the traditional ResBlock, and the number of channels in each convolutional layer is reduced by adjusting the width multiplier hyperparameter, thus decreasing the network's parameter count. In the enhanced feature extraction network, channel attention and spatial attention mechanisms (CBAM) are incorporated to better focus on key feature regions, ensuring detection accuracy remains largely unaffected after lightweighting.

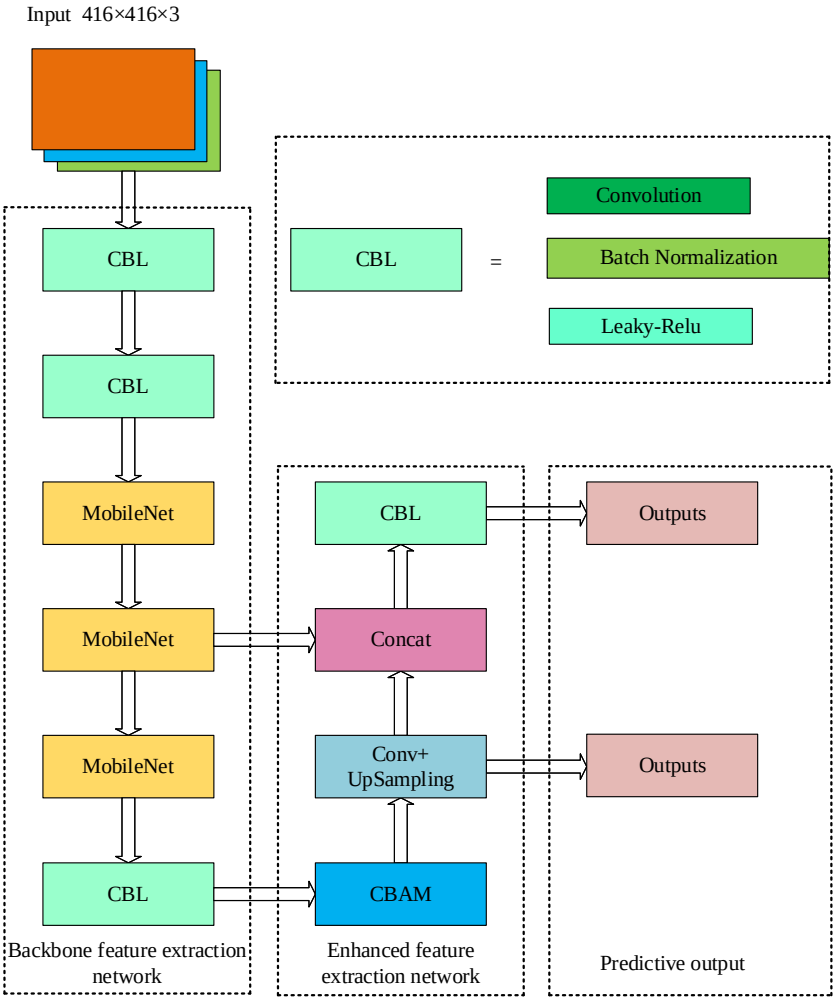


Figure 1. Structure of the improved YOLOv4-Tiny model

2.1 Network pruning

To reduce the computational and parameter load and enhance the inference speed of the model, this study implements network pruning on the improved YOLOv4-Tiny structure, introducing a channel pruning strategy. Compared to weight pruning, channel pruning significantly reduces both computational demands and model size; in terms of hardware-friendliness, channel pruning does not require specific hardware support to effectively utilize sparsity and simplifies hardware acceleration by eliminating entire channels. The pruning strategy outlined in this paper is divided into three steps: (1) Sparse Training: Introducing an L1 regularization term into the loss function to achieve sparsity in scaling factors. (2) Channel Pruning: Based on preset thresholds and scaling factor comparisons, selective pruning of the neural network is conducted. (3) Model Fine-Tuning: The pruned model is retrained using a low learning rate. This design employs structured pruning based on the scaling factor Gamma in the batch normalization (BN) layers. L1 regularization drives the BN layer's Gamma coefficients towards zero, enabling the model to automatically identify and eliminate channels of low importance. An L1 norm is introduced into the loss function to constrain the weights, structured as follows:

$$L = \sum_{(x,y)} l(f(x, W), y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma) \tag{1}$$

In the formula (1), the parameters (x, y) represent the input and output of training, and W represents the corresponding weights. The first term is the loss after a regular iteration, and λ is the coefficient that balances the sparsity-inducing factor. Before pruning, we set g(s) = |s| (L1 norm) to ensure that the solution for L achieves sparsity.Gamma coefficients correspond directly to the dimensions of the feature maps post-convolution, and their values indicate the importance of the respective channels. Channels with Gamma coefficients close to zero undergo sparsity-induced pruning in practice, affecting the convolution kernels of the feature maps. By sorting the global Gamma coefficients by absolute value and selecting a threshold based on the percentage of channels to be pruned, channels below this threshold are pruned. These measures effectively reduce the computational load on the FPGA during the model inference stage, lowering the consumption of DSP and LUT resources. Additionally, the reduction in computational steps also decreases inference latency, thereby enhancing the overall performance of the system.

2.2 Dynamic fixed point quantization.

Appropriate quantization can significantly enhance the efficiency of hardware resource utilization. Assuming that model parameters and computation results undergo N-bit fixed-point quantization, the quantization process can be divided into the following four steps:

I. Sort the original data by absolute value and identify the maximum value:

$$|Max| = \max(|x_i|) \tag{2}$$

II. Get decimal bit width: value:

$$f_{x_i} = ceil\left(\log_2 \frac{|Max|}{2^{N-1}-1}\right) \tag{3}$$

III. Scale each input data linearly and round it to a fixed-point number:

$$x_i' = round(x_i \cdot 2^{-f_{x_i}}) \tag{4}$$

IV. To prevent data overflow, truncate the output data to N bits:

$$x_i' = \begin{cases} 2^{N-1}-1, & x_i' > 2^{N-1}-1 \\ x_i', & -2^{N-1} \le x_i' \le 2^{N-1}-1 \\ -2^{N-1}, & x_i' < -2^{N-1} \end{cases} \tag{5}$$

On the other hand, when using a dynamic fixed-point quantization strategy to process weights and bias parameters, misalignment of decimal places may occur during the convolution process. Misalignment during addition can lead to computational errors. There are primarily two scenarios of decimal misalignment:

The decimal places of the multiply-accumulate results from the input features and weights do not align with those of the biases, which means:

$$f_{in} + f_w[n] \neq f_b[n] \tag{6}$$

The decimal places of the multiply-accumulate results of the input features and weights do not align with those of the output feature maps, which means:

$$f_{in} + f_w[n] \neq f_{out} \tag{7}$$

These issues can be resolved using shift operations. Quantization performance results are shown in Table 1. Compared to the methods in references [8] and [9], our method achieves better results. Although reference [10] yields a smaller model size, it suffers a significant drop in accuracy, whereas our approach maintains nearly the same accuracy as the original network. This balance between accuracy and model size makes our quantization method ideal for resource-constrained FPGA platforms.

Table 1 . Quantified comparison of performance results

	Target network	**mAP/%**	**volume /mb**
Literature [8]	YOLOv3-Tiny	55.6	20.2
Literature [9]	YOLOv4-Tiny	69.2	24
Literature [10]	YOLOv4-Tiny	41.3	12.6
This text	YOLOv4-Tiny	79.1	13.8

3. HARDWARE IMPLEMENTATION

3.1 Overall scheme

The hardware-software co-design scheme is shown in Figure 2. The design task is divided into two parts: model compression and hardware deployment. In the model compression process, sparse training is first performed to obtain scaling factors. Then, based on the size of these scaling factors, channel importance is evaluated, and connections with lower importance are pruned. Precision recovery training follows to compensate for accuracy loss caused by channel pruning. Finally, quantization is applied, converting data from 32-bit floating-point to 16-bit fixed-point, generating weight and bias files.The hardware component is optimized with parallel and pipelined design, replacing the traditional serial computation process. The hardware deployment process on the FPGA proceeds as follows: first, using High-Level Synthesis (HLS) tools to write and test the YOLO module code, perform functionality verification, and optimize performance. RTL-level simulation is then conducted, and these modules are encapsulated as reusable IP cores with interface optimization to enhance data transfer efficiency. The designed IP cores are connected with other modules in Vivado to create the final hardware structure. Lastly, the weight and bias files, along with the hardware configuration files of the optimized YOLOv4-Tiny model, are downloaded to the ZYNQ board. Using Xilinx SDK software, the system's functionality and performance are tested and evaluated.

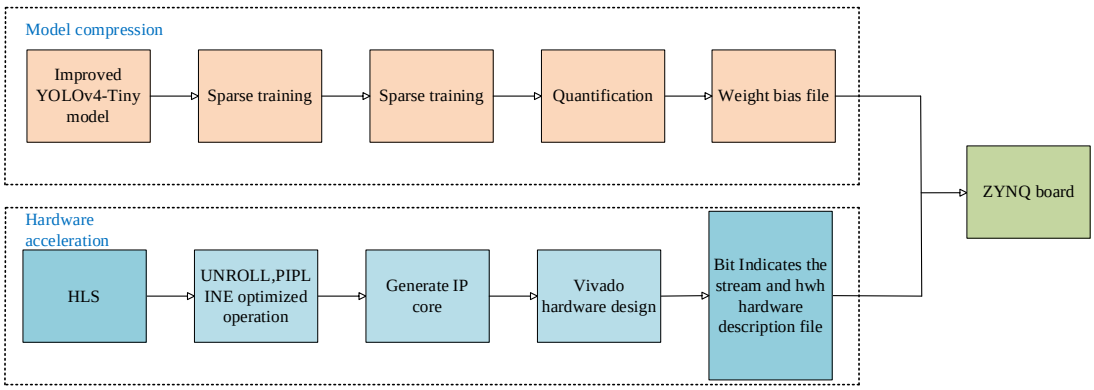


Figure 2. Software and hardware design flow chart

3.2 Overall Framework of Hardware Design

The overall framework of the hardware design is illustrated in Figure 3. The entire algorithm deployed on the ZYNQ platform is divided into two components: one part operates on the PL (Programmable Logic) side, which houses the neural network accelerator, while the other part runs on the PS (Processing System) side, primarily responsible for acquiring input images and handling general computations. Communication and data transfer between the PS and PL sides are facilitated by the AXI bus. The PS side retrieves image data and weight parameters from the memory card and sends them via the AXI bus to the compiled neural network accelerator on the PL side for inference. Following operations such as classification and bounding box prediction, the results are transmitted back to the PS side over the AXI bus and are finally read and displayed on a PC.

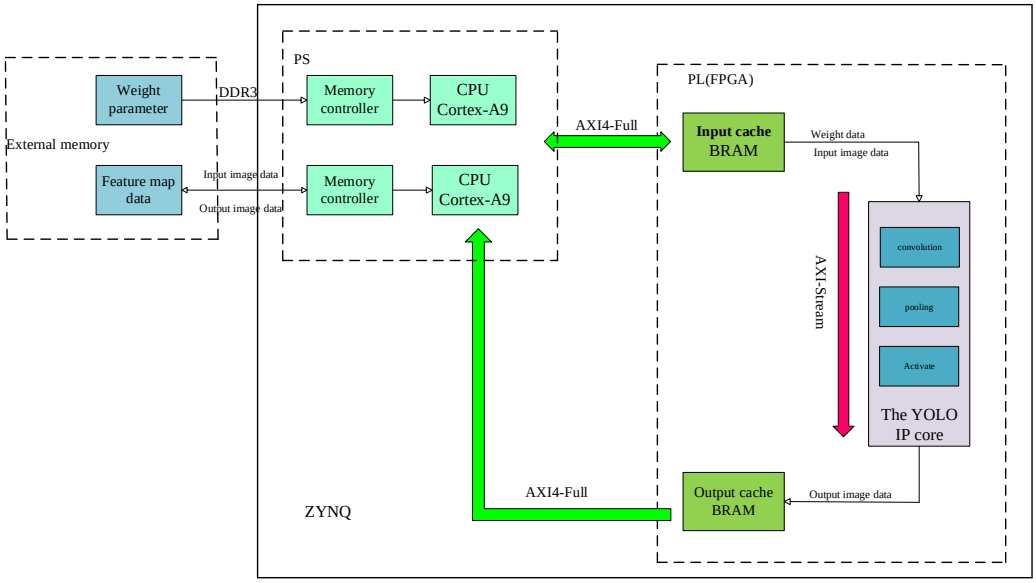


Figure 3. Hardware design structure diagram

3.3 Design of Grouping Between Convolutional Layers

The convolutional layer grouping design is shown in Figure 4. Frequent data retrieval from off-chip memory can lead to significant transmission delays. To address this issue, based on the different types of convolutional layers in the YOLOv4-Tiny architecture, the network layers are divided into four groups using Vivado SDK software, with each group containing different types of layers. This grouping, along with a selective activation strategy, effectively reduces delays caused by frequent DDR access.

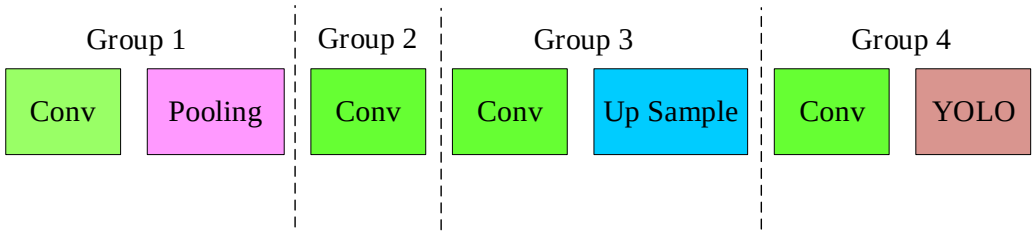


Figure 4. Diagram of Convolutional Layer Grouping Design

4. EXPERIMENTAL ANALYSIS

4.1 Experimental environment setting

The hardware system design in this paper is based on Xilinx's ARM+FPGA architecture, utilizing the Zynqxc7z020clg400-2 chip. The development environment for the hardware system included Vivado 2019.2 and Vivado HLS 2019.2. The hardware design results are shown in Table 2. The SoC operates at a frequency of 120 MHz with a power consumption of 3.083 W, utilizing approximately 70% of the LUT and BRAM resources, and with a DSP utilization rate of 98%. Experimental results indicate that the design effectively utilizes the limited on-board resources.

Table 2. Resource utilization

resource	used	usable	occupy
LUT	36708	53200	69%
FF	39368	106400	37%
BRAM	170	280	68%
DSP	214	220	98%
frequency /Mhz	120	/	/

4.2 Performance comparison

The performance comparison of the improved YOLOv4-Tiny algorithm on different processors is shown in Table 3, which records the inference time for a single run on each platform. On the ARM+FPGA platform, a single inference takes only 0.39 seconds, achieving a speedup of 172 times compared to using ARM alone. The results show that the energy efficiency of FPGA is 47 times higher than that of the CPU platform and 1.5 times higher than that of the GPU platform.

To comprehensively evaluate the system's overall performance, a comparative analysis was conducted with previous studies. We selected several convolutional neural network accelerators based on different design approaches, all using the Zynq-7020 hardware platform, and compared their performance metrics from various perspectives, as summarized in Table 4. Reference [8] proposed a general convolutional neural network accelerator that performed well in accelerating YOLOv3-Tiny but showed lower DSP utilization efficiency compared to our design. References [9] and [10] targeted the same algorithm as this work. Reference [9] employed 16-bit fixed-point quantization to map the algorithm to FPGA but lacked specific structures for inference acceleration, resulting in a significant performance gap compared to our approach. Reference [10] designed a specialized accelerator with highly parallel pipelined traditional and depthwise convolutions to improve computational efficiency; however, its performance was mainly constrained by bandwidth, limiting effective utilization of processing resources. Our method achieves approximately ten times the computational capability of reference [10].

Reference [11] introduced a shared convolutional hardware structure that showed advantages over other references in terms of energy efficiency and resource utilization. While the performance of reference [11] is comparable to ours, our major advantages lie in scalability and the reduced need for multiple retraining processes, thus lowering training costs. In terms of computational power, energy efficiency, and DSP utilization efficiency, our design demonstrates significant improvements over existing studies.

Table 3. Comparison of models on different hardware platforms

Hardware platform	CPU	GPU	ARM	ARM+FPGA
Model	i5-12600KF	RTX3060Ti	Cortex-A9	Zynq7020
Clock frequency	3.7GHz	1800MHz	667MHz	120MHz
Identification time /ms	465	6.75	67308	390
Frame rate /FPS	2.15	148	0.015	2.56
Power dissipation /W	115	242	1.9	3.2
Energy efficiency ratio	0.019	0.61	0.0079	0.8

Table 4. Comparison between this paper and other literatures on the same hardware platform

	Algorithm model	FPGA frequency/ *MHz*	DSP Usage amount	Power dissipation / *W*	Computing power / *GOPS*	Energy efficiency / $GOPS \cdot W^{-1}$	DSP efficiency/ $DSP \cdot W^{-1}$
Reference [8]	YOLOv3-Tiny	100	173/220	2.53	10.6	4.23	0.062
Reference [9]	YOLOv4-Tiny	100	149/220	2.38	0.94	0.39	0.0063
Reference [10]	YOLOv4-Tiny	180	196/220	2.81	4.04	1.44	0.021
Reference [11]	YOLOv5s	180	196/220	3.04	26.0	8.56	0.133
This text	YOLOv4-Tiny	120	214/220	3.2	43.4	15.5	0.231

5. CONCLUSION

This work presents a hardware-software co-design based on the YOLOv4-Tiny network. The software aspect combines model pruning and quantization techniques to enhance inference efficiency. The hardware design is based on the Xilinx Zynq-7020 platform, with custom optimizations tailored to fully utilize on-chip resources. System performance evaluation covered key metrics such as resource utilization, computation latency, power consumption, and GOPS. Experimental results show that the design achieves a good balance between resource usage, inference speed, and power consumption, delivering an effective compute power of 43.4 GOPS with an inference time of just 390 ms per image and a power consumption of 3.2 W. Compared to existing deployment schemes, our system demonstrates significant advantages in energy efficiency and computational efficiency, making it more suitable for edge computing scenarios with limited processing power. Future work will explore lightweighting the network through techniques such as knowledge distillation to make it even more suitable for deployment on FPGA devices.

ACKNOWLEDGEMENTS

This work was supported by the Research Start-up Fund for Talent Introduction of Shanxi Electronic Science and Technology Institute under Grant 2023RKJ018, and the Key Technology Research on Safety Risk Identification and Operation Control of Power Engineering Field Based on 3D Spatiotemporal Fusion under Grant 2024TYJB0133.

REFERENCES

[1] Guo J M, Yang J S, Seshathiri S, et al. A light-weight CNN for object detection with sparse model and knowledge distillation[J]. Electronics, 2022, 11(4): 575.
[2] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
[3] Xu S, Zhou Y, Huang Y, et al. YOLOv4-tiny-based coal gangue image recognition and FPGA implementation[J]. Micromachines, 2022, 13(11): 1983.
[4] Xiao C, Xu D, Qiu S, et al. FGPA: Fine-grained pipelined acceleration for depthwise separable CNN in resource constraint scenarios[C]//2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, 2021: 246-254.
[5] Liao J X, Wei S L, Xie C L, et al. BearingPGA-Net: A lightweight and deployable bearing fault diagnosis network via decoupled knowledge distillation and FPGA acceleration[J]. IEEE Transactions on Instrumentation and Measurement, 2023.
[6] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size[J]. arxiv preprint arxiv:1602.07360, 2016.
[7] Chen Zhuo, Chen Yiduo, Tian Chunsheng, et al. "Cache Optimization Structure Design for ZynqNet Hardware Accelerator" [J]. Microelectronics, 2023, 53(5): 841-845.
[8] Chen Haomin, Yao Senjing, Xi Yu, et al. "Hardware Acceleration Design and FPGA Implementation of YOLOv3-Tiny" [J]. Computer Engineering and Science, 2021, 43(12): 2139-2149.
[9] LI P,CHE C.Mapping YOLOv4-tiny on FPGA-based DNN Accelerator by Using Dynamic Fixed-Point Method[J],12th International Symposium on Parallel Architectures,AlgorithmsandProgramming,2021,1:125-129.
[10] Cao Yuanjie, Gao Yuxiang, Du Xinchang, et al. "FPGA Acceleration Method Based on Improved YOLOv4-Tiny" [J]. Radio Engineering, 2022, 52(04): 604-611.
[11] Liu Qian, Wang Linlin, Zhou Wenbo. "Efficient Convolutional Accelerator Design for YOLOv5s Network Based on FPGA" [J]. Telecommunication Engineering, 2024, 64(03): 366-375.

MTOClus: Multi-Type Objects Clustering in Heterogeneous Information Networks

Yongjie Liang[a], Wujie Hu[b], Junjie Wu[c], and Jinzhao Wu[d]

[a,c]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China
[b]School of Electric Engineering, Guangxi University, Nanning, China
[d]Guilin University of Aerospace Technology, Guilin, China

ABSTRACT

Clustering is crucial for analyzing heterogeneous information networks (HINs). Mainly state-of-the-art algorithms often focus on single-type node clustering, overlooking the clustering of multiple node types and requiring manual setting of cutoff distances and clustering centers. This paper introduces a **clus**tering algorithm integrating **m**ulti-**t**ype **o**bjects, named MTOClus, which generates multiple similarity matrices for different node types and assigns weights to each matrix to indicate their significance. By aggregating weighted similarity matrices, it creates a distance matrix. MTOClus automates cluster center selection through node sorting and mitigates the impact of outliers by utilizing the K-Nearest Neighbors approach to compute the cutoff distance. Experimental results across four datasets consistently show that MTOClus outperforms six other algorithms, underscoring its superior clustering efficacy.

Keywords: Heterogeneous information networks; Multi-type objects clustering

1. INTRODUCTION

In recent years, HINs have garnered significant attention as a versatile framework for modeling complex systems.[1] These networks find applications in diverse real-world domains such as social networks, academic networks, e-commerce platforms, and biological molecular structures.[2] HINs offer the advantage of effectively integrating a wealth of information, capturing richer semantics in both node attributes and edge relationships. This advancement has spurred innovation in data mining, prompting researchers to delve into novel tasks tailored for HINs, including similarity search,[3] clustering,[4] and classification.[5]

Cluster analysis, also known as clustering, refers to the task of partitioning a dataset into groups or clusters, where objects within a cluster share similarities among themselves but are dissimilar to objects in other clusters. This unsupervised data analysis technique holds significant importance in data mining and is widely applied across diverse domains, including image pattern recognition, web search, biology, and security.[6] Most HIN clustering algorithms primarily focus on clustering nodes of a single type,[7] lacking algorithms capable of clustering nodes of integrating multiple types within HINs. However, some challenges persist. Firstly, an important challenge is automatically selecting cluster centers when clustering nodes from multiple types. Secondly, the sparsity inherent in HINs poses another challenge. It can lead to a large number of outliers during the clustering process, thereby affecting the quality of the clustering result. Density-based clustering algorithms are effective in detecting outliers, but determining the appropriate cutoff distance remains a challenge. The selection of the cutoff distance is crucial for effectively handling outliers and ensuring the robustness of the clustering algorithm. However, it requires further investigation and development.

Further author information: (Send correspondence to Jinzhao Wu)
Yongjie Liang: E-mail: 22032202017@mails.guet.edu.cn
Wujie Hu: E-mail: hwj@st.gxu.edu.cn
Junjie Wu: E-mail: 23032201031@mails.guet.edu.cn
Jinzhao Wu: E-mail: wjzguet@163.com

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190C · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057528

To address these issues, we propose a novel algorithm for clustering in HINs, named MTOClus (Multi-Type Objects Clustering). MTOClus integrates multiple metapaths for each node type. These matrices are then weighted and aggregated through summation to form the final similarity matrix. When converted into a distance matrix, MTOClus utilizes node sorting to choose cluster centers and establishes cutoff distances through the K-Nearest Neighbors (KNN) approach. Subsequently, nodes are allocated to clusters based on the distance matrix, resulting in the final clustering outcome.

The main contributions of this paper are outlined as follows:

1. This paper acknowledges the challenges faced by current clustering methods for HINs, especially in integrating multi-type information and addressing outliers caused by sparsity during clustering.
2. This paper introduces the MTOClus algorithm, which effectively integrates multiple metapaths and employs a weighted aggregation approach to enhance clustering outcomes. Additionally, it utilizes node sorting to select cluster centers and determines the cutoff distance based on the KNN approach to improve overall efficiency.
3. MTOClus consistently outperforms six baseline algorithms based on publicly classic evaluation metrics in a series of comprehensive experiments. These results further demonstrate the effectiveness and necessity of analyzing heterogeneous information networks.

The paper is organized as follows: Section 2 presents the definitions and concepts employed in this study. Section 3 details the MTOClus algorithm proposed in this research. Section 4 discusses the experimental results, and Section 5 summarizes the key findings and contributions of this study.

2. PRELIMINARIES

Heterogeneous information network (HIN) is characterized by the tuple $G = (V, E, O, R)$, where V denotes the set of nodes and E represents the set of edges. O and R refer to the sets of node types and edge types, respectively. A node type mapping function $\varphi : V \to O$ is employed to link each node $v \in V$ to a specific node type in O. Similarly, an edge type mapping function $\psi : E \to R$ establishes a correspondence between each edge $e \in E$ and a specific edge type in R, where $|O|$ denotes the cardinality of O. If $|O| + |R| > 2$, the network is commonly referred to as a heterogeneous information network; otherwise, it is a homogeneous information network.

EXAMPLE 2.1. *An example of a heterogeneous information network (HIN) is illustrated in Figure 1. This network consists of three actors, two movies, and two directors. For instance, node a_1 is connected to node m_1 through an 'actor-movie' link, indicating that actor a_1 participates in movie m_1. Formally, the set of nodes in this network is denoted as $V = \{a_1, a_2, a_3, m_1, m_2, d_1, d_2\}$, and the set of edges is represented as $E = \{(a_1, m_1, 0), (a_2, m_1, 0), (a_2, m_2, 0), (a_3, m_1, 0), (a_3, m_2, 0), (m_1, a_1, 1), (m_1, a_2, 1), (m_1, a_3, 1), (m_2, a_2, 1), (m_2, a_3, 1), (m_1, d_1, 2), (m_2, d_2, 2), (d_1, m_1, 3), (d_2, m_2, 3)\}$.*

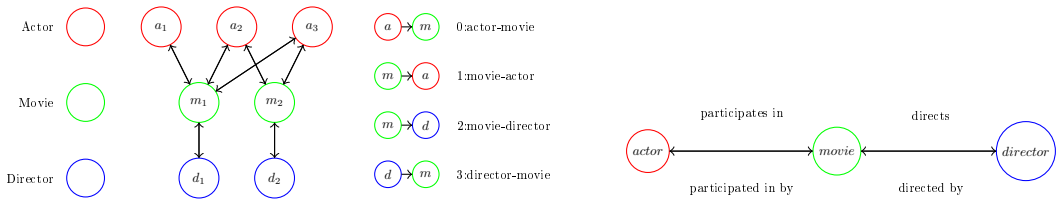


Figure 1. Example of HIN.

Network schema serves as a meta-template for a HIN $G = (V, E, O, R)$, where all elements in O are treated as nodes, and all elements in R are viewed as edges.

The network schema for Example 2.1 comprises three node types: actor (A), movie (M), and director (D). Additionally, it includes four types of edges connecting these nodes. For example, an edge between an actor and a movie denotes an actor participates in a movie.

Metapath is a connected path defined within a network schema and is formally defined as $O_1 \rightarrow O_2 \rightarrow ... \rightarrow O_k$, where $O_i \in O, i = 1, 2, \cdots k$, and k is an integer denoting its length.

In Figure 1, multiple distinct metapaths can be derived. For instance, the metapath A-M (actor $\overset{participates\ in}{\longrightarrow}$ movie) indicates an actor participates in a movie, while A-M-D (actor $\overset{participates\ in}{\longrightarrow}$ movie $\overset{is\ directed\ by}{\longrightarrow}$ director) signifies an actor participates in a movie directed by a director.

Cluster center type is characterized as a specific type within O having the fewest nodes among all node types, while any type other than the cluster center type is designated as the cluster target type.

3. MTOCLUS: A MULTI-TYPE OBJECT CLUSTERING ALGORITHM

This section proposes MTOClus algorithm, including information integration of multi-type object based on diverse metapaths, solution optimization of cluster center.

3.1 Integrating similarity matrices

In this subsection, we first determine cluster center and target types, then establish a set of similarity matrices $P = \{P_1, P_2, ..., P_n\}$ based on metapath set $mp = \{p_1, p_2, \cdots, p_n\}$. Each similarity matrix is then obtained based on corresponding metapath $p_n = (O_1 O_2 ... O_{l_n})$ and HeteSim algorithm,[8] where similarity between two nodes v_i and v_j is defined as:

$$HeteSim(v_i, v_j|p) = \frac{PM_{p_L}(i,:)PM'_{p_R^{-1}}(j,:)}{|PM_{p_L}(i,:)||PM'_{p_R^{-1}}(j,:)|} \tag{1}$$

Where $PM_{p_L} = U_{O_1 O_2} ... U_{O_{mid-1} O_{mid}}$ and $PM'_{p_R^{-1}} = U_{O_{l_n} O_{l_n - 1}} ... U_{O_{mid} O_{mid+1}}$, U is the transition probability matrix, $PM_p(a,:)$ means the a-th row in PM_p and $mid = \frac{l_n + 1}{2}$.

Now, we aim to distribute the weights $\alpha = [\alpha_1, \alpha_2, ..., \alpha_n]$ for each similarity matrix by employing the entropy method. Next, we propose concrete operation processes.

Firstly, compute the proportion pt_{ij} of the i-th number in the j-th column for each matrix M in P using the following formula:

$$pt_{ij} = \frac{M_{ij}}{\sum_{i \in [0, |t|-1]} M_{ij}}, j \in [0, |c| - 1] \tag{2}$$

Where $|c|$ represent the number of cluster center type nodes and $|t|$ represent the number of nodes for cluster target type t.

Next, calculate the entropy value e_j for the j-th column in M using the formula:

$$e_j = -k \times \sum_{i \in [0, |t|-1]} pt_{ij} \times log(pt_{ij}), k = \frac{1}{ln(|c|)} \tag{3}$$

Once the entropy value for each column in M is obtained, calculate the average entropy M_e of M:

$$M_e = \frac{\sum_{j \in [0, |c|-1]} e_j}{|c| - 1} \tag{4}$$

Then, calculate the coefficient of variation g for each matrix:

$$g = 1 - M_e \tag{5}$$

Finally, determine the weight of each matrix:

$$\alpha_i = \frac{g_i}{\sum_{i\in[1,n]} g_i}. \tag{6}$$

With the set of weight values α and the sets of similarity matrices in P, we can obtain the final comprehensive similarity matrix S through weighted aggregation as follows:

$$s = \sum_{i\in[1,n]} \alpha_i \cdot P_i \tag{7}$$

The pseudocode for computing and aggregating the similarity matrices is outlined in Algorithm 1.

Algorithm 1: Calculating and aggregating similarity matrixs.

Input : Hin $G = (V, E, O, R)$, cluster center type c, cluster target type t, the set of metapaths mp between c and t
Output: the final similarity matrice s between c and t

```
P = [ ];
n ← length of mp;
for i ← 1 to n do
    Generate probability transition matrixs based on path_i;
    Calculate P_i = HeteSim(t, c|path_i) in (1);
    Add P_i to P;
end
Calculating α in (6);
Aggregating similarity matrixs in (7);
Normalize s;
```

3.2 Determining cluster centers

In this subsection, we will calculate the number of clusters and then choose cluster center nodes.

We used the elbow method[9] to compute the number of clusters. The core idea of the elbow method is that as the number of clusters k increases, the sample partition becomes finer. Consequently, the cohesion of each cluster gradually increases, leading to a gradual decrease in the sum of squared errors (SSE). Additionally, when k is less than the true number of clusters, increasing k significantly enhances the cohesion of each cluster, leading to a large decrease in SSE. However, when k reaches the actual number of clusters, the additional improvement in cohesion achieved by increasing k further diminishes rapidly, leading to a significant decrease in the rate of SSE reduction. Subsequently, as k continues to increase, the decrease in SSE becomes less pronounced, indicating a flattening of the SSE versus k curve.

We used DPRank[10] to select cluster center nodes. DPRank is a node ranking method designed to identify important nodes within a network. DPRank considers not only the immediate local environment of nodes but also broader contexts, such as the neighbors of their neighbors, resulting in a more accurate assessment of node importance. The transition probability matrix TP for DPRank is defined as follows:

$$TP_{i,j} = \frac{d_i}{\sum_{v\in N(i)} d_v} \cdot A_{i,j}, \tag{8}$$

where A represent the adjacency matrix of G, d represent the node degree and $N(i)$ represent the neighbors of node. Then we list the pseudocode for above process in Algorithm 2.

3.3 Multi-type objects clustering

In this subsection, we will calculate the cutoff distance for each cluster center node and then proceed with clustering.

Given the sparsity of the HIN, the abundance of outliers is a major concern. Determining outliers is a crucial problem to address. Drawing inspiration from the K-Nearest Neighbors (KNN)[11] approach, we adaptively select the cutoff distance based on the K nearest neighbors of the cluster center nodes. The adaptive cutoff distance d_{ij} between the cluster center c_i and the target type t_j is defined as follows:

Algorithm 2: Selecting cluster center nodes.

Input : Hin $G = (V, E, O, R)$, adjacency matrix A, cluster center type c, the set of cluster target type T
Output: the set of distance matrices D, the set of cluster center nodes $cnodes$

$D = [\,]$;
$n \leftarrow$ length of T;
for $i \leftarrow 1$ *to* n **do**
  Calculating s_i in Algorithm 1;
  Add $1 - s_i$ to D;
end
$D_{mat} \leftarrow$ splicing all matrices in D;
$k_{num} \leftarrow$ using elbow method on D_{mat};
Calculating TP in (8);
Calculating the eigenvector vec corresponding to the greatest eigenvalue 1 of TP^T;
Extracting the sub-vector vec_c from vec corresponding to the nodes of c;
Selecting the nodes corresponding to the top k_{num} maximum values in vec_c as the cluster center nodes;

$$d_{ij} = \frac{dist_{ij}}{N_j^{\frac{1}{n}}}, n = \{1, 2, 3, ...\} \tag{9}$$

Where N represent the number of target type node, and $dist_{ij}$ represent the average distance of $N_j^{\frac{1}{n}}$ nearest neighbor of cluster center c_i.

After obtaining the set of distance matrix D, cutoff distance d, and cluster centers nodes $cnodes$, we can proceed with clustering. Using each cluster center as the center of a circle and d as the radius, we draw circles. By using the distance matrix, we can calculate the distance from each node to the cluster centers, which helps us determine whether a node belongs to a specific cluster. If a node is not within any of the circles, it is considered an outlier and is not included in the clustering results. If a node is within the radius of multiple circles, it will be assigned to the cluster with the smallest distance to the cluster center, as determined by the distance matrix. Algorithm 3 provides pseudocode for MTOClus.

Algorithm 3: MTOClus.

Input : Hin $G = (V, E, O, R)$, adjacency matrix A, cluster center type c, the set of cluster target type T
Output: cluster results

Calculating D and $cnodes$ in Algorithm 2;
$n \leftarrow$ length of D;
$m \leftarrow$ length of $cnodes$;
for $i \leftarrow 1$ *to* m **do**
  for $j \leftarrow 1$ *to* n **do**
    Calculating d_{ij} in (9);
  end
end
for $j \leftarrow 1$ *to* n **do**
  Clustering based on $D_j[:, cnodes]$, and d_{ij};
end

4. EXPERIMENTS

4.1 Datasets, baseline algorithms, and evaluation methods

We utilize four real-world datasets in our study, which are:

ACM[12*]: a bibliographic network containing academic publications; **DBLP**[12 *]: a bibliographic network dedicated to academic publications; **IMDB**[†]: a movie information network; **KG20C**[13‡]: a scholarly data analysis support knowledge graph. Figure 2 illustrates the network schema alongside the respective sizes of each dataset.

*https://github.com/BUPT-GAMMA/OpenHGNN
†https://www.kaggle.com/datasets/karrrimba/movie-metadatacsv
‡https://www.kaggle.com/datasets/tranhungnghiep/kg20c-scholarly-knowledge-graph

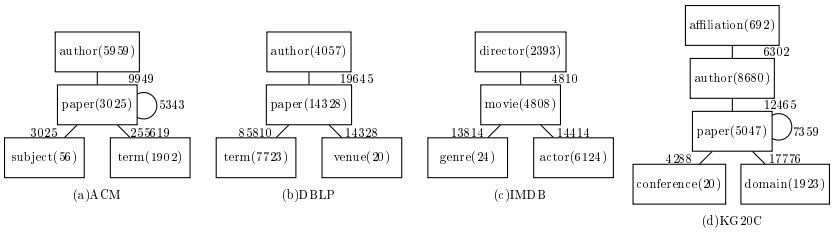


Figure 2. Network schema and size of datasets.

To evaluate the effectiveness of object algorithm, we compare it with six commonly used clustering algorithms that serve as baselines in the field.

K-means:[14] a straightforward method for partitioning data into K groups based on similarity is often employed in tasks such as customer segmentation and image compression.

X-means:[15] a variant of the K-means algorithm dynamically determines the optimal number of clusters during the clustering process.

FCM:[16] a method that allows data points to belong to multiple clusters with varying degrees of membership, which is useful for handling ambiguous cluster assignments.

Bsas:[17] a sequential clustering method processes data points one by one based on a predefined order, facilitating efficient organization and analysis of data.

Mbsas:[17] A variant of the Bsas, this algorithm efficiently clusters data points sequentially.

SClump:[18] a method utilizes spectral clustering and metapaths to create effective similarity matrices. It iteratively enhances these matrices and optimizes metapath weights.

Since partial datasets lack true labels, making direct comparisons of clustering results with ground truth unfeasible, thus we further employ five internal evaluation metrics to assess the performance among diverse algorithms:

Sum of Squares due to Error (SSE)[9]: assesses the dispersion of data within clusters by summing the squared distances between each data point and its cluster centroid. Smaller SSE values indicate tighter and more cohesive clusters, calculated as $SSE = \sum_{k\in[1,K]} \sum_{p\in C_k} |p - m_k|^2$. Here, k, p, and m_k represent clusters, nodes within clusters, and cluster centers, respectively.

Silhouette Coefficient (SC):[19] measures the cohesion within clusters and the separation between clusters, aiming for a higher value. It is calculated as $SC = \frac{\sum_{i\in[1,n]} \frac{b(i)-a(i)}{max(a(i),b(i))}}{n}$, where n, $a(i)$, and $b(i)$ represent the number of nodes, average distance within the same cluster, and minimum average distance to nodes in other clusters, respectively.

Davies-Bouldin Index (DBI):[20] quantifies clustering based on the compactness within clusters and the separation between clusters. A lower index indicates more effective clustering, as demonstrated by $DBI = \frac{\sum_{i,j\in[1,K] max_{i\neq j}} \frac{s_i+s_j}{d_{ij}}}{K}$, where s_i, d_{ij}, and K represent the average distance within clusters, the distance between clusters, and the total number of clusters, respectively.

4.2 Experimental setup

All baselines use the $D_{mat} \times D_{mat}^T$ calculated by Algorithm 3 as input. K-means, X-means, and FCM require the selection of initial cluster centers and the specification of the number of clusters before initiating the clustering process. Bsas, Mbsas and SClump require specifying the number of clusters before conducting the clustering process. We utilize the K-Means++[21] initialization technique to select the initial cluster centers. K-Means++ improves the convergence and quality of the final clustering results by strategically selecting initial cluster centers. This initialization method helps achieve better convergence and avoid encountering suboptimal solutions. We use the Elbow[9] to determine the optimal number of clusters. This involves running algorithms for a range of cluster numbers and plotting the results of the within-cluster sum of squares based on the cluster

numbers. This method helps strike a balance between minimizing the within-cluster sum of squares and avoiding overfitting caused by using an excessive number of clusters.

4.3 Experimental results

In this subsection, we present the experimental results obtained from applying our proposed MTOClus algorithm to various datasets and evaluation methods.

4.3.1 Weight value study

The clustering results are influenced by various factors, primarily the multiple metapaths. The assigned weight values to each factor represent their relative importance. If the weight of a particular factor surpasses that of others, it indicates its greater impact on the final clustering results. Table 1 presents the weights of each similarity matrix calculated by executing Algorithm 1 across multiple datasets.

For the ACM target type paper, there are three metapaths. Among them, S-P and S-P-A-P have larger weights, indicating that these two metapaths contain more information in the corresponding similarity matrices, while S-P-T-P has a smaller weight, indicating less information content. Similarly, for the ACM target type author, three metapaths are identified. S-P-A and S-P-P-A have larger weights, suggesting more information in the corresponding similarity matrices, whereas S-P-T-P-A has a smaller weight, indicating less information content. Additionally, for the ACM target type term, S-P-T has a larger weight, indicating more information content in its corresponding similarity matrix, while S-P-P-T and S-P-A-P-T have relatively smaller weights, suggesting less information content.

Similarly, for the DBLP, the target type paper exhibits three metapaths. V-P and V-P-A-P have larger weights, indicating more information content in their respective similarity matrices, while V-P-T-P has a smaller weight, suggesting less information content. Regarding the target type author, V-P-A and V-P-A-P-A have larger weights, signifying more information content, whereas V-P-T-P-A has a smaller weight, indicating less information content. For the target type term, V-P-T and V-P-A-P-T have larger weights, indicating more information content, while V-P-T-P-T has a smaller weight, indicating less information content.

In the IMDB, for the target type movie, there are three metapaths. G-M and G-M-D-M have larger weights, indicating more information content, while G-M-A-M has a smaller weight, indicating less information content. Regarding the target type director, G-M-D and G-M-D-M-D have larger weights, indicating more information content, whereas G-M-A-M-D has a smaller weight, indicating less information content. For the target type actor, G-M-A and G-M-D-M-A have larger weights, indicating more information content, while G-M-A-M-A has a smaller weight, indicating less information content.

Lastly, in the KG20C, for the target type paper, there are four metapaths. C-P and C-P-A-P have larger weights, indicating more information content, while C-P-D-P and C-P-A-AFF-A-P have smaller weights, indicating less information content. Regarding the target type author, C-P-A has a larger weight, indicating more information content, whereas C-P-D-P-A and C-P-A-AFF-A have smaller weights, indicating less information content. Similarly, for the target types domain and affiliation, the patterns in the weights of their respective metapaths follow a similar trend. Specifically, metapaths such as C-P-D, C-P-A-P-D, C-P-A-AFF, and C-P-P-A-AFF display larger weights, indicating a higher amount of information content. Conversely, the metapath C-P-A-AFF-A-P-D, and C-P-D-P-A-AFF have relatively smaller weights, indicating less information content.

4.3.2 Clustering quality

Table 2 shows the clustering results produced by MTOClus. To evaluate the impact of n on the clustering outcomes, we executed the MTOClus algorithm with $n \in 1, 2, 3$. Following this, we assessed the clustering results, as depicted in Figure 3.

As shown in Figure 3, when a larger value of n is selected, the SSE and DBI values of the clustering results show a decreasing trend, while the SC values exhibit an increasing trend. This indicates that a larger n leads to better clustering results. A larger value of n results in the exclusion of more outliers during the clustering process, leading to a decrease in the number of nodes in the clustering results. This reveals the potential trade-off between clustering quality and the number of nodes in clustering results. We can set the value of n according to our specific requirements.

Table 1. Weight of each similarity matrix.

Dataset	Center type	Target type	Metapath	Weight	Dataset	Center type	Target type	Metapath	Weight
ACM	subject	paper	S-P	0.5569	DBLP	venue	paper	V-P	0.5077
			S-P-A-P	0.4423				V-P-A-P	0.4286
			S-P-T-P	0.0008				V-P-T-P	0.0637
		author	S-P-A	0.5706			author	V-P-A	0.5937
			S-P-P-A	0.4276				V-P-A-P-A	0.3442
			S-P-T-P-A	0.0018				V-P-T-P-A	0.0621
		term	S-P-T	0.4361			term	V-P-T	0.5548
			S-P-P-T	0.2914				V-P-A-P-T	0.3541
			S-P-A-P-T	0.2724				V-P-T-P-T	0.0911
IMDB	genre	movie	G-M	0.3865	KG20C	conference	paper	C-P	0.4226
			G-M-D-M	0.3950				C-P-A-P	0.3484
			G-M-A-M	0.2185				C-P-D-P	0.1157
		director	G-M-D	0.4383				C-P-A-AFF-A-P	0.1133
			G-M-D-M-D	0.3356			author	C-P-A	0.5714
			G-M-A-M-D	0.2261				C-P-D-P-A	0.1710
		actor	G-M-A	0.4045				C-P-A-AFF-A	0.2576
			G-M-D-M-A	0.3269			domain	C-P	0.4767
			G-M-A-M-A	0.2686				C-P-A-P-D	0.4014
								C-P-A-P-D	0.4014
								C-P-A-AFF-A-P-D	0.1219
							affiliation	C-P-A-AFF	0.3903
								C-P-P-A-AFF	0.4672
								C-P-D-P-A-AFF	0.1425

Table 2. Cluster outcomes derived from MTOClus.

Dataset	Center type	Number of target type nodes	Number of clusters obtained
ACM	subject	10886	5
DBLP	venue	26108	7
IMDB	genre	13325	15
KG20C	conference	16342	6

Table 3 presents the experimental results derived from the datasets. These results are generated by executing each algorithm five times. Subsequently, three evaluation metrics (SSE, SC, DBI) are computed for each algorithm on every dataset. The values displayed in Table 3 represents the averages of the five values obtained during multiple runs.

The experimental results showcased in Table 3 indicate that MTOClus consistently outperforms the six baseline algorithms across all three evaluation metrics. This underscores the superior clustering quality achieved by MTOClus compared to the six baseline algorithms.

For SSE, the results of MTOClus($n = 3$) on ACM, DBLP, IMDB, and KG20C are 53.51, 90.49, 155.24 and 84.03, respectively. These results significantly outperform the optimal values of 10601.41, 25006.62, 12808.00 and 15474.67 for the same task baselines. This indicates that MTOClus can produce more compact and cohesive clusters.

Regarding SC, MTOClus($n = 3$) achieves results of 0.12, 0.07, 0.04 and 0.11 on ACM, DBLP, IMDB, and KG20C, respectively, which are significantly better than the optimal values of 0.00, 0.00, 0.00 and 0.00 achieved by the baseline models for the same task. This demonstrates that MTOClus produces more effective and well-separated clustering results.

In terms of DBI, MTOClus($n = 3$) shows results of 1.85, 1.83, 1.90 and 1.77 on ACM, DBLP, IMDB, and KG20C, respectively. These results were significantly better than the optimal values of 2.00, 2.00, 2.00 and 1.95 achieved by the same task baselines. This indicates that MTOClus's clusters are more compact, better separated, and overall more effective.

The experimental results may be attributed to the sparse relationships between entities in HIN, leading to generally lower similarity values between nodes and, consequently, larger distances between nodes. Traditional clustering algorithms such as K-means, X-means, FCM, Bsas, Mbsas and SClump do not explicitly identify outliers, which can result in suboptimal values for internal evaluation metrics. Even if only a small number of outliers are identified when $n = 1$, the metric value is still better than the baselines.

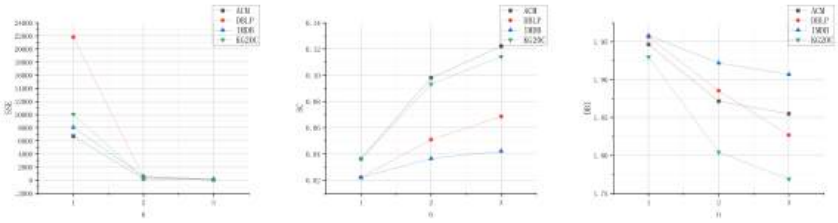


Figure 3. Trend of evaluation methods of different n.

Table 3. Experimental result.

Dataset	Algorithm	SSE	SC	DBI	Dataset	Algorithm	SSE	SC	DBI
ACM	K-means	10608.48	-0.04	2.15	DBLP	K-means	25086.48	-0.04	2.09
	X-means	10606.77	-0.03	2.15		X-means	25079.83	-0.04	2.03
	FCM	10601.41	-0.03	2.15		FCM	25073.36	-0.04	2.09
	Bsas	10602.01	-0.01	2.05		Bsas	25006.62	-0.03	2.06
	Mbsas	10612.49	-0.05	2.15		Mbsas	25006.62	-0.03	2.06
	SClump	10881.49	0.00	2.00		SClump	26103.30	0.00	2.00
	MTOClus($n=1$)	6700.57	0.04	1.95		MTOClus($n=1$)	21813.25	0.02	1.96
	MTOClus($n=2$)	213.25	0.10	1.87		MTOClus($n=2$)	488.61	0.05	1.89
	MTOClus($n=3$)	**53.51**	**0.12**	**1.85**		MTOClus($n=3$)	**90.49**	**0.07**	**1.83**
IMDB	K-means	12897.41	-0.02	2.08	KG20C	K-means	15492.36	-0.04	2.10
	X-means	12857.36	-0.02	2.09		X-means	15479.60	-0.03	2.10
	FCM	12861.33	-0.02	2.08		FCM	15474.67	-0.03	2.10
	Bsas	12808.00	-0.01	2.04		Bsas	15481.52	-0.02	1.95
	Mbsas	12808.00	-0.01	2.04		Mbsas	15481.51	-0.02	1.95
	SClump	13319.97	0.00	2.00		SClump	16337.19	0.00	2.00
	MTOClus($n=1$)	8065.52	0.02	1.96		MTOClus($n=1$)	10034.51	0.04	1.93
	MTOClus($n=2$)	597.24	0.03	1.92		MTOClus($n=2$)	371.46	0.09	1.80
	MTOClus($n=3$)	**155.24**	**0.04**	**1.90**		MTOClus($n=3$)	**84.03**	**0.11**	**1.77**

Table 4. Partial cluster result of ACM.

Center node	target type	Similar node
DATABASE MANAGEMENT	paper	On demand classification of data streams
		Mining massively incomplete data sets by conceptual reconstruction
		XRules: an effective structural classifier for XML data
	author	Jiawei Han
		Christos Faloutsos
		Jian Pei
COMPUTER-COMMUNICATION NETWORKS	paper	A scalable content-addressable network
		A Layered Naming Architecture for the Internet
		HLP: a next generation inter-domain routing protocol
	author	Scott Shenker
		Yang (Richard) Yang
		Dina Katabi
INFORMATION STORAGE AND RETRIEVAL	paper	Automatic labeling of multinomial topic models
		Mining multi-faceted overviews of arbitrary topics in a text collection
		Structured metric learning for high dimensional problems
	author	ChengXiang Zhai
		K. Selçuk Candan
		Abdulmohsen Algarni
PATTERN RECOGNITION	paper	SyMP: an efficient clustering approach to identify clusters of arbitrary shapes in large data sets
		Mining rare and frequent events in multi-camera surveillance video using self-organizing maps
		Model compression
	author	Marcos Salganicoff
		Jinbo Bi
		Rebecca Castano
ARTIFICIAL INTELLIGENCE	paper	Using graph-based metrics with empirical risk minimization to speed up active learning on networked data
		Machine learning for online query relaxation
		A heuristic method for optimizing an intercity data transmission network
	author	Naoki Abe
		Amr Ahmed
		Robert F. Murphy

In contrast, our proposed MTOClus adopts a comprehensive approach by aggregating similarity matrices, assigning weights to each matrix, computing the final similarity matrix, and identifying cluster centers. Furthermore, we introduce a method for calculating the cutoff distance. This holistic methodology consistently outperforms other algorithms in clustering HIN, resulting in more robust and effective clustering outcomes.

4.3.3 Case study

Taking MTOClus($n = 3$) as an example, clustering ACM can yield partial clustering results as shown in Table 4. In the provided table, five nodes have been identified as central nodes in ACM, DATABASE MANAGEMENT, COMPUTER-COMMUNICATION NETWORKS, INFORMATION STORAGE AND RETRIEVAL, PATTERN RECOGNITION and ARTIFICIAL INTELLIGENCE, respectively. We also presented the three papers and three authors with the highest similarity to each central node in the clustering results. After conducting online research, we discovered that these papers and authors have a strong correlation with the subject of their center node.

5. CONCLUSION

A novel HIN clustering algorithm, MTOClus, is proposed. MTOClus clusters nodes of multi-type objects in HIN and assigns weights to differentiate the importance of each metapath. Additionally, it automatically selects cluster centers through node sorting and identifies outliers in the network. Experimental results demonstrate that MTOClus outperforms six other algorithms in terms of SSE, SC, and DBI. Therefore, we can conclude that the proposed algorithm achieves a relatively higher clustering quality.

Acknowledgement

This work was supported by the National Natural Science Foundation of China grant No. 12261027.

REFERENCES

[1] Huang, Y. and Gao, X., "Clustering on heterogeneous networks," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **4**(3), 213–233 (2014).

[2] Das, S. and Biswas, A., "Deployment of information diffusion for community detection in online social networks: a comprehensive review," *IEEE Transactions on Computational Social Systems* **8**(5), 1083–1107 (2021).

[3] Shi, C., Kong, X., Yu, P. S., Xie, S., and Wu, B., "Relevance search in heterogeneous networks," in [*Proceedings of the 15th international conference on extending database technology*], 180–191 (2012).

[4] Sun, Y., Norick, B., Han, J., Yan, X., Yu, P. S., and Yu, X., "Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)* **7**(3), 1–23 (2013).

[5] Kong, X., Yu, P. S., Ding, Y., and Wild, D. J., "Meta path-based collective classification in heterogeneous information networks," in [*Proceedings of the 21st ACM international conference on Information and knowledge management*], 1567–1571 (2012).

[6] Madhulatha, T. S., "An overview on clustering methods," *arXiv preprint arXiv:1205.1117* (2012).

[7] Shi, C., Li, Y., Zhang, J., Sun, Y., and Philip, S. Y., "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering* **29**(1), 17–37 (2016).

[8] Shi, C., Kong, X., Huang, Y., Philip, S. Y., and Wu, B., "Hetesim: A general framework for relevance measure in heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering* **26**(10), 2479–2492 (2014).

[9] Thorndike, R. L., "Who belongs in the family?," *Psychometrika* **18**(4), 267–276 (1953).

[10] Liu, M., Xiong, Z., Ma, Y., Zhang, P., Wu, J., and Qi, X., "Dprank centrality: finding important vertices based on random walks with a new defined transition matrix," *Future Generation Computer Systems* **83**, 376–389 (2018).

[11] Fix, E., [*Discriminatory analysis: nonparametric discrimination, consistency properties*], vol. 1, USAF school of Aviation Medicine (1985).

[12] Han, H., Zhao, T., Yang, C., Zhang, H., Liu, Y., Wang, X., and Shi, C., "Openhgnn: an open source toolkit for heterogeneous graph neural network," in [*Proceedings of the 31st ACM International Conference on Information & Knowledge Management*], 3993–3997 (2022).

[13] Tran, H.-N. and Takasu, A., "Exploring Scholarly Data by Semantic Query on Knowledge Graph Embedding Space," in [*Proceedings of the 23rd International Conference on Theory and Practice of Digital Libraries*], 154–162 (2019).

[14] MacQueen, J. et al., "Some methods for classification and analysis of multivariate observations," in [*Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*], **1**(14), 281–297, Oakland, CA, USA (1967).

[15] Pelleg, D., Moore, A. W., et al., "X-means: Extending k-means with efficient estimation of the number of clusters.," in [*Icml*], **1**, 727–734 (2000).

[16] Bezdek, J. C., Ehrlich, R., and Full, W., "Fcm: The fuzzy c-means clustering algorithm," *Computers & geosciences* **10**(2-3), 191–203 (1984).

[17] Koutroumbas, K. and Theodoridis, S., [*Pattern recognition*], Academic Press (2008).

[18] Li, X., Kao, B., Ren, Z., and Yin, D., "Spectral clustering in heterogeneous information networks," in [*Proceedings of the AAAI Conference on Artificial Intelligence*], **33**(01), 4221–4228 (2019).

[19] Rousseeuw, P. J., "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics* **20**, 53–65 (1987).

[20] Davies, D. L. and Bouldin, D. W., "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence* (2), 224–227 (1979).

[21] Arthur, D. and Vassilvitskii, S., "K-means++ the advantages of careful seeding," in [*Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*], 1027–1035 (2007).

FPFS:Federated Privacy-Preserving Feature Selection with Privacy Techniques for Vertical Federated Learning

Linlong Wang[a], Chungen Xu[a], Pan Zhang[b], and Yiting Liu[a]

[a]School of Mathematics and Statistics, Nanjing University of Science and Technology, NanJing, China
[b]School of Cyber Science and Engineering, Nanjing University of Science and Technology, NanJing, China

ABSTRACT

In recent years, the increasing demand for data privacy has positioned federated learning (FL) as a promising approach for collaborative machine learning, allowing participants to preserve privacy while jointly training models. Vertical federated learning (VFL), where participants hold distinct feature sets for the same data cohort, introduces unique privacy challenges. While VFL prevents the sharing of raw data, intermediate results such as the gini coefficient can still reveal sensitive information. We presents a privacy-preserving feature selection framework tailored for VFL, designed to prevent the server from inferring the client's feature distribution through intermediate computation parameters. By integrating homomorphic encryption (HE) and Differential Privacy (DP), the framework enables collaborative computation without exposing raw data, while the added noise enhances data privacy protection. Experimental evaluations demonstrate that the proposed feature selection method FPFS consistently achieves superior accuracy and efficiency across various datasets, particularly excelling in effective feature selection and privacy preservation. Compared to other methods, FPFS maintains higher model performance across multiple experimental scenarios and effectively mitigates both internal and external attacks.

Keywords: Vertical federated learning, Feature selection, Data security, Differential privacy

1. INTRODUCTION

In recent years, the increasing demand for data privacy protection has made federated learning (FL)[1–3,14] a promising approach for collaborative machine learning, allowing participants to preserve privacy while jointly training models.[4,5] In traditional centralized learning paradigms, participants' data is collected by a central server, which significantly increases the risk of privacy breaches.[12] FL addresses this issue by enabling local model training at each participant's site, sharing only the model parameters and thus reducing direct exposure of raw data.[16]

VFL in which participants hold different feature sets for the same cohort of data samples, presents additional privacy challenges.[5,13] While VFL avoids sharing raw data, the intermediate results exchanged during collaborative learning, such as Gini coefficients, can still leak sensitive information about the participants' private datasets.[7,15] This risk becomes particularly concerning when the server or other participants can infer sensitive feature distributions from shared intermediate results.[13] Additionally, achieving computational efficiency in VFL remains a challenge, as privacy-preserving mechanisms like HE and DP often introduce substantial computational and communication overhead, particularly when applied to large-scale datasets.[5,10]

To mitigate these challenges, we propose a novel privacy-preserving feature importance[8,9,11] initialization scheme designed for VFL, which integrates HE and DP.[7,19] In our scheme, participants encrypt their local feature data using HE, allowing secure computations to be performed on encrypted data without revealing sensitive

Further author information:
Linlong Wang: E-mail: wanglinlon2000@163.com;
Chungen Xu: E-mail: xuchungen@gmail.com;
Pan Zhang: panzhang@njust.edu.cn;
Yiting Liu: 919107810310@njust.edu.cn;

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190D · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058016

information. DP is used to introduce controlled noise into intermediate results, preventing the server or any participant from inferring sensitive information through inference attacks.[15,18] This design ensures that no participant, including the server, can deduce private data, such as feature distributions, from Gini coefficients.[15,17]

The main contributions of this paper are summarized as follows:

- We propose a comprehensive privacy-preserving scheme that combines HE and DP, ensuring that in VFL settings, no participant, including the server, can infer the private data of others. The scheme prevents the server from deducing feature distributions from Gini coefficients and ensures that clients cannot access the server's label data.
- We enhance computational efficiency by incorporating feature selection and dimensionality reduction techniques, which reduces the complexity of feature importance computations. Additionally, distributed batch processing and parallelism significantly improve scalability for handling large datasets.
- Despite the introduction of DP noise, our empirical evaluations demonstrate that the proposed scheme maintains high accuracy in feature importance rankings, preserving the integrity of the feature selection process without introducing significant distortions due to noise.

In conclusion, the proposed scheme offers a robust and efficient privacy-preserving solution for feature selection and feature importance evaluation in vertical federated learning. It is particularly well-suited to scenarios involving large-scale datasets and strict privacy requirements. Our experimental results validate the effectiveness of the scheme in balancing strong privacy protection with computational efficiency, highlighting its potential for practical applications in real-world federated learning systems.

2. PRELIMINARIES

Before introducing the proposed feature importance initialization scheme, it is crucial to review key cryptographic and privacy-preserving techniques relevant to federated learning, particularly HE and DP.

2.1 Homomorphic Encryption

HE allows computations on encrypted data, with the results remaining valid upon decryption, as if operations had been performed on the plaintext. This is essential in federated learning, where participants can collaboratively compute on distributed data without exposing their raw data. HE ensures secure aggregation and processing, preserving privacy throughout the computation.

2.1.1 CKKS Encryption Scheme

The CKKS (Cheon-Kim-Kim-Song) scheme is optimized for operations on floating-point numbers, supporting addition and multiplication on encrypted data. CKKS is efficient for large-scale federated learning tasks due to its ability to perform approximate computations with negligible error and its support for batch processing, allowing multiple data items to be encrypted and processed simultaneously.

2.2 Differential Privacy

DP introduces noise to computation results to protect individual data points from being inferred. It ensures that the inclusion or exclusion of a single data point does not significantly alter the outcome. In federated learning, DP prevents attackers from deducing private information from shared results by adding noise proportional to the privacy budget ϵ, maintaining a balance between privacy and accuracy. The noise-added result is:

$$\hat{y} = y + \text{Lap}\left(\frac{1}{\epsilon}\right), \tag{1}$$

where $\text{Lap}\left(\frac{1}{\epsilon}\right)$ represents the Laplace noise with scale $\frac{1}{\epsilon}$.

2.3 Gini Impurity

Gini impurity is a metric used to evaluate the importance of features in classification tasks by measuring the probability of misclassification. The formula for Gini impurity is:

$$G(f_j) = 1 - \sum_{k=1}^{c} p_k^2, \tag{2}$$

where p_k represents the proportion of samples in class k. Lower Gini impurity indicates higher feature importance, making it useful for selecting the most discriminative features.

In federated learning, Gini impurity can be securely computed using HE and DP, allowing feature importance evaluation without exposing raw data. This section outlines the essential cryptographic techniques used in our scheme to ensure privacy-preserving feature importance computation in VFL.

3. PROBLEM FORMULATION

In this section, we define the threat model and outline the Security and Accuracy Objectives for our proposed scheme. These elements are essential for understanding potential security risks, adversary capabilities, and the requirements our scheme must meet to ensure data privacy and computational correctness.

3.1 Threat Model

In VFL, multiple participants and potential adversaries require a clear threat model to address potential attacks. Adversaries can be categorized as internal malicious participants or external eavesdroppers.

Malicious clients may attempt to infer private data by analyzing encrypted data or reverse-engineering computation results, aiming to extract both feature and label information. Conversely, a malicious server could deduce client-specific data by decrypting intermediate results or analyzing statistical outputs to infer feature distributions or label patterns. Addressing these threats is key to maintaining data privacy and integrity in federated learning.

3.2 Security and Accuracy Objectives

To counter the outlined threats, we define the following security and accuracy objectives:

3.2.1 Security Goals

Our scheme ensures data privacy through HE, allowing computations on encrypted data, while DP adds noise to prevent inference attacks on client data distributions. Additionally, the server's label data remains protected through HE, ensuring no unauthorized access during computation.

3.2.2 Accuracy Goals

While privacy is maintained through DP, the introduction of noise is carefully calibrated according to the privacy budget ϵ, ensuring that model performance remains accurate without significant degradation. The scheme also emphasizes computational efficiency, minimizing overhead in large-scale federated learning settings.

4. OUR PROPOSED SCHEME

In VFLsecure collaboration between the server and clients is crucial to protect data privacy. This section details the process of how the server and clients perform collaborative computations to ensure data security.

Initially, the server holds the label matrix A and encrypts it using a homomorphic encryption scheme, such as CKKS, to ensure the privacy of the label data during subsequent computations. The encrypted matrix $[A]$ is computed as follows:

$$[A] = Enc(A, pk), \tag{3}$$

The server uses the public key pk to encrypt the label matrix, ensuring data security during transmission. After encryption, the encrypted label matrix $[A]$ is generated and sent to the clients.

Upon receiving the encrypted matrix $[A]$, the clients begin processing their feature data $f_{m,j}$. This involves two main steps: variance threshold filtering and PCA for dimensionality reduction.

Algorithm 1 Privacy-Preserving Feature Selection

Input: Server: Label matrix A; **Clients:** Feature matrices $f_{m,j}$, privacy budget ϵ.
Output: Final Gini coefficients $G(f_j)$ with privacy protection.
Server:
Encrypts the label matrix A using HE: [A] = Enc(A, pk)
Sends $[A]$ to all clients.
Clients:
for each client $m \in [M]$ **do**
    Compute variance σ_j^2 for each feature $f_{m,j}$.
    Filter out features where $\sigma_j^2 < \theta_{\sigma^2}$.
    Apply Principal PCA to reduce feature dimensions, selecting top k principal components.
    Encrypt reduced feature matrix $f'_{m,j}$:

$$[f'_{m,j}] = Enc(f'_{m,j}, pk),$$

    Perform HE with $[A]$ to compute Gini coefficients:

$$[G(f_j)] = \text{HE\_Compute}([f'_{m,j}], [A]),$$

    Add Laplace noise :

$$[G(f_j)_{\text{private}}] = [G(f_j)] + \text{Lap}\left(\frac{1}{\epsilon}\right),$$

    Send $[G(f_j)_{\text{private}}]$ to the server.
end for
Server:
Decrypt the Gini coefficients:

$$G(f_j)_{\text{decrypted}} = Dec([G(f_j)_{\text{private}}], sk).$$

Aggregate results and send final Gini coefficients $G(f_j)$ back to the clients for further computation.
Return final Gini coefficients with privacy protection.

The client first computes the variance σ_j^2 of each feature to evaluate its contribution. Features with lower variance indicate less variation across samples and are considered less important. A threshold θ_{σ^2} is set, and features with variance below this threshold are filtered out. The variance is calculated as:

$$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(f_{j,i} - \mu_j)^2, \tag{4}$$

where N is the number of samples, $f_{j,i}$ represents the value of feature f_j for sample i, and μ_j is the mean of feature f_j. After filtering, the remaining features undergo PCA for dimensionality reduction. PCA computes the covariance matrix Σ of the data:

$$\Sigma = \frac{1}{N-1}X^T X, \tag{5}$$

where X is the centralized data matrix. The eigenvalue decomposition of the covariance matrix is then performed:

$$\Sigma v = \lambda v. \tag{6}$$

The top k eigenvectors corresponding to the largest eigenvalues are selected as the new projection basis. The original data is projected onto this basis, yielding the reduced feature matrix $f'_{m,j}$.

After dimensionality reduction, the clients encrypt their reduced feature matrix $f'_{m,j}$ using the same homomorphic encryption scheme to produce $[f'_{m,j}]$. The encrypted feature matrix is then used to perform secure computations with the server's encrypted label matrix $[A]$, resulting in encrypted Gini coefficients.

The Gini coefficient is used to measure the importance of features. Each client computes the Gini coefficient in the encrypted domain by performing homomorphic operations. For a feature f_j in batch b, the Gini coefficient is calculated as:

$$G(f_j) = 1 - \sum_{k=1}^{c} p_k^2, \tag{7}$$

where p_k is the probability of category k, and c is the number of categories. The client performs homomorphic operations to compute the squared probabilities:

$$[p_k^2] = [p_k] \cdot [p_k]. \tag{8}$$

The result is an encrypted Gini coefficient, denoted as $[G(f_j)]$. To prevent the server from inferring the clients' feature distributions from the computed Gini coefficients, the clients apply differential privacy by adding noise to the computed Gini coefficients. Specifically, Laplace noise is added to each Gini coefficient:

$$G(f_j)_{\text{private}} = G(f_j) + \text{Lap}\left(\frac{1}{\epsilon}\right), \tag{9}$$

where ϵ is the privacy budget controlling the scale of the noise, and $\text{Lap}\left(\frac{1}{\epsilon}\right)$ represents the Laplace distribution with scale parameter $\frac{1}{\epsilon}$. The Gini coefficients with added noise remain encrypted, denoted as $[G(f_j)_{\text{private}}]$. The clients then send the encrypted, noisy Gini coefficients back to the server.

Upon receiving the encrypted Gini coefficients from the clients, the server uses its private key sk to decrypt the Gini coefficients:

$$G(f_j)_{\text{decrypted}} = Dec([G(f_j)_{\text{private}}], sk). \tag{10}$$

The decrypted Gini coefficients, still containing differential privacy noise, ensure that the server cannot infer the exact feature distributions of the clients. The server can either analyze the decrypted Gini coefficients or send them back to the clients for further computation or feature importance analysis. This workflow combines homomorphic encryption and differential privacy to ensure the security and privacy of the data during the computation of feature importance in VFL. The clients' feature data and the server's label data remain secure throughout the process, enabling secure and privacy-preserving collaborative computation.

5. SECURITY ANALYSIS

This section evaluates the security of the proposed scheme, focusing on preventing data leakage, mitigating inference attacks, and defending against internal and external adversaries. By integrating HE and DP, the scheme enhances privacy protection in the federated learning framework.

5.1 Preventing Data Leakage and Inference Attacks

In traditional distributed computing, the server and clients risk exposing sensitive information. For example, the server could infer the client's data distribution from intermediate results, while the client may deduce the server's label distribution.

To mitigate these risks, the scheme uses HE, allowing the server to compute on encrypted client data without accessing plaintext. Additionally, DP adds noise to results, preventing the server from reliably inferring the true Gini coefficient or the underlying data distribution:

$$G' = G + \epsilon. \tag{11}$$

This ensures that even if the server receives noisy results, inference attacks are mitigated.

5.2 Preventing Internal and External Attacks

Internal attacks occur when participants attempt to infer others' private data, while external attacks involve intercepting communication. The scheme addresses both threats through HE, ensuring computations occur on encrypted data, and DP, adding uncertainty to results. Even intercepted communications yield only encrypted data, making decryption infeasible without the private key.

6. PERFORMANCE EVALUATION

6.1 Experimental Setup

Experiments were conducted on a workstation with a 12 vCPU Intel Xeon Platinum 8255C CPU @ 2.50GHz and an NVIDIA GeForce RTX 3090 GPU. All models were implemented using PyTorch.

6.2 Datasets and Models

We utilized four image datasets: CIFAR-10, CIFAR-100, CINIC-10 , and BHI for image recognition tasks in VFL model structures. CIFAR-10 and CIFAR-100 consist of 60,000 samples each, while CINIC-10 contains 270,000 samples. BHI includes 86,000 medical images for breast cancer prediction. Additionally, the MADELON, RELATHE, and FRIEDMAN datasets were used for binary classification tasks, featuring both synthetic and real-world data. For the image datasets, each client operates on one half of the image, and the server uses a four-layer fully connected network (FCNN-4) to aggregate results.

6.3 Test Accuracy and R^2 Score

This section analyzes the performance of various feature selection methods based on test accuracy (testAcc%) and R^2 scores, with the horizontal axis indicating the number of features used for training.

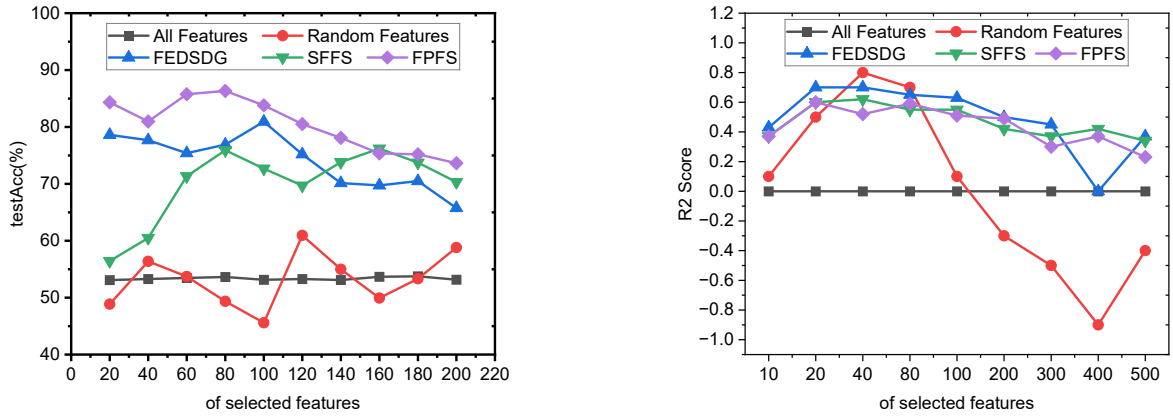
FPFS consistently achieved higher test accuracy, particularly when using fewer features. For instance, with 20 and 40 features, FPFS attained accuracies of 84.33% and 80.97%, outperforming other methods. As feature counts increased, FPFS maintained strong performance, demonstrating its efficiency in extracting critical features and enhancing overall model accuracy.

FedSDG also showed relatively high accuracy but was generally inferior to FPFS. For example, at 60 and 100 features, FPFS achieved 85.75% and 83.81%, compared to FEDSDG's 75.4% and 80.93%, suggesting that FedSDG is less effective at noise reduction.

As in Fig.1 SFFS, random features, and All features exhibited lower performance. Particularly, Random and All Features methods showed poor generalization, with accuracies of only 49.36% and 53.64% at 80 features, highlighting the limitations of unoptimized feature selection.

R^2 score analysis further confirmed FPFS's advantage. At 20 and 100 features, FPFS achieved R^2 scores of 0.6 and 0.63, indicating strong noise filtering capabilities. Conversely, FedSDG had scores of 0.7 and 0.45, revealing less effective noise suppression. SFFS and random features demonstrated significant declines in R^2 scores, particularly when more features were selected.

In summary, FPFS excels at maintaining high accuracy and R^2 scores with fewer selected features, proving its effectiveness in extracting relevant features and reducing noise. While FedSDG showed some feature selection capability, it was less efficient than FPFS. SFFS, random Features, and a lll features faced substantial limitations, particularly with increased feature counts, underscoring the need for optimized selection strategies. These findings affirm FPFS as a robust and efficient method for feature selection and model optimization.

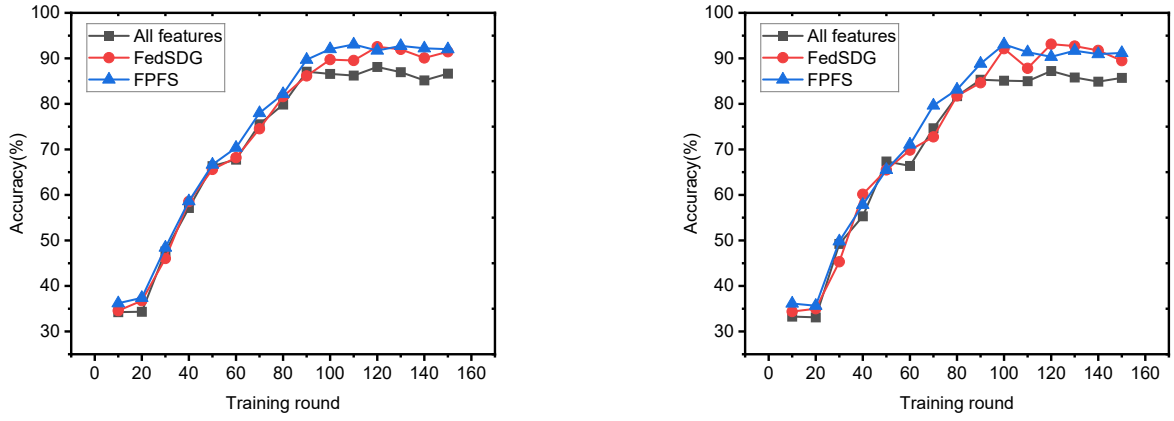


Testacc-VFLNN-MADELON R2-VFLNN-MADELON

Figure 1. Test accuracy and R2 scores vs. number of selected features on synthetic datasets.

In terms of R^2 scores, FPFS also outperformed other methods, achieving scores of 0.6 and 0.63 with 20 and 100 features. FedSDG exhibited lower scores, while random features saw a significant decline as the number of features increased, highlighting the benefits of noise reduction in FPFS.

6.4 Learning Accuracy Analysis



VFLNN-MADELON VFLNN-RELATHE

Figure 2. Learning accuracy on various datasets.

In Fig. 2, we conducted experiments to assess the impact of different learning rates across varying training epochs, using the MADELON and RELATHE datasets to compare the accuracy trends of FPFS, FedSDG, and an all-features approach.

On the MADELON dataset, FPFS consistently outperformed the other methods across most epochs. At 50 and 100 epochs, FPFS achieved accuracies of 66.66% and 92.05%, surpassing FedSDG (65.59% and 89.73%) and

the all-features approach (66.34% and 86.59%). These results highlight FPFS's capacity for optimizing learning via effective feature selection and adaptive learning rate adjustments.

While FedSDG was competitive, particularly in early epochs, it generally trailed FPFS, achieving 91.45% accuracy at 150 epochs compared to FPFS's 92.02%. In contrast, the all-features approach lagged significantly, with only 79.86% accuracy at 80 epochs, reflecting the limitations of including unfiltered features.

A similar pattern emerged on the RELATHE dataset, where FPFS led in accuracy, achieving 79.64% and 93.08% at 70 and 110 epochs, respectively, outperforming both FedSDG and the all-features method. Although FedSDG remained close, FPFS consistently maintained a slight advantage, while the all-features approach demonstrated weaker performance, with 91.19% accuracy at 150 epochs—still below FPFS's 92.02%.

Overall, FPFS achieved higher accuracy across training epochs, emphasizing the benefits of targeted feature selection and adaptive learning. FedSDG proved competitive but less robust than FPFS, while the all-features approach suffered from performance issues, underscoring the importance of optimized feature selection.

6.5 Efficiency Analysis

We conducted experiments to evaluate the impact of different learning rates across varying training epochs, using the MADELON and RELATHE datasets. The results compare the accuracy trends of three methods: FPFS, FedSDG, and all features.

For the MADELON dataset, FPFS consistently outperformed the other methods across most epochs. At 50 and 100 epochs, FPFS achieved accuracies of 66.66% and 92.05%, higher than FedSDG (65.59% and 89.73%) and All Features (66.34% and 86.59%). This indicates FPFS's efficiency in optimizing learning through effective feature selection and learning rate adaptation.

FedSDG performed competitively, especially in early stages, but generally fell short compared to FPFS. For instance, at 150 epochs, FPFS maintained 92.02% accuracy, slightly ahead of FedSDG's 91.45%. all features lagged behind, reaching only 79.86% at 80 epochs, showing the drawbacks of including unfiltered features.

A similar pattern was observed for the RELATHE dataset. FPFS led in accuracy, achieving 79.64% and 93.08% at 70 and 110 epochs, surpassing FedSDG and All Features. While FedSDG remained close, FPFS maintained a slight edge throughout. all features showed weaker performance, with 91.19% accuracy at 150 epochs, still lower than FPFS's 92.02%.

Overall, FPFS consistently achieved higher accuracy across training epochs, demonstrating effective feature selection and adaptive learning. While all features suffered from performance issues due to the inclusion of irrelevant data, highlighting the need for optimized feature selection.

7. CONCLUSION

In this paper, we introduced a privacy-preserving feature selection framework specifically designed for vertical federated learning. The primary objective is to prevent the server from inferring the client's feature distribution based on intermediate parameters transmitted during collaborative computation. Our framework robustly protects client data while ensuring high model accuracy and computational efficiency. Experimental results across multiple datasets validate the effectiveness of the proposed scheme.

For future work, optimizing homomorphic encryption and secure multiparty computation will be essential to reduce computational and communication overhead, especially for larger datasets. Additionally, extending the framework to multi-party settings and exploring more adaptive privacy-preserving mechanisms to balance privacy and accuracy across diverse scenarios are key directions for further improvement.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62072240, by the National Natural Science Foundation of China under Grant 62202228, and by the Natural Science Foundation of Jiangsu Province under Grant BK20210330.

REFERENCES

[1] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

[2] Li, Anran, et al. "Privacy-preserving efficient federated-learning model debugging." IEEE Transactions on Parallel and Distributed Systems 33.10 (2021): 2291-2303.

[3] Li, Anran, et al. "Efficient federated-learning model debugging." 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021.

[4] Khan, Afsana, Marijn ten Thij, and Anna Wilbik. "Communication-efficient vertical federated learning." Algorithms 15.8 (2022): 273.

[5] Chen, Tianyi, et al. "Vafl: a method of vertical asynchronous federated learning." arXiv preprint arXiv:2007.06081 (2020).

[6] Li, Xiling, Rafael Dowsley, and Martine De Cock. "Privacy-preserving feature selection with secure multiparty computation." International Conference on Machine Learning. PMLR, 2021.

[7] Li, Anran, et al. "FedSDG-FS: Efficient and secure feature selection for vertical federated learning." IEEE INFOCOM 2023-IEEE Conference on Computer Communications. IEEE, 2023.

[8] Song, Le, et al. "Feature Selection via Dependence Maximization." Journal of Machine Learning Research 13.5 (2012).

[9] Roy, Debaditya, K. Sri Rama Murty, and C. Krishna Mohan. "Feature selection using deep neural networks." 2015 international joint conference on neural networks (IJCNN). IEEE, 2015.

[10] Zhang, Yifei, and Hao Zhu. "Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning." arXiv preprint arXiv:2007.06849 (2020).

[11] Guyon, Isabelle, et al. "Result analysis of the nips 2003 feature selection challenge." Advances in neural information processing systems 17 (2004).

[12] Erkin, Zekeriya, et al. "Privacy-preserving face recognition." Privacy Enhancing Technologies: 9th International Symposium, PETS 2009, Seattle, WA, USA, August 5-7, 2009. Proceedings 9. Springer Berlin Heidelberg, 2009.

[13] Gu, Bin, et al. "Federated doubly stochastic kernel learning for vertically partitioned data." Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery data mining. 2020.

[14] Melis, Luca, et al. "Exploiting unintended feature leakage in collaborative learning." 2019 IEEE symposium on security and privacy (SP). IEEE, 2019.

[15] Fu, Chong, et al. "Label inference attacks against vertical federated learning." 31st USENIX security symposium (USENIX Security 22). 2022.

[16] Li, Liping, et al. "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets." Proceedings of the AAAI conference on artificial intelligence. Vol. 33. No. 01. 2019.

[17] Ryffel, Theo, et al. "A generic framework for privacy preserving deep learning." arXiv preprint arXiv:1811.04017 (2018).

[18] Chen, Xiangquan, et al. "PPAPAFL: A Novel Approach to Privacy Protection and Anti-poisoning Attacks in Federated Learning." International Conference on Testbeds and Research Infrastructures. Cham: Springer Nature Switzerland, 2023.

[19] Tao, Hongyao, Chungen Xu, and Pan Zhang. "A Fast and Accurate Non-interactive Privacy-Preserving Neural Network Inference Framework." International Conference on Testbeds and Research Infrastructures. Cham: Springer Nature Switzerland, 2023.

Renal Tumor Classification and Detection Based on Artificial Intelligence

Raflaa Hilmi Al-taie[a], Nashwan J. Hussein[b]
[a]Physiology and Organ Functions Department, College of Medicine, University of Babylon, Iraq;
[b]Department of Electrical Power Engineering, AL-Hussain University College, Department of Programming, Collage of Information Technology, University of Babylon, Iraq

ABSTRACT

The kidney is a crucial organ in the human body, co-operating billions of pipelines to cleanse the body's water. Kidney failure, renal tumor happens when cells divide uncontrollably and form an aberrant collection of cells surrounding or within the kidney. This cell type has the ability to disrupt regular kidney activity and destroy healthy cells. The prompt diagnosis of renal tumor is critical since they can be fatal if left untreated. Because it is dependent on the skill of the person analyzing the images, the traditional approach of manually checking the MR image may not be very accurate. The study concentrated on the diagnosis of normal and normal renal tumor. To enhance accuracy and expedite diagnosis, using publicly accessible individual records, the present research used methodologies for machine learning, involving support vector machine learning (SVM), adaptive optimization (AO), as well as gradient enhancement (GE). It uses an approach for making the dataset reduced multidimensional. Image feature extraction is a data preprocessing method that minimizes the time required to train the proposed algorithm. The accuracy rates of the algorithms for diagnosing normal and up normal are reported to be 88.8 % for GB, 83.8 % for ADA, 86.1 % for SVM and 93.3% for KNN and for up normal diagnosis tumor are reported to be 55.3 % for GB, 55.4 % for SVM, 55.3 % for ADA and 51.0% for KNN.

Keywords: Renal Tumor Detection, K-nearest Neighbors, Magnetic Resonance Imaging, Gradient Boosting GB, Boosting Algorithm.

1 INTRODUCTION

Medical image of RMI, it is a technical medical image used magnetic field with non –invasive procedure and radio waves. With to be a detailed image and produce deep information images of internal structure. It is a type of image uses the power of result of these images (MRI) contain tissues and organ details. The main purpose of this kind of this photo to medical diagnose for many unhealthy cases such as, Breast cancer, suddenly stroke problem, neurological problem and other illness of heart disease. The main idea of using magnetic powerful in MRI image to utilizing from the align characteristics of hydrogen atoms in the human body. Based on emitting the signal which can be detected by the RMI device. Nevertheless, the MRI devices support a high degree of human body and save of it without and reflect radio to the body with high degree of accuracy to allowing the doctor and physician with wide range of medical condition. MRI can be used to detect tumors, diagnose heart disease, and monitor the progression of neurological disorders. It can also be used to detect abnormalities in the kidney , spine, and other organs [1]. A renal tumor is an anomalous proliferation of cells in the kidney. It can be either benign or malignant. The factors that can cause a renal tumor are diverse, including genetics, radiation, and environmental exposures. And many others can make the abnormal cells in kidney happen also can be the foods. Diagnosis of a Renal Tumor typically involves imaging tests such as a CT scan or MRI. These tests can help doctors regulate the magnitude, shape, and location of the carcinoma. An examination may too be accomplished to govern if the tumor is mortal.[2] Machine learning manners have shown tremendous promise in medicinal concept analysis, containing the discovery and categorization of various afflictions, in the way that kidney tumors, pleura malignancy, bosom cancer, and more, Machine learning possibly used to organize brain Cancer MRI countenances into diverse categories founded on the closeness or absence of tumors and their essence. Machine learning algorithms, such as decision trees, support vector machines (SVMs), and random forests, can be used for classifying Renal malignant MRI images, based on two parameters first parameter is the size of dataset and the second is the complexity of it and these parameters will be decided about the requirement of it [3]. This is a common application of machine learning in the

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190E · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058690

medical field; where it may assist doctors create more accurate and efficient diagnoses. The methods that follow could be used for building a machine learning model for this task:

1- Data collection: Collect MRI images of Renal Tumor, including both benign and malignant tumors. These images should be labeled with the corresponding tumor type or grade.
2- Pre-processing: Pre-process the MRI images by removing noise, normalizing intensity levels, and segmenting the tumor region from the background.
3- Feature extraction: Extract relevant features such as texture, shape and intensity from pre-processed MRI images. These features should capture the characteristics that distinguish different types of Renal tumor.
4- Choose the model: Specify a machine learning algorithm, such as supported vector machines (SVM), Random Forests, or Convolutional neural networks, also called neural networks, (CNN), that is appropriate for the classification problem.
5- Train the model: Train the selected machine learning model using the tagged MRI images and extracted features. The model should learn to classify new MRI images into the correct tumor type or grade.
6- Assess the trained model's effectiveness using an independent collection of MRI pictures that were not utilised i n training.
   Utilise indicators like accuracy, precision, recall, and F1 score to assess the performance of the model.
7- Improve performance on the evaluation set by optimizing model parameters.

The model can be used to classify new renal tumor MRI images and support clinical decision making once trained and optimized. In [4] created a system with four stages. First, a normalization process and Binarization are accomplished. Next, GLCM processes morphological features extracted from Binarized images. Lastly, Anisotropic Diffusion is applied to produce the final result. A second Neural Network, called the LVQ for classification, has been added to the process of classification. This is because Ad boost NN was implemented as the first Neural Network. Clinical kidney MRI images obtained 95% accuracy and 80.6% success rates with 79.3% and 69.9% DDSM results.[5] In [6] use Adaptive Threshold Selection Network, or ATSN, to address lack of data. There is essentially a pair of phases to ATSN: testing and training. For the purpose to generate a flexible threshold, we pre-process simultaneously our test and real-world pictures during the training phase. Using thresholding, a tumor segment is obtained from the test image throughout the testing phase. We used the 2295 test images in three renal cancer categories. In [7] A classification system is presented in this work for analyzing Kidney images captured in different colors. GMB Renal tumor, with a grade IV rating, has been chosen to use as the basis for classification. Models that can subsequently categories fresh photographs have been trained using four selected photos from each of the four patients. Authorities identified five distinct categories for the photos. These comprised Dura mater, tumor, venous and arterial blood vessels, and healthy tissue. SVM provided better results than RF in most cases. The general accuracy achieved by SVM was 97%. In [8] in the current review, several methods have been used to classify brain cancer. These methods enable the successful pre-processing of images, the extraction of features and the subsequent classification of renal cancer. Three distinct machine learning approaches—Hybrid Classifier (SVM-KNN), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM)—are used to categories fifty pictures. Based on the results, SVM-KNN outperformed the rest, correctly categorizing 93% of the cases. This paper's primary objective is to provide a good MRI Renal cancer categorization result. In [9], the authors proposed to separate renal tumor tissue from renal MRI images using renal tumor segmentation. The pre-processing of the MRI images should include skull stripping and median filtering, and the thresholding process is carried out on the provided MRI images using the watershed segmentation approach. Finally, a segmented tumor area is discovered. In a further step, features were retrieved using GLCM techniques using MATLAB software. Support vector machine (SVM) categorization of some of the photos then had an average accuracy of 90%. In [10] the development of a method for the identification of renal tumors (BT) is presented in this paper. Figure 1 depicts the color image for kidney tumor and the right of image explains the tumor (red color tumor).

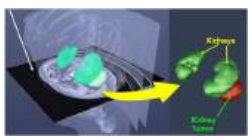


Figure1: explain the color Image of Kidney Tumor left Side Healthy and Right Side with Tumor which growth and spread through the kidney tissue

And figure 2 Explain the Dataset collected from available education sites using after Processing Algorithm to prepare the comfortable.

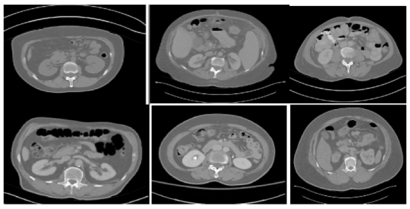


Figure 2: Explain the Dataset collected from available education sites using after Processing Algorithm to prepare the comfortable Images.

2 PROPOSED METHOD

The primary goal and motivation for this research paper are to develop a system for classifying renal tumors using Kidney images. Figure 3 depicts the proposed methodology's block diagram. The following sub-sections are discussed in depth in this section: the used dataset and the proposed methodology.

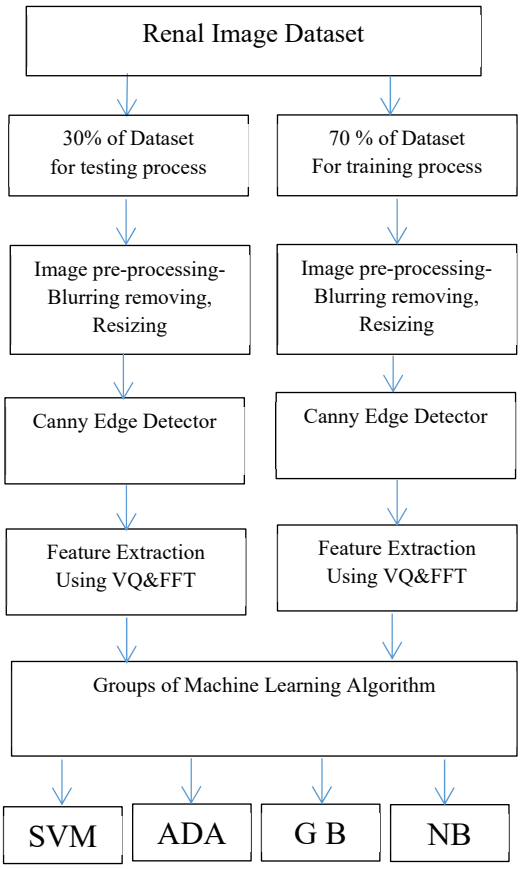


Figure 3: Explain Proposed System and draw the algorithm and Method Using to Detect Kidney Tumor with procedure of using of tumor detection by SVM, ADA, GB, and NB algorithms.

3 BOOSTING ALGORITHM

3.1 HOW TO CHANGE A PICTURE FROM RGB TO GREYSCALE

When an RGB image is transformed to a greyscale image, each pixel's colour information is reduced to a single intensity value. The main image colour or RGB channels of each pixel are typically combined [11-12], and the luminosity method formula is

Gray-value = 0.21 * R + 0.72 * G + 0.07 * B (1)

3.2 USING HISTOGRAM

The occurred at each pixel value in the captured image. It requires an integer value in the range [0-255], where 0 represents pure black and represents 255 pure white. All values in this range represent different shades of Gray. One pre-processing method for enhancing low-contrast photos is histogram equalization. One pre-processing method for enhancing low-contrast photos is histogram smoothing. Increases the dynamic range of the pixel values and equalizes all pixels to produce a uniform and smooth histogram with high-contrast images [13]. Image histogram shows the number of times the frequency.

3.3 GAUSSIAN FILTERING

Gaussian blur is commonly used in MRI image processing to process of reduce of noising. MRI images are often affected by noise due to the complex nature of the imaging process; it may cause unintentional distortions and artifacts in the picture. Nonetheless, it's crucial to remember that the blur level should be properly adjusted to prevent losing crucial details from the picture. Over-blurring can result in loss of detail and may affect the accuracy of subsequent image analysis. The size of the Gaussian blur kernel should therefore be carefully selected in accordance with the specific requirements of the application. To clear up, Gaussian blur is a beneficial approach when analyzing MRI images. It improves image quality and decreases noise. Before using other image analysis techniques, it can be employed as a pre-processing step, however precaution should be used to avoid over-blurring and the loss of crucial information [14-15].

3.4 FEATURE EXTRACTION

It is the process of using by the important, selecting and transforming data or information with relevant details and changing after raw data case to the set of features, which can be used for algorithm of machine learning, statistical analysis, or other applications. Therefor the process of decrease and enhance the performance respectively can named as feature extraction, of machine learning models. In image processing, feature extraction involves analysing an image to identify relevant characteristics, such as edges, corners, shapes, textures, or colors.

3.5 THE FFT-BASED FEATURE EXTRACTION

The Fast Fourier Transform (FFT) is a mathematical algorithm that is commonly used to analyses signals and images in various fields, including medical imaging. In MRI imaging, the FFT will be work as the extraction of frequency information from the image. MRI images are essentially 3D arrays of data, where each data point represents the depth of a pixel in the image. The FFT can be applied to each dimension of the image (x, y, and z) separately, or to a 2D slice of the image. The FFT works by converting a signal from time or space to frequency. In MRI images, the spatial domain refers to the location of each pixel in the image, while the frequency domain refers to the frequency content of the image.[16]applying the FFT to an MRI image can reveal important frequency information about the image, such as the presence of periodic patterns or the distribution of high and low frequency components. This can be useful for tasks such as image enhancement, noise reduction, or feature extraction. To apply the FFT to an MRI image, the following steps can be taken: Convert the image to gray scale if it is in color.

Take a 2D slice of the image or apply the FFT to each dimension of the image separately.

Apply the FFT using a suitable algorithm such as the Fast Fourier Transform.
Analyse the frequency spectrum to extract relevant information about the image. Transition process: - it a process to get the inverse Fourier transform. Overall, the FFT can be a powerful tool for analyzing and processing MRI images, and can provide valuable frequency to obtain the important information that disappear in spatial domain. [17]: the equation bellows (2) explain the expression for the operation of FFT.

$X(k) = \Sigma n=0^{(N-1)} x(n) \exp(-2\pi ink/N)$ (2)

3.6 K-MEANS CLUSTER

In machine learning applications, K-means clustering can also be utilized for feature extraction. Instead of using the actual raw data points, the k-means method is applied to a collection of input features in this case. Finding a set of k centroids representing the most significant or discriminative features in the input data is the aim [18-19]. The k-means algorithm can be used to iteratively update the centroids based on the input features, and the resulting centroids can be used as a reduced set of features for downstream analysis tasks, such as classification or regression. This approach can be particularly useful when dealing with high-dimensional data, where the original feature space is too large to be manageable [20-21]. One potential issue with using k-means for feature extraction is that the resulting centroids may not always be interpretable or meaningful in the context of the original data [22-23]. Additionally, the choice of the number of centroids (k) can have a significant impact on the resulting feature set and downstream performance. Therefore, it is important to carefully tune the hyper parameters of the k-means algorithm when using it for feature extraction.

3.7 THE K- NEAREST NEIGHBOUR ALGORITHM

Data can be categorized using the non-parametric technique K-Nearest neighbor's (KNN). Given a new instance, it will assign a class to it based on the majority class of it is k- nearest neighbor's. The KNN algorithm can be implemented in different ways, but one common approach is to use the Euclidean distance as a measure of The steps for implementing the KNN classifier can be summarized as follows: -

First: - Selection of number of neighbor's-K The main work here, all of this process will implement by using training data to find the new value between the other instances and the new instances.

Second: - Make the new instance's class the majority class among the k-nearest neighbor's by choosing the k instances that are closest to it. Compute the Euclidean distance between x and y using the following equation: -In the event when y= (y1, y2, yn) and x= (x1, x2, xn) is decided by:

(x1-y1) ^2 + (x2-y2) ^2 +... + (xn-yn) ^2 = sqrt(d(x,y)). (3)

Third: After calculating the distances, we must find the k-nearest term by utilizing the shortest distances.

It is possible to choose the k-nearest neighbours based on the shortest distances. Ultimately, the majority class among the K-nearest neighbours can be given to the projected class for the new instances. It's crucial to remember that KNN requires the full training set to be kept in memory and is not computationally efficient for large datasets.

4 RESULTS AND DISCUSSION

4.1. EVALUATION METRICS

It used to classify or divide the tasks, which used to measure the model labels of a given performance in a way of predicting the class dataset. Some of the commonly used evaluation metrics for classification tasks are: Accuracy: for the classification tasks, is the most basic evolution metrics? It measures the model's percentage of correct predictions, Figure 4, depicts The Left Side Tumor Place with region of interest that place is the area tumor or the ROI and Right Side with Normal Kidney, the image in white and black color.

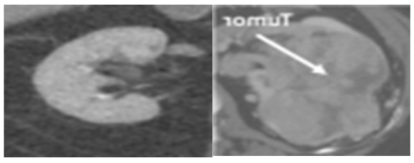


Figure 4: Explain The Left Side explains Tumor Place with region of interest that place is the area tumor or the ROI and Right Side with Normal Kidney.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$precision = \frac{tp}{tp+fp} \quad (5)$$

$$recall = \frac{tp}{tp+fn} \quad (6)$$

$$F1_{score} = \frac{2 \times precision \times recall}{precision+recall} \quad (7)$$

4.2. RESULT

Table (1) and (2) present two cases of classification, and their outcomes have been expounded. The conducted process of evaluation to multi-classification methods has been doing to compute matrix of precision, to find out the precocious value, the recall evaluation and F-matrix score. The process of multiclassification done with three main classification methods SVM, GB advanced and ADA. A training set of 4916 instances and an attest set of 2017 cases make up the two sets of data. The performance of each algorithm is measured and reported based on factors including the parameters mentioned in equation 4, 5, 6, and 7with process of, training and testing. The following lists the performance of each model: - Case 1: classify the tumor.

Table 1: Result for classification tumor to manifest the normal case of tumor and abnormality case.

Method	Accuracy	Precision	Recall	F-score
SVM	86%	80%	86%	82%
ADA	82%	75%	81%	77%
GB	89%	84%	90%	86%
KNN	93%	91%	93%	92%

Case 2: classify the abnormality Tumor.

Table 2: Result to classify the abnormal renal tumor.

Method	Accuracy	Precision	Recall	F-score
SVM	0.55	0.55	0.51	0.52
ADA	0.55	0.55	0.52	0.52
GB	0.55	0.55	0.52	0.52
KNN	0.51	0.51	0.51	0.51

Figure5 Explain Left Sides with Gray details of tumor that can clearly give the malignant or benign region of interest and Right Side with Region of tumor.

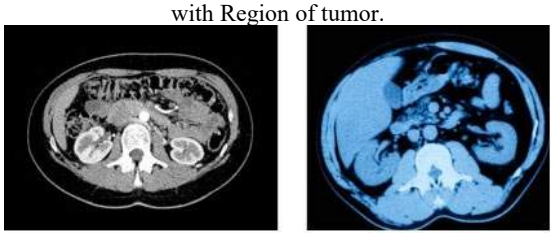


Figure 5: Explain Left Sides with Gray details of tumor that can clearly give the malignant or benign region of interest and Right Side with Region of tumor.

CONCLUSION

Classification of Renal Tumor using machine learning algorithms has gained significant attention in the medical field. It is a challenging task as kidney tumors can be of different types and may vary in size, shape, location, and other characteristics. However, as technology advances and large datasets become available, machine learning techniques have proven to be effective in diagnosing in summary, data collection and pre-processing, feature extraction, selection and training of a model are required for machine learning classification of renal tumor, evaluating its performance, and deploying it for clinical use. The efficacy of the selected algorithm and the caliber of the data determine the model's success. The suggested method goes through several stages; including picture pre-processing, feature extraction using the FFT technique, and classification using machine learning. The algorithms' accuracy rates for diagnosing normal and up normal are indicated to be 89.8% for GB, 84.8% for ADA, 87.1% for SVM, and 94.3% for KNN, and 55.3% for GB, 55.4% for SVM, 55.3% for ADA, and 51.0% for KNN for up normal diagnosis of Renal Tumor.

REFERENCES

[1] M. A. Hussain, A. Amir-Khalili, G. Hamarneh, and R. Abugharbieh, Segmentation-free kidney localization and volume estimation using aggregated orthogonal decision CNN, in International Conference on Medical Image Computation and Computer Assisted Intervention, Springer, 2017.

[2] M. A. Hussain, G. Hamarneh, T. W. O'Connell, M. F. Mohammed, and R. Abugharbieh, Segmentation-free estimation of kidney volumes in CT with dual regression forests, in International Workshop on Machine Learning in Medical Imaging, pp. 156–163, Springer, 2016.

[3] M. A. Hussain, A. Amir-Khalili, G. Hamarneh, and R. Abugharbieh, Collage CNN for renal cell carcinoma detection from CT, in International Workshop on Machine Learning in Medical Imaging, Springer, 2017.

[4] M. A. Hussain, G. Hamarneh, and R. Garbi, ImHistNet: Learnable image histogram based DNN with application to noninvasive determination of carcinoma grades in CT scans, in International Conference on Medical Image Computing and Computer Assisted Intervention, pp. 1–8, Springer, 2019.

[5] M. A. Hussain, G. Hamarneh, and R. Garbi, Renal cell carcinoma staging with learnable image histogram-based deep neural network, in 103 International Workshop on Machine Learning in Medical Imaging, pp. 533–540, Springer, 2019.

[6] K. He, G. Gkioxari, P. Dollar, and R. Girshick, Mask R-CNN, in Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969, 2017.

[7] V. Wang, H. Vilme, M. L. Maciejewski, and L. E. Boulware, The economic burden of chronic kidney disease and end-stage renal disease," in Seminars in Nephrology, vol. 36, pp. 319–330, Elsevier, 2016.

[8] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sanchez, ´ A survey on deep learning in medical image analysis, Medical Image Analysis, vol. 42, pp. 60–88, 2017.

[9] K. M. Krajewski and I. Pedrosa, Imaging advances in the management of kidney cancer,Journal of Clinical Oncology, vol. 36, no. 36, pp. 3582–3590, 2018.

[10] D. B. Rukstalis, J. Simmons, and P. F. Fulgham Renal ultrasound, in Practical Urological Ultrasound, pp. 51–76, Springer, 2017.

[11] T. J. van Oostenbrugge, J. J. Futterer, and P. F. Mulders, Diagnostic ¨ imaging for solid renal tumors: A pictorial review, Kidney Cancer, no. Preprint, pp. 1–15, 2018.

[12] X. Xu, F. Zhou, B. Liu, D. Fu, and X. Bai, Efficient multiple organ localization in CT image using 3D region proposal network, IEEE Transactions on Medical Imaging, 2019.

[13] J. Wang and D. Fleischmann, Improving spatial resolution at ct: Development, benefits, and pitfalls, 2018.

[14] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, Multiple instance learning: A survey of problem characteristics and applications, Pattern Recognition, vol. 77, pp. 329–353, 2018.

[15] O. Z. Kraus, J. L. Ba, and B. J. Frey, Classifying and segmenting microscopy images with deep multiple instance learning, Bioinformatics, vol. 32, no. 12, pp. i52–i59, 2016.

[16] J. Ding, Z. Xing, Z. Jiang, J. Chen, L. Pan, J. Qiu, and W. Xing, CT-based radiomic model predicts high grade of clear cell renal cell carcinoma, European Journal of Radiology, vol. 103, pp. 51–56, 2018.

[17] S. Oh, D. J. Sung, K. S. Yang, K. C. Sim, N. Y. Han, B. J. Park, M. J. Kim, and S. B. Cho, Correlation of CT imaging features and tumor size with fuhrman grade of clear cell renal cell carcinoma, Acta Radiologica, vol. 58, no. 3, pp. 376–384, 2017.
[18] J. Shu, Y. Tang, J. Cui, R. Yang, X. Meng, Z. Cai, J. Zhang, W. Xu, D. Wen, and H. Yin, Clear cell renal cell carcinoma: CT-based radiomics features for the prediction of Fuhrman grade, European Journal of Radiology, vol. 109, pp. 8–12, 2018.
[19] Z. Wang, H. Li, W. Ouyang, and X. Wang, Learnable histogram: Statistical context features for deep neural networks, in European Conference on Computer Vision, pp. 246–262, Springer, 2016.
[20] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature, vol. 521, no. 7553, p. 436, 2015.
[21] A. K. AAlAbdulsalam, J. H. Garvin, A. Redd, M. E. Carter, C. Sweeny, and S. M. Meystre, Automated extraction and classification of cancer stage mentions from unstructured text fields in a central cancer registry, AMIA Summits on Translational Science Proceedings, vol. 2018, p. 16,2018.
[22] B. Escudier, C. Porta, M. Schmidinger, N. Rioux-Leclercq, A. Bex, V. Khoo, V. Gruenvald, and A. Horwich, Renal cell carcinoma: ESMO clinical practice guidelines for diagnosis, treatment and follow-up, Annals of Oncology, vol. 27, no. suppl 5, pp. v58–v68, 2016.
[23] X. Li, X. Chen, J. Yao, X. Zhang, and J. Tian, Renal cortex segmentation using optimal surface search with novel graph construction, in International Conference on Medical Image Computing and Computer Assisted Intervention, pp. 387–394, Springer, 2011.

Stat-Net: Spatio-Temporal Aggregation Transformer Network for Skeleton-based Few-shot Action Recognition

Dazhi Ren, Shengli lv, Jinlin Li, Naining Li, Lin Dang
Chinenergy (shandong) New Energy Co.,Ltd, Jinan, Shandong, China 250109

ABSTRACT

Few-shot action recognition predicts new classes without labels and has received widespread attention for practical systems. The skeleton is a sparse representation of human actions, and existing spatio-temporal based models by training a strong encoder network could make the skeleton graph very dense with edges, which may lead to the over-smoothing problem. To address this issue, we propose the Spatio-Temporal Aggregation Transformer Network (STAT-Net) as a general backbone for skeleton-based few-shot action recognition. In the spatio-temporal aggregation transformer modules, the spatial multi-head self attention for modeling the connection of different joints in the same frame, while the temporal multi-head self attention for modeling the skeleton sequence between two adjacent frames. The extracted features between the three parts are aggregated by Adaptive Fusion technique to obtain a high dimensional embedding. Extensive experiments on two benchmarks demonstrate that our proposed model achieves better recognition results com- pared with other existing methods.

Keywords: Skeleton-based, action recognition, few- shot learning, Transformer, spatio-temporal network

1. INTRODUCTION

Action recognition has made great progress due to deep learning model development and enrichment of action capture methods invented. Currently, a mainstream direction of work is to learn effective representations for video or image classification using large amounts of labeled data [1]. However, when a pretrained model needs to be adapted to recognize an invisible category, especially in the medical image domain, it usually requires many medical experts to manually collect hundreds of video samples for knowledge transfer, and such a process is quite tedious and labor-intensive [2]. Note that labeling videos is much more difficult and costly than images. In order to solve this data problem, action recognition based on few-shot learning has been proposed and received much attention.

Given a few labeled actions, the aim of few-shot action recognition is to predict new categories that are unlabeled. The existing methods are mainly grouped into video based and skeleton based methods. In video-based methods [3, 4], 3D convolution and optical flow are the two dominant methods being used to model short-term temporal connections. However, long-term temporal sequences are often ignored. Moreover, high-dimensional redundant information, such as luminance and background, is usually unreliable in a few-shot scene. In skeleton-based methods [5, 6], skeleton sequences provide dense and background-strong robust action representations. Existing methods mainly use Spatial-Temporal Graph Convolution (ST-GCN [7]) as the backbone network to capture the spatio-temporal relationships of the skeleton. How- ever, excessive smoothing of the graph convolution leads to indistinguishable node representations and difficult to deter- mine the weights of edges, which ultimately leads to the loss of important node and edge information after ST-GCN [7]. To reduce the edge loss problem caused by transition smoothing, we adopt a Transformer-based multi-head self-attention mechanism. Besides, in order to learn the global context in- formation with long temporal dependency, we combine Spa- tial multi-head attention mechanism and temporal multi-head attention mechanism.

In this paper, we propose the Spatio-Temporal Aggregation Transformer Network (STAT-Net) as a backbone for skeleton-based few-shot action recognition. In the spatio-temporal aggregation transformer modules, the spatial multi-head self attention for modeling the connection of different joints in the same frame, while the temporal multi-head self attention for modeling the skeleton sequence between two adjacent frames. The extracted features between the three parts are aggregated by Adaptive Fusion technique to obtain a high dimensional embedding. We train the model to employ supervised contrastive learning techniques. In the evaluation stage, we calculate the similarity and a 1-nearest neighbor approach is used to determine the category. We demonstrate the effectiveness of our method through extensive experimental evaluations on widely used datasets: NTU-60 [8] and NTU- 120 [9]. Furthermore, Experimental results

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190F · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058329

show that our proposed model achieves better recognition results compared with other other state-of-the-art (SOTA) methods [9–12].

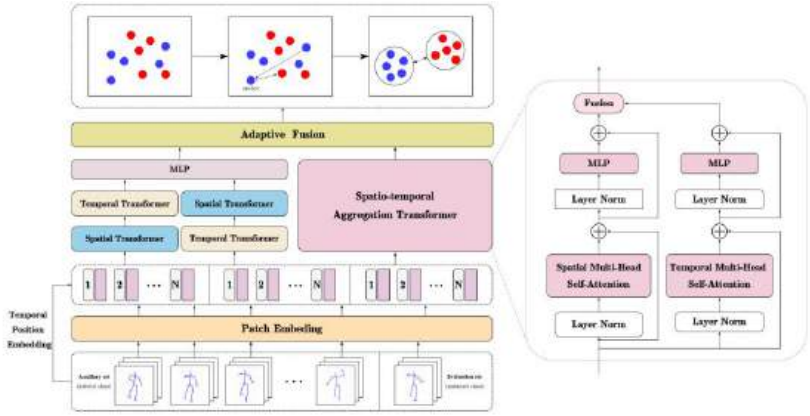


Fig. 1: Overview of the proposed network. We propose the Spatio-Temporal Aggregation Transformer Network (STAT-Net) as a general backbone for skeleton-based few-shot action recognition. STAT-Net consists of three components, which are spatial transformer modules, temporal transformer modules, and spatio-temporal aggregation transformer modules. Each module contains multiple Multi-Head Self Attention (MHSA) blocks and Multilayer Perception (MLP). In the spatio-temporal aggregation transformer modules, the spatial multi-head self attention for modeling the connection of different joints in the same frame, while the temporal multi-head self attention for modeling the skeleton sequence between two adjacent frames. The extracted features between the three parts are aggregated by Adaptive Fusion technique to obtain a high dimensional sequence- to-sequence embedding.

Overall, the contributions of our work are in the following three folds:

• We design a Spatio-Temporal Aggregation Transformer Network (STAT-Net) for skeleton-based few-shot action recognition, effective and efficient modeling is achieved through parallel and cascaded multi-head self-attentive mod- ules.

• We introduce a supervised contrastive learning tech- nique for the few-shot action recognition problem to maxi- mize the information of each sample embedded in the output of the backbone network.

• We further provide an evaluation method for skeleton- based few-shot action learning.

2. RELATED WORK

2.1 Skeleton-based Action Recognition

Traditionally, RGB images, depth sequences, sensor data, or fusion of these modalities have been used for human action recognition tasks, with notable achievements in human- computer interaction, virtual reality, and robotics [1]. However, compared to skeleton sequences, a topological representation of the position of human gestures (joint points and bones), the preceding modalities tend to be computationally intensive and highly sensitive to background noise, viewpoint changes, and motion speed [2]. In addition, advances in depth cameras and pose estimation algorithms [13, 14] make it easy to acquire 2D or 3D skeleton data.

2.2 Transformer-based Action Recognition

The skeleton-based human action recognition is mainly a long temporal problem, so traditional skeletal-based methods usually need to extract the movement patterns into specific skeletal sequences, such as pseudo-images in Convolution Neural Networks (CNNs) [15], one-dimensional sequential data in Long-Short Term Memory (LSTM) or Gate Recurrent Unit (GRU) [16]. Because of the unique advantage of transformer with multi-head self-attention mechanism in processing long-term time series, it has been widely noticed by researchers in the field of human action recognition. Most of the action recognition methods using the transformer treat the video frames as tokens and then add positional encoding information [17–19]. However, there are few approaches that perform tokenization operations on the skeleton sequence of the transformer. Due to the significant computational cost of transformer-based action recognition, in our

work, instead of using the 3D skeleton data from the original dataset, but rather 2D skeleton data extracted by pose estimation methods to minimize training time and model parameters.

2.3 Few-shot Action Recognition

The aim of few-shot learning is to identify novel categories from a small number of labeled samples, and in few- shot action recognition, the model only needs to compare the different features between samples without the participation of labels. In particular, metric-based few-shot classification focuses on learning a similarity function that measures the similarity between two samples, where a larger similarity indicates that the two samples are more similar and a smaller similarity indicates that the two samples are more divergent. Recently, few-shot action recognition has received more and more widespread attention due to the lack of large-scale la- beled data and the need to recognize actions that have not been seen in daily life. For example, Liu et al. [9] proposed an Action-Part Semantic Relevance-aware (APSR) network for few-shot action recognition, which achieves relevance modeling between actions and semantics by outputting hu- man part of skeleton data and a pretrained action description. Memmesheimer et al. [10] considered one-shot action recognition as a metric learning problem and proposed a model that would be an image-based representation of the skeleton. Considering the similarity between actions, Li et al. [11] pro- posed a self and mutual adaptive matching (SMAM) module to transform feature maps into distinguishable feature vectors. In this paper, we consider the few-shot action recognition task as a contrastive learning process to find discriminative fea- tures, measured by comparing the similarity between skeleton data.

3. METHODS

3.1 Problem Definition

Traditional human action recognition learns a large training set $\mathcal{D}$, and then the weights obtained from the training set are used to classify samples in the test set $\mathcal{T}$. The samples in the test set are not seen in the training set, but are of the same class as the samples in the training set. However, in the few- shot action recognition setting, the training set $\mathcal{D}$ containing N classes is known, the test set $\mathcal{T}$ is a new K-class sample, and only a small number of samples of each category are in the auxiliary set $\mathcal{A}$, the size of A is equal to K. And the other number of samples $N-K$ is attributed to the evaluation set $\mathcal{E}$. We attribute the few-shot action recognition problem to the metric learning problem. Since our goal is to project the feature embeddings of each sample into a feature representation Z. We train the model to employ supervised contrastive learning techniques on the auxiliary set $\mathcal{A}$. In the evaluation stage, we calculate the similarity $\mathcal{S}$ between each Z. To maximize this $\mathcal{S}$, a 1-nearest neighbor approach is used to determine the category.

3.2 Network Architecture

As shown in Fig.1, we propose a Spatio-Temporal Aggre- gation Transformer Network (STAT-Net) as a general back-bone for skeleton-based few-shot action recognition. Given a 2D skeleton sequence $x \in \mathbb{R}^{T\times J\times C_{in}}$, we first through spatial patch embedding $P^S \in \mathbb{R}^{1\times J\times C_f}$ and temporal posi- tion embedding $P^T \in \mathbb{R}^{T\times 1\times C_f}$, which are then projected to an initial sequence $F_0 \in \mathbb{R}^{T\times J\times C_f}$ by full connection, and N in the figure denoting the relative position information. Then, the features F_0 are passed through the STAT-Net en- coding network to obtain high-dimensional embeddings $F^i \in \mathbb{R}^{T\times J\times C_f}(i = 1,2,\cdots,n)$.Each embedding is projected to a feature representation Z. Finally,we train the model to employ supervised contrastive learning techniques on the auxiliary set $\mathcal{A}$. On the evaluation set $\mathcal{E}$, we compute the similarity $\mathcal{S}$ between each Z, and maximize this $\mathcal{S}$, 1-nearest neighboris used to determine the classification category. Here, we focus on three basic backbone modules of STAT-Net, which are spatial multi-head self attention (Spatial MHSA), temporal multi-head self attention (Temporal MHSA) and Spatio- Temporal Aggregation Transformer.

• **Spatial MHSA:** The main concern of Spatial MHSA (SMHSA) block is to compute the self attention of the joints with the objective of learning the joint connections for each frame. The formulation are shown below:

$$SMHSA = Concat[head_1,\ldots,head_h]W_s^p \tag{1}$$

$$head_i = Attention(Q_s^i, K_s^i, V_s^i), i \in h \tag{2}$$

$$Attention(Q_s^i, K_s^i, V_s^i) = soft\max(\frac{Q_s^i (K_s^i)^{'}}{\sqrt{d_m}})V_s^i \tag{3}$$

where W_s^p is a linear projection weight and his the number of heads, $^{'}$ denotes the matrix transpose. In the transformer encoder of our method, given the input as per-frame spatial features $F_s \in R^{J \times C_e}$ for each $head_i$, we employ multi-head self-attention to compute Q_s^i, K_s^i and V_s^i:

$$Q_s^i = F_s W_s^{Q,i}, K_s^i = F_s W_s^{K,i}, V_s^i = F_s W_s^{V,i} \tag{4}$$

where Q_s^i, K_s^i and V_s^i are learnable matrices. Furthermore, we employ residual connection and layer normalization (Layer Norm) as input of SMHSA, and Layer Norm, multi- layer perception (MLP) and residual connection are applied to the output of SMHSA.

• **Temporal MHSA:** Temporal MHSA (TMHSA) mainly computes the self attention between neighboring frames and learns the global dynamic relationship of each joint point. Its computational procedure is similar to SMHSA, except that the inputs are per-joint temporal features $\mathrm{F_t} \in \mathbb{R}^{\mathrm{T} \times \mathrm{C_e}}$ and are parallel in spatial dimension. Specifically, the formulation are as follows:

$$TMHSA = Concat[head_1, \ldots, head_h] W_t^p \tag{5}$$

$$head_i = Attention(Q_t^i, K_t^i, V_t^i), i \in h \tag{6}$$

$$Attention(Q_t^i, K_t^i, V_t^i) = soft\max(\frac{Q_t^i (K_t^i)^{'}}{\sqrt{d_m}})V_t^i \tag{7}$$

$$Q_t^i = F_t W_t^{Q,i}, K_t^i = F_t W_t^{K,i}, V_t^i = F_t W_t^{V,i} \tag{8}$$

• **Spatio-Temporal Aggregation Transformer:** Since com- prehensive context information should be available to be modeled between the multi streams and each stream plays a special role, we could train more reasonable by fusing the parameters of these different streams. In our skeleton action recognition task, Spatial MHSA and Temporal MHSA could learn inter-joint connections and joint global temporal con- nections each frame, respectively. Therefore, we propose a sequence-to-sequence Spatio-Temporal Aggregation Trans- former network. Specifically, we parallelize Spatial MHSA and Temporal MHSA, and obtain a spatio-temporal output through a Fusion module. In addition, we stack the temporal and spatial Transformers in different orders, where the Spa- tial Transformer module mainly consists of Spatial MHSA, Layer Norm and MLP, and the Temporal Transformer module mainly consists of Temporal MHSA, Layer Norm and MLP. Finally, the threes are further fused by Adaptive Fusion to obtain high dimensional embeddings Zi.The fusion formula is defined as follows:

$$F_{st}^i = \lambda_s \cdot SMHSA\ (F^{i-1}) + \lambda_t \cdot TMHSA\ (F^{i-1}) \tag{9}$$

$$Z^i = \alpha F_{st}^{i-1} + \beta_1 T_1^i (S_1^i (Z^{i-1})) + \beta_2 S_1^i (T_1^i (Z^{i-1})) \tag{10}$$

where F_{st}^i denotes the spatio-temporal features output from layer i, $i \in (1, \ldots, n)$, n is the depth of the model, S and T denote the spatial transformer module and temporal transformer module respectively, and Z_i is the embeddings obtained after adaptive fusion,$\alpha, \beta_1, \beta_2 \in \mathbb{R}^{\mathrm{N} \times \mathrm{T} \times \mathrm{J}}$are adaptive fusion weights.

4. EXPERIMENTAL RESULTS

4.1 Datasets

• NTU-60:NTU-60 dataset is a video dataset for action recog- nition and behavior analysis released by Nanyang Technolog- ical University (NTU), Singapore [8]. NTU-60 is a depth video dataset containing multi-camera, multi-modal (RGB images, depth images, skeleton data) video data. It contains 60 different human movement categories such

as handshaking, waving, and running. The acquisition environment of the dataset may include both indoor and outdoor scenes in order to better simulate different real-world situations, thus this dataset is used by researchers to evaluate and advance research in the field of action recognition and behavioral analysis.

• NTU-120:As another version of the NTU-60,contains more action categories and more samples than the NTU-60 [9]. It contains 120 different action categories covering a wide range of everyday life actions. Therefore, it can be considered more challenging for human action classification tasks.

4.2 Training and Implementation Details

For the aforementioned NTU-60 and NTU-120, we em- ploy HRNet [13] to extract the 2D skeleton information. Thanks to the backbone based on the Transformer model, we could handle different data lengths, and the input length T=243, depth=5, number of heads=8, feature size C_f of 512, and embeddings size C_e of 512 for our sequences. As in Paper [20], we also evaluate 20 novel classes in the set to report the results, and each class includes either 1 or 5 labeled samples, namely 1-shot and 5-shot. the auxiliary dataset in- cludes the remaining classes, and all samples are used during training.

During training, we use a supervised contrastive learn- ing technique, where samples of the same class are brought closer together and samples of different classes are moved away from each other based on an anchor in a high dimen- sional embedding space. In the evaluation phase, we compute the cosine similarity between the test set and the exemplars and predict the results based on 1-nearest neighbor.

4.3 Results Comparison

We experimentally verify the accuracy under 1-shot and 5-shot metrics on NTU-60 and NTU-120. In particular, note that these results are achieved with the model trained on the 100-class auxiliary dataset and tested on the 20-new-class evaluation dataset. As shown in Table 1, on NTU-60, our proposed STAT-Net achieves 58.9% and 70.5% accuracies at 1-shot and 5-shot respectively, exceeding the accuracy of the existing method SMAM-Net by up to 2.5% and 4.6%. As shown in Table 2, on NTU-120, our proposed STAT-Net achieves 67.0% and 79.3% under 1-shot and 5-shot metrics, respectively. It validates the effectiveness and efficiency of our method employing Transformer-based network for few- shot skeleton-based action recognition.

The size of the training classes influences the performance of the model on the evaluation set. In practical life, since the amount of training data is limited, this evaluation has an important reference for environments where only a small amount of data is provided. Following Liu et al. [10], we use a training set including 20, 40, 60, 80, 100 training classes and keep the evaluation set of 20 classes constant. In our experiments, our STAT-Net performs incrementally better as the training classes size gets larger, as shown in Fig.2, where it is clearly seen that our method outperforms the state-of-the-art method when the training sample class reaches 80.

In Fig.3, we show the visualization of UMAP [21] for 1- shot and 5-shot action classification on NTU-60 and NTU-120, which provides an idea of the model discrimination capability. The distance in the embedding space captures the number of identities well. From a qualitative point of view, we could see that some clusters are well separated from each other and easily and clearly distinguished from each other intuitively, so in our proposed STAT-Net approach, these clusters can be separated very well.

Table 1: Few-shot action recognition results on the NTU-60 dataset

Methods	**Accuracy(%)**	
	1-shot	**5-shot**
Attention Network	41.0	-
Fully Connected	42.1	52.4
Average Pooling	42.9	51.1
APSR	45.3	-
TCN-OneShot	46.5	60.3
SL-DML	50.9	64.0
Skeleton-DML	54.2	65.5
SMAM-Net	56.4	65.9
STAT-Net(ours)	**58.9**	**70.5**

Table 2: Few-shot action recognition results on the NTU-120 dataset

Methods	Accuracy(%)	
	1-shot	5-shot
Fully Connected	60.9	64.2
Average Pooling	59.8	61.2
TCN-OneShot	64.8	66.8
SL-DML	71.4	77
Skeleton-DML	71.8	77.6
SMAM-Net	73.6	79
STAT-Net(ours)	**67**	**79.3**

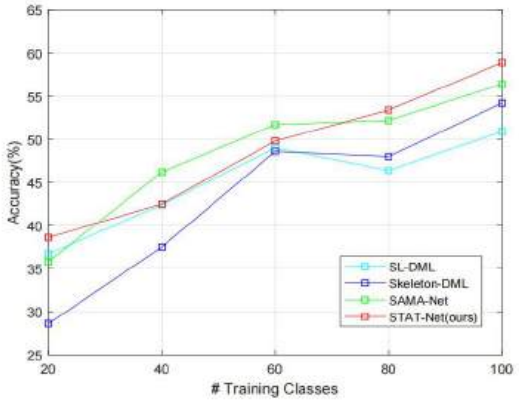


Fig. 2: Results comparison for increasing auxiliary set A sizes

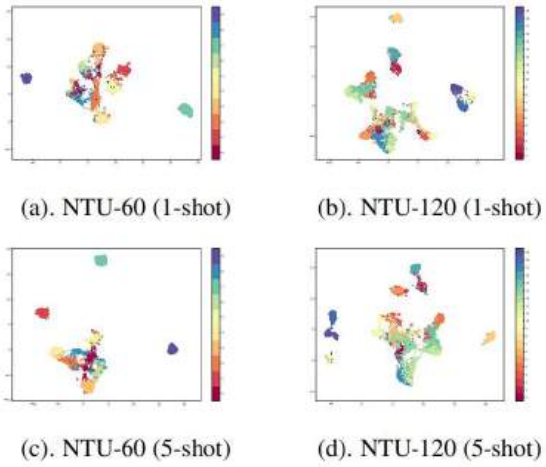


Fig. 3: The UMAP embedding visualization for 1-shot and 5-shot action classification on NTU-60 and NTU-120.

5. CONCLUSION

In this paper, we propose the Spatio-Temporal Aggregation Transformer Network (STAT-Net) for skeleton-based few-shot action recognition. STAT-Net consists of three components, which are spatial transformer modules, temporal transformer modules, and spatio-temporal aggregation trans- former modules. Each module contains multiple Multi-Head Self Attention (MHSA) blocks and Multilayer Perception (MLP). In the spatio-temporal aggregation

transformer modules, the spatial multi-head self attention for modeling the connection of different joints in the same frame, while the temporal multi-head self attention for modeling the skeleton sequence between two adjacent frames. The extracted features between the three parts are aggregated by Adaptive Fusion technique to obtain a high dimensional embedding. Experimental results show that our proposed model achieves better recognition results compared with other existing methods.

REFERENCES

[1] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu, "Human action recognition from various data modalities: A review," pp. 1–20.

[2] Bin Ren, Mengyuan Liu, Runwei Ding, and Hong Liu, "A survey on 3d skeleton-based action recognition us- ing learning method," arXiv preprint arXiv:2002.05907, 2020.

[3] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles, "Few-shot video classification via temporal alignment," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10618–10627

[4] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Changxin Gao, Yingya Zhang, Deli Zhao, and Nong Sang, "Molo: Motion-augmented long-short contrastive learning for few-shot action recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18011–18021.

[5] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang, "Motionbert: A unified perspective on learning human motion representations," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.

[6] Ning Ma, Hongyi Zhang, Xuhui Li, Sheng Zhou, Zhen Zhang, Jun Wen, Haifeng Li, JingjunGu, and Jiajun Bu, "Learning spatial-preserved skeleton representations for few-shot action recognition," in European Conference on Computer Vision. Springer, 2022, pp. 174–191.

[7] Sijie Yan, Yuanjun Xiong, and DahuaLin, "Spatial tem- poral graph convolutional networks for skeleton-based action recognition," in Proceedings of the AAAI confer- enceon artificial intelligence, 2018, vol. 32.

[8] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in Proceedings of the IEEE confer- enceon computer vision and pattern recognition, 2016, pp. 1010–1019.

[9] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity under- standing," IEEE transactions on pattern analysis and machine intelligence, vol. 42, no. 10, pp. 2684–2701, 2019.

[10] Raphael Memmesheimer, Simon Hring, Nick Theisen, and Dietrich Paulus, "Skeleton-dml: Deep metric learn- ing for skeleton-based one-shot action recognition," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 3702–3710.

[11] Zhiheng Li, Xuyuan Gong, Ran Song, Peng Duan, Jun Liu, and Wei Zhang, "Smam: Self and mutual adap- tive matching for skeleton-based few-shot action recog- nition," IEEE Transactions on Image Processing, vol. 32, pp. 392–402, 2022.

[12] Alberto Sabater, Laura Santos, Jos Santos-Victor, Alexandre Bernardino, Luis Montesano, and Ana C Murillo, "Oneshot action recognition towards novel assistive therapies," arXiv preprint arXiv:2102.08997, 2021.

[13] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang, "Deep high-resolution representation learning for hu- man pose estimation," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5693–5703.

[14] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," IEEE Trans- actions on Pattern Analysis and Machine Intelligence, 2019.

[15] Kailin Xu, Fanfan Ye, Qiaoyong Zhong, and Di Xie, "Topology-aware convolutional neural network for effi- cient skeleton-based action recognition," in Proceedings of the AAAI Conference on Artificial Intelligence, 2022, vol. 36, pp. 2866–2874.

[16] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recog- nition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 1227–1236.

[17] Vittorio Mazzia, Simone Angarano, Francesco Salvetti, Federico Angelini, and Marcello Chiaberge, "Action transformer: A self-attention model for short-time pose- based human action recognition," Pattern Recognition, vol. 124, pp. 108487, 2022.
[18] Jiewen Yang, Xingbo Dong, Liujun Liu, Chao Zhang, Jiajun Shen, and Dahai Yu, "Recurring the transformer for video action recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14063–14073.
[19] Zhimin Gao, Peitao Wang, Pei Lv, Xiaoheng Jiang, Qi- dong Liu, Pichao Wang, Mingliang Xu, and Wanqing Li, "Focal and global spatial-temporal transformer for skeleton-based action recognition," in Proceedings of the Asian Conference on Computer Vision, 2022, pp. 382–398.
[20] Raphael Memmesheimer, Nick Theisen, and Dietrich Paulus, "Sl-dml: Signal level deep metric learning for multimodal one-shot action recognition," in 2020 25th International conference on pattern recognition (ICPR). IEEE, 2021, pp. 4573–4580.
[21] Leland McInnes, John Healy, and James Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," arXiv preprint arXiv:1802.03426, 2018.

Dynamic Feedback-Based Vulnerability Mining Method for Highly Closed Terminal Protocols

Yong Wang[a], Wenting Wang[b], and Dongchang Li[c]

[a]State Grid Shandong Electric Power Company, Jinan, China
[b]State Grid Shandong Electric Power Research Institute, Jinan, China
[c]Beijing KeDong Electric Power Control System Co.,Ltd., Beijing, China

ABSTRACT

This paper introduces a dynamic feedback-based vulnerability mining method tailored for highly closed terminal protocols, addressing the limitations of traditional fuzz testing methods which struggle with closed-source protocols due to the lack of accessible code or protocol specifications. The proposed method overcomes these barriers by generating test cases using Large Language Models (LLMs) and optimizing them through real-time execution feedback without a deep understanding of the protocol. The primary contributions include a balanced training set construction method for LLMs, integration of LLMs with fuzz testing to generate test cases without relying on protocol knowledge, and a real-time feedback mechanism from a state machine to LLMs for continuous test case optimization.The method's effectiveness is validated through experiments on a closed-source protocol, MQTT, and SSH, demonstrating significant improvements over conventional AFL fuzz testing. The results show that the proposed method can identify up to 4.34 times more valid cases in closed-source protocols, highlighting its efficiency in vulnerability detection.

Keywords: Closed-source Protocols, Fuzz Testing, Large Language Models, Vulnerability Mining

1. INTRODUCTION

As highly closed systems continue to evolve, an increasing number of heterogeneous terminal devices are integrated into these systems, leading to a sharp increase in the diversity of terminal types. This diversification trend significantly raises the risk of threatening system security by exploiting terminal vulnerabilities, posing a severe challenge to the security protection of highly closed systems. Faced with such a large number of terminals, traditional manual security testing methods can no longer meet the demands, as their efficiency and coverage are difficult to ensure. Therefore, there is an urgent need to develop an automated and efficient vulnerability mining technology specifically targeted at the vast number of terminal devices in highly closed systems to ensure the security and stability of network communications.

In previous research, several methods have been proposed to identify and eliminate vulnerabilities in the implementation of system protocols, each addressing different aspects of the problem. If the implementation of a system protocol is easily accessible, different program analysis techniques can be applied to find security vulnerabilities, such as white-box fuzz testing,[1] dynamic taint tracking,[2] symbolic execution,[3] and static code analysis.[4] However, when facing highly closed terminals, neither the code nor the protocol specifications can be directly accessed, making it difficult to conduct existing vulnerability mining methods. Although there are methods to retrieve protocol implementations in some cases, such as by reading firmware images or reverse-engineering binary packages, the complexity of this work may still hinder a thorough security analysis.

Fuzz testing involves inputting abnormal test cases into the software being tested in an attempt to trigger crashes.[5] Compared to traditional techniques such as static and dynamic analysis, fuzz testing does not require the analysis of source code and is not limited by the explosion of path and state spaces, while also reducing

Further author information:
Yong Wang: E-mail: wangyong@sgcc.com.cn
Wenting Wang: E-mail: wangwenting@sgcc.com.cn
Dongchang Li: E-mail: ldc9211@sina.com

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190G · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058050

false positives and resource consumption. It discovers security vulnerabilities in protocols by simulating various edge cases and abnormal inputs. For open-source protocols, traditional fuzz testing methods can instrument the source code and compile it for vulnerability mining.[6,7] However, for closed-source protocols, due to the inability to obtain source code, traditional fuzz testing methods can only disassemble binary files and then instrument them. This process requires the execution of a large number of test cases but only finds a small number of effective ones, resulting in low vulnerability mining efficiency.

To address the aforementioned issues, this paper proposes a dynamic feedback-based vulnerability mining method for highly closed terminal protocols. This method does not rely on an in-depth understanding of the protocol. It generates test cases using large language models(LLMs) and provides real-time feedback of the execution results back to the model, thereby optimizing the test cases. This approach can significantly improve the efficiency of generating effective test cases and greatly enhance the speed and accuracy of vulnerability detection in closed-source protocols. The primary contributions of this paper are as follows:

1. To address the issue of imbalance between the number of valid and invalid test cases in the training samples, this paper proposes a method for constructing a balanced training set of positive and negative samples to train a large language model. By employing a combination of oversampling and random undersampling strategies, the method ensures that the quantities of valid and invalid test cases are balanced, thereby enhancing the accuracy of the large language model in generating test cases.
2. Fuzz testing methods that generate test cases based on existing documentation are ineffective for vulnerability mining in highly closed terminal protocols, as closed-source protocols may be proprietary and lack documentation support. This paper integrates LLMs with fuzz testing to generate test cases without relying on prior knowledge of the protocols.
3. The process of generating test cases by LLMs can be inaccurate in predicting the validity of test cases, leading to the generation of a large number of invalid test cases. This paper feeds back the results from the testing process in real-time to the LLMs, optimizing the generation of test cases through real-time feedback from a state machine. This approach not only improves the quality of the test cases but also focuses the testing process more on potential vulnerability areas through continuous iterative optimization, resulting in the efficient and accurate generation of test cases.

2. BACKGROUND INFORMATION

2.1 Fuzz Testing

Contemporary fuzz testing has transcended the realms of conventional white-box and black-box methodologies, giving rise to a novel approach termed gray-box fuzz testing.[8] This innovative technique offers a potent strategy for uncovering vulnerabilities within closed-source protocols of highly sealed systems. Distinct from its predecessors, gray-box testing employs a unique set of testing methods and strategies. By amalgamating the advantages of white-box and black-box testing, gray-box testing endeavors to inform the fuzz testing process with a partial understanding of the internal mechanisms and operational nuances of the system under scrutiny. Employing techniques such as static analysis, dynamic analysis, and symbolic execution, this method garners essential system insights and achieves a level of code coverage. Consequently, it allows for the crafting of fuzz test cases that are more precisely aimed at uncovering potential weaknesses.

A prominent gray-box testing methodology is the American Fuzzy Lop (AFL).[9] AFL has garnered considerable success within the domain of fuzz testing and is extensively applied for software security assessments and vulnerability unearthing.[10] By instilling random mutations into input data to probe for potential program vulnerabilities, AFL also leverages code coverage data to direct the creation of test cases. Comprising two principal components—an execution engine and a fuzz manager,[8,11,12] the execution engine is tasked with running the target program's test cases, while the fuzz manager generates novel test cases through genetic algorithm-based mutations, coordinates the execution, and analyzes the outcomes. A distinguishing feature of this approach is its employment of dynamic binary instrumentation to track the program's execution trajectories and code coverage, facilitating the detection of a broader array of vulnerabilities.The AFL fuzz testing procedure is depicted in Figure1.

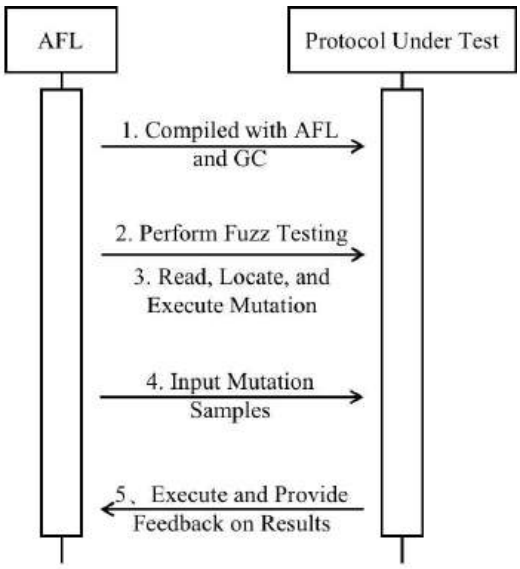


Figure 1. AFL Algorithm Flowchart

Step 1: Compilation Phase: The target protocol's code is compiled using an AFL-specific compiler to ensure effective fuzz testing;

Step 2: Test Execution: AFL is initiated and runs the target program, which functions as a subprocess of AFL;

Step 3: Seed Mutation: AFL extracts initial test cases from a seed library and alters them based on predefined mutation strategies to produce new test cases;

Step 4: Test Case Transmission: The mutated test cases are transmitted to the target program, operating as a subprocess, via a dedicated piping mechanism;

Step 5: Execution and Feedback: Upon receiving the test cases, the subprocess executes them and relays the results back to AFL.

AFL logs the outcomes of these test executions, retaining the effective test cases. It then reverts to step 3, perpetuating the cycle of mutation and testing until a set number of iterations is completed or other termination criteria are satisfied.

2.2 Large Language Models

Large Language Models (LLMs) have become avant-garde artificial intelligence systems, dedicated to processing and generating text while maintaining coherent communication.[13–15] The escalating demand for LLMs is driven by the growing necessity for machines to manage sophisticated language tasks, encompassing translation, summarization, information retrieval, and interactive conversations. Recent notable advancements in language models are largely due to the evolution of deep learning technologies, neural architectures such as transformers, augmented computational capabilities, and the accessibility of training datasets. Pre-trained Language Models (PLMs),[16,17] particularly those fine-tuned through self-supervised learning on extensive textual repositories, demonstrate superior text comprehension and generation skills. They are capable of accomplishing an array of tasks, including code and text generation, tool manipulation, reasoning, and comprehension, employing zero-shot and few-shot learning approaches.

Drawing inspiration from the recent triumphs of LLMs in comprehending natural language, the strategy of regarding fuzz testing feedback as an alternative natural language input has been embraced.[18–20] By developing a corpus and training the LLM, the process of generating test cases is rendered more efficient. However, there are intrinsic disparities between protocol feedback and conventional text, with the former often demanding strict adherence to structure and syntax. Consequently, a novel approach will be proposed to integrate these nuances into the learning process, thereby enhancing the LLM's capacity to interpret and understand protocols.[11]

3. DYNAMIC FEEDBACK-BASED VULNERABILITY MINING METHOD

3.1 Training Sample Preprocessing

Given the inherent unpredictability and randomness of fuzz testing's mutation process, the vast majority of test cases generated by AFL prove to be ineffective, with a mere handful capable of inducing crashes or timeouts in the target program. This dynamic results in a significant numerical imbalance within the training dataset, with far fewer valid test cases that can successfully traverse new code paths compared to the invalid ones.

To mitigate the scarcity of valid test cases, this study adopts a hybrid strategy that leverages both comprehensive oversampling and random undersampling for the preprocessing of training samples, prior to training the LLMs with this refined dataset. The process begins by retaining all valid cases derived from fuzz testing and calculating the necessary number of invalid test cases based on the disparity between the overall counts of valid and invalid test cases. By applying a specific matching coefficient in conjunction with the volume of valid test cases, the total number of invalid test cases is ascertained. Thereafter, a random selection of invalid test cases is conducted to equalize the number of valid and invalid test cases. Through this approach, a balanced training sample set is ultimately assembled, ensuring an equitable representation of both valid and invalid test cases.The specific construction method is as follows:

Let the set of seed cases be denoted by Z, and the number of elements in the set be t, then

$$Z = \{z_1, z_2, z_3, \ldots, z_t\} \tag{1}$$

In line with AFL's mutation protocols, each seed instance z_i is paired with a collection of valid test cases P_i and a collection of invalid test cases N_i. Assuming the count of valid test case sets to be k_i and the count of invalid test case sets to be l_i, then

$$P_i = \{p_i, p_{i2'}, p_{i3'}, \ldots, p_{ik_i}\} \tag{2}$$

$$N_i = \{n_i, n_{i2'}, n_{i3'}, \ldots, n_{il_i}\} \tag{3}$$

Let p_{ij} denote the set of valid test cases and n_{ij} denote the set of invalid test cases, both resulting from the mutation of the $j-th$ seed instance. Prior to data matching, the aggregate counts of the valid test case sets P_i and the invalid test case sets N_i are designated as K and L respectively

$$K = \sum_{i=0}^{t} k_i, \quad L = \sum_{i=0}^{t} l_i \tag{4}$$

Subsequently, a matching factor is determined based on the aggregate count of valid and invalid test cases, referred to as ϑ:

$$\vartheta = f(M, N) \tag{5}$$

The value of ϑ increases with a larger disparity between the total counts of valid and invalid test cases. Nevertheless, to maintain sample balance, the quantity matching factor ϑ is capped at 5.

Proceeding with oversampling of the valid test cases, we begin by compiling all valid cases into a training set. Following this, we replicate each element in the aggregated set according to the quantity matching coefficient, thereby constructing the comprehensive set of valid test cases, labeled as O. Within this set, O_{ij} represents the $j-th$ copy of the subset O_i.

$$O = (O_{11} \cup O_{12} \cup O_{1\vartheta}) \cup (O_{21} \cup O_{22} \cup O_{2\vartheta}) \cup \ldots \cup (O_{t1} \cup O_{t2} \cup O_{t\vartheta}) \tag{6}$$

Given that each seed case is associated with k_i valid test cases and the quantity matching factor is ϑ, we need to randomly extract ϑ_{ki} negative samples from each negative sample set to form the comprehensive subset N:

$$N = N_1' \cup N_2' \cup N_3' \cup \ldots \cup N_t' \tag{7}$$

In which, N_t' denotes the collection of negative samples for the $i-th$ seed case following random sampling, with the number of samples in the set being ϑ_{ki}. Upon completion of the sample processing steps described, a balanced training dataset is achieved. With the balanced training dataset in hand, the LLM is then trained to produce the initial seed cases.

3.2 Method Design

The AFL fuzz testing tool, previously discussed, employs a random mutation strategy to generate a vast array of potential test cases from a set of seed cases, aiming to cover a broad input space. While this method is commendable for its comprehensive exploration of the input space, it suffers from a notable efficiency issue: among the multitude of test cases, only a minuscule fraction actually reveal the hidden flaws within the program. This inefficient testing process implies that capturing rare vulnerabilities requires a substantial investment, including computational resources, time, and human effort. Therefore, despite AFL's strength as a powerful tool, its efficiency in filtering effective test cases needs further enhancement.

To address the aforementioned issues, this paper introduces a dynamic feedback-based vulnerability mining method tailored for highly closed terminal protocols, designed to efficiently uncover vulnerabilities within these protocols. The method integrates AFL with two specialized LLMs: a Test Case Generation LLM for generating and evaluating test cases, and a State Machine Construction LLM for building a protocol state machine based on interaction data. Initially, the Test Case Generation LLM is trained with balanced positive and negative sample datasets to produce high-quality initial seed cases. These test cases are then utilized for fuzz testing of the closed-source protocol, while the State Machine Construction LLM constructs the state machine based on the interaction data and relays real-time feedback back to the Test Case Generation LLM. This cyclic feedback mechanism guides the Test Case Generation LLM to generate more effective test cases. This approach not only enhances the overall quality of test cases and reduces the generation of ineffective tests but also, through continuous iterative optimization, focuses the testing process more intently on potential vulnerability areas. The detailed principles of this method are illustrated in Figure 2.

Step 1: Draw base samples from the initial seed case ensemble, then employ mutation techniques to refine and diversify these samples. Utilize the resulting mutated seeds to calculate and obtain a balanced training sample set for training the Test Case Generation LLM;

Step 2: Kick off fuzz testing by introducing the initial seed cases crafted by the Test Case Generation LLM into AFL, and set up AFL's fuzz testing parameters. Conduct fuzz testing to scrutinize the closed-source protocol, with vigilant monitoring to ensure extensive coverage of the input space;

Step 3: Should the testing process uncover crashes, timeouts, or the initiation of new code execution paths within the protocol object, the respective test cases are confirmed as valid and incorporated into the ultimate collection of effective cases;

Step 4: Simultaneously with the fuzz testing execution, leverage the State Machine Construction LLM to dissect protocol requests and responses, constructing a protocol state machine to capture the protocol's dynamic behaviors and state transitions;

Step 5: Engage a dynamic feedback loop by conveying real-time feedback from the State Machine Construction LLM back to the Test Case Generation LLM. The Test Case Generation LLM fine-tunes its generation strategy based on this feedback to yield more potent test cases. Through persistent iterative optimization of the test case generation, enhance the quality of test cases, curtail the proliferation of ineffective tests, and concentrate on areas likely to harbor vulnerabilities, analyzing test outcomes to uncover and confirm potential security flaws;

Step 6: Iteratively perform steps one to five until no new states are activated within an established timeframe, indicating the program's consistent performance in successive tests, thereby concluding the vulnerability mining endeavor.

By consistently learning from the data gathered throughout the fuzz testing process, this method allows Large Language Models to evolve, identifying and anticipating effective test cases with greater precision. This

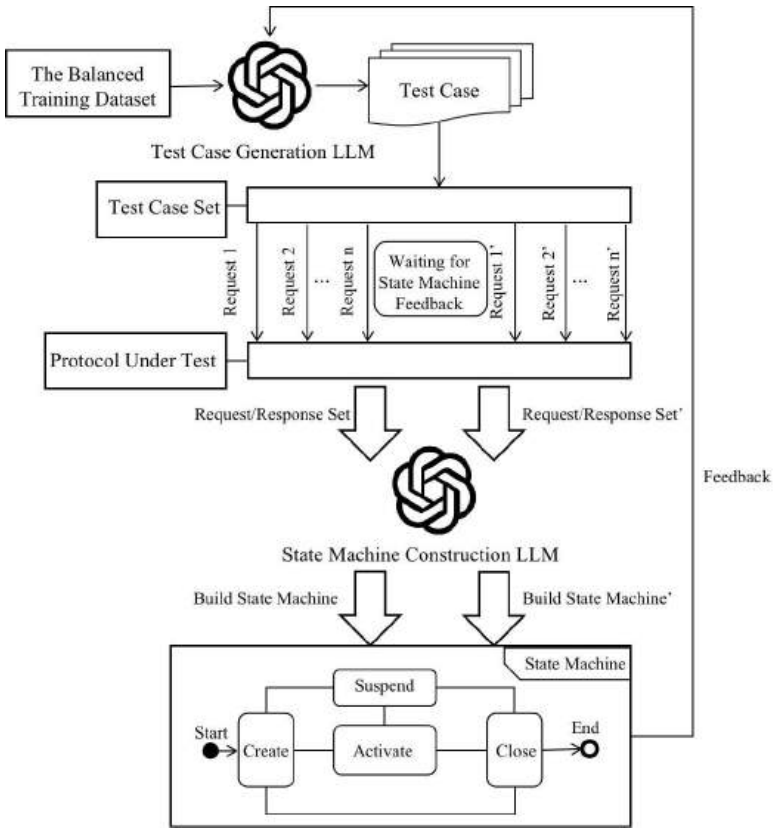


Figure 2. The Principle of Dynamic Feedback-Based Vulnerability Mining Method

self-optimization trait makes the dynamic feedback-driven fuzz testing approach more adaptable and efficient against intricate and highly sealed network protocols than traditional tools. In the end, this methodology not only boosts the efficacy of vulnerability detection but also markedly diminishes the false positive rate, introducing a novel perspective and solution to the domain of software security testing.

3.3 Vulnerability Mining Framework

Addressing the dynamic feedback-based fuzz testing methodology, this paper introduces a protocol-focused vulnerability mining framework illustrated in Figure3. This framework is composed of four core components: the test case generation module for generating and evaluating test cases, the fuzz testing module for executing the tests, the adapter module for interfacing with the testing environment, and the module representing the protocol under test.

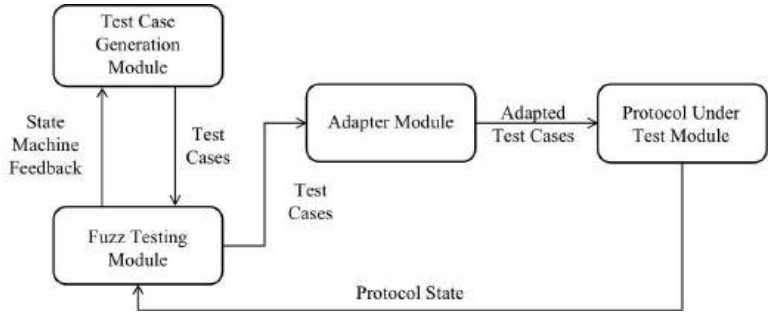


Figure 3. Vulnerability Mining Framework

3.3.1 Test Case Generation Module

The primary function of this module is to generate test cases and refine them based on the real-time feedback from the protocol's state machine, with an emphasis on generating test cases more efficiently. To this end, a compact-scale model is utilized as the Test Case Generation LLM, aiming to boost the productivity of fuzz testing. Copilot is selected for its role as the generation LLM due to its support for various programming languages and its capacity to comprehend contextual information, enabling it to produce code snippets that align with the existing code logic.

The module is bifurcated into training and application stages. In the training stage, a balanced dataset of collected test cases is assembled, which is pivotal for the model to learn the differentiation between effective and ineffective test cases. This enables the model to recognize features and patterns correlated with the discovery of vulnerabilities. During the application stage, the model not only generates initial test cases but also conducts real-time analysis of feedback to predict whether existing cases can elicit anomalous behavior from the target program. Leveraging these predictions, the model filters out ineffective test cases and selectively forwards only those effective cases that have the potential to expose program flaws back to the fuzz testing module, thus enhancing the overall testing efficiency.

3.3.2 Fuzz Testing Module

The principal function of this module is to facilitate the transmission of test cases to the adapter module and to use fuzz testing outcomes to construct a protocol state machine. It also serves as a conduit for real-time feedback from the state machine to the Test Case Generation LLM. To this end, the module is equipped with superior data processing capabilities, enabling it to thoroughly comprehend and refine intricate input data. GPT-4 is selected as the State Machine Construction LLM due to its advanced language understanding capabilities. Together with Copilot's expertise in code generation, this combination ensures that the generated test cases comply with syntactic rules and are adept at uncovering potential flaws in the target system.

The fuzz testing module plays a dual role: it aids in the construction of the state machine while also offering intelligent support throughout the testing process. The state machine dynamically modifies itself based on the feedback from test executions, guiding the ongoing refinement of test cases and thus enhancing the likelihood of uncovering new states or latent vulnerabilities. In close cooperation with the adapter module, the fuzz testing module guarantees that the generated test cases are accurately adapted and delivered to the protocol under test module.

3.3.3 Adapter Module

The adapter module's primary purpose is to act as an intermediary between the fuzz testing module and the protocol under test module under scrutiny. Its central role is to receive test cases generated by the fuzz testing module and to ensure their proper conveyance and execution in accordance with the specific format and methods demanded by the protocol under test. Given the diversity in input interfaces and communication protocols that various network protocol modules might employ, such as interactions via sockets, the adapter module offers indispensable conversion and adaptation functionalities.

This module is indispensable as it guarantees the adaptability and compatibility of fuzz testing, enabling the testing framework to integrate fluidly with a broad spectrum of testing environments. Thanks to the adapter module's sophisticated processing capabilities, test cases are accurately interpreted and executed—even when confronted with network modules that have unique input specifications or communication protocols—thus bolstering the efficiency and dependability of the testing process.

3.3.4 Protocol Under Test Module

The Protocol Under Test Module plays an essential role in fuzz testing, tasked with emulating real-world network services or applications that function according to defined communication protocols. Upon receiving the test cases from the adapter module, it processes them in line with the protocol standards, ensuring security and stability benchmarks equivalent to those in a live environment. The module reacts to the input test cases, manifesting either typical service responses or atypical error behaviors, which enables the fuzz testing module to evaluate the efficacy of the test cases through its reactions. The Protocol Under Test Module is engineered

to offer a dependable testing milieu, facilitating the thorough examination of potential security vulnerabilities without compromising the integrity of the actual system.

4. EXPERIMENTS AND ANALYSIS

In order to substantiate the viability of the dynamic feedback-driven protocol vulnerability mining method put forth in this paper and to conduct a comprehensive assessment of the optimization outcomes of the presented technique, three distinct yet representative network protocols with broad applications have been chosen for validation. These include a proprietary closed-source protocol, the prevalent MQTT protocol, and the security-intensive SSH protocol.

By conducting experimental confirmations across these three diverse network protocols, the paper seeks to evaluate extensively the efficacy and the enhancement delivered by the proposed methodology. Such an evaluation is poised to contribute a novel tool for vulnerability mining within the cyber security domain and to provide both theoretical underpinnings and practical directives for the fortification of the security measures associated with these protocols.

4.1 Experiment 1: Vulnerability Mining in Closed-Source Protocol

The proprietary protocols utilized in highly sealed terminal communications within power systems are among the most extensively implemented. The closed nature of these protocols keeps their internal workings and mechanisms largely undisclosed, amplifying the challenge of conducting vulnerability mining. Given the paramount importance of the secure and stable operation of power systems to societal functioning, the efficacy of the method proposed in this paper in unearthing potential vulnerabilities within such protocols is of substantial practical significance for bolstering the security of power systems, marking a pivotal aspect of our research.

This study contrasts the efficiency of two approaches to vulnerability mining. The experimental outcomes, depicted in Figure4, evaluate the efficiency of both methods by assessing the quantity of valid cases. The horizontal axis of the chart signifies the iterations conducted on the closed-source protocol, whereas the vertical axis represents the count of cases deemed valid during the actual execution process.

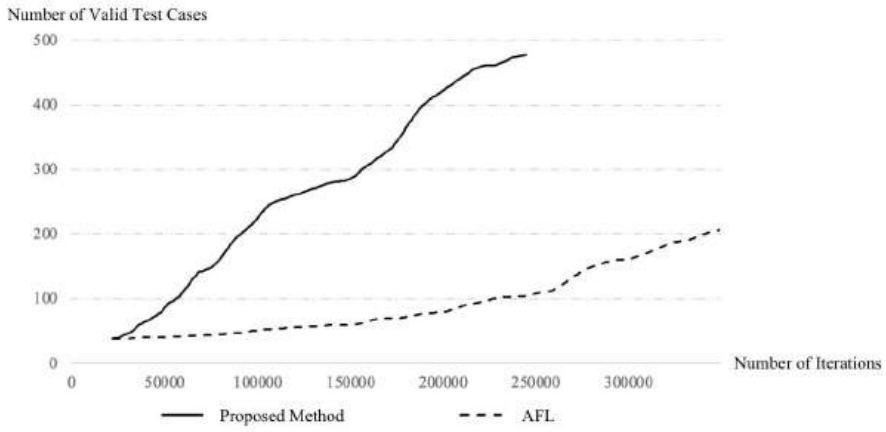


Figure 4. The Result of Experiment 1

The results clearly demonstrate the superiority of the method presented in this paper over the conventional AFL fuzz testing tool. With the same iteration count, the number of valid cases identified by our approach can exceed that of the AFL method by a factor of up to 4.34. This signifies that the enhanced fuzz testing technique is more efficient and precise in uncovering vulnerabilities within closed-source protocols. The dynamic feedback mechanism for optimizing test cases allows for a more exhaustive coverage of the code, leading to the detection of a greater number of vulnerabilities. Comparatively, the AFL method yields a significantly lower count of valid cases within the equivalent number of iterations.

4.2 Experiment 2: Vulnerability Mining in MQTT Protocol

The MQTT protocol, a lightweight publish/subscribe messaging protocol, is extensively utilized in the Internet of Things (IoT) sector. Celebrated for its minimal bandwidth usage, low power consumption, and swift real-time performance, MQTT is instrumental in facilitating data exchanges among IoT devices. Yet, amid the surge in IoT devices, concerns over the security of the MQTT protocol have correspondingly escalated. Thus, this study has chosen the MQTT protocol as one of the subjects for validation, with the goal of evaluating the proposed method's proficiency in uncovering vulnerabilities within lightweight protocols.

The study executed fuzz testing on the MQTT protocol using the 1.3.0 version of the Eclipse Paho MQTT client library, written in C. The Eclipse Paho MQTT client library is a prevalent open-source initiative that delivers MQTT client implementations across multiple platforms and supports an array of programming languages. This particular version was selected for fuzz testing due to its prevalence as a stable and widely embraced release, offering both representativeness and ubiquity. The library's open-source characteristic also simplifies the research endeavor by enabling a thorough examination of its underlying mechanisms and potential security vulnerabilities.

Fortunately, adapters for the 1.3.0 version of the Eclipse Paho MQTT client library have been previously developed, permitting the direct application of the method outlined in this paper to conduct fuzz testing on the open-source codebase of this version. Leveraging these adapters, the study performed vulnerability mining experiments on the MQTT protocol and compared the efficiency of the two methods in unearthing protocol vulnerabilities. The outcomes of these experiments are depicted in Figure 5.

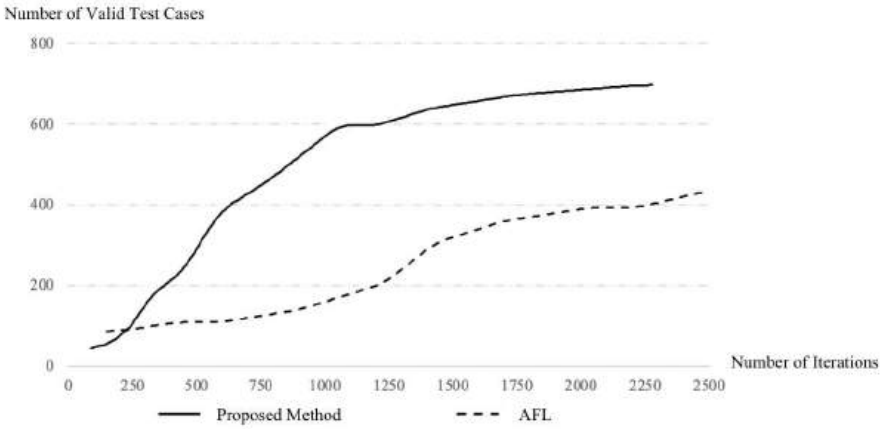


Figure 5. The Result of Experiment 2

During the initial iterations, both methodologies exhibit a low count of valid test cases. However, the approach presented in this paper increases the number of valid cases more swiftly than AFL, indicating a more efficient exploration of the input space in the early stages. With the increment of iterations, the valid case count for both methods rises, yet the method proposed here shows a more significant upward trend, especially at higher iteration counts. This suggests that our method is capable of discovering new valid cases more reliably over prolonged testing periods. Moreover, for an equivalent number of iterations, the maximum number of valid cases identified by our method is 2.73 times greater than that of the AFL method. Consequently, the dynamic feedback-based fuzz testing approach introduced in this paper has proven to be particularly effective in excavating vulnerabilities within the MQTT protocol, showcasing enhanced performance in detecting new valid cases as the iteration count increases.

4.3 Experiment 3: Vulnerability Mining in SSH Protocol

The SSH protocol is a secure, encrypted network protocol designed for safeguarding data communication between computers. It ensures the confidentiality and integrity of data in transit through encryption, positioning itself as the protocol of choice for operations such as remote login and command execution. Despite these strengths, the SSH protocol confronts significant security challenges amidst the escalating sophistication of cyber-attack

techniques. This study has chosen the SSH protocol as a subject for validation to assess the efficacy of the proposed method in vulnerability mining, particularly against encrypted protocols. We performed fuzz testing on the SSH protocol using version 7.6 of the OpenSSH source code, implemented in C.

OpenSSH, an extensively utilized open-source initiative, offers robust SSH client and server implementations compatible with multiple operating systems. The selection of this version for fuzz testing is justified by its prevalence as a stable release, making it both representative and widely applicable. The open-source characteristic of OpenSSH also aids in research by enabling a detailed examination of its internal workings and potential security vulnerabilities. Consequently, we applied both methods to mine for vulnerabilities in the SSH protocol, with the experimental findings depicted in Figure 6. The results indicate that the method introduced in this paper exhibits enhanced performance in the context of SSH protocol vulnerability mining, outperforming the AFL method at equivalent iteration counts and identifying up to 1.38 times more valid cases than the AFL method.

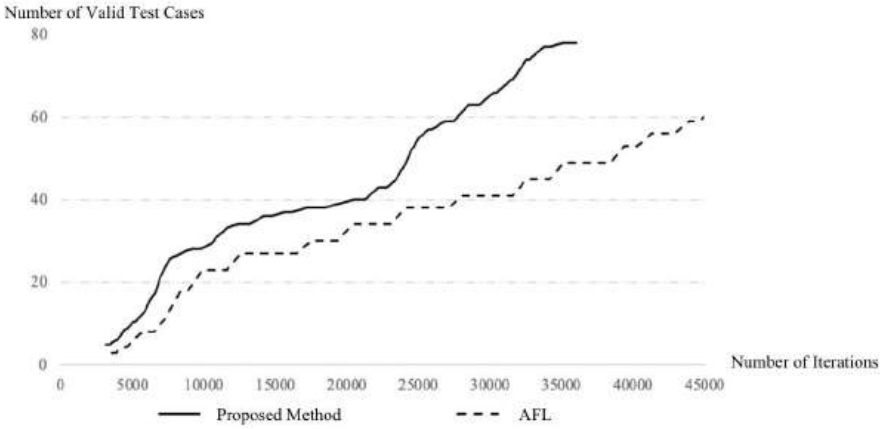


Figure 6. The Result of Experiment 3

5. CONCLUSION

This paper addressed the critical challenge of vulnerability mining in highly closed terminal protocols, which has been a daunting task due to the lack of accessibility to the source code and protocol specifications. Traditional fuzz testing methods, while effective for open-source protocols, have been less successful with closed-source protocols due to the limitations in understanding the internal workings of the protocols. This paper proposed a dynamic feedback-based vulnerability mining method that leverages LLMs to generate and optimize test cases, significantly improving the efficiency and accuracy of vulnerability detection in closed-source protocols.

The proposed method effectively addresses the imbalance between the number of valid and invalid test cases in the training samples by employing a combination of oversampling and random undersampling strategies. This ensures a balanced dataset, which is crucial for training the LLM to accurately generate test cases. The integration of LLMs with fuzz testing overcomes the limitations of traditional methods by generating test cases without relying on prior knowledge of the protocols, making it highly effective for highly closed terminal protocols that lack documentation support.

The experimental results demonstrated the superiority of the proposed method over the conventional AFL fuzz testing tool. The method showed a remarkable increase in the number of valid test cases found, with a maximum of 4.34 times more valid cases discovered compared to the AFL method in the case of closed-source protocols. Similarly, the method outperformed AFL in both the MQTT and SSH protocols, with improvements of up to 2.73 times and 1.38 times in the number of valid cases found, respectively.

The dynamic feedback mechanism, which provides real-time execution results back to the LLM, optimizes the test case generation process. This approach not only enhances the quality of the test cases but also focuses the testing process more intently on potential vulnerability areas through continuous iterative optimization. The

self-optimization trait of the proposed method makes it more adaptable and efficient against complex and highly sealed network protocols, offering a significant advancement over traditional tools.

In conclusion, the dynamic feedback-based vulnerability mining method presented in this paper has proven to be a powerful tool for uncovering vulnerabilities in highly closed terminal protocols. It offers a significant improvement over existing methods, providing a more efficient and accurate approach to ensure the security and stability of network communications in power systems and other critical infrastructures. The method's ability to adapt and evolve through continuous learning from fuzz testing data makes it a robust solution for the ever-evolving landscape of cyber threats.

ACKNOWLEDGMENTS

This work is supported by the science and technology project of State Grid Corporation of China: "Research and Application of Fuzzy Testing Technology for Power System Terminals" (Grand No. 5700-202316312A-1-1-ZN).

REFERENCES

[1] Fuzzing, I. W., "Sage: whitebox fuzzing for security testing," *SAGE* **10**(1) (2012).

[2] Wondracek, G., Comparetti, P. M., Kruegel, C., Kirda, E., and Anna, S. S. S., "Automatic network protocol analysis.," in [*NDSS*], **8**, 1–14, Citeseer (2008).

[3] Cha, S. K., Avgerinos, T., Rebert, A., and Brumley, D., "Unleashing mayhem on binary code," in [*2012 IEEE Symposium on Security and Privacy*], 380–394, IEEE (2012).

[4] Ozturk, O. S., Ekmekcioglu, E., Cetin, O., Arief, B., and Hernandez-Castro, J., "New tricks to old codes: can ai chatbots replace static code analysis tools?," in [*Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference*], 13–18 (2023).

[5] Miller, B. P., Zhang, M., and Heymann, E. R., "The relevance of classic fuzz testing: Have we solved this one?," *IEEE Transactions on Software Engineering* **48**(6), 2028–2039 (2020).

[6] Gao, Z., Dong, W., Chang, R., and Wang, Y., "Fw-fuzz: A code coverage-guided fuzzing framework for network protocols on firmware," *Concurrency and Computation: Practice and Experience* **34**(16), e5756 (2022).

[7] Gorbunov, S. and Rosenbloom, A., "Autofuzz: Automated network protocol fuzzing framework," *Ijcsns* **10**(8), 239 (2010).

[8] Lu, Y., Shao, K., Sun, W., and Sun, M., "Rgchaser: A rl-guided fuzz and mutation testing framework for deep learning systems," in [*2022 9th International Conference on Dependable Systems and Their Applications (DSA)*], 12–23, IEEE (2022).

[9] Fioraldi, A., Mantovani, A., Maier, D., and Balzarotti, D., "Dissecting american fuzzy lop: a fuzzbench evaluation," *ACM transactions on software engineering and methodology* **32**(2), 1–26 (2023).

[10] Godboley, S., Dutta, A., Pisipati, R. K., and Mohapatra, D. P., "Ssg-afl: Vulnerability detection for reactive systems using static seed generator based afl," in [*2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*], 1728–1733, IEEE (2022).

[11] Zhao, J., Rong, Y., Guo, Y., He, Y., and Chen, H., "Understanding programs by exploiting (fuzzing) test cases," *arXiv preprint arXiv:2305.13592* (2023).

[12] Li, Y., Chen, B., Chandramohan, M., Lin, S.-W., Liu, Y., and Tiu, A., "Steelix: program-state based binary fuzzing," in [*Proceedings of the 2017 11th joint meeting on foundations of software engineering*], 627–637 (2017).

[13] Chernyavskiy, A., Ilvovsky, D., and Nakov, P., "Transformers:"the end of history" for natural language processing?," in [*Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*], 677–693, Springer (2021).

[14] Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al., "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068* (2022).

[15] Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., Bahri, D., Schuster, T., et al., "Ul2: Unifying language learning paradigms," *arXiv preprint arXiv:2205.05131* (2022).

[16] Ahmad, W. U., Chakraborty, S., Ray, B., and Chang, K.-W., "Unified pre-training for program understanding and generation," *arXiv preprint arXiv:2103.06333* (2021).

[17] Guo, D., Lu, S., Duan, N., Wang, Y., Zhou, M., and Yin, J., "Unixcoder: Unified cross-modal pre-training for code representation," *arXiv preprint arXiv:2203.03850* (2022).

[18] Oliinyk, Y., Scott, M., Tsang, R., Fang, C., Homayoun, H., et al., "Fuzzing busybox: Leveraging llm and crash reuse for embedded bug unearthing," *arXiv preprint arXiv:2403.03897* (2024).

[19] Gong, X., Li, M., Zhang, Y., Ran, F., Chen, C., Chen, Y., Wang, Q., and Lam, K.-Y., "Effective and evasive fuzz testing-driven jailbreaking attacks against llms," *arXiv preprint arXiv:2409.14866* (2024).

[20] Xia, C. S., Paltenghi, M., Le Tian, J., Pradel, M., and Zhang, L., "Fuzz4all: Universal fuzzing with large language models," in [*Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*], 1–13 (2024).

TFOEE—An Event Extraction Model for Police Text

Zirong Su[a], Yongbing Gao[*a], Xiaoang Chen[b], Lidong Yang[a]
[a]School of Numerical Industry, Inner Mongolia University of Science and Technology, Baotou, Inner Mongolia 014010; [b]Baotou Public Security Bureau Information Center, Baotou, Inner Mongolia 014010
suzr2024@163.com, *gaoyongbing@163.com, yunchouweibo@qq.com, yld_nkd@imust.edu.cn

ABSTRACT

In the event extraction task, the existing models use trigger words as a bridge to extract structured information, but the extraction effect is not ideal when faced with police texts without trigger words or fixed trigger words.To solve this problem, an end-to-end trigger-free word overlapping event extraction model was proposed—TFOEE. In this model, the task of extracting overlapping events without triggering words is transformed into a task of identifying relationships based on grid filling strategy, event types and word fragments. Experiments show that the accuracy, recall rate and F1 value of TFOEE model are better than those of baseline model on police text dataset. And the F1 value of the TFOEE model reached 94.1%.

Keywords: Overlap Event Extraction; Trigger-free; Grid Fill strategy; End-to-End

1. INTRODUCTION

In public security, vast amounts of oral alarm information are stored as unstructured data with high value but low accessibility. Given event extraction techniques can extract structured info from unstructured text, this paper applies it to police texts. Police texts have two key features: (i) An alarm may contain multiple events with shared arguments (e.g., time, name, address). (ii) Trigger words for the same event type may vary or be entirely absent. Texts with these characteristics are classified as overlapping event texts (two or more event types) or single event texts (one event type), and cases with multiple or no trigger words are termed unfixed or no trigger words. This is shown in Table 1. Addressing these, we propose the Trigger-Free Overlapping Event Extraction (TFOEE) model.

Table 1 Examples of single events and overlapping events and examples of cases where the event type is the same but the trigger word is not fixed and there is no trigger word

	Police text	Extraction results
Single event	On February 16th,2022, Ren Xiaoxia alarm, reported on February 16th,2022 in pine village area 41 building 4 home through the TikTok group of code share join WeChat group to do brush single task, and through the group site link to download the "Cardi" app, in the "Cardi" app do task top-up 8 times, each brush is under the guidance of the number of goods and goods, click after completing the task, the customer service is to continue to brush single transfer for task completed refused to let withdrawal, was cheated 2390 yuan.	Event Type: Brush cash back Trigger Word: do brush single task Arguments and roles: Alarm Time: February 16th,2022 Alarm Person: Ren Xiaoxia Cheated Time: February 16th,2022 Cheated Address: pine village area 41 building 4 Way: join WeChat group to do brush single task Cheated Money: 2390 yuan

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190H · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057733

	On May 12th,2022, report Hao Jing alarm said, on May 7th,2022 in Baotou Kundu district, Xinxing Yipin 1 building 201 home, through the Himalayan app, contact voice recording, through private chat, the other party provide links: https://btrry.cn/BNUs, use mobile phone browser download "day orange media" app, registered login account, under the guidance to buy goods, promised to reward each order 20 yuan, after providing bank account prepaid phone 54553 yuan, after the platform cannot be withdrawal and no reward.	Event Type: Brush cash back Trigger Word: Arguments and roles: Alarm Time: May 12th,2022 Alarm Person: Hao Jing Cheated Time: May 7th,2022 Cheated Address: Baotou Kundu district, Xinxing Yipin 1 building 201 Way: under the guidance to buy goods Cheated Money: 54553 yuan
Overlap events	On February 17th, 2022, The reporter Yin Xiaoying reported to the police that on February 16th,2022, in his house 5-1504,Qiankouzi Kundulun District, Baotou City, Being pulled into a wechat group by your wechat friends, Publish tasks in the group, Like and Follow to receive red packets, Then provide a link: https:\/\/a.fkccbh.xyz, Download the "yo cool" APP with your own browser, Contact the receptionist, The other side led to the landing of "Jinxiu Group" to bet on "large and small single and double", Transfer of 148492 yuan from a bank account provided to the other party, After the withdrawal cannot be made, Found out that being cheated, Was cheated 148316.2 yuan.	Event Type 1: Brush cash back Trigger Word: Like and Follow to receive red packets Arguments and roles: Alarm Time: February 17th,2022 Alarm Person: Yin Xiaoying Cheated Time: February 16th,2022 Cheated Address: 5-1504,Qiankouzi Kundulun District, Baotou City Way: Like and Follow to receive red packets Cheated Money: 148316.2 yuan Event Type 2: Online gambling Trigger Word: Bet on Arguments and roles: Alarm Time: February 17th,2022 Alarm Person: Yin Xiaoying Cheated Time: February 16th,2022 Cheated Address:5-1504,Qiankouzi Kundulun District, Baotou City Way:landing of "Jinxiu Group" to bet on "large and small single and double" Cheated Money: 148316.2 yuan

2. RELATED WORK

The evolution of event extraction technology has shifted from pattern matching to traditional machine learning and now to deep learning. Deep learning has marked a new milestone, with Chen. proposing DMCNN to enhance event extraction by retaining key sentence information. The traditional method is based on the pipe method and the cascade method, but either way will cause error accumulation, resulting in incorrect transmission. Thus, this paper adopts an end-to-end extraction method to avoid such errors. Grid filling has proven effective in overlapping event extraction tasks by illustrating the relationships between word pairs. Wang utilized link tags to label these word pairs, enabling single-stage joint relationship extraction. Ning further refined this by using triple label tags that incorporated trigger words, arguments, and their respective types or roles, thereby achieving one-step event extraction. Many event extraction models prioritize trigger word extraction, as this serves as the foundation for subsequent tasks. However, these models often encounter limitations when dealing with texts where trigger words are not fixed or may not exist at all, such as in alarm information. This can lead to inaccuracies in role classification or missed argument extractions. To address this issue, this paper proposes a grid filling strategy that incorporates the concept of no trigger words. By designing two types of relationship labels, the model can handle cases where trigger words are unfixed or absent.

3. TFOEE MODEL

The goal of police text event extraction is to identify the event types, arguments and roles of the text. In this paper, the extraction task is transformed into a relationship recognition task based on grid filling strategy, event types and word fragments. The TFOEE model structure is shown in Figure 1, which is divided into three parts: the coding layer, the event type perception layer, and the relationship prediction layer.

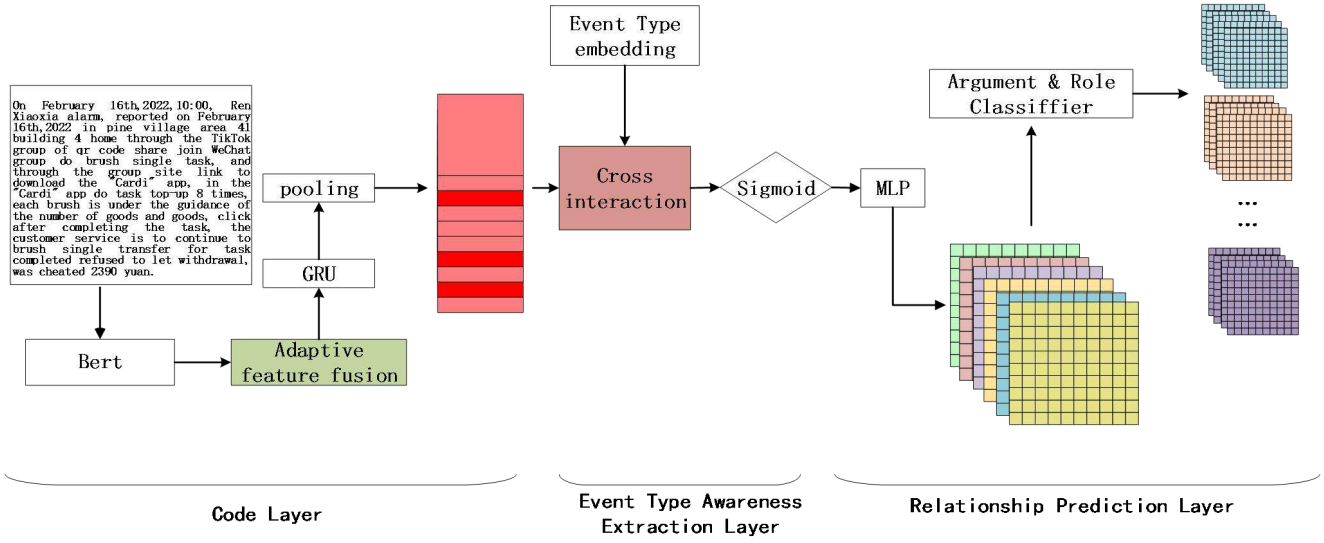


Figure 1 Structure diagram of TFOEE model

3.1 Code Layer

Given that the input text $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$ is divided into a character sequence, the word vector, text vector and position vector of the character sequence are obtained by loading the pre-trained language model BERT to obtain the vector:

$$\boldsymbol{S}_{BERT} = \boldsymbol{BERT}(\mathbf{S}) \tag{1}$$

Through the adaptive feature fusion module designed by us, we carry out feature fusion on the text. The adaptive fusion layer fully connects the hidden information of the last six layers obtained after the model BERT and then outputs it. Namely:

$$\begin{cases} \boldsymbol{H}_1 = \boldsymbol{S}_{BERT} \times \mathbf{W}_1 \\ \boldsymbol{H}_2 = \boldsymbol{H}_1 \times \mathbf{W}_2 \\ \boldsymbol{H}_3 = \boldsymbol{H}_2 \times \mathbf{W}_3 \\ \boldsymbol{H}_4 = \boldsymbol{H}_3 \times \mathbf{W}_4 \\ \boldsymbol{H}_5 = \boldsymbol{H}_4 \times \mathbf{W}_5 \\ \boldsymbol{H}_6 = \boldsymbol{H}_5 \times \mathbf{W}_6 \end{cases} \tag{2}$$

and

$$\boldsymbol{S}_x = \boldsymbol{ReLU}(\boldsymbol{H}_6 + \mathbf{b}) \tag{3}$$

Among them, $\mathbf{W}_{1-6}$ is the learnable parameter and $\mathbf{b}$ is the bias vector. Because the police text is too long, GRU is used to strengthen the information interaction of word representation, and the features of deep word representation are extracted. GRU forgets $\boldsymbol{z}_t$ and $\boldsymbol{r}_t$ selects memory of input content through update gate and reset gate. Its formula is:

$$\begin{cases} z_t = \mathbf{sigmoid}\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \\ r_t = \mathbf{sigmoid}\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \end{cases} \tag{4}$$

Among them, the input information $\boldsymbol{x}_t$ is the current time, and the hidden state $\boldsymbol{h}_{t-1}$ is the previous time. Then the word expression of the vector $\boldsymbol{S}_x$ after passing through GRU is:

$$\boldsymbol{G} = \mathbf{GRU}(\boldsymbol{S}_x) = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_N\} \tag{5}$$

Finally, a maximum pooling layer is used to reduce vector dimensions to improve the generalization ability of the model. That is, the generated word is expressed as:

$$\boldsymbol{H} = \mathbf{maxpool}(\boldsymbol{G}) = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N\} \tag{6}$$

At this point, the encoding layer is completed, and the output $\boldsymbol{H}$ is a word representation.

3.2 Event Type Awareness Extraction Layer

The event type awareness extraction layer includes a cross-attention mechanism and a Sigmoid function. We pass the word representation output by the coding layer through a cross-attention mechanism with each event type. The cross-attention mechanism can selectively pay attention to the important information of the text. The $Q(query)$ is the feature that needs attention, the $K(key)$ and $V(value)$ is the global feature. Its formula is:

$$Attention(Q,K,V) = softmax\left(\frac{QK^{T}}{\sqrt{d_h}}\right)V \tag{7}$$

At this time, you will get group word representations $\mathbf{M}$, and each group word representation incorporates the information characteristics of an event type e_m. That is, the word in which the event type information is fused is expressed as:

$$V^{e_m} = Attention(H,e_m,e_m) = softmax\left(\frac{He_m^{T}}{\sqrt{d_h}}\right)e_m \tag{8}$$

Where, $e_m \in \mathbf{E}$, $\mathbf{E} = \{e_1, e_2, \ldots, e_M\}$, $\mathbf{M}$ is the number of event types. Then, through the Sigmoid function, the word representations that do not meet the event type e_m are filtered, and the output result is that all word representations that meet the event type e_m are:

$$V^{t} = sigmoid\left(V^{e_m}, e_m\right) \tag{9}$$

and

$$sigmoid \begin{cases} \geq 0.5, & 1 \\ other, & 0 \end{cases} \tag{10}$$

That is, the output vector V^t is all word representations that match the event type e_m. Similarly, the of the group $\mathbf{M}$ event type can be obtained after $\mathbf{M}$ event type passes through the awareness extraction layer in turn.

3.3 Relationship Prediction Layer Based on Grid Fill

3.3.1 Grid fill Strategy and Relationship Label

The grid filling method is:: (i) Constructing multiple matrices of $\mathbf{N \times N}$ for multiple word fragments composed of word representations of each event type, and determining the event type of the text by whether the number of matrices matches the number of predefined role labels of the event type; (ii) Using the relationship between event types and word fragments, design two kinds of relationship tags, argument relationship tags $\mathbf{A\text{-}R}$ and role relationship tags $\mathbf{A\text{-}R(i)}$.

Firstly, a plurality of matrices $\mathbf{N \times N}$ are constructed for the word fragments composed of each event type word representation. For example, when the event type is e_m, the word representation and the text are matched $V^t = \{s_3, \ldots, s_i, \ldots, s_j, \ldots\}$ and $\mathbf{S} = \{s_1, s_2, \ldots, s_N\}$, judged to determine whether the maximum boundary between the words $\mathbf{S_i}$ and $\mathbf{S_j}$ is continuous. If it is not continuous, it is considered that the word fragment $\langle s_i, s_j \rangle$ cannot be formed, and if it is continuous, the word fragment composed between the maximum continuous words and words is constructed into a matrix $\mathbf{N \times N}$, all the maximum word fragments V^t are used to form a plurality of matrices $\mathbf{N \times N}$, and all the matrices are regarded as a set $\mathbf{A}$, $\mathbf{A} = \langle s_i, s_j \rangle_k = \langle X, Y \rangle_k$, where in the position $s_i = X$ is the head of the word fragment,and in the position $s_j = Y$ is the tail of the word fragment. And the number of matrices is $K =$ the set.

At this time, the event type is determined, whether the text belongs to the event type e_m is determined by using whether the number of predefined role tags $\mathbf{I}$ is the same with K. When $\mathbf{K = I}$, we think that the text belongs to the event type. When $\mathbf{K \neq I}$, there will be two situations. One is that the text does not belong to the event type; Second, the text belongs to this event type, but the number of arguments in the text does not match the number of roles of the event type. Because most of the police texts are complete texts, that is, the number of arguments in the text is mostly the same as the number of event type roles, the probability of the second situation is very low. In the real police information, this police text lacking necessary information is divided into special text, so this special text is temporarily ignored in the model design,

that isn't. while $\mathbf{K} \neq \mathbf{I}$, the text is considered not to belong to the event type and discarded. We judge the word representation of multiple groups of event types in the text in turn, that is, judge the group $\mathbf{M}$ event types, and think that the qualified event types are all the event types of the text, so as to solve the problem of overlapping events in the text. The advantage of this method is that each group of event types is independently verified and does not affect each other, and there will be no improper matching of the number of labels due to the same argument between overlapping events.

When $\mathbf{K} = \mathbf{I}$, we designed the following labels for the relationship between arguments and roles:

$\mathbf{A-R}$: Represents an argument of an event type e_m, that is, a word fragment in the word fragment set $\mathbf{A}$ that is an argument of an event type e_m;

$\mathbf{A-R(i)}$: indicates that the argument is this role, that is, the role corresponding to the highest probability score of the argument in the predefined role label $\mathbf{i}$ of the event type e_m;

None: Represents an argument that is not an event type e_m, that is, a word fragment in the word fragment set $\mathbf{A}$ that is not an event type argument of e_m.

The predefined role label $\mathbf{i}$ indicating the event type e_m, for example, when the event type is "Brush cash back" of $\mathbf{i}$ = {Alarm Time;Alarm Person;Cheated Time;Cheated Address;Way;Cheated Money}.

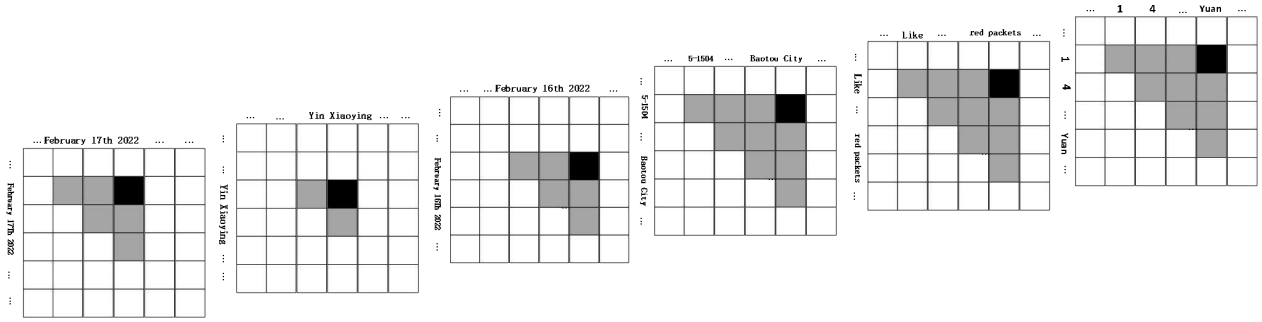


Figure 2 Example of grid filling strategy

As shown in Figure 2, the text "On February 17th, 2022, The reporter Yin Xiaoying reported to the police that on February 16th,2022, in his house 5-1504,Qiankouzi Kundulun District, Baotou City, Being pulled into a wechat group by your wechat friends, Publish tasks in the group, Like and Follow to receive red packets, Then provide a link: https:\/\/a.fkccbh.xyz, Download the "yo cool" APP with your own browser, Contact the receptionist, The other side led to the landing of "Jinxiu Group" to bet on "large and small single and double", Transfer of 148492 yuan from a bank account provided to the other party, After the withdrawal cannot be made, Found out that being cheated, Was cheated 148316.2 yuan. "Examples of grid filling strategy and relationship prediction. Where in, the horizontal direction represents the head $\mathbf{X}$ of the word fragment, the vertical direction represents the tail $\mathbf{Y}$ of the word fragment, the matrix cell $\langle \mathbf{X}, \mathbf{Y} \rangle$ represents the word fragment composed of the head and tail, and the word fragment set $\mathbf{A}$ includes a plurality of word fragments $\langle \mathbf{X}, \mathbf{Y} \rangle_K$ including arguments, that is, the colored box represents the set $\mathbf{A}$, and the black box represents the set $\mathbf{A-R}$; Gray boxes indicate None; Different matrices represent different roles, and black boxes on different matrices represent the role of this argument $\mathbf{A-R(i)}$. Therefore, $\mathbf{A-R}$ namely "February 17th,2022", "Yin Xiaoying", "February 16th,2022", "5-1504,Qiankouzi Kundulun District, Baotou City","Like and Follow to receive red packets" and "148316.2yuan", of which 6 is the number of predefined labels $\mathbf{i}$ for event type "Brush cash back". $\mathbf{A-R(i)}$ is that the corresponding roles of "February 17th,2022", "Yin Xiaoying", "February 16th,2022", "5-1504,Qiankouzi Kundulun District, Baotou City", "Like and Follow to receive red packets", and "148316.2yuan" The sequence is "Alarm Time","Alarm Person", "Cheated Time", "Cheated Address", "Way", and "Cheated Money". At this time, for this overlapping event, the arguments and roles whose event type is "Brush cash back" are extracted. Similarly, we traverse the word representations of all event types in turn $\mathbf{M}$, that is, we can judge that the event types of this text belong to "Brush cash back" and "Online gambling".

3.3.2 Argument relationship prediction based on grid fill

After passing through the event type awareness extraction layer, a text can get a word representation of the group event type $\mathbf{M}$. First, the word representation V' is passed through a Multilayer Perceptron (MLP). The MLP can learn the hidden feature representation of the word.:

$$V = \mathrm{MLP}(V') \tag{11}$$

Then, according to the network filling strategy, all event types and corresponding word fragment sets that match the text are found out, that is, all event types and word fragment sets that match the text are screened out by using the number $\mathbf{K}$ of word fragment sets and the number of predefined role labels $\mathbf{I}$ of event types.

For each event type that fits the text, that is, all cases that satisfy the condition $\mathbf{K} = \mathbf{I}$, we use the additive attention scoring function to score all candidate word fragments in each word fragment $\langle \mathbf{X}, \mathbf{Y} \rangle$. The additive attention scoring function can measure the correlation between the query vector $\mathbf{x}$ and $\mathbf{q}$ the input vector. The formula is:

$$\mathbf{s}(\mathbf{x}, \mathbf{q}) = \mathbf{V}^{\mathrm{T}} tanh\left(\mathbf{W}\mathbf{x} + \mathbf{U}\mathbf{q}\right) \tag{12}$$

Among them, $\mathbf{V}, \mathbf{W}, \mathbf{U}$ is a learnable parameter. That is e_m, in the event type, the score $\mathbf{r_m}$ of each candidate word fragment $\langle \mathbf{X}, \mathbf{Y} \rangle$ in the word fragment and the event type e_m is:

$$\mathbf{r}_{\mathrm{m}}(e_m, \langle \mathbf{X}, \mathbf{Y} \rangle) = \mathbf{W}_{\mathbf{Y}}{}^{\mathrm{T}} tanh\left(\mathbf{W}_{\mathbf{e}_m} e_m + \mathbf{W}_{\mathbf{X}} \langle \mathbf{X}, \mathbf{Y} \rangle\right) \tag{13}$$

Among them, $e_m \in \mathbf{R}^{\mathrm{E}}$ is the event type, $\langle \mathbf{X}, \mathbf{Y} \rangle \in \mathbf{R}^{\mathrm{X} \times \mathrm{Y}}$ is the candidate word fragment, $\mathbf{W}_{\mathbf{e}_m} \in \mathbf{R}^{\mathrm{Y} \times \mathbf{e}_m}$ $\mathbf{W}_{\mathbf{X}} \in \mathbf{R}^{\mathrm{Y} \times \mathrm{X}}$ $\mathbf{W}_{\mathbf{Y}} \in \mathbf{R}^{\mathrm{Y}}$ is the learnable parameters. The scores of all candidate word segments $\langle \mathbf{X}, \mathbf{Y} \rangle$ except entities in each word segment $\langle \mathbf{X}, \mathbf{Y} \rangle^{\mathbf{n}}$ are arranged in descending order, and the highest score is judged as the argument that conforms to the event type, which $\mathbf{n}$ is the number of candidate segments.

That is, in a case where the event type is e_m and $\mathbf{K} = \mathbf{I}$, an argument whose group $\mathbf{I}$ matches the event type e_m can be obtained.

3.3.3 Role relationship prediction based on grid fill

After obtaining the group arguments that match the event type e_m, the role relationship $\mathbf{I}$ is predicted.First, argument $\mathbf{I}$ is matched with the predefined role label $\mathbf{i}$ of the event type e_m one by one, which is expressed as $\mathbf{y}$:

$$\mathbf{y}(\langle \mathbf{X}, \mathbf{Y} \rangle) = argmax\, \mathbf{r}_{\mathrm{m}} \tag{14}$$

The role corresponding to the highest probability score of the argument on a certain role is judged as the correct role:

$$\mathbf{P}_{\mathrm{m}}\left(\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathrm{i}}\right) = \frac{e^{r_m\left(e_m, \langle X, Y \rangle_i\right)}}{\sum_{\mathrm{i}=1}^{\mathrm{I}} e^{r_m\left(e_m, \langle X, Y \rangle_i\right)}} \tag{15}$$

At this point, the relationship of event types, arguments and roles has been identified, that is, an event extraction has been completed. After $\mathbf{M}$ times, all the event information of a text can be extracted.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Dataset construction and environment configuration

This paper selects the real telecommunications network fraud alarm information of a city public security bureau, and after desensitization processing, constructs a Police Text event extraction dataset (PT-data). The data set PT-data defines 9 types of events according to the actual situation. There are 9 types of events and 9 arguments, with a total of 3,165 samples and 1,045 overlapping samples, accounting for about 33%. According to the labeling rules, during the experiment on the event extraction model based on trigger words, the data set PT-data is labeled with trigger words, and the trigger word with the largest proportion of all trigger words corresponding to an event type is identified as the trigger word of the event type. The remaining trigger words are identified as unfixed or no trigger words, with 348 samples, accounting for about 11%. All the above are constructed through manual annotation. In the experiment, the training set, the verification set and the test set were randomly selected according to the ratio of 8: 1: 1.

The TFOEE model uses Bert-Base-Chinese as the encoder, AdamW as the optimizer, the hidden layer dimension is set to 768, the batch _ size is 8, the dropout is 0.5, the learning rate is 2e-5, and the learning rate of other modules is 1e-3.

4.2 Evaluation indicators

The evaluation of the whole incident is divided into three items:

(i) Event Type Classification (TC): determine whether the event type is correctly classified;

(ii) Argument Identification (AI): determine whether the binary group ("event type", "argument") is correctly identified;

(iii) Argument Classification (AC): Determine whether the triple ("event type", "argument", "argument role") is correctly extracted.

Finally, the accuracy rate P, recall rate R and F1 values are used for comparison in these three evaluations.

4.3 Comparative experiments and analysis

To verify the effect of the TFOEE model, it was compared with the following baseline model on the PT-data dataset.

Bert-CRF: Using Conditional Random Field (CRF) to capture label dependencies works well for single event extraction tasks, but cannot solve overlapping event extraction tasks.

PLMEE[1]: Based on the two-pipeline method, the overlapping argument problem is solved by extracting arguments of specific roles by triggering words.

MQAEE[2]: Based on reading comprehension event extraction, multiple MRC BERTs are trained to perform overlapping event extraction.

CasEE[3]: Based on the three-segment joint extraction of the cascading pointer marking scheme, and the overlapping targets are separately extracted according to the previous prediction.

OneEE[4]: A labeling scheme based on label chain simultaneously identifies the relationship between trigger words and arguments and argument roles to solve the problem of overlapping events.

ODEE[5]: Vertex-based labeling scheme, using two auxiliary tasks to predict the breadth and type of event trigger words and argument entities to solve the overlapping event problem.

ChatGPT[6]: By setting appropriate prompt words, formulating prompt templates, and classifying and extracting texts.

Table 2 Performance of the model on the P T-data dataset

Model	TC			AI			AC		
	P	R	F1	P	R	F1	P	R	F1
Bert-CRF	80.2	65.6	72.2	75.1	64.3	69.3	71.5	63.7	67.4
PLMEE	86.8	85.4	86.1	86.6	85.2	85.9	84.9	84.1	84.5
MQAEE	88.9	85.6	87.2	88.9	85.1	87.0	87.7	84.3	86.0
CasEE	89.5	86.4	87.9	88.2	85.8	87.0	87.6	85.1	86.3
OneEE	91.9	87.1	89.4	90.3	86.2	88.2	88.6	86.0	87.3
ODEE	**92.1**	87.5	89.7	90.5	86.9	88.7	90.2	86.2	88.2
ChatGPT	55.6	49.2	52.2	55.1	49.0	51.9	55.1	48.5	51.6
TFOEE (ours)	91.8	**97.3**	**94.5**	**91.6**	**97.1**	**94.3**	**91.5**	**96.8**	**94.1**

Table 2 shows the results of the PT-data dataset on different models. The results showed that the TFOEE model outperformed all baselines. Through these data, we can draw the following conclusions:

(1) From the perspective of sequence labeling strategy and grid filling strategy, the F1 score of TFOEE model is 26.7% higher than that of Bert-CRF model. This proves that sequence labeling can only solve single event extraction tasks, while grid filling can solve both single event extraction tasks and overlapping event extraction tasks.

(2) Compared with PLMEE, MQAEE, CasEE, OneEE and ODEE models, the F1 score of TFOEE model is 9.6%, 8.1%, 7.8%, 6.8% and 5.9% higher than that of PLMEE, MQAEE, CasEE, OneEE and ODEE models, respectively.

Among them, the P score was improved by 1.3% and the R score was improved by 10.6% compared to the ODEE model of SOTA. It can be seen that our model is better than other models in recall rate. This is because the serious colloquialization of alarm texts leads to some text trigger words that are not fixed or have no trigger words, and the existing models to solve overlapping event extraction tasks are based on fixed trigger words, and follow-up tasks are performed through fixed trigger words. Therefore, these models will predict positive samples that are different from the trigger words specified in the trigger vocabulary but have the same event type as negative samples, resulting in low R scores. This proves that the model not based on trigger word extraction is more suitable for tasks such as alarm text.

(3) Compared with end-to-end and multi-stage perspectives, the end-to-end TFOEE, OneEE, ODEE models, the two-stage PLMEE model and the three-stage CasEE model have improved F1 scores. This proves that the end-to-end approach can avoid error propagation in overlapping event extraction tasks.

(4) Compared with OneEE and ODEE models, the F1 score of TFOEE model is still improved by 6.8% and 5.9% respectively from the perspective of relationship label of grid filling strategy. This proves that the labeling strategy of the relationship between event types and word fragments designed in this paper is more helpful to the relationship identification of arguments and roles than the labeling strategy of the relationship between trigger words and word fragments.

(5) From a model perspective, the TFOEE model is superior to ChatGPT in P, R, and F1. Although the autoregressive large model Performed better than existing models on general extraction tasks, but performed poorly on vertical domain tasks. This proves that the self-coding class pre-trained model. It is more suitable for vertical event extraction tasks in specific fields.

4.4 Ablation experiment and analysis

In order to illustrate that each module in the TFOEE model plays different roles on the experimental results on the PT-data dataset, we conducted ablation experiments on the TFOEE model, and the experimental results are shown in Table 3. Where -* indicates removal of * modules.

Table 3 Performance of ablation experiments on the P T-data dataset

	TC			**AI**			**AC**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
TFOEE	**91.8**	**97.3**	**94.5**	**91.6**	**97.1**	**94.3**	**91.5**	**96.8**	**94.1**
-Adaptive feature	88.9	96.6	92.5	88.2	95.9	91.8	86.7	95.2	90.7
-GRU	88.5	96.2	92.2	88.3	95.7	91.8	87.4	95.0	91.0
-Cross Attention	86.0	96.9	91.1	85.8	96.9	91.0	85.6	96.8	90.8

It can be seen from the results that compared with TFOEE, the R value of -GRU decreases significantly, which shows that the gated loop unit is very effective in contacting context information in the police text. -Compared with TFOEE, the P score of Cross Attention decreases significantly, indicating that the Cross Attention mechanism can effectively identify the boundary of police text entities. In addition, the F1 scores of TFOEE are higher than those of-Adaptive feature, -GRU and -Cross Attention, which shows that the combined use of the three types of enhancement modules is beneficial to improving the performance of the TFOEE model.

5. CONCLUDING REMARKS

In this paper, a police data set is constructed, and an end-to-end relationship recognition model based on grid filling strategy, event types and word fragments is proposed, which solves the problems of overlapping events and unfixed trigger words or no trigger words in warning texts. In addition, the relationship label between event types and word fragments designed in this paper proves that it is more conducive to the relationship identification of arguments and roles in police texts. The experimental results show that the F1 score of TFOEE model reaches 94.1% on the police data set, which is very suitable for texts with too many overlapping events and unfixed trigger words.

ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China (NO. 62161040), the Science and Technology Project of Inner Mongolia Autonomous Region (NO. 2021GG0023), the Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (NO. NJYT22056), the Fundamental Research Funds for Autonomous Region Directly Affiliated Universities (NO. 209-2000026), the Natural Science Foundation of Inner Mongolia Autonomous Region (NO. 2021MS06030), the Fundamental Research Funds for Inner Mongolia University of Science and Technology (NO. 2023RCTD029), and the Science and Technology Project of Inner Mongolia Autonomous Region (NO. 2023YFSW0006).

REFERENCES

[1] Yang, Sen, et al, "Exploring pre-trained language models for event extraction and generation." Proceedings of the 57th annual meeting of the association for computational linguistics, pages 5284–5294, (July 2019)

[2] Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu, "Event Extraction as Multi-turn Question Answering.", EMNLP 2020, pages 829–838,(November 2020)

[3] Sheng J W, Guo S, Yu B W, et al. "CasEE: A joint learning framework with cascade decoding for overlapping event extraction ", ACL-IJCNLP 2021,pages 164-174, (2021)

[4] Cao, Hu, et al. "OneEE: A one-stage framework for fast overlapping and nested event extraction." arXiv preprint arXiv:2209.02693 (2022)

[5] Jinzhong Ning, Zhihao Yang, Zhizheng Wang, Yuanyuan Sun, and Hongfei Lin. " ODEE: a one-stage object detection framework for overlapping and nested event extraction. " , In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, Article 574, pages 5170 – 5178. (19 August 2023); https://doi.org/10.24963/ijcai.2023/574

[6] Tong, Bao, and Zhang Chengzhi. "Extracting Chinese Information with ChatGPT: An Empirical Study by Three Typical Tasks." Data Analysis and Knowledge Discovery , pages 1-11,(9 July 2023)

Central Bank Digital Currency Design architecture: A Systematic review using text mining

SOMDA Metouole Mwinbe Yves Ghislain[a], Samuel OUYA[a], and Gervais MENDY[a]

[a]LITA (Laboratoire D'informatique, de Télécommunication et Applications), Université Cheikh Anta Diop, Dakar, Sénégal

ABSTRACT

The evolution of the monetary system has been historically catalyzed by technological advancements, socioeconomic shifts, and evolving consumer needs. With the rise of technology-driven payment systems like cryptocurrencies, instant payments, and blockchain, central banks globally have increasingly explored the potential benefits of issuing digital forms of their fiat currency, known as CBDC. Despite central banks' traditional roles in ensuring financial stability, managing currency circulation, and supporting state financing needs, they have sometimes failed to prevent macroeconomic crises and ensure price stability, leading to a decline in trust in national currencies. This has been evidenced by the emergence of private cryptocurrencies, particularly in developing countries where traditional economic systems have faltered. A well-designed CBDC holds the promise of addressing multiple goals, enhancing financial stability, promoting inclusivity, and fostering innovation in economic systems. While numerous studies have explored CBDC design since 2020, there remains a significant demand for knowledge due to the ongoing exploration and implementation of CBDCs worldwide. This paper presents a systematic literature review utilizing text mining to analyze CBDC design architecture. By examining abstracts, white papers, and conference articles, we identify common design features and topics, offering insights into the evolving landscape of CBDC design and implementation particularly for developing countries.

Keywords: TEXT MINING, DATA MINING, SLR, BLOCKCHAIN, CENTRAL BANK DIGITAL CURRENCY, CBDC ARCHITECTURE

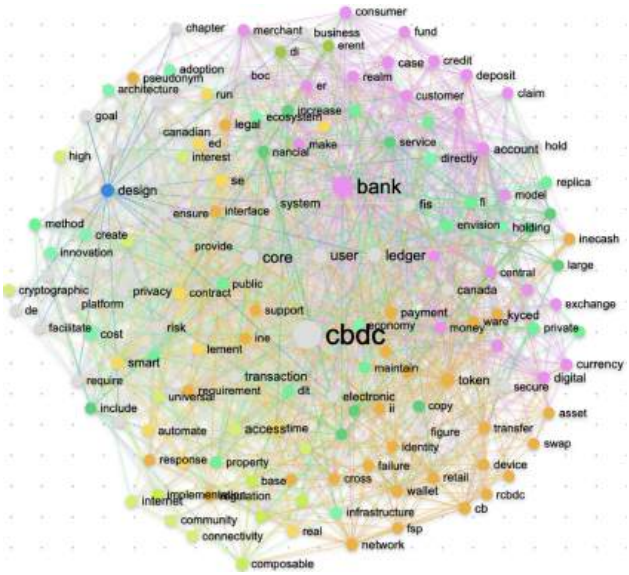
1. INTRODUCTION

The evolution of the monetary system has been driven by technological advancements, socioeconomic factors, and changing consumer needs. Recently, central banks have been exploring Central Bank Digital Currencies (CBDCs) to address challenges such as declining cash use, ineffective monetary policies, and the rise of cryptocurrencies, particularly in developing countries. CBDCs aim to enhance financial stability, inclusion, and economic flexibility, by replicating the benefits of physical cash while adapting to modern financial demands.[1] Increased competition among financial institutions, global disruptions like COVID-19, and population displacement have further emphasized the need for agile monetary policies to address emerging financial challenges. Central banks see CBDCs as a response to the proliferation of cryptocurrencies, which threaten national economic sovereignty.[2] Researchers have explored CBDC's potential for improving financial inclusion, facilitating cross-border payments, and integrating online and offline payment modes.[3] Additionally, proposals for CBDC design and implementation models suggest that CBDCs could enhance currency functionality and enable direct transfers to citizens and businesses.[4] Initially, research focused on how CBDCs could complement or replace conventional money. However, given the potential for CBDCs to disrupt financial systems, careful analysis of different design architectures is essential. This study systematically reviews 1,287 abstracts from journals, central bank white papers, and conference articles to identify common CBDC design features using text mining. Key themes include retail vs. wholesale CBDCs, account-based vs. token-based models, core technology, distribution architecture, security, and integration with existing payment ecosystems.[5] While substantial research exists on CBDC, a systematic literature review focused solely on CBDC design architecture has been lacking. Our

Further author information: (Send correspondence to Metouole Mwinbe Yves Ghislain SOMDA)
E-mail: yvesomda93@yahoo.fr, Telephone: +221 773352896

Third International Conference on Communications, Information System, and Data Science (CISDS 2024), edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190I · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3057590

(a) Word cloud of the most frequent terms in CBDC design abstracts



(b) Network diagram of CBDC design architecture

Figure 1: CBDC design analysis: Word cloud and network diagram

study aims to fill this gap by providing a comprehensive review of the state-of-the-art research on CBDC design architecture, covering technologies, functionalities, security measures, offline modes, and potential research perspectives. Our primary contribution is a rich, concise literature review that serves as a valuable resource for academics, central banks, and technology providers. It enables them to compare different CBDC architectural approaches and provides crucial insights for guiding future implementations. The goal is to address the research gap in CBDC design architecture and offer a foundational reference for stakeholders in the field. In the next line we will first focus on the methodology used after we will explain the results and will end by the conclusion.

2. METHODOLOGY

Over the past decade, text-based information retrieval has become vital across analytics domains due to the exponential growth of text content, particularly on social media and the internet. The need for robust text mining frameworks has driven the development of methodologies such as descriptive, discovery, and predictive analytics.[6] Text mining is increasingly adopted for systematic reviews, offering advantages over traditional methods, including efficient identification of relevant literature and key concepts .Systematic reviews require minimizing bias when identifying research themes, and text mining aids in quickly scanning large datasets . It also enables trend identification and relationship analysis between key concepts. Sentiment analysis offers additional insights into authors' opinions on specific topics. This section is dedicated to the research framework, including data collection, frequency analysis, topic modeling, and network-based topic analysis.

2.1 Data Collection, Inclusion and Exclusion

To analyze CBDC topics, we utilized abstracts from academic publications and central bank articles on CBDC design. We limited our search to Scopus, Google Scholar, and central bank white papers, using the Octoparse tool[7] for web scraping. Keywords such as "CBDC design," "CBDC architecture," "CBDC implementation," and "CBDC prototype" yielded 1287 relevant publications. A clear inclusion and exclusion criteria were applied to ensure data relevance. We removed incomplete abstracts, non-English papers, and duplicates, resulting in 1136 papers being excluded. A second review round disqualified 61 papers unrelated to CBDC design. Abstracts focusing on policy, financial impact, or unrelated topics were also excluded. In total, 90 abstracts were retained for text mining.

2.2 Frequency Analysis

Text mining was applied to the 90 abstracts to visualize term frequency using a word cloud, as shown in *Fig. 1.a.* Dominant terms include CBDC, Blockchain, privacy, security, distributed ledger technology (DLT), and offline functionality. These terms highlight key themes such as user-centric design, network architecture, and the potential of programmable CBDCs for the financial sector. From 2023 to early 2024, research has increasingly focused on CBDC design, particularly the technological choices between token-based and account-based systems, privacy, and security. The growing interest in CBDC design since 2020 likely stems from early prototype launches, with notable research increases in late 2023.

2.3 Topic Modeling with R and by Using Network Diagrams

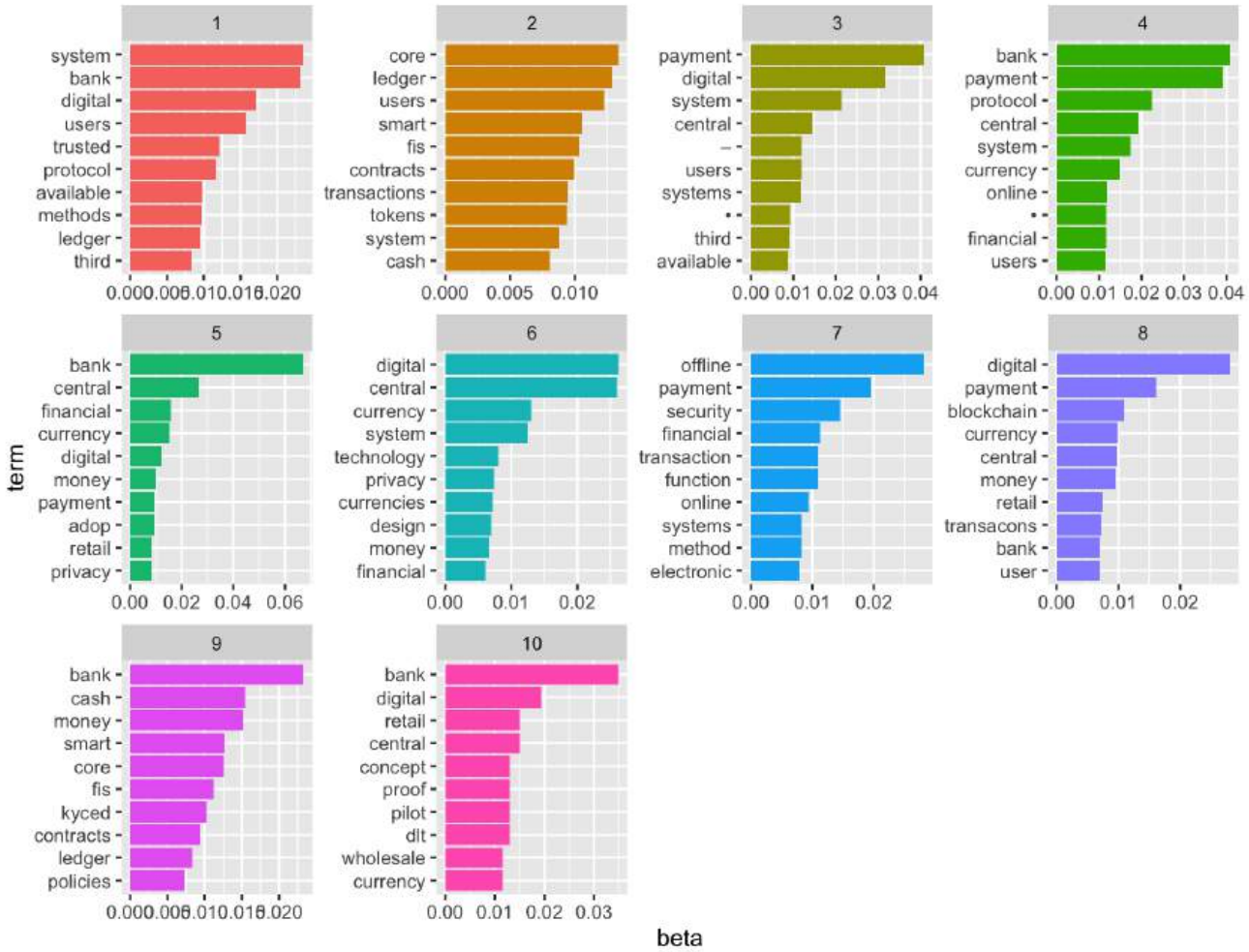


Figure 2: Topic modeling using LDA algorithm for papers abstract

Topic modeling is a powerful text mining technique for systematically analyzing large document collections. Latent Dirichlet Allocation (LDA) is a widely used model that identifies clusters of semantically related terms across texts, forming coherent themes.[8] LDA works by assessing the probability of words given a topic, P(W—T), and topics given a document, P(T—D). The most representative words for each topic are ranked by P(Wi—Tk). In this study, we employed bi-grams to enhance topic understanding, using an R script with the LDA package on over 90 abstracts [*]. To trace the evolution of CBDC design architecture, we applied an inductive approach,

[*]https://gist.github.com/metouole/ed0f8918803ded4af7d45f2583d7b143

using network visualization to map the relationships between key terms identified through topic modeling. These networks were generated with a Sass solution,[9] and the results are displayed in the network analysis.

3. RESULTS

Fig. 2 highlights the ten predominant topics in CBDC-related articles, featuring the top 10 bi-grams per topic. Themes include (1) DLT-based CBDC protocols, (2) core design using ledgers and smart contracts, (3) digital payment systems, (4) online payment protocols, (5) retail adoption and privacy, (6) financial system design, (7) secure transactions, (8) blockchain transactions, (9) smart ledger KYC policies, and (10) proof-of-concept pilot projects.Topic modeling reveals a shift in focus over time. Prior to 2020, studies centered on CBDC definitions and motivations, including financial inclusion. From 2020 onward, attention shifted to technical aspects, impacts on financial systems, payment systems, and cross-border exchanges.[5] The results from network algorithms shown in fig. 1.b align closely with LDA outcomes, confirming key themes and their evolution in CBDC research. We will deep dive in each themes in the following lines.

3.1 Retails or Wholesales CBDC

In the evolving CBDC landscape, a key distinction exists between **retail** and **wholesale CBDCs**, based on their target users and purposes. Retail CBDCs are designed for the general public, offering a digital alternative to cash, enhancing payment efficiency, accessibility, and security in everyday transactions. Wholesale CBDCs, however, are restricted to financial institutions for improving interbank payments, settlements, and high-value transactions. Fig.3 show that most of the CBDC architecture is based on the retails followed by the Wholesale. But Both retail and wholesale CBDCs offer distinct advantages and face unique challenges. Table 1 present a comparaison between the two in terms of characteristics, advantages and challenges.

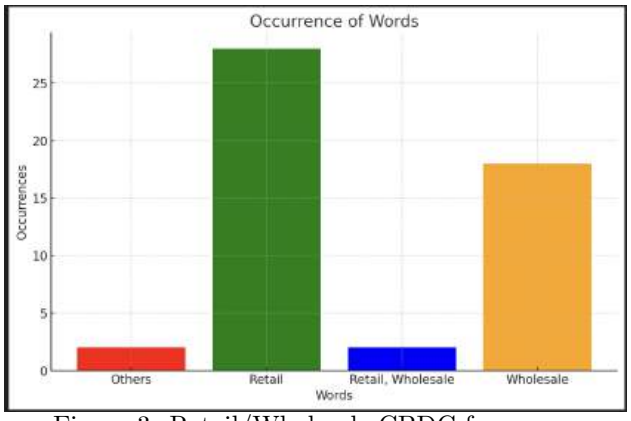


Figure 3: Retail/Wholesale CBDC frequency

Aspect	Retail CBDC	Wholesale CBDC
Characteristics	- For general public use - Enhances payment efficiency - Accessible by individuals and businesses	- Restricted to financial institutions - Improves interbank payments and settlements - Handles high-value transactions
Advantages	- Increases financial inclusion[10] - Spurs innovation and competition[11]	- Reduces settlement risk[12] - Enhances financial stability
Challenges	- Balancing privacy with AML/CTF[13] - Infrastructure requirements	- Integration with interbank systems - Cybersecurity risks (BIS, 2020)

Figure 4: table
Comparison of Retail and Wholesale CBDCs

Retail CBDCs focus on financial accessibility, while wholesale CBDCs enhance financial system operations. The choice between them depends on a country's financial system, regulations, and policy goals.[14] Both represent a shift toward digitizing finance, with implications for monetary policy and financial stability.

3.2 Structure: Account Based or Token Based

The implementation of Central Bank Digital Currencies (CBDCs) can follow one of two primary models: account-based or token-based. These models represent fundamentally different approaches to how digital currencies are held and transferred, each with its distinct characteristics, advantages, and challenges. Understanding the choice between these models is crucial for central banks in designing a CBDC system that aligns with their objectives regarding security, privacy, inclusivity, and efficiency.

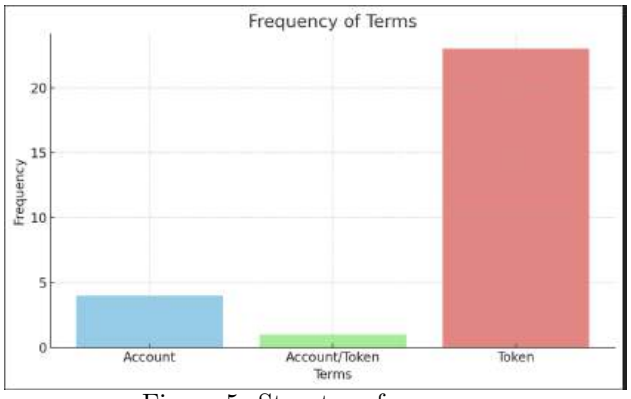


Figure 5: Structure frequency

Aspect	Account-Based CBDC	Token-Based CBDC
Characteristics	- Requires identity verification - Centralized oversight by banks	- Anonymity in transactions - Relies on cryptography for security
Advantages	- Enhanced security, compliance with AML/CFT - Direct control over monetary policy	- Greater privacy, potentially more inclusive for unbanked populations
Challenges	- Privacy concerns, regulatory complexity - Operational challenges in managing accounts	- Risk of loss/theft - Balancing anonymity with regulatory requirements

Table 1: Comparison of Account-Based and Token-Based CBDCs

Account-Based CBDC ties ownership to the identity of the account holder, requiring identity verification for transactions. **Token-Based CBDC** operates like physical cash, where ownership is determined by possession of the token, with transactions relying on cryptographic verification. Fig.4 shows that the majority of central bank prefered the Token based models than the account based. You can have an overview of they motivation on table 2 which compare the two models. The choice between account-based and token-based CBDCs depends on central bank objectives, regulations, and infrastructure. Hybrid models, combining elements of both, may offer tiered privacy levels to balance anonymity with regulatory compliance.[15] Ultimately, CBDC design requires careful consideration of trade-offs in privacy, security, compliance, and financial inclusion.

3.3 Choice of Technology Provider in CBDC Implementation Design

The implementation of CBDCs requires careful selection of technology providers and frameworks. Fig. 5 shows that R3 Corda is the most popular choice, followed by IBM, Soramitsu, G+D, and others.**R3's Corda** is a blockchain platform focused on private, secure transactions for businesses. It has been widely adopted for CBDC implementation, offering efficiency, security, and scalability.[16]**IBM's CBDC Solution** uses blockchain for both retail and wholesale CBDCs, emphasizing security, privacy, and interoperability. Its modular design allows for customization and integration into existing financial systems.[17]**Giesecke+Devrient (G+D)** introduced "Filia," a secure and flexible CBDC solution that supports both retail and wholesale applications. Filia is designed to be technology-agnostic and can operate across various platforms, enhancing accessibility.[18] Ghana's central bank has adopted this solution.

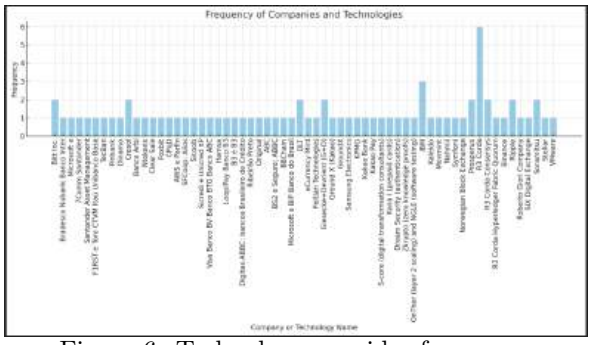


Figure 6: Technology provider frequency

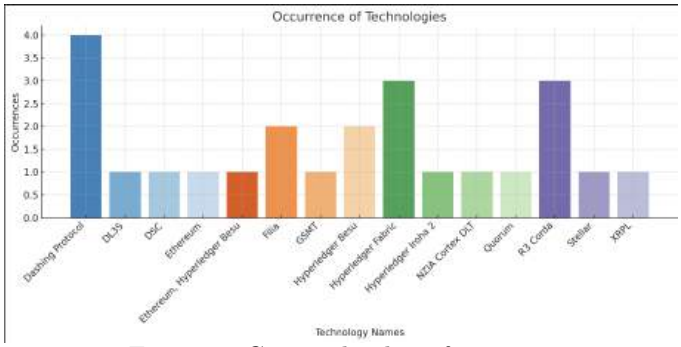


Figure 7: Core technology frequency

While R3 Corda leads with its focus on privacy, security, and regulatory compliance, IBM and G+D also offer strong features for CBDC systems. Corda's design specifically caters to financial institutions, making it a

preferred choice.[19] Additionally, Soramitsu's solution, based on Hyperledger Iroha, was adopted by the central bank of Cambodia.[20]In conclusion, technology provider selection in CBDC implementation depends on factors such as security, interoperability, privacy, and regulatory compliance. These considerations will shape the future of CBDCs in central banking.

3.4 Core Technology Solutions for CBDC Implementation

Fig. 6 shows that the most used technology in CBDC implementation is the Dashing protocol, followed by R3 Corda, Hyperledger Fabric, Besu, Filia, and others. While the Dashing protocol is primarily a consensus mechanism developed in Asia,[21] Hyperledger solutions (Fabric, Besu, Iroha) are widely adopted depending on the project's requirements.**Consensus Mechanisms in DLT for CBDC**: Consensus mechanisms are essential for CBDC transaction security. Platt and McBurney (2023)[22] emphasize the importance of Sybil attack resistance, highlighting Proof-of-Work (PoW) and Proof-of-Stake (PoS) as crucial for ensuring reliability in permissionless networks. Zhou et al. (2023)[23] review various blockchain consensus mechanisms, providing insights into their selection based on the specific needs of CBDC systems.**Smart Contracts and Consensus in CBDC Transactions**: Du et al. (2023)[24] discuss the integration of blockchain with smart contracts and PoS consensus mechanisms, improving the management of CBDC transactions and ensuring better social welfare and rationality.**Blockchain Applications for CBDC**: Sethaput and Innet (2021)[25] explore how blockchain can be tailored for CBDC, providing secure, efficient, and transparent solutions for central bank digital currencies.In conclusion, core technology solutions for CBDC implementation rely heavily on blockchain advancements, focusing on consensus mechanisms, smart contracts, and cryptographic security to ensure the integrity, efficiency, and innovation of digital currency systems.

3.5 CBDC Distribution Architecture: Integration and Implications

The implementation of CBDC merges innovative technologies with traditional financial systems, ensuring scalability and reliability in digital currency deployment. This review explores CBDC distribution architecture, focusing on the integration of Distributed Ledger Technologies (DLTs), non-DLT systems, and permissioned blockchain networks, and their broader implications for financial inclusion and stability.**Integration of DLT and Non-DLT Systems:** Sasongko and Yazid (2020)[26] propose a hybrid architecture where wholesale DLT networks are combined with retail systems. This design allows digital tokens to be accessed through commercial bank accounts, ensuring scalability, user accessibility, and operational reliability in CBDC systems.**Permissioned Blockchain Networks for CBDC:** Bhawana and Kumar (2021)[27] suggest a two-layer architecture, employing a permissioned blockchain network (PBN) for distributing digital currency between central banks and commercial banks. This model enhances regulatory control over the issuance and circulation of digital tokens, while maintaining user privacy and compliance.**CBDC Policies in Open Economies:** Kumhof et al. (2023)[28] highlight the effects of interest-bearing retail CBDCs on bank deposits, using a two-country DSGE model. Their findings indicate that optimized CBDC interest rates, along with fiscal stabilizers, can reduce exchange rate volatility and provide welfare gains, demonstrating the importance of carefully calibrated monetary policies in CBDC deployment.**Impact on Financial Inclusion and Stability:** Fullerton and Morgan (2022)[29] analyze the role of China's digital yuan in enhancing financial inclusion and stability, emphasizing the significance of CBDC design in supporting broader access to financial services and maintaining system stability.In conclusion, the success of CBDC distribution lies in the seamless integration of DLT with traditional systems, the deployment of permissioned blockchain networks for regulated issuance, and the formulation of policies that foster financial inclusion and stability. These studies underscore the need for a well-structured architecture to ensure scalability, security, and regulatory compliance in CBDC systems.

3.6 Security Measures and functionalities

Implementing a reliable CBDC system requires robust security strategies to safeguard the integrity, confidentiality, and availability of digital transactions. Key measures include:**Fraud Detection and Machine Learning:** Techniques such as feature extraction, data sampling, and machine learning algorithms are essential for detecting and preventing fraudulent activities within CBDC systems.[30]**Cryptographic Security:** Cryptographic techniques are vital for protecting against hardware vulnerabilities like Rowhammer attacks, ensuring the integrity of the system.[31] A reliable CBDC system should also support essential functionalities for efficiency and security,

including secure digital wallets, seamless payment integration, and privacy through smart contracts. It should enable cross-border payments and support both online and offline transaction capabilities. An online approach enables real-time transactions, while an offline system ensures accessibility in areas with limited connectivity, balancing both to maximize accessibility and security, as demonstrated by China's DCEP project.[18]

3.7 Integration with Existing Payment Infrastructure

Integrating CBDC into existing financial systems requires careful planning: **Interoperability:** CBDCs must integrate with existing payment networks (e.g., SWIFT, ACH) using common standards and APIs.[32]**Security and Privacy:** Security measures such as encryption and privacy protections are essential to safeguard user data during transactions.[33]**Regulatory and Technical Considerations:** Adapting regulatory frameworks and conducting pilot tests ensures smooth integration into existing systems, minimizing risks and enhancing user adoption.[?] In conclusion, a successful CBDC system requires integration with current payment infrastructures, advanced security protocols, and a phased rollout approach to ensure stability and inclusivity.

4. CONCLUSION AND FUTURE WORK

This systematic literature review explores the architecture of CBDC design through text mining, analyzing various studies, white papers, and technical documents to understand the evolving landscape of CBDC implementation. CBDC implementation involves balancing security, privacy, regulatory compliance, efficiency, and inclusivity. Key design decisions include choosing between retail and wholesale models, account-based or token-based systems, and selecting appropriate technology providers. Major focuses are integration with existing payment systems, robust security measures, and enabling both online and offline transactions. Leading technology platforms like R3 Corda, IBM's blockchain solution, and G+D's Filia offer enhanced privacy, interoperability, and offline support. Challenges include complex integration, adoption hurdles, balancing privacy with regulation, and addressing technical issues. Developing countries face additional concerns like financial inclusion and infrastructure readiness. CBDCs represent a significant advancement in monetary systems, potentially transforming financial transactions, monetary policy, and economic inclusivity. Future research should address knowledge gaps in global financial stability, offline transactions, and universal interoperability standards.

REFERENCES

[1] Choi, K. J., Henry, R., Lehar, A., Reardon, J., and Safavi-Naini, R., "A Proposal for a Canadian CBDC," *SSRN Electronic Journal* (2021).

[2] Dionysopoulos, L., Marra, M., and Urquhart, A., "Central Bank Digital Currencies: A Critical Review," *International Review of Financial Analysis* **91,2024**, 55 (Feb. 2023).

[3] Chu, Y., Lee, J., Kim, S., Kim, H., Yoon, Y., and Chung, H., "Review of Offline Payment Function of CBDC Considering Security Requirements," *Applied Sciences* **12**, 4488 (Apr. 2022).

[4] Furtado, F. R., Costa, A., and Guimarães, C., "An Architecture Proposal to Provide Interoperability Between DLT Platforms and Legacy Systems in the Financial Ecosystem," 13 (2023).

[5] BANK FOR INTERNATIONAL SETTLEMENT, "High-level technical requirements for a functional central bank digital currency (CBDC) architecture," 20 (Dec. 2023).

[6] Valdez, D., Pickett, A. C., and Goodson, P., "Topic Modeling: Latent Semantic Analysis for the Social Sciences," *Social Science Quarterly* **99**, 1665–1679 (Nov. 2018).

[7] "Web Scraping Tool & Free Web Crawlers | Octoparse." https://www.octoparse.com/ (Mar. 2024).

[8] Mcauliffe, J. and Blei, D., "Supervised Topic Models," in [*Advances in Neural Information Processing Systems*], **20**, Curran Associates, Inc. (2007).

[9] "How InfraNodus Works: AI Text Network Analysis - InfraNodus.Com." https://infranodus.com/about/how-it-works (2024).

[10] Mancini-Griffoli, T., Peria, M. S. M., Agur, I., Ari, A., Kiff, J., Popescu, A., and Rochon, C., [*Casting Light on Central Bank Digital Currency*], 307–340, Oxford University Press (Oct. 2019).

[11] Auer, R., Cornelli, G., and Frost, J., "Central bank digital currencies: Drivers, approaches, and technologies,"

[12] Bech, M. L. and Garratt, R., "Central Bank Cryptocurrencies," (Sept. 2017).
[13] Sveriges Riksbank, "E-krona pilot phase 1," (2021).
[14] Maryaningsih, N., Nazara, S., Kacaribu, F. N., and Juhro, S. M., "CENTRAL BANK DIGITAL CURRENCY: WHAT FACTORS DETERMINE ITS ADOPTION?," *Buletin Ekonomi Moneter dan Perbankan* **25**, 1–24 (June 2022).
[15] Auer, R. and Boehme, R., "The technology of retail central bank digital currency," (2020).
[16] Calle, G. and Eidan, D., "Central Bank Digital Currency: An innovation in payments," (2020).
[17] Androulaki, E., Brandenburger, M., Caro, A. D., Elkhiyaoui, K., Filios, A., Funaro, L., Manevich, Y., Natarajan, S., and Sethi, M., "A Framework for Resilient, Transparent, High-throughput, Privacy-Enabled Central Bank Digital Currencies," (2023).
[18] Stefan, H., "GD Filia Whitepaper,"
[19] Corda, "The Internet of Value," tech. rep. (2020).
[20] soramitsu, "The National Bank of Cambodia boosts financial inclusion with Hyperledger Iroha." https://www.hyperledger.org/case-studies/soramitsu-case-study (2019).
[21] Duan, S., Zhang, H., Sui, X., Huang, B., Mu, C., Di, G., and Wang, X., "Dashing and Star: Byzantine Fault Tolerance with Weak Certificates," (2022).
[22] Platt, M. and McBurney, P., "Sybil in the Haystack: A Comprehensive Review of Blockchain Consensus Mechanisms in Search of Strong Sybil Attack Resistance," *Algorithms* **16**, 34 (Jan. 2023).
[23] Zhou, S., Li, K., Xiao, L., Cai, J., Liang, W., and Castiglione, A., "A Systematic Review of Consensus Mechanisms in Blockchain," *Mathematics* **11**, 2248 (May 2023).
[24] Du, Y., Wang, Z., Li, J., Shi, L., Jayakody, D. N. K., Chen, Q., Chen, W., and Han, Z., "Blockchain-Aided Edge Computing Market: Smart Contract and Consensus Mechanisms," *IEEE Transactions on Mobile Computing* **22**, 3193–3208 (June 2023).
[25] Sethaput, V. and Innet, S., "Blockchain application for central bank digital currencies (CBDC)," *Cluster Computing* **26**, 2183–2197 (Aug. 2023).
[26] Sasongko, D. T. and Yazid, S., "Integrated DLT and non-DLT system design for central bank digital currency," in [*Proceedings of the 5th International Conference on Sustainable Information Engineering and Technology*], 171–176, ACM, Malang Indonesia (Nov. 2020).
[27] Bhawana and Kumar, S., "Permission Blockchain Network based Central Bank Digital Currency," in [*2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*], 1–6, IEEE, Kuala Lumpur, Malaysia (Sept. 2021).
[28] Kumhof, M., Pinchetti, M., Rungcharoenkitkul, P., and Sokol, A., "CBDC Policies in Open Economies," *SSRN Electronic Journal* (2023).
[29] Fullerton, E. J. and Morgan, P. J., "THE PEOPLE'S REPUBLIC OF CHINA'S," 35 (2022).
[30] Srividya, M., Sumedha, K., Shraddha, M., and Samhitha, V. V., "Credit Card Fraud Detection Using State-of-the- Art Machine Learning and Deep Learning Algorithms," **5**(3) (2023).
[31] Juffinger, J., Lamster, L., Kogler, A., Eichlseder, M., Lipp, M., and Gruss, D., "CSI:Rowhammer – Cryptographic Security and Integrity against Rowhammer," in [*2023 IEEE Symposium on Security and Privacy (SP)*], 1702–1718, IEEE, San Francisco, CA, USA (May 2023).
[32] Boar, C., Holden, H., and Wadsworth, A., "Impending arrival - a sequel to the survey on central bank digital currency," (Jan. 2020).
[33] Bindseil, U., "Tiered CBDC and the Financial System," *SSRN Electronic Journal* (2020).

A RevVIT-Based Discrimination Model for Concrete Crack Images

JiaWen Zhao*

City University of Macau, China

E-mail: D23091110059@cityu.edu.mo

Abstract

Cracks in concrete are a major hazard to the safety and durability of buildings. Efficient detection and timely repair of these cracks have become pressing issues in the field of civil engineering. Existing research suffers from shortcomings such as insufficient data, difficulties in feature extraction, and an imbalance between accuracy and computational cost, hindering the practical application of models. This paper presents an automated crack detection model based on the Revisiting Vision Transformer (RevVIT), with deep learning optimizations tailored to the complexity and diversity of crack images. Various data augmentation techniques were applied to preprocess raw images from highways, bridges, dams, and other structures to create a high-quality crack dataset. The RevVIT model was then introduced to address the inadequacies of traditional models in feature extraction under complex environments, and its performance was compared against VGG19 and three other baseline models on this dataset. Experimental results show that the RevVIT model achieved a classification accuracy of 99.03% in crack detection tasks, demonstrating high robustness and significantly outperforming existing methods, while also delivering superior efficiency in training and inference times.

Keywords: Concrete crack detection, Deep learning, Revisiting Vision Transformer, Image augmentation, Feature extraction

1. Introduction

Concrete is the most widely used material in civil engineering today, accounting for more than 90% of the three major materials in construction projects. Due to improper construction, environmental erosion, geological movements, increased loads, and other factors, concrete structures inevitably exhibit surface defects such as cracks, spalling, exposed reinforcement, leakage, and structural deformation. Among these defects, cracks are the most common and most harmful, as their presence severely impacts structural strength[1], leading to reduced building performance and shortened service life. Therefore, efficiently detecting these cracks and ensuring precise maintenance has become a critical issue in the field of construction engineering.

Traditional concrete crack detection relies on manual inspection or specialized equipment, which is not only inefficient and costly but also poses safety risks. In recent years, with the rise of deep learning (DL), accident analysis, and intelligent detection in various civil engineering fields[2][3], researchers have turned towards developing automated and intelligent crack detection programs, yielding significant results. However, there are still limitations in the current research on crack recognition using deep learning algorithms. First, extracting image features remains a challenge. Due to the high variability of crack characteristics in real-world scenarios：being small, with irregular edges, and subject to noise interference and lighting changes：effective feature extraction is difficult. Second, the lack of data resources limits

Third International Conference on Communications, Information System, and Data Science (CISDS 2024),
edited by Cheng Siong Chin, Shensheng Tang, Daniele Giusto, Yonghui Li, Proc. of SPIE
Vol. 13519, 135190J · © 2025 SPIE · 0277-786X · doi: 10.1117/12.3058011

the model's ability to adapt to complex real-world environments. Third, there is a lack of balance between model performance and efficiency. Most current research focuses on improving model performance, which has led to increased model size and significant computational resource demands, making it difficult to deploy models on mobile devices, thus hindering practical applications.

Against this backdrop, this is essential for improving infrastructure maintenance and ensuring public safety. Based on a large-scale dataset, this paper builds and trains a crack recognition classification model using the Revisiting Vision Transformer (REVVIT), achieving accurate and efficient crack image recognition while balancing both model accuracy and efficiency.

2. Literature Review

Methods for detecting bridge cracks can be categorized into four types: manual inspection, machine-based inspection, digital image processing-based inspection, and deep learning-based inspection. In recent years, with the increase in data and computational power, deep learning has demonstrated performance far exceeding that of machine learning models[4][5] , offering new solutions for automatic crack detection.

Since the breakthrough success of AlexNet[6] in the ImageNet competition in 2012, Convolutional Neural Networks (CNNs) have quickly become a core technology in image recognition[7]. AlexNet significantly improved image classification accuracy and computational efficiency through its deep convolutional layer structure and the introduction of the ReLU activation function, laying the foundation for subsequent models. In 2016, Zhang et al. used deep convolutional neural networks for road crack detection, demonstrating the robustness and efficiency of CNNs in handling large-scale traffic infrastructure images[8].

As deep learning technology advances, the limitations of CNNs in capturing global image information have become apparent. To address this issue, attention mechanisms[9], which have emerged in natural language processing, along with their variants such as Squeeze-and-Excitation Networks (SE)[10] and Efficient Channel Attention Modules (ECA)[11] \, have gradually been introduced into this field. Liu et al. (2021) introduced the Swin Transformer into crack detection and proposed the Swin-Unet model. This model combines the global feature extraction capability of the Swin Transformer with the local feature fusion capability of U-Net, demonstrating outstanding precision and robustness in crack detection tasks[12]. Similarly, Yang et al. optimized the attention mechanism based on Swin-Unet, implementing an improved segmentation algorithm to achieve accurate labeling and segmentation of dam cracks, effectively enhancing detection accuracy and recall[13]. Xia et al. combined attention mechanisms with deep feature optimization to propose a new method for detecting concrete pavement cracks. This method significantly improved the model's robustness and precision in complex backgrounds through multi-level feature extraction and attention mechanism optimization[14]. Furthermore, the introduction of attention mechanisms not only enhances detection precision but also provides new ideas for model lightweight design. Xu (2021) proposed a lightweight concrete crack detection method based on a multi-dimensional attention module, achieving efficient crack detection in resource-constrained environments by integrating multi-dimensional attention mechanisms into a lightweight network structure[15].

Considering that the performance improvement of single models often comes with increased computational overhead,

researchers have gradually explored multi-model fusion methods to balance detection accuracy and computational efficiency, and to enhance model robustness and adaptability. Piyathilaka et al. introduced YOLACT instance segmentation technology combined with traditional image processing methods to develop a real-time concrete crack detection and instance segmentation model. This model effectively leverages the advantages of deep learning models in feature extraction and the simplicity and efficiency of traditional methods, thus achieving a balance between real-time performance and high precision[16] .

Despite the significant progress made with deep learning technology in crack detection, there are still four key areas of limitation. Firstly, the scarcity of data resources remains a significant factor limiting model generalization ability. Most current research relies on private datasets, which are often small in scale and lack diversity, making it difficult to cover various complex scenarios in practical applications[17] .Secondly, deficiencies in feature extraction remain a major issue when models handle complex scenes[18]. Concrete cracks typically have fine, irregular edges and are easily affected by noise and lighting changes. These factors make models prone to false positives or missed detections in practical applications[19] .Lastly, the demand for model lightweight design is increasingly urgent[20] . With the development of edge computing and Internet of Things (IoT) technologies[21], crack detection models need to operate efficiently on devices with limited computational resources. However, most current research still focuses on improving model accuracy while relatively neglecting the model's computational complexity and size.

3. Crack Recognition Algorithm

Revision Transformer model, introduced in 2023[22], shows superior performance in feature extraction and handling complex image structures due to its unique multi-path parallel architecture and self-distillation technique, which significantly improves the model’s performance, adaptability, and robustness. This makes it particularly well-suited for detecting complex crack images.

Instead of the conventional approach with Multi-Head Self-Attention (MHSA) and Feed-Forward Networks (FFN) in series, this model transforms them into multiple parallel pathways. These paths are optimized using techniques like path pruning and weighted adjustments (EnsembleScale) to reduce redundant computations while boosting performance.These parallel paths function as distinct sub-networks, each employing unique feature extraction strategies that enhance the model’s overall capability. Additionally, the model applies knowledge distillation, transferring insights from longer paths to shorter ones, thus improving the feature representation quality of the shorter paths.

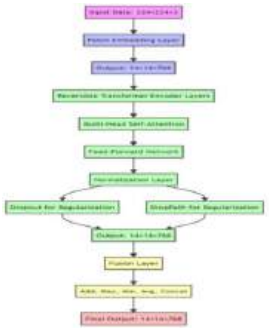


Figure 1. Structure diagram of the backbones

Note:In the model used in this study (see Figure 1), the input data size is 224×224×3. The data first passes through the Patch Embedding layer, which divides the image into 16×16 patches, with each patch embedded into a 768-dimensional vector space. The output size of the Patch Embedding layer is 14×14×768 (i.e., 196 embedded patches, each with 768 dimensions). In the Reversible Transformer Encoder Layers, the data size remains 14×14×768. The output from the encoder layers is passed through the final Fusion Layer, yielding a final output with the same size (14×14×768). These features are then fed into the model's head (with weights of 2×1536) for further classification tasks.

Crack detection is framed as a binary classification problem, where cross-entropy is frequently utilized as the loss function.To mitigate the effects of data imbalance, this paper implements a weighted cross-entropy loss function, as defined below:

$$L_{\mathrm{WCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\beta y_i \log \hat{y}_i + (1-\beta)(1-y_i)\log\left(1-\hat{y}_i\right)\right]$$

Where N represents the number of pixels, y_i represents the label of the i-th pixel, $\hat{y}_i$ represents the predicted result of the i-th pixel, $\beta = |G^-|/(|G^+| + |G^-|)$ and $|G^+|$ and $|G^-|$ represent the number of crack and non-crack pixels, respectively.

4. Experimental Design

The crack detection image dataset used in this study was sourced from monitoring points on various infrastructure types, including roads, bridges, dams, and building walls.To ensure the accuracy and reliability of the data, all images were inspected and verified by an expert team. Blurry or unclear images were removed to maintain high-quality standards. The final dataset comprises 6,058 non-crack images and 5,958 cracked images, creating a relatively balanced dataset that provides sufficient samples for model training and testing.

To simulate the various situations encountered in real-world applications during model training, data augmentation techniques were applied to the training data. Specifically, this study employed five methods: Random Resized Crop[23], Random Horizontal Flip[24], Color Jitter, Gaussian Blur[25], and Colorspace Conversion[26]. Additionally, combinations of these methods (i.e., randomly applying a mix of these five techniques) were used to process the existing dataset, generating more training samples.Figure 2 illustrates the effects of the aforementioned image augmentation techniques.The final database includes the 18,172 non-crack images and 17,874 cracked images.

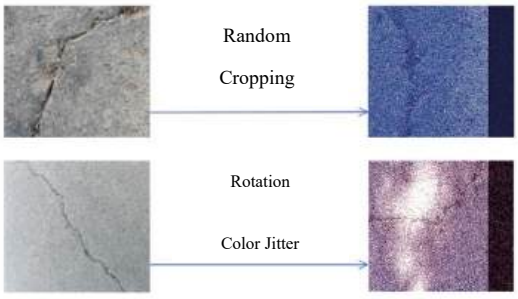


Figure 2. Image Augmentation Effects

VGG[27] (Visual Geometry Group Network) was introduced in 2014 by the Visual Geometry Group from the University of Oxford. VGG19, in particular, enhances image classification accuracy by increasing network depth using small 3×3 filters.While preserving the receptive field, these smaller kernels reduce the model's parameter count and computational load. VGG19 performed exceptionally well in the ILSVRC-2014 competition, securing second place in both the classification and localization tasks.

ResNet[28] (Residual Neural Network) made its debut at the ILSVRC-2015 competition.The key innovation of ResNet lies in addressing the degradation problem in deep neural networks, where increasing the network depth leads to higher training errors. ResNet tackles this issue through Residual Learning, which employs skip connections (also called shortcut connections) to bypass certain layers, allowing input data to flow directly to deeper layers. This forms the basis of residual blocks, the fundamental building blocks of ResNet. In this paper, we utilize the ResNet-18 variant.

ConvMixer is a novel convolutional neural network architecture proposed in 2021 by a team from the University of Tennessee and Microsoft Research[29].The main innovation of ConvMixer lies in its modular design: each ConvMixer block consists of a depthwise convolution layer and a pointwise convolution layer. This design effectively decouples the learning of spatial features from channel features.The depthwise convolution layer focuses on capturing spatial information, while the pointwise convolution layer handles the fusion of features across channels.

When evaluating classifier performance, multiple metrics are commonly used.Accuracy measures the proportion of correctly classified samples, with higher values indicating better performance. Precision reflects the proportion of true positives among the predicted positive samples, while recall assesses the proportion of actual positive samples correctly identified. Precision and recall often trade off against each other, and the F-Score (which becomes the F1-Score when $\beta=1$) balances both, providing a comprehensive evaluation.The ROC curve assesses model performance by plotting the true positive rate against the false positive rate. AUC is unaffected by class imbalance[30], making it suitable for cross-task comparisons, with an AUC of 0.5 indicating that the model performs no better than random guessing.Additionally, training time reflects the computational efficiency of the model, while testing time measures inference speed, which is critical for real-time applications. Together, these metrics offer a thorough comparison of models.

After applying image augmentation, the dataset comprises a total of 36, 046 images. The data is split into training, validation, and test sets in a ratio of 8:1:1. The input image size is set to 224×224 pixels. The hyperparameters for model training are configured as follows: the batch size is set to 32, and the optimizer used is the Cosine Annealing Learning Rate (LR) Scheduler. The parameters for the scheduler include T_max=50 and eta_min=1e-6. The initial learning rate is set to 0.001, with a minimum learning rate of 0.00001, and the number of training epochs is set to 100.

5.Experimental Results

In this evaluation, the RevVIT model demonstrated outstanding performance in terms of classification accuracy. It achieved a high accuracy rate, precision, recall, and F1 score, all reaching 99.03%, with an AUC value of 0.998. These metrics indicate that the RevVIT model is highly accurate and stable, capable of correctly classifying input images in the vast majority of cases. In comparison, other models such as VGG19, ResNet, and ConvMixer performed less impressively.Even the highest-performing model among them, ConvMixer, did not come close to RevVIT in any of the key metrics.

Tab1 Model performance and efficiency comparison

Model	Accuracy	Precision	Recall	F1	AUC	train time(min)	test time(min)
VGG19	93.31	92.67	91.88	92.27	0.924	988.6	15.2
ResNet	93.28	91.22	92.04	91.83	0.918	497.1	9.6
ConvMixer	93.44	92.98	91.74	92.36	0.916	489.5	8.4
RevVIT	99.03	99.03	99.03	99.03	0.998	432.3	8.8

In terms of model efficiency, RevVIT's training time was 432.3 minutes, and its testing time was 8.8 minutes. Although not the shortest, considering its exceptional classification performance, this level of efficiency is still noteworthy. By contrast, VGG19 and ResNet required longer training and testing times, while MobileNetV3, though achieving the best training and testing times, lagged behind in terms of accuracy and other performance indicators.

Considering both classification accuracy and model efficiency, the RevVIT model achieved the best balance, emerging as the most optimal model in terms of overall performance. However, in scenarios where rapid deployment and fast response are critical, MobileNetV3, with its superior training and testing speed, would be recommended, especially since it still maintains a relatively high level of classification accuracy.

6.Results and Discussion

Compared to traditional deep learning models, RevVIT demonstrates significant advantages across multiple metrics. On the dataset, this model achieved an accuracy rate of 99.03%, with precision, recall, and F1 score all exceeding 99%. Additionally, its ROC curve's Area Under the Curve (AUC) reached 0.998, indicating exceptional robustness and accuracy in classification tasks. Moreover, the RevVIT model also shows high efficiency in training and inference times, at 432.3 minutes and 8.8 minutes, respectively, outperforming traditional models like VGG19 and ResNet in balancing performance.

There are two main innovations in this work. One side, it introduces the RevVIT model for concrete crack detection for the first time, leveraging its multi-path parallel structure and self-distillation mechanism to address the issue of inadequate handling of crack edge details by traditional CNNs. This innovative design significantly enhances the model’s robustness and detection accuracy in complex backgrounds, particularly excelling in capturing key features when cracks are blurred or have irregular boundaries. On the other hand， the study proposes an efficient data augmentation strategy that combines methods such as random cropping, horizontal flipping, and color jittering, greatly expanding the dataset’s diversity and enhancing the model’s generalization capability in various real-world scenarios.

Reference

[1]Jin, W. L., & Zhao, Y. X. (2002). Review and prospects of research on the durability of concrete structures. *Journal of Zhejiang University (Engineering Science)*, *36*(4), 371-380.

[2] Li, X.-R., Ban, X.-J., Yuan, Z.-L., et al. (2022). Temporal prediction methods and applications based on deep learning in industrial scenarios. *Journal of Engineering Sciences, 44*(4), 757-766.

[3] Liu, T., Zhang, S.-R., Wang, C., et al. (2022). BERT-BiLSTM hybrid model for intelligent analysis of hydraulic construction accident texts. *Journal of Hydropower Engineering, 41*(7), 1-12.

[4] Li S, Zhao X. Convolutional neural networks-based crack detection for real concrete surface[C]//Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018. SPIE, 2018, 10598: 955-961.

[5] Lee D, Kim J, Lee D. Robust concrete crack detection using deep learning-based semantic segmentation[J]. International Journal of Aeronautical and Space Sciences, 2019, 20: 287-299.

[6] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[7] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[8] Zhang L, Yang F, Zhang Y D, et al. Road crack detection using deep convolutional neural network[C]//2016 IEEE international conference on image processing (ICIP). IEEE, 2016: 3708-3712.

[9] Vaswani A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.

[10]] Hou Y, Liu S, Cao D, et al. A deep learning method for pavement crack identification based on limited field images[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(11): 22156-22165.

[11] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11534-11542.

[12] Liu, Z., Lin, Y., & Cao, Y. (2021). Application of hierarchical self-attention mechanism based on Swin Transformer in crack detection. *Journal of Image and Graphics*, 26(12), 1563-1575.

[13] Yang, H.-L., Zhang, W., & Li, Q. (2024). Research on automatic annotation and segmentation method for dam cracks based on Swin-Unet. *Computer Applications and Research*, 41(03), 789-796.

[14] Xia, S.-F., Li, Z.-J., & Wang, Q. (2024). Detection of concrete pavement cracks based on attention mechanism and deep feature optimization. *Road, Bridge, and Architectural Engineering*, 38(02), 205-213.

[15] Xu, H.-J., Li, X.-L., & Zhang, Y. (2024). Lightweight concrete crack detection method based on multidimensional attention module. *China Journal of Highway and Transport*, 37(04), 287-296.

[16] Piyathilaka L, Preethichandra D M G, Izhar U, et al. Real-time concrete crack detection and instance segmentation using deep transfer learning[J]. Engineering Proceedings, 2020, 2(1): 91.

[17] Zhang K, Zhang Y, Cheng H D. CrackGAN: Pavement crack detection using partially accurate ground truths based on generative adversarial learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(2): 1306-1319.

[18] Long, T. (2023). Intelligent identification of concrete surface cracks based on deep learning and image processing technology. *Advances in Applied Mathematics*, 12, 1130.

[19] Li, Y., Yang, H.-J., Liu, H., et al. (2021). Research on concrete crack detection based on deep learning. *Information Technology and Informatization*, 12, 233.

[20] Zhang K, Zhang Y, Cheng H D. CrackGAN: Pavement crack detection using partially accurate ground truths based on generative

adversarial learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(2): 1306-1319.

[21] Hassan N, Gillani S, Ahmed E, et al. The role of edge computing in internet of things[J]. IEEE communications magazine, 2018, 56(11): 110-115.

[22] Chang S, Wang P, Luo H, et al. Revisiting vision transformer from the view of path ensemble[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 19889-19899.

[23] Takahashi R, Matsubara T, Uehara K. Ricap: Random image cropping and patching data augmentation for deep cnns[C]//Asian conference on machine learning. PMLR, 2018: 786-798.

[24] Bozorgzad A. Consistent distribution of air voids and asphalt and random orientation of aggregates by flipping specimens during gyratory compaction process[J]. Construction and Building Materials, 2017, 132: 376-382.

[25] Flusser J, Farokhi S, Höschl C, et al. Recognition of images degraded by Gaussian blur[J]. IEEE transactions on Image Processing, 2015, 25(2): 790-806.

[26] Lee D J, Archibald J K, Chang Y C, et al. Robust color space conversion and color distribution analysis techniques for date maturity evaluation[J]. Journal of Food Engineering, 2008, 88(3): 364-372.

[27] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[28] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385.

[29] Trockman, A., & Kolter, J. Z. (2021). Patches Are All You Need? In Proceedings of the 9th International Conference on Learning Representations (ICLR).

[30] Carrington A M, Manuel D G, Fieguth P W, et al. Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(1): 329-341.