Publisher: State and Provincial Joint Engineering Lab. of Advanced Network Monitoring and Control (ANMC)

Cooperate:

Xi'an Technological University (CHINA)
West Virginia University (USA)
Huddersfield University of UK (UK)
Missouri Western State University (USA)
James Cook University of Australia
National University of Singapore (Singapore)

Approval:

Library of Congress of the United States

Shaanxi provincial Bureau of press, Publication, Radio and Television

Address:

4525 Downs Drive, St. Joseph, MO64507, USA

No. 2 XueFu Road, WeiYang District, Xi’an, 710021, China

Telephone: +1-816-2715618 (USA) +86-29-86173290 (CHINA)

Website: www.ijanmc.org

E-mail: ijanmc@ijanmc.org

xxwlcn@163.com

ISSN: 2470-8038

Print No. (China): 61-94101

Publication Date: March 28, 2024

## Editor in Chief

Ph.D. Xiangmo Zhao
Prof. and President of Xi'an Technological University, Xi'an, China
Director of 111 Project on Information of Vehicle-Infrastructure Sensing and ITS, China

## Associate Editor-in-Chief

Professor Xiang Wei
Electronic Systems and Internet of Things Engineering
College of Science and Engineering
James Cook University, Australia

Dr. Chance M. Glenn, Sr.
Professor and Dean
College of Engineering, Technology, and Physical Sciences
Alabama A&M University
4900 Meridian Street North Normal, Alabama 35762, USA

Professor Zhijie Xu
University of Huddersfield, UK
Queensgate Huddersfield HD1 3DH, UK

Professor Jianguo Wang
Vice Director and Dean
State and Provincial Joint Engineering Lab. of Advanced Network and Monitoring Control, China
School of Computer Science and Engineering, Xi'an Technological University, Xi'an, China

Ph. D Natalia Bogach
Director of Computer Science Department

Peter the Great St. Petersburg Polytechnic University, Russia

## Administrator

Dr. & Prof. George Yang
Department of Engineering Technology
Missouri Western State University, St. Joseph, MO 64507, USA

Professor Zhongsheng Wang
Xi'an Technological University, China
State and Provincial Joint Engineering Lab. of Advanced Network and Monitoring Control, China

## Associate Editors

Prof. Yuri Shebzukhov
International Relations Department, Belarusian State University of Transport, Republic of Belarus.

Dr. & Prof. Changyuan Yu
Dept. of Electrical and Computer Engineering, National Univ. of Singapore (NUS)

Dr. Omar Zia
Professor and Director of Graduate Program
Department of Electrical and Computer Engineering Technology
Southern Polytechnic State University
Marietta, Ga 30060, USA

Dr. Baolong Liu
School of Computer Science and Engineering
Xi'an Technological University, CHINA

Dr. Mei Li
China university of Geosciences (Beijing)
29 Xueyuan Road, Haidian, Beijing 100083, P. R. China

Dr. Ahmed Nabih Zaki Rashed
Professor, Electronics and Electrical Engineering
Menoufia University, Egypt

Dr. Rungun R Nathan
Assistant Professor in the Division of Engineering, Business and Computing
Penn State University - Berks, Reading, PA 19610, USA

Dr. Taohong Zhang
School of Computer & Communication Engineering
University of Science and Technology Beijing, China

Dr. Haifa El-Sadi
Assistant professor
Mechanical Engineering and Technology
Wentworth Institute of Technology, Boston, MA, USA

Huaping Yu
College of Computer Science
Yangtze University, Jingzhou, Hubei, China

Ph. D Yubian Wang
Department of Railway Transportation Control
Belarusian State University of Transport, Republic of Belarus

Prof. Mansheng Xiao
School of Computer Science
Hunan University of Technology, Zhuzhou, Hunan, China

Prof. Ying Cuan
School of Computer Science, Xi'an Shiyou University, China

Qichuan Tian
School of Electric & Information Engineering
Beijing University of Civil Engineering & Architecture, Beijing, China

Ph. D MU JING
Xi'an Technological University, China

## Language Editor

Professor Gailin Liu
Xi'an Technological University, China

Dr. H.Y. Huang
Assistant Professor
Department of Foreign Language, the United States Military Academy, West Point, NY 10996, USA
Would you like to be an Associate Editor? Simply send a request together with your Curriculum Vitae to xxwlcn@163.com. We will have a team of existing editors or at least three experts in your field to review your request and make a decision as soon as we can. The criteria to be an associate editor are: 1. must have advanced degree; 2. must be a leader or have outstanding achievements in the specific research field; 3. must be recommended by the review team.

# Table of Contents

# A Target Recognition Method of Small Sample Based on RCS Data

Ruocheng Ma
Technology Center
Beijing Qihu Technology Co., Ltd.
Beijing, China
E-mail: mrtn@qq.com

Haoyang Liu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 502339341@qq.com

Jun Yu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: yujun@xatu.edu.cn

Zhiyi Hu
Engineering Design Institute
Army Research Loboratory
Beijing, China
E-mail: 18992899862@163.com

***Abstract*—During the training of target recognition models based on Radar Cross Section (RCS) data, a persistent challenge arises in sampling due to the inherent difficulty in acquiring a sufficient number of samples. This scarcity of data poses a significant impediment to the effective training of models, resulting in diminished accuracy in target recognition. To address this issue, this article proposes a target classification method based on RCS data under small sample conditions. The approach adopts the fundamental concept of Model-Agnostic Meta-Learning (MAML) to train the target recognition model, enhancing the structure of MAML model. An hourglass-shaped convolution layer is introduced to the input layer, with an additional convolution layer preceding the output layer, and a switch to a central loss function. To substantiate the efficacy of the improved MAML model, comprehensive comparative analyses are conducted with benchmark models, including MAML, ResNet 18-layers, Long Short-Term Memory (LSTM), among others. Experimental results conclusively demonstrate the superior performance of the refined MAML model in target recognition under conditions of limited samples, attaining an average prediction accuracy of 85.62%. This signifies a noteworthy 5-percentage-point improvement compared to the baseline model prior to the introduced enhancements.**

***Keywords-RCS Data; Small Sample; Target Recognition; MAML Model***

## I. INTRODUCTION

Radar systems detect targets by capturing the electromagnetic waves reflected from those targets. The Radar Cross Section (RCS) [1] quantifies a target's capacity to reflect radar signals in the direction of radar reception [2]. RCS data finds extensive utility in both military and civilian contexts, serving to evaluate and identify distant targets. In military applications, RCS data finds relevance in tasks such as military ship type identification and the recognition of air-launched decoys. In the civilian domain, RCS data proves valuable for anticipating islands or reefs and assessing fog severity, among various other applications.

Traditionally, the prevalent approach for target identification using RCS data involves extracting periodic features, size features, statistical features, and discrete wavelet energy features from RCS sequences. Subsequently, single-classifier algorithms like KNN, correlation matching, support vector machines, and random forests are employed for target identification. Alternatively, fusion algorithms amalgamate multiple single classifiers to create a more robust fusion classifier. Despite leveraging the distinct advantages of different classifiers, these methods often exhibit

DOI: 10.2478/ijanmc-2024-0001

limited discriminability and lower identification rates. The advent of neural network classifiers rooted in deep learning has significantly enhanced the accuracy of RCS target identification. Neural networks offer advantages such as distributed information storage, parallel computation, integrated storage and processing, rapid processing speed, robust fault tolerance, self-learning, self-organization, and adaptability. The typical paradigm involves constructing an RCS dataset for targets and training a target identification model using deep learning methods, thereby achieving recognition of target objects.

The effective deployment of deep learning for RCS-based target identification typically necessitates an extensive dataset for model learning and training. However, the practical acquisition of RCS data samples poses challenges, leading to a scarcity of samples that impedes the training of deep learning models and consequently results in diminished target identification accuracy. Although dataset expansion is a conventional strategy, its high associated costs render it impractical.

In addressing the constraint of limited RCS data samples [3], our research reveals that applying the concept of "meta-learning" can facilitate model training with small samples [4, 5, 6]. Conventional small sample learning algorithms often encounter issues where models become entrapped in local optima, consequently, necessitating a training method that considers the global context.

To overcome the limitations of existing methodologies, CHELSEA F et al. [7] introduced an enhancement to MAML algorithm by establishing a model initialization representation adaptable to multiple tasks. When confronted with a new task, only parameter fine-tuning is required to achieve satisfactory training results [8, 9, 10], minimizing the demand for a large sample size and addressing the small sample problem. However, the shallow layer structure of MAML's network model will impact recognition accuracy. Additionally, the model presents challenges such as high computational load and training instability during the training process.

To address these issues, this paper proposes enhancements to the network model of MAML algorithm. Experimental results substantiate that the refined MAML model significantly improves target recognition accuracy for RCS data in scenarios with limited samples.

## II. THE BASIC IDEA OF MAML ALGORITHM

MAML, as a model-agnostic meta-learning algorithm distinct from a deep learning model, functions as a method for training practical mathematical models. The primary objective is to cultivate models capable of transcending dependence on extensive data volumes, thereby facilitating swift adaptation to new tasks. MAML distinguishes itself by exhibiting notable proficiency in the context of novel tasks, owing to its provision of substantial prior knowledge.

### *A. The core idea of MAML*

MAML algorithm operates as a meta-learning framework, distinguished by its role as a facilitator for the training of mathematical models rather than constituting a deep learning model per se. Its primary objective is to cultivate models that can transcend reliance on extensive data volumes, demonstrating a capacity for rapid adaptation to novel tasks.

The foundational concept of MAML algorithm involves training a meta-learning model, denoted as M, on tasks of similar nature. Subsequently, through fine-tuning on a limited dataset specific to a new task, a distinct mathematical model, denoted as m, is derived—effectively adapted to the nuances of the novel task. The loss function, denoted as $L_{Ti}$ , in the context of MAML is articulated in Formula (1). In this expression, $\theta$ signifies the initial parameter of the network model, $\theta_i'$ represents the parameter acquired through learning for the i-th sub-task based on the initial parameter of the network model, and $L_{Ti}$ denotes the loss function characterizing the sub-task, parameterized by $\theta_i'$ .

$$L(\theta) = \sum_{i=1}^{N} L_{Ti}\left(\theta_i^{'}\right) \tag{1}$$

MAML diverges from the conventional approach of traditional pre-training. In the traditional paradigm, the same parameter $\theta$ undergoes updates across various sub-tasks, resulting in an initialization parameter $\theta$ optimized to minimize the cumulative losses across all sub-tasks. However, such an approach does not ensure the attainment of a global optimal solution for each individual sub-task. In contrast, MAML algorithm adapts the initialization parameter $\theta$ based on the parameter $\theta_i'$ associated with each sub-task. At this stage, the model shifts its focus from the losses of individual sub-tasks, prioritizing the maximization of its overall learning capability. With minimal training on new tasks, it exhibits a rapid convergence towards a global optimal solution.

### *B. The network model of MAML*

The fundamental architecture of the deep neural network model employed in MAML algorithm plays a pivotal role in the training process. As depicted in Figure 1, the network structure adheres to a specific configuration, encompassing a total of five layers distinguished by different colors. A notable characteristic of this structure is the composition of the initial four layers, which consist of convolutional layers and batch normalization layers. In contrast, the final layer is exclusively comprised of fully connected layers. This design choice results in a shallow network configuration, characterized by a reduced parameter count, expeditious convergence, and commendable fitting performance.

However, the inherent nature of shallow networks imposes certain constraints, notably in terms of their limited feature extraction capabilities and a deficiency in establishing correlations between data points. These limitations, intrinsic to shallow networks, consequently impact the recognition accuracy of MAML model. Addressing these constraints is imperative for enhancing the overall performance and efficacy of the algorithm.

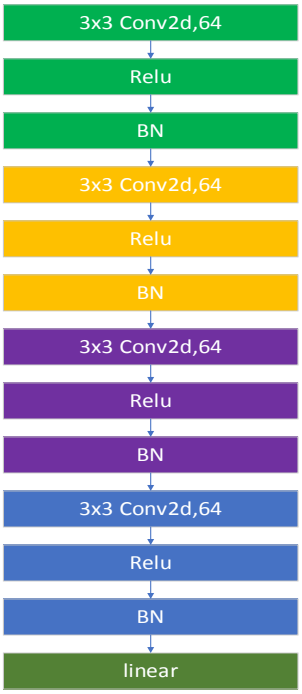



Figure 1. The general form of MAML model

## III. IMPROVEMENT OF MAML ALGORITHM AND IMPLEMENTATION STEPS

### *A. Improvements to the network model*

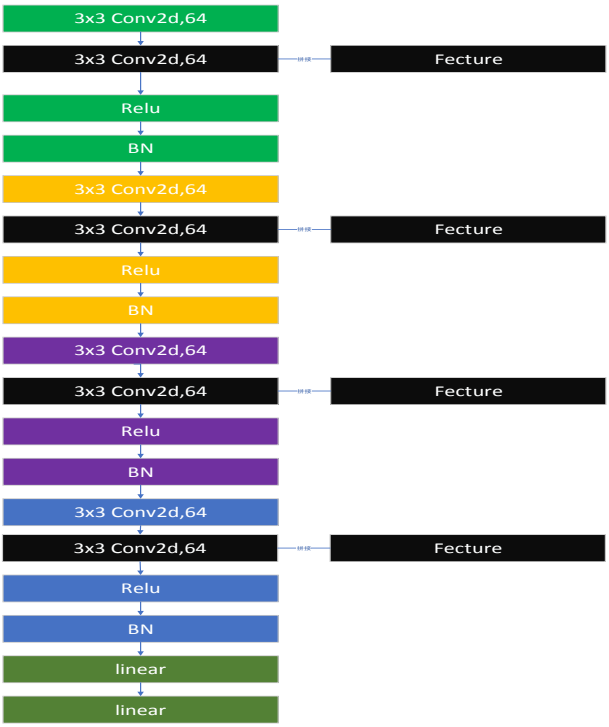



Figure 2. Improved MAML model

To enhance the precision of target recognition within MAML model, a series of structural refinements have been implemented, as illustrated in Figure 2. The key enhancements are outlined as follows:

*1)* Addition of hourglass-shaped convolutional layer in the input layer:

A distinctive hourglass-shaped convolutional layer has been incorporated into the input layer to augment the feature extraction capability specific to target data. This augmentation aims to capture more representative feature parameters, thereby enhancing the model's ability to discern critical patterns.

*2)* Additional convolutional layer in the layer preceding the output layer:

An supplementary convolutional layer has been introduced just before the output layer to fortify the inter-neuronal correlations, facilitating the network's descent along the global optimum gradient. This augmentation is designed to improve the model's ability to capture nuanced relationships and intricacies in the data.

*3)* Modification of loss function to central loss function:

The original loss function has been revamped to incorporate a central loss function, designed to gauge the proximity between instances belonging to the same class. This modification contributes significantly to elevating target recognition accuracy by emphasizing the inherent similarities within classes.

### *B. Implementation of improvement algorithm*

The refined MAML algorithm necessitates specific configurations within the training dataset. The pre-training dataset is meticulously organized on a task-by-task basis. To initiate training, the model requires a task distribution, and concurrently, two hyperparameters must be specified.

The enhancement algorithm is implemented through the following specific steps:

*1)* Randomly initialize the model parameters.

*2)* Set the number of epochs for a training round.

*3)* Sample multiple tasks to form a batch.

*4)* Calculate the loss $L_{Ti}$ on the support set of a task using Formula (2):

$$L_{Ti}(f_\theta) = \sum_{x^{(j)}, y^{(j)} \sim T_i} y^{(j)} \log f_\theta\left(x^{(j)}\right) + \left(1 - y^{(j)}\right) \log\left(1 - f_\theta\left(x^{(j)}\right)\right) \quad (2)$$

where $f_\theta$ represents the model, x is the input training sample, y is the label of the training sample.

*5)* Calculate the parameter $\theta_i'$ after a gradient update using Formula (3):

$$\theta_i' = \theta - \alpha \nabla_\theta L_{Ti}(f_\theta) \quad (3)$$

*6)* Iterate through Steps 4 to 5 until all tasks in the current batch are traversed, completing the first gradient update.

*7)* Upon acquiring parameters from the initial gradient update, a subsequent gradient update is executed via a procedure commonly referred to as "gradient by gradient." The gradients for the complete batch are computed by employing the query set from each task. Subsequently, these gradients are directly employed in modifying the original model through the application of Stochastic Gradient Descent (SGD), thereby updating the parameter $\theta$ in accordance with Formula (4):

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} L_{Ti}(f_{\theta_i'}) \quad (4)$$

*8)* Continue sampling the next batch and iterate through Steps 3 to 7 until all batches are traversed.

## IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to ascertain the efficacy of the improved MAML algorithm, a horizontal comparison was executed among four distinct models: the original MAML model, the improved MAML model denoted as MAML-New, ResNet 18-layers model, and Long Short-Term Memory (LSTM) model. Comprehensive analyses, encompassing both qualitative and quantitative

assessments, were conducted on the experimental results to validate the advancements introduced by MAML-New model.

The comprehensive procedural steps of the experimental investigation are outlined as follows:

*1)* Preparation of experimental datasets:

This phase involves the meticulous preparation of Radar Cross Section (RCS) data for four distinct models: MAML, MAML-New, ResNet 18-layers [11], and LSTM [12]. This encompasses the curation of both training and testing datasets specifically tailored for MAML model.

*2)* Model Training:

Initial preprocessing of the experimental dataset, encompassing critical tasks such as dataset classification, data filling, normalization, and data standardization.

The dataset is partitioned on a per-task basis, serving as input for the respective models.

Subsequently, the models undergo comprehensive training.

*3)* Experimental results and comparative analysis:

Execution of experiments to elicit results that reflect the models' performance.

Rigorous comparative analysis is conducted to assess and contrast the efficacy of MAML, MAML-New, ResNet 18-layers, and LSTM models.

### *A. Preparation of experimental data*

The raw RCS dataset was generated using the FEKO software simulation method [13]. To facilitate computation, the dataset was stratified into 12 categories, delineated by unique external features of the targets and labeled as category 1 to category 12. Notably, category 1 to category 4 constituted the experimental test dataset, while category 5 to category 12 comprised the pre-training dataset. Figure 3 and 4 provide simplified models representing the 12 different categories. The RCS data for these models were exported utilizing the FEKO software, with each category containing 200 models of varied sizes. As depicted in Figure 5 and 6, the cumulative incident angles for a single model resembled a hemisphere. Drawing an analogy to Earth's longitude and latitude, the longitude spanned from $0°$ to $360°$, and the latitude ranged from $0°$ to $90°$. Each degree of latitude corresponded to 360 incident angles, resulting in a model encompassing $90 \times 360$ incident angles. Each incident angle correlated with a specific RCS value, yielding RCS data for each model sized at $90 \times 360$. Consequently, the 12 categories contributed to a total of $12 \times 200$ RCS datasets. Figure 7 illustrates an instance of RCS data for a category 3 model in Cartesian coordinates with a size of 85 cm.

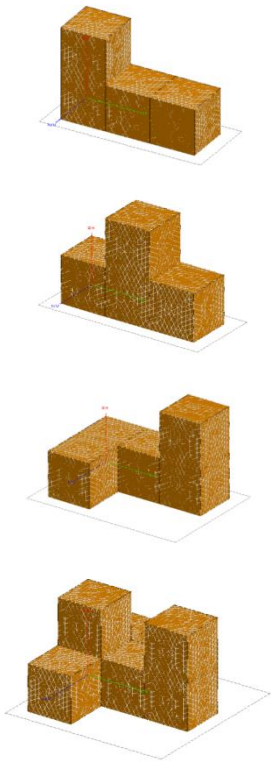


Figure 3. Category 1 - 4, the experimental test dataset

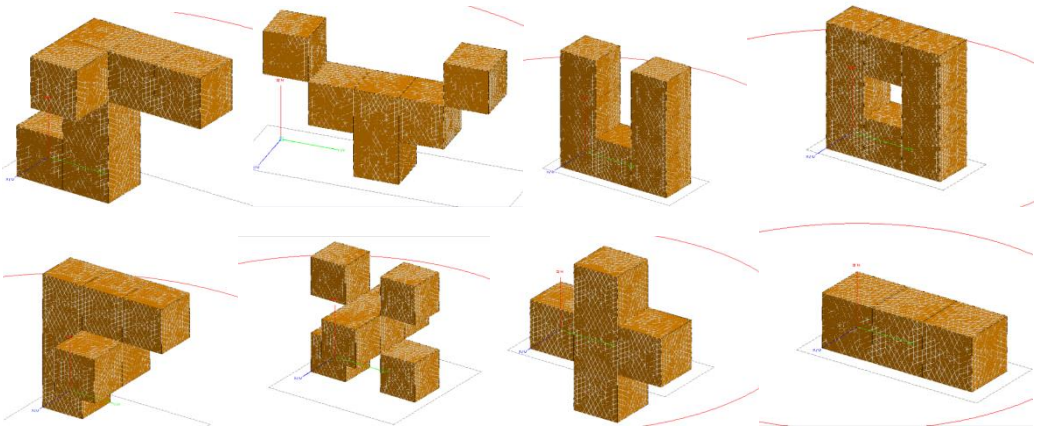


Figure 4. Category 5 - 12, the pre-training dataset

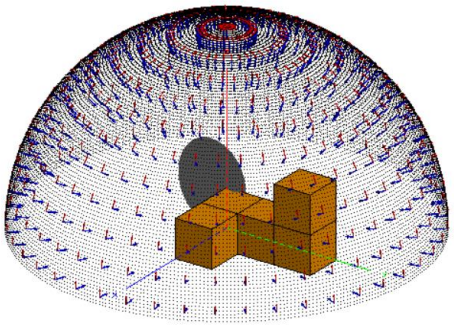


Figure 5. All incident angles of a model

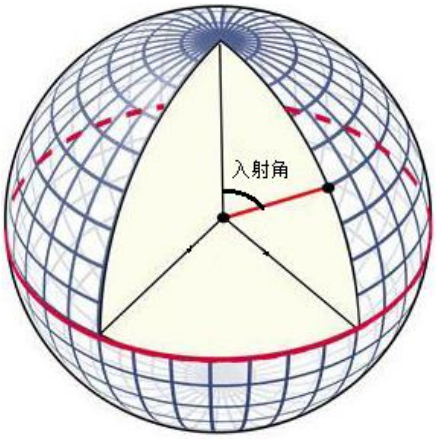



Figure 6. Schematic diagram of incidence angle

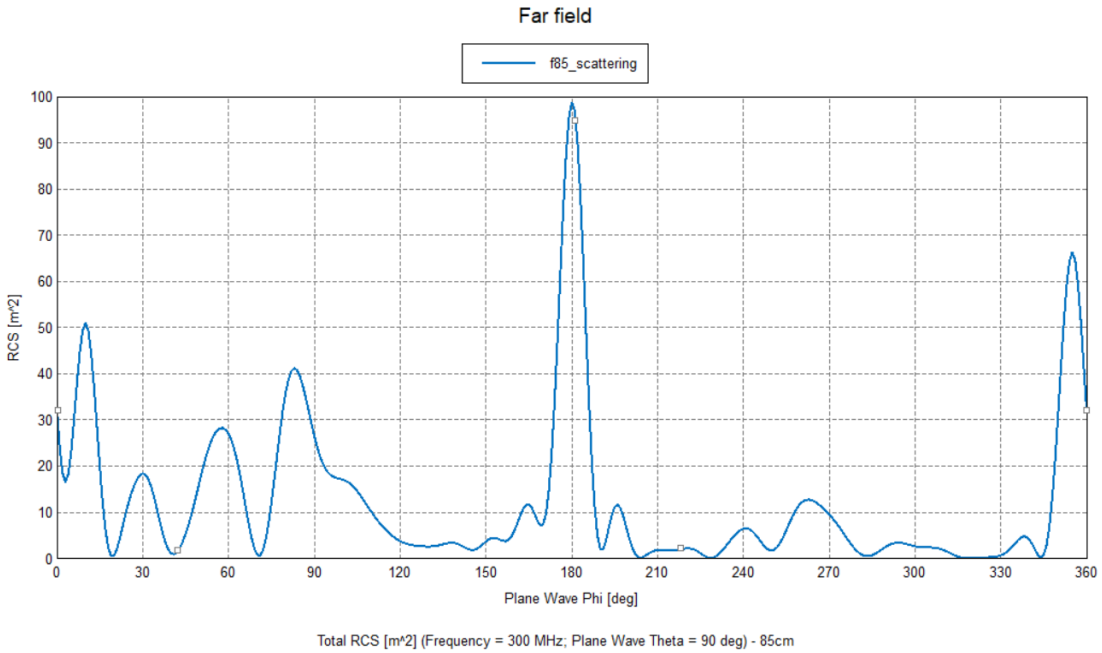



Figure 7. RCS data of 85CM category 3 model in Cartesian coordinate system

TABLE I. PARTIAL NETWORK PARAMETER VALUES FOR MAML AND MAML-NEW

| Parameter | Value | Meaning |
|---|---|---|
| epoch | 600 | Training epochs |
| k | 4 | Number of sample categories |
| k_spt | 20 | Number of support set samples |
| k_qry | 30 | Number of query set samples |
| imgsz | 180 | Dimension of input data |
| imgc | 1 | Number of channels for input data |
| task_num (batch_size) | 16 | Training batch of samples |
| meta_lr | 1e-3 | First gradient update learning rate |
| update_lr | 0.01 | Second gradient update learning rate |

### *B. Training of MAML model*

The training procedures for both MAML and MAML-New model can be delineated as follows:

*1)* Preliminary to the training process, employ preprocessing techniques such as data padding, normalization, and data standardization on the pre-training dataset $I$ and the test dataset $J$.

*2)* Define key parameters: n_way signifies the number of sample categories in each task, k_spt represents the number of support set samples, k_qry denotes the number of query set samples, and task_numtask_num stands for the number of training batches of samples. Randomly select n_way (n_way < 8) categories from the pre-training dataset $I$. For each category, randomly choose k_spt + k_qry (k_spt + k_qry ≤ 200) labeled samples, thereby constituting a task $T_i$ with n_way × (k_spt + k_qry) samples. From each category's k_spt + k_qry samples in the current task, designate k_spt samples as the support set $T_{is}$ and k_qry samples as the query set $T_{iq}$. Each task is tantamount to a data point in training. Randomly extract task_num such tasks to form a batch. Concurrently specify the hyperparameters meta_lr and update_lr, where meta_lr and update_lr denote the learning rates for the two-stage gradient iterations.

Table I enumerates certain network parameter values pertinent to training the model using MAML method.

*3)* Employ the same procedure as delineated in step (2) to partition the test dataset $J$ into tasks, selecting $J_s$ and $J_q$ as the support set and query set, respectively, for all tasks in the test dataset.

*4)* Following the steps outlined in MAML algorithm's section III.B, train the meta-learning model $M_{meta}$ using the pre-training dataset $I$.

*5)* Fine-tune the trained meta-learning model $M_{meta}$ on the support set $J_s$ of the test data, thereby obtaining the target recognition model $M$ adapted to the current task.

*6)* Input the query set $J_q$ into the well-trained target recognition model $M$ and ultimately obtain a prediction result $R$.

At this juncture, the training of MAML model concludes.

### *C. Experimental results and comparative analysis*

Figure 8 through 11 individually delineate the training progression of ResNet 18-layers, LSTM, MAML, and MAML-New models. These subfigures present their respective prediction accuracy and loss, with blue curves representing accuracy and yellow curves representing loss rates. The horizontal axis denotes training batches, while the vertical axis spans ratio values from 0 to 1.

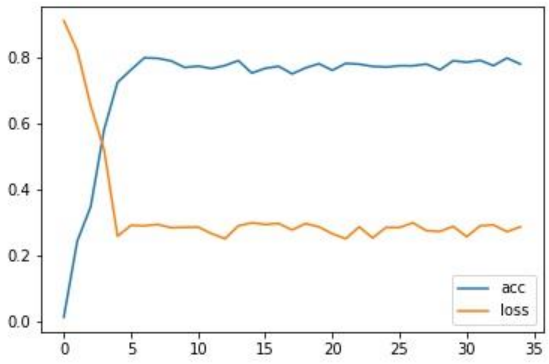



Figure 8. ResNet 18-layers

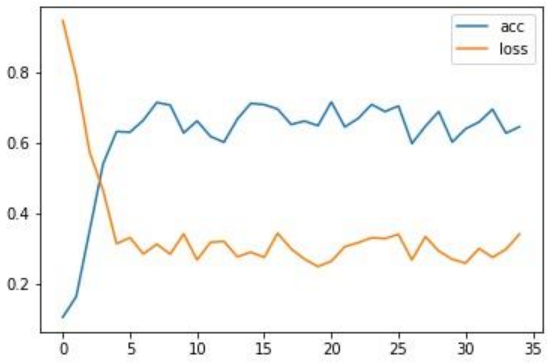



Figure 9. LSTM

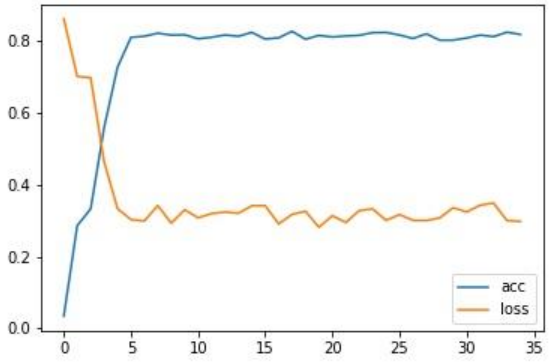



Figure 10. MAML

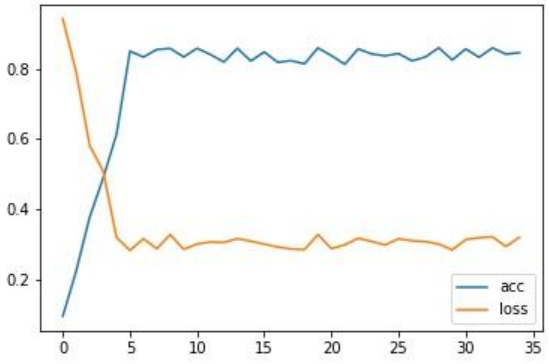



Figure 11. MAML-New

Certainly, the observations suggest that the accuracy of ResNet 18-layers and LSTM remains below 0.8, whereas the accuracy of both MAML and MAML-New exceeds 0.8. Additionally, MAML-New exhibits a slightly higher accuracy than MAML. Noteworthy is the observation that MAML-New experiences lower training losses in comparison to MAML.

TABLE II. COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT MODELS

| Accuracy / Model | Category 1 Accuracy | Category 2 Accuracy | Category 3 Accuracy | Category 4 Accuracy | Average accuracy |
|---|---|---|---|---|---|
| MAML | 82.16% | 72.45% | 81.3% | 85.97% | 80.47% |
| MAML-New | 86.42% | 79.70% | 87.17% | 89.19% | 85.62% |
| ResNet 18-layers | 81.7% | 62.1% | 82.4% | 90.1% | 73.45% |
| LSTM | 81.1% | 68.0% | 80.8% | 80.3% | 77.55% |

The recognition accuracy for the four models corresponding to the four categories in the test dataset, as illustrated in Figure 3, has been computed and is presented in Table II.

Table II reveals that the average recognition accuracy for MAML model, ResNet 18-layers model, and LSTM model is 80.47%, 73.45%, and 77.55%, respectively—substantially lower than the recognition accuracy achieved by MAML-New model at 85.62%. The recognition accuracy hierarchy, from highest to lowest, is as follows: MAML-New > MAML > ResNet 18-layers > LSTM. In scenarios with limited samples, MAML model demonstrates superior recognition capability compared to conventional deep neural networks such as ResNet 18-layers and LSTM models. Furthermore, MAML-New model exhibits an average improvement of 5-percentage-point in recognition accuracy over MAML model.

## V. CONCLUSIONS

To address the challenge of a small sample size in training a target recognition model based on RCS data, MAML algorithm was employed. Structural modifications to the network included the incorporation of an hourglass-shaped architecture and the addition of convolutional operations at the output layer. Simultaneously, adjustments were applied to the loss function, and experiments were systematically conducted on the RCS dataset. The resulting model effectively recognizes targets using RCS data, with empirical results indicating a notable improvement in recognition performance, particularly in scenarios characterized by a small sample size.

## REFERENCES

[1] W. Hu, X. Du, L. Zhang, et al, "Radar target recognition theory." National Defense Industry Press. Beijing, 2015.

[2] E. Wengrowski, M. Purri, K. Dana. and A. Huston, "Deep CNNs as a method to classify rotating objects based on monostatic RCS." IET Radar, vol. 13, Jul. 2019, pp. 1092-1100, doi: 10.1049/iet-rsn.2018.5453.

[3] K. Zhao, X. Jin, Y. Wang, "Review of small sample learning research." Journal of Software, vol. 32, Feb. 2021, pp. 349-369 , doi: 10.13328/j.cnki.jos.006138.

[4] J. Sun, S. Yu, and J. Sun, "Radar small sample target recognition method based on meta learning and its improvement." Systems Engineering & Electronics, vol. 44, Jun. 2022, pp. 1837-1845.

[5] H. Li, L. Wu, Y. Niu and C. Wang, "Research on meta learning method for UAV small sample target recognition." Unmanned system technology, vol. 2, Nov. 2019, pp. 17-22, doi: 10.19942/j.issn.2096-5915.2019.06.003.

[6] W. Cao, "Design of twin network for small sample underwater target recognition." Mechanical design, vol. 37, Dec. 2020, pp. 203-207. doi: 10.13841/j.cnki.jxsj.2020.s2.050.

[7] C. Finn, P. Abbeel and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks." International conference on machine learning, Jul. 2017, pp. 1126-1135.

[8] R . Puri, A. Zakhor, and R . Puri, "Few shot learning for point cloud data using model agnostic meta learning." 2020 IEEE International Conference on Image Processing (ICIP), IEEE, Oct. 2020, pp. 1906-1910, doi: 10.1109/ICIP40778.2020.9190819.

[9] Q. Zhong, L. Chen and Y. Qian, "Few-shot learning for remote sensing image retrieval with maml." 2020 IEEE International Conference on Image Processing (ICIP), IEEE, Oct. 2020, pp. 2446-2450, doi: 10.1109/ICIP40778.2020.9191042.

[10] D. Wang, Y. Cheng, M. Yu, X. Guo. and T. Zhang, "A hybrid approach with optimization-based and metric-based meta-learner for few-shot learning." Neurocomputing, vol. 349, Jul. 2019, pp. 202-211, doi: 10.1016/j.neucom.2019.03.085

[11] Y. Zhou, F. Ren, S. Nishide and X. Kang, 2019, " Facial sentiment classification based on resnet-18 model." 2019 International Conference on electronic engineering and informatics (EEI), IEEE, Nov. 2019, pp. 463-466, doi: 10.1109/EEI48997.2019.00106

[12] F. Zhang, C. Hu, Q. Yin, W. Li, H.C. Li and W. Hong, "Multi-aspect-aware bidirectional LSTM networks for synthetic aperture radar target recognition." Ieee Access, vol. 5, Nov. 2017, pp. 26880-26891, doi: 10.1109/ACCESS.2017.2773363.

[13] L. Zhou, C. Wang, "Accuracy and efficiency selection of FEKO simulation RCS." 2019 Altair Technology Conference, Altair engineering software (Shanghai) Co., Ltd., Jul. 2019, pp. 1358-1362, doi: 10.26914/c.cnkihy.2019.090153.

# Indoor Robot SLAM with Multi-Sensor Fusion

Jionglin He
School of Computer Science & Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1013598511@qq.com

Jiaxiang Fang
School of Computer Science & Engineering
Xi'an Technological University
Xi'an, China
E-mail: 2293950221@qq.com

Shuping Xu
School of Computer Science & Engineering
Xi'an Technological University
Xi'an, China
E-mail: 563937848@qq.com

Dingzhe Yang
School of Computer Science & Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1820522365@qq.com

***Abstract*—In order to solve the problem of large positioning error and incomplete mapping of SLAM based on two-dimensional lidar in indoor environment, a multi-sensor fusion SLAM algorithm for indoor robots was proposed. Aiming at the mismatch problem of the traditional ICP algorithm in the front end of the lidar SLAM, the algorithm adopts the PL-ICP algorithm that is more suitable for the indoor environment, and uses the extended Kalman filter to fuse the wheel odometer and IMU to provide the initial motion estimation value. Then, during the mapping phase, the pseudo 2D laser data converted from the 3D point cloud data obtained by the depth camera is fused with the data obtained from the 2D lidar to compensate for the lack of vertical field of view in the 2D lidar mapping. The final experimental results show that the fusion odometer data has improved the positioning accuracy by at least 33% compared to a single wheeled odometer, providing a higher initial iteration value for the PL-ICP algorithm. At the same time, fusion mapping compensates for the shortcomings of a single two-dimensional lidar mapping, and constructs an environmental map with more complete environmental information.**

***Keywords-SLAM; Indoor Robot; Extended Kalman Filter; PL-ICP***

## I. INTRODUCTION

With various types of indoor mobile robots being widely used in human life and production, robot Simultaneous Localization and Mapping (SLAM) technology, as the basis for robots to complete service work, has gradually become a research hotspot [1]. Affected by the diversity and complexity of the indoor environment, the information that can be obtained by the simultaneous localization and mapping of the common single radar is limited, so it cannot construct complete three-dimensional map information [2]. Therefore, multi-sensor fusion has become a new method to solve the defects of indoor robot SLAM. At present, the mainstream sensors applied to solve SLAM problem are lidar and vision camera, each of which has different advantages [3]. Among them, two-dimensional lidar has the advantages of small error and high reliability, but it has no vertical direction information and is easy to cause waste of resources. The depth camera has the advantages of a wide field of view and can identify specific objects, but the accuracy and stability of mapping are poor, and the probability of deviation is high [4-5]. At the moment, many scholars have proposed many fusions SLAM schemes based on the complementarity of perception sensors.

In terms of the fusion of vision and pose data Measurement sensors, literature [6] uses the coupling method to fuse the camera with the Inertial Measurement Unit (IMU) and the odometer, so as to obtain more accurate pose data. However, because the use is not tightly coupled, the vision sensor is prone to photosensitivity and instability. Literature [7] transfers the data

collected by the IMU as the initial pose to the vision sensor, which reduces the data pressure of the depth camera in the mapping to a certain extent, but it is prone to data disorder. If the data in the IMU unit is wrong, it will lead to problems in the vision part. In terms of wheeled odometer and laser sensor, literature [8] uses the Extended Kalman Filter (EKF) algorithm to allow the odometer to provide pose while fusing the point cloud data of lidar to make the obtained data more accurate and reliable. In literature [9], key frame data are identified by lidar and weighted and fused with odometer data by Iterative Closest Point (ICP) algorithm, so as to obtain more accurate pose information. However, these two algorithms will be too traditional due to the implementation method. It is easy to have problems such as low efficiency and poor data fusion effect. In the fusion of laser and attitude measurement sensor. Literature [10] tightly coupled the lidar and IMU fusion, which provided a higher frequency of real-time pose data for the robot, but there was no way to eliminate the invalid data resulting in data redundancy. Literature [11] tries to introduce odometer data to assist IMU for fusion, but there are still some errors. In the fusion of vision and laser sensor, literature [12] fuses laser and camera to realize loop closure detection and complete global map construction, but the stability is not good. Literature [13] uses the fusion of vision and laser based on static scenes to improve the accuracy of mapping, but it cannot achieve dynamic real-time performance. In literature [14], the overall fusion data of odometry, IMU, visual camera and lidar were analyzed, and the feasibility of multi-sensor fusion SLAM was verified, but there were problems such as accuracy decline and partial map missing.

Based on the above literature, it can be seen that SLAM with multi-sensor fusion can provide more accurate pose data, and can improve the accuracy of mapping and navigation. However, there are also many urgent problems to be solved, such as odometer slippage and map fusion defects. In this paper, a new fusion method is proposed, which uses two-dimensional lidar, depth camera, wheeled odometer and IMU unit to carry out multi-sensor fusion, and uses the improved Point-to-line ICP (PL-ICP) algorithm to intercept key frames. To make up for the shortcomings of the common ICP algorithm in data interception and fusion. EFK algorithm is directly applied to IMU to ensure the simplification and effectiveness of data acquisition. Finally, the Bayesian method is used to fuse the data of vision and laser sensors, and the fusion two-dimensional raster map is constructed to make up for the shortcomings of single sensor mapping. This paper aims to improve the accuracy of indoor robot pose information, establish a global raster map with more complete information elements and more complete data, and prove the effectiveness of the fusion in the actual environment.

## II. Inter-Frame Matching of Laser Point Cloud

The position and orientation information of the mobile robot can be calculated by wheeled odometry or inertial odometry, but the odometry information obtained by these two methods has large errors. In fact, the laser odometry calculation method uses multiple sub-images to complete the map. In fact, it is impossible to insert all the point clouds identified in each frame into the submap, so the laser odometry only selects the key frame data for insertion, and the unimportant frame data will be discarded. The data of the traditional wheel odometer is used as the initial iteration value for the laser odometer to solve the robot's position and orientation information.

In the laser inter-frame matching solution, the Iterative Closest Point (ICP) algorithm can obtain a good matching effect by iterating the initial value without point cloud segmentation and feature extraction, so ICP algorithm has become one of the most studied and most mature algorithms [15].

### *A. ICP Algorithm*

The basic principle of ICP algorithm is as follows:

Let the set of spatial coordinates of the two-point cloud frames of the laser be: Starting frame $X = \{x_1, x_2, ...x_i\}$ , $x_i = \begin{bmatrix} d_i \cos\theta_i \\ d_i \sin\theta_i \end{bmatrix}$ . Target

frame $P=\{p_1,p_2,...p_j\}$, $p_j=\begin{bmatrix} d_j\cos\theta_j \\ d_j\sin\theta_j \end{bmatrix}$. Where $d_i$, $\theta_i$ and $d_j$, $\theta_j$ are the distance and the corresponding angle of the environmental information acquired by the two frames of lidar, respectively. Then, by finding $k$ groups of corresponding points, the rotation $R$ and translation $t$ of the two laser frames can be solved. Finally, the error function $E(R,t)$ is constructed and iterated continuously to make the error function result meet the set threshold, and the optimal $R$ and $t$ can be obtained.

The error function is as follows.

$$E(R,t)=\frac{1}{k}\sum_{n=1}^{k}\| x_n-(Rp_n+t)\|^2 \tag{1}$$

Where $x_n$ and $p_n$ denote a certain set of corresponding points in $k$ groups, and then the mean value of the point cloud of the two frames is denoted by $u_x$ and $u_p$ respectively:

$$\begin{aligned} E(R,t)=\frac{1}{k}\sum_{n=1}^{k}(\| x_n-u_x-R(p_n-u_p)\|^2 \\ +\| u_x-Ru_p-t\|^2) \end{aligned} \tag{2}$$

For any $R$ in the right term above, a $t$ can be found to make the right term overall 0, so the left term above can be transformed into the maximum value of $Trace\sum_{n=1}^{k}Rp_n^{'}x_n^{'T}$, Where $x_i^{'}=x_i-u_x$ and $p_i^{'}=p_i-u_p$ are two point clouds respectively subtracting their respective point cloud geometric center to form a new point cloud. Decentralizing the point clouds is equivalent to performing a translation, which shortens the distance between two point clouds. The purpose is to approximately convert two point clouds that may be in different coordinate systems to the same coordinate system, and also to prevent local optima. The SVD decomposition yields the following.

$$W=\sum_{n=1}^{k}p_n^{'}x_n^{'T}=U\sum V^T \tag{3}$$

When W has full rank:

$$R=UV^T \tag{4}$$

$$t=u_x-Ru_p \tag{5}$$

In summary, each iteration of ICP algorithm will traverse every point in the origin set until it finds the closest point to the target point [16]. Therefore, the basic process of ICP algorithm is as follows: firstly, for the two laser point clouds that need to be matched, the nearest associated point of the point cloud is found. Usually, the initial corresponding point is obtained by using the data of the wheeled mileage meter. Then based on the corresponding points, $R$ and $t$ are further solved. After that, the point cloud is transformed, the matching error is calculated and whether the error meets the set threshold is judged. If it does, the $R$ and $t$ solved are output, if not, the iteration continues until the error meets the set threshold.

### *B. PL-ICP Algorithm*

In the ICP algorithm in the previous section, in the process of finding corresponding points, the point with the closest Euclidean distance is considered to be the corresponding point of the point cloud. However, in the actual indoor environment, the distance between the laser point and the actual environmental surface is the best error scale, so the standard ICP algorithm will cause a certain number of wrong corresponding points [17]. To solve this problem, many improved versions of ICP algorithm have been derived. The iterative closest point from point to line improves the error equation of the standard ICP algorithm, and uses the distance between the laser point and the line of the nearest two points of the laser point cloud in the next frame to approximate the real scale relationship, which is more suitable for indoor scenes [18].

Therefore, the error equation of PL-ICP algorithm is as follows.

$$J\left(R_{k+1},t_{k+1}\right)= \sum_{i}\left(n_i^T\left[R_{k+1}p_i+t_{k+1}-p_{j_1^i}\right]\right)^2 \quad (6)$$

Where $p_i$ represents the $i-th$ sampling point, $p_{j_1^i}$ is the nearest matching point of the sampling point under the target point cloud, $n_i$ is the normal vector of the two nearest matching points, $R_{k+1}$, $t_{k+1}$ represent the transformation parameters, and the optimal transformation parameters can be obtained by minimizing the objective function $J$.

Therefore, the basic process of PL-ICP algorithm is as follows: firstly, the initial rotation matrix is obtained according to the wheel odometer data, and then the current frame laser data is converted to the reference coordinate system. Then for each point of the current laser frame, the two closest points in the reference frame are found and the error is calculated. The error equation is constructed by removing the point with too large error. Finally, $R$ and $t$ are solved and the error is judged. If the set threshold is not satisfied, the $R$ and $t$ are used to go back to Step1 and continue to iterate. If they are satisfied, the output $R$ and $t$ are used to obtain the pose of the robot.

PL-ICP is an algorithm [18] with high matching accuracy and robustness. Compared with ICP algorithm, PL-ICP algorithm has higher solution accuracy and is more suitable for indoor environment [20]. Therefore, the PL-ICP algorithm will also be used in the laser front-end of this paper to solve the inter-frame matching of the laser. However, it is also relatively more sensitive to the initial value and requires a higher accuracy initial value. Although the wheel odometer has the advantages of high frequency and low negative environmental impact, it also has the problems of low test accuracy and the wheel is prone to deformation and slip, which leads to large errors in the test. If only the data of the wheel odometer is used as the initial value of matching, the algorithm is easy to fall into a local cycle due to the large cumulative error of the wheel odometer.

## III. SLAM WITH MULTI-SENSOR FUSION

### *A. Extended Kalman Filter Fusing Wheeled Odometry and IMU*

Since the PL-ICP algorithm has higher initial value requirements than the ICP algorithm, and the wheel odometer will cause more and more cumulative errors with the movement of the robot and tire slip, if only the data of the wheel odometer is used as the initial value of the PL-ICP algorithm iteration, the PL-ICP algorithm may fall into a local loop. Although IMU has the problem of integral pose divergence, its instantaneous pose is accurate. Therefore, extended Kalman filter can be used to fuse wheel odometry and IMU to improve the accuracy of pose, and the fused pose data can be used as the initial value of PL-ICP algorithm to avoid falling into local circulation [21].

The extended Kalman filter realizes the linearization of the nonlinear function by performing the first-order Taylor expansion of the nonlinear function. The basic process is as follows:

The state equation is as follows.

$$x_k = f\left(x_{k-1},u_{k-1},w_{k-1}\right) \quad (7)$$

The observation equation is:

$$z_k = h\left(x_k,v_k\right) \quad (8)$$

Where $x_k$ and $u_{k-1}$ are the state variable and the control of the system respectively, $w_{k-1}$ and $v_k$ are the process noise and measurement noise and meet normal distribution, and 1 in the subscript $k$ and $k-1$ is the time unit, indicating the sampling time. Because of the state equation and observation equation is nonlinear, so a posteriori state estimation on the system of $\hat{x}_{k-1}$ place for the first order Taylor expansion for linearization, the results are as follows:

$$\begin{aligned} x_k &= f\left(\hat{x}_{k-1},u_{k-1},w_{k-1}\right) \\ &+A\left(x_k-\hat{x}_{k-1}\right)+Ww_{k-1} \end{aligned} \quad (9)$$

Because of $w_{k-1}$ error cannot be computed, so assume it is 0, the $\tilde{x}_k = f\left(\hat{x}_{k-1}, u_{k-1}, 0\right)$, in which $A$ is the Jacobian of the partial derivative of the function $f$ with respect to $x$, $A = \frac{\partial f}{\partial x}|_{\hat{x}_{k-1}, u_{k-1}}$, $W$ is the Jacobian of the partial derivative of the function $f$ with respect to $w$, $W = \frac{\partial f}{\partial_w}|_{\hat{x}_{k-1}, u_{k-1}}$, in this way:

$$x_k = \tilde{x}_k + A\left(x_k - \hat{x}_{k-1}\right) + Ww_{k-1} \tag{10}$$

The observation equation is $\tilde{x}_k$ linearized at, and the result is as follows:

$$z_k = h\left(\tilde{x}_k, v_k\right) + H\left(x_k - \tilde{x}_k\right) + Vv_k \tag{11}$$

Because $v_k$ for error cannot be computed, assuming it is 0, let $\tilde{z}_k = h\left(\tilde{x}_k, 0\right)$. $H$ is the Jacobian of the partial derivative of the function $h$ with respect to $x$, $H = \frac{\partial h}{\partial x}|_{\tilde{x}_k}$, V is the Jacobian of the partial derivative of the function $h$ with respect to $v$, $V = \frac{\partial h}{\partial v}|_{\tilde{x}_k}$, $P\left(Vv_k\right) \sim N\left(0, VRV^T\right)$.

$$z_k = \tilde{z}_k + H\left(x_k - \tilde{x}_k\right) + Vv_k \tag{12}$$

So the prediction stage of the extended Kalman filter is as follows:

*1)* The prior state estimate is:

$$\tilde{x}_k^- = f\left(\hat{x}_{k-1}, u_{k-1}, 0\right) \tag{13}$$

*2)* The covariance matrix of the error is:

$$P_k^- = AP_{k-1}A^T + WQW^T \tag{14}$$

correction stage is as follows:

*1)* Kalman gain:

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + VRV^T} \tag{15}$$

*2)* Posterior state estimation results:

$$\tilde{x}_k = \tilde{x}_k^- + K_k\left(z_k - h\left(\tilde{x}_k^-, 0\right)\right) \tag{16}$$

*3)* Update the error covariance matrix:

$$P_k = \left(I - K_k H\right)P_k^- \tag{17}$$

Therefore, the fusion process of wheel odometer and IMU using extended Kalman filter is as follows: When the observation equation of the first sensor is updated, the state quantity and the covariance matrix of the system are obtained as the system predicted state quantity and the system predicted covariance matrix of the second sensor correction process, and then the updated system state estimation result and the covariance matrix are used as the fused output. And they are used in the prediction process for iteration at the next moment [22]. In order to ensure that the subsequent fusion experiment in the real environment can be carried out smoothly, the algorithm is first tested in the simulation software to verify whether the algorithm can correctly subscribe the information of IMU and odometer node and output the fused odometer data. Fig. 1 shows the simulation fusion calculation diagram of the algorithm under ROS:

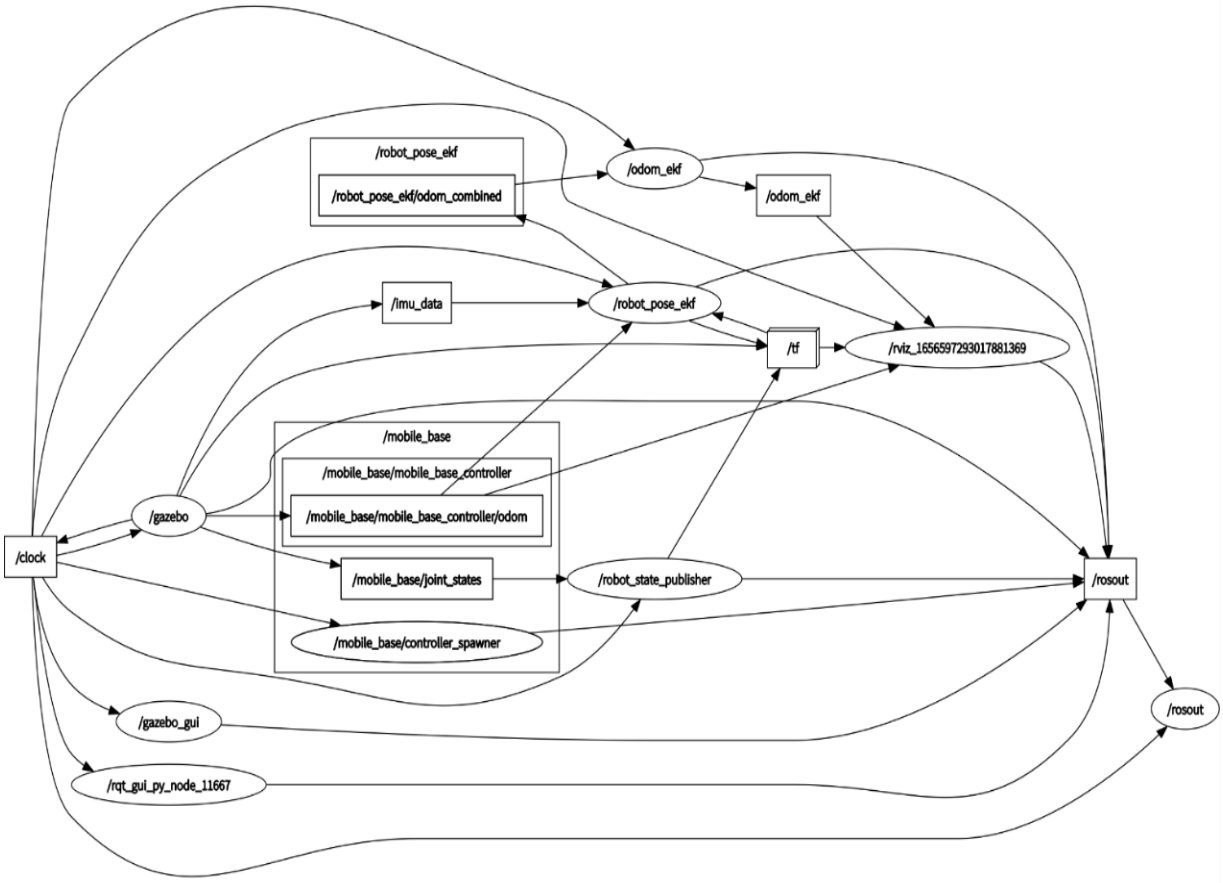



Figure 1. Simulation fusion calculation diagram

As can be seen from the arrow pointing in the Fig. 1, the fusion node robot_pose_ekf subscribing the odom topic and imu_data topic of the chassis respectively, and then publishing the fusion result as odom_combined topic and the tf transformation of the robot. Finally, the odom_ekf node converts the odom_combined format and publishes it as odom_ekf topic. The result of the final fusion is shown in Fig. 2:

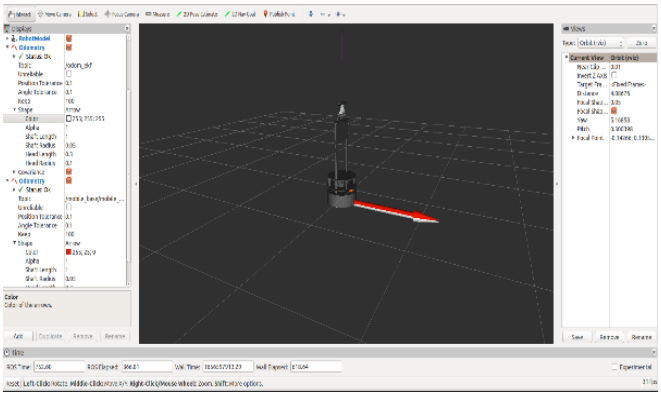


Figure 2. Extended Kalman filter fusion results

As can be seen from the results in Fig. 2, subscription is available in rviz odometry after successful fusion, where the white arrow represents the fused odometry information, and the red arrow represents the original chassis odometry information. In the simulation software, the movement of the simulation robot can be controlled by the keyboard control node, and the fusion node can output the fused odometry data in real time. The output test of odom_ekf with fused odometry data is shown in FIG. 3:

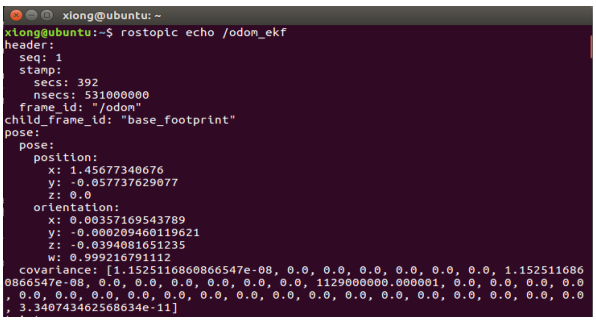



Figure 3. Integrating odometer data

The above experiments are simulation tests of the fusion algorithm. In order to further verify the effectiveness of the extended Kalman filter for the fusion of IMU and wheeled odometer data, the positioning accuracy of the robot is tested in the real scene as shown in Fig. 4. The starting position in the figure is the position where the robot is turned on and the origin of the world coordinate system. The tile size of the test scene is 65cm×65cm. For the convenience of recording, let the robot move from the starting position in the following figure to the end position in the figure along the red rectangle trajectory.

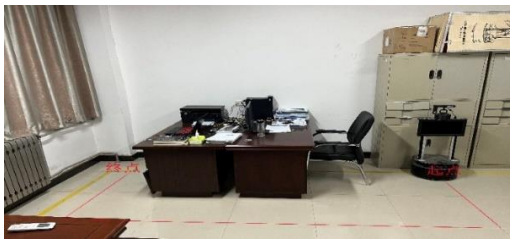


Figure 4. Experimental scenario

In order to facilitate the comparison of odometer data, the remaining three vertices in the rectangular trajectory except the starting point were selected as the target points during the test, and the odometer data was reserved for three decimal places, and the unit was m. The final test results are shown in Table 1 below:

TABLE I. EXPERIMENTAL RESULT

| Starting point | Target point | Wheel odometry | Error | Fusion odometry | Error |
|---|---|---|---|---|---|
| (0,0) | (1.3,0) | (1.347,0.035) | (0.047,0.035) | (1.324,0.029) | (0.024, 0.029) |
| | (1.3,-3.25) | (1.391,-3.373) | (0.091,0.123) | (1.386,-3.322) | (0.086, 0.072) |
| | (0,-3.25) | (0.152,-3.387) | (0.152,0.137) | (0.085,-3.316) | (0.085, 0.066) |
| Mean error | | | (0.097,0.098) | | (0.065, 0.056) |

It can be seen from the test data in the table above that in the rectangular area trajectory of 1.3m×3.25m, the positioning accuracy of the robot is improved by at least 33% compared with the positioning accuracy of the wheeled odometer by fusing the data of the wheeled odometer and IMU. Since the PL-ICP algorithm requires higher initial values than ICP, poor initial values may cause the iterative solution process to fall into a local cycle, so this paper uses the extended Kalman fusion wheel odometer and IMU data, and takes the relatively more accurate data after fusion as the initial value of the PL-ICP algorithm. To a certain extent, the problem that PL-ICP may fall into a local cycle is avoided.

### *B. Creation of Fused Two-Dimensional Raster Map*

Because the simple two-dimensional laser SLAM mapping can only scan the information of installation height and lack the information of vertical direction, there may be missing information when scanning and mapping the desk, stool and other items in the indoor environment, such as: When the height of the desktop is higher than the installation height of the lidar, the lidar can only scan the information of the table leg, and there is no obstacle in the middle of the table leg. If the actual height of the robot is higher than the height of the table, the subsequent navigation, because the laser radar can not scan the table, it is determined that there is no obstacle to pass through the middle of the table leg, this situation may lead to damage to the robot. At present, the visual SLAM algorithms based on feature points and direct methods are established in a static environment for measurement, but there are many variables in the real scene, which lead to a serious decline in the positioning accuracy and robustness of the SLAM system, and even lead to the failure of mapping. And pure visual SLAM, due to the limitation of the sensor itself, has low accuracy, is greatly affected by ambient light, and the error of the lack of texture environment is very large. The traditional visual SLAM algorithm does not have the ability to perceive the target in real time, so it is not suitable for simple visual mapping for navigation. However, visual SLAM can scan the vertical information, so the local grid map

established by vision and the local grid map established by laser can be fused. On the one hand, it can make up for the lack of vertical information in laser mapping, and on the other hand, it can also improve the low accuracy of visual mapping. The schematic diagram of the scanning range of laser and vision is shown in Fig. 5:

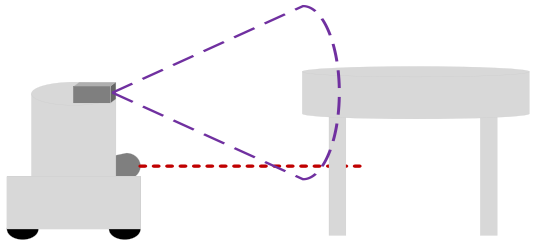


Figure 5. Laser and visual scanning range

The red dotted line in the figure above represents the scanning range of the two-dimensional laser sensor installation position, which can only scan objects in the installation height plane, and the purple dotted line of the cone represents the scanning range of the vision sensor. The Xtion Pro live depth camera used in this paper has a vertical field of view of 45 °and a horizontal field of view of 58 °.

Since the final established is a two-dimensional raster map, and the depth point cloud information of the depth camera is three-dimensional, it is necessary to project the three-dimensional depth information into two-dimensional pseudo-laser data, and then construct a two-dimensional raster map. Therefore, the final fusion process of the local raster map established by laser and vision is as follows: after the point cloud information is obtained by two-dimensional lidar, the obstacle information obtained by laser is transformed into the raster map coordinate system according to the solved robot pose, and the local two-dimensional raster map of the environment obstacles is formed. At the same time, the data of the depth camera is projected into a pseudo-two-dimensional laser data, which is also transformed into a raster map coordinate system to form a local two-dimensional raster map. Then, the local 2D raster maps of the two are fused to supplement the vertical environmental obstacle information that is not obtained by the 2D laser radar, and a global raster map is formed.

For the fusion of local raster maps, according to the basic principle of raster maps, the Bayesian method is continued to be used for the fusion of raster maps, and the fusion formula [23]:

$$P^0 = \frac{P_s^0 P_m^0}{P_s^0 P_m^0 + \left(1-P_s^0\right)\left(1-P_m^0\right)} \tag{18}$$

In the above equation, $P_m^0$ and $1-P_m^0$ represent the prior probability of occupied and unoccupied grid before fusion respectively, $P_s^0$ represents the conditional probability of grid state obtained by distance sensor, and $P^0$ represents the estimated value updated by distance sensor according to the current measurement distance after the obstacle is measured. In the fusion process, the fusion is performed according to the coordinates of the raster according to the rules shown in Table 2 below.

TABLE II. LOCAL MAP FUSION RULES

| 2D excitation Optical radar | Depth camera | | |
|---|---|---|---|
| | Occupy | empty | Uncertain |
| Occupied | Occupy | Occupy | Occupy |
| empty | Occupy | empty | empty |
| Uncertain | Occupy | empty | Uncertain |

If the obtained probability of grid occupation $P^0$ is greater than the initial threshold of the grid $T_0$, then the probability of the current grid occupation is set to 1, otherwise it is still $P^0$, where $T_0$ is 0.5. Then the probability value of each grid after the distance sensor measurement is:

$$P_{n=1,2}^0 = \begin{cases} 1 & , \quad P^0 > T_0 \\ P^0 & , \quad P^0 \le T_0 \end{cases} \tag{19}$$

In the equation, $P_{n=1}^0 = P_{n=2}^0 = P_2^0$ is the grid probability value corresponding to different sensors, so the probability of grid being occupied after fusion can be obtained by using Bayesian estimation:

$$P_f^0 = \frac{P_1^0 P_2^0}{P_1^0 P_2^0 + \left(1 - P_1^0\right)\left(1 - P_2^0\right)} \tag{20}$$

## IV. Experimental Verification and Analysis

### A. Simulation Environment Analysis

The robot used in this paper is Handsfree, so firstly, Gazebo 3D simulation software is used to create a simulation robot model with the same sensors as Handsfree platform, as shown in Fig. 6 after the creation is completed:

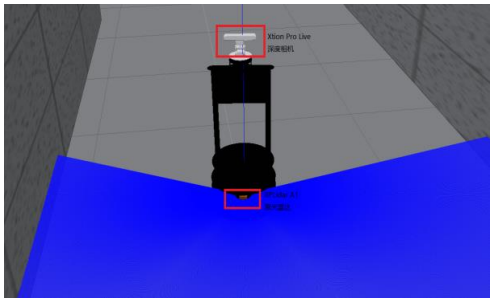


Figure 6. Robot model

The above figure is the overall model of the simulated robot, in which the red boxes are the simulated Xtion Pro Live depth camera and RPLidar A1 lidar model respectively. At the same time, in order to ensure better follow-up experiments, the same sensor parameters of each sensor of the simulated robot and the real robot are set respectively. Then, a simulation experiment environment is established for the experimental effects that need to be verified by the algorithm in this paper. Obstacles such as four-legged desks, T-shaped tables (the height of the desktop is higher than the installation height of the two-dimensional lidar), trucks (the height of the truck body is also higher than the installation height of the two-dimensional lidar) and solid spheres for convenient comparison are set in the simulation experiment environment. The setting of the truck model is to better show the fusion effect on the final created environment map, and the final created simulation experiment environment is shown in Fig. 7:

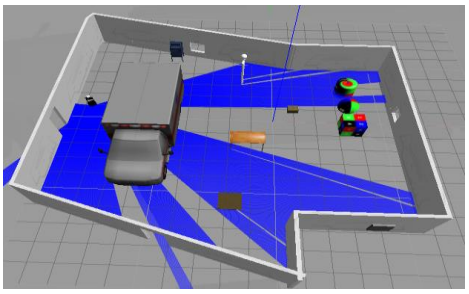
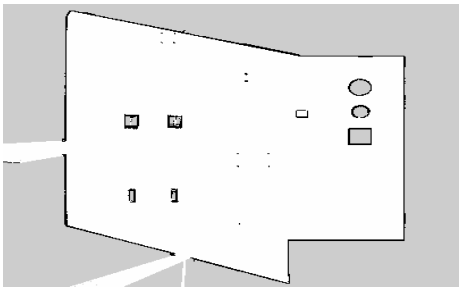
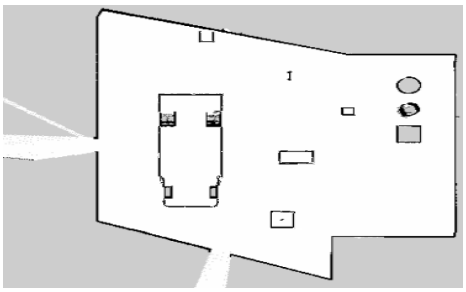


Figure 7. Simulation experimental environment

It can be seen from the above figure that the height of the table and the truck body are higher than the installation height of the 2D lidar, so the scanning point of the 2D lidar passes through the table and the truck. If it is a single 2D LIDAR SLAM, it will be considered that there are no obstacles here, which may lead to the collision between the robot and the table and the truck in the subsequent navigation. The effectiveness of the fusion is verified through the simulation experiment of the single two-dimensional laser sensor mapping and the multi-sensor fusion scheme in this paper, and the angular velocity and linear velocity of the robot motion are kept the same during the experiment. The results are shown in Fig. 8:



(a) Single 2D laser mapping



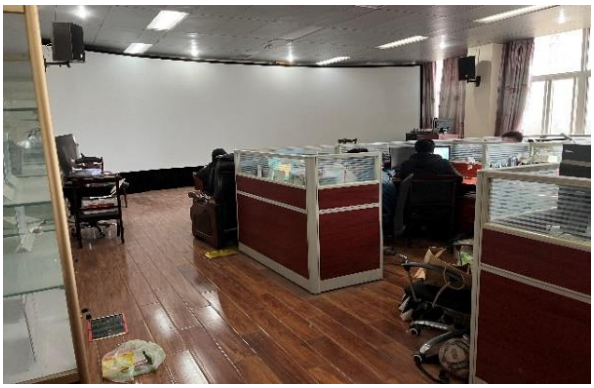
(b) Fusion mapping

Figure 8. Comparison of simulation experiments

Through the mapping results of the above figure combined with the simulation experiment environment, it can be seen that the two-dimensional lidar can only scan the object with the installation height, and cannot scan the main body of the truck and the desktop with the installation height higher than the lidar in the simulation environment. For example, the blue range in the simulation environment in Fig. 7 is the scanning point of the lidar. As there is no laser information of the main truck and the desktop in the simulation environment, it will be considered that there are no obstacles here. The final mapping result is shown in Fig. 8 (a), and the complete obstacle map information cannot be established for the table and the truck in the simulation environment. For the fusion mapping, since the depth camera can scan the information of the main body of the truck and the table, and project these point cloud information into a fake two-dimensional laser data for mapping, it is very good to establish the complete information of the table and the truck on the final grid map, as shown in Fig.8 (b) , so as to avoid the risk of collision between the robot and the obstacles in the subsequent navigation.

### *B. Experimental Verification and Analysis in Real Environment*

In order to further verify the effect of fusion, this section will test the algorithm in the real environment of about 70m2 as shown in Fig. 9 below to verify the effectiveness of the algorithm in the actual environment.



(a) Overall environment

(b) Environmental front

Figure 9. Real experimental environment

FIG. 9 (a) shows the overall situation of the experimental environment, and (b) shows the front part of the experimental environment. In (b), there is a row of conference tables on the right, and the height of the gap left under the table top is higher than the lidar installation height of the mobile robot. This is shown in Fig. 10:


Figure 10. Installation height of LiDAR

The red circle in the figure above is the comparison between the height of the lidar and the gap under the conference table. Since the gap height of the conference table is larger than the installation height of the lidar, the information above the conference table will not be built into the final grid map of the environment when the single 2D lidar is used for mapping.

During the experiment, keep the linear speed of the robot at 0.2m/s and the angular speed of the steering at 30 ° per second. Start at the position shown in Fig. 11, circle around the experimental environment clockwise and return to the origin.

Figure 11. Robot departure position

As a comparison, the Gmapping algorithm based on a single two-dimensional lidar and the fusion algorithm in this paper are used for mapping tests, and the motion speed of the robot is kept unchanged during the experiment, and the results are shown in Fig. 12:

(a) Single 2D laser mapping (b) Fusion mapping

Figure 12. Real environment mapping results

Fig 12 (a) shows the mapping results of Gmapping. It can be seen that part of the information of the conference table is missing in the final raster map, and only the leg information of the conference table is available. In (b), the complete information of the conference table is added to the final raster map, and this information is obtained by fusing the local raster map established by the projection of the depth camera into a pseudo-two-dimensional laser data, so as to verify the effectiveness of the fusion mapping.

In addition, in order to facilitate the recording of the robot's pose during operation, the timing control is used to release forward and turn commands to the mobile robot chassis to control the robot to reach the specified position. Five points at the same position in the two experiments were selected to analyze the accuracy of the pose, and the results are shown in Table 3 below.

TABLE III. POSITIONING RESULTS

| Actual location (m) | Actual pose ( ) | Gmapping | | | | Vision + Laser | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Estimated position (m) | Estimated pose ( ) | Root mean square error of position (cm) | Attitude error ( ) | Estimated position (m) | Estimated pose ( ) | Root Mean square error of position (cm) | Pose Error ( ) |
| (6,0) | 30 | (6.085, 0.079) | 33.149 | 11.604 | 3.149 | (6.059, 0.037) | 31.092 | 5.9 | 1.092 |
| (6,-3) | 90 | (6.094, -3.081) | 94.634 | 12.408 | 4.634 | (6.064, -3.051) | 92.043 | 8.184 | 2.043 |
| (6,-6) | 90 | (6.103, -6.089) | 95.005 | 13.612 | 5.005 | (6.072, -6.063) | 92.729 | 9.567 | 2.729 |
| (0,-6) | 180 | (0.135, -6.112) | 186.024 | 17.541 | 6.024 | (0.089, -6.075) | 184.007 | 11.639 | 4.007 |
| (0,0) | 0 | (0.156, 0.127) | 7.678 | 20.116 | 7.678 | (0.091, 0.08) | 4.96 | 12.117 | 4.96 |

According to the results in the above table combined with the mapping results, the Gmapping algorithm based on a single two-dimensional lidar relies heavily on the wheeled odometer, and the cumulative error of the wheeled odometer information increases with the movement of the robot, so when the robot finally moves back to the origin, the established environment map has been misaligned. The root mean square error of the position reaches about 20cm. The fusion algorithm in this paper uses the laser odometer, and uses the extended Kalman filter to fuse the data of the wheel odometer and IMU as the iterative initial value of the laser odometer to obtain a more accurate robot pose, so the overall pose error is significantly smaller than Gmapping, and the root mean square error of the robot's position is about 12cm when it returns to the origin. The positioning error of the robot is effectively reduced.

## V. CONCLUSIONS

In this paper, SLAM based on single 2D LiDAR is analyzed and optimized. Extended Kalman filter is used to fuse wheel odometer and IMU data to provide initial iteration values for the laser interframe matching algorithm PL-ICP, and at the same time, the fusion visual projection is built into a local 2D raster map of pseudo-2D laser data. The results show that compared with the single two-dimensional laser SLAM, the multi-sensor fusion SLAM scheme in the paper can improve the pose accuracy of the robot and build a more complete global map of the environment details.

## ACKNOWLEDGMENT

The authors wish to thank the cooperators. This research is partially funded by the Innovation and Entrepreneurship Project Fund for College Students (202310702040).

## REFERENCES

[1] Randall Smith, Matthew Self. Estimating Uncertain Spatial Relationships in Robotics [J]. CoRR, 2013, abs/1304.3111.

[2] HAO Rui, Li Rui, SHI Yingjing et al. System self-localization and map reconstruction based on multi-sensor fusion [J/OL]. Radio Engineering :1-9[2023-09-04].

[3] ZHAO Shaoan. Simultaneous Localization and Mapping for Mobile Robot Navigation with 3D Laser Point Clouds [D]. University of Electronic Science and Technology of China, 2018. (in Chinese)

[4] ZHAO Yancheng, WEI Tianxu, TONG Di et al. Research on Visual SLAM algorithm based on YOLOv5s in dynamic scenarios [J/OL]. Radio Engineering :1-10[2023-08-09].

[5] DENG Zibin. Research on Dynamic target tracking algorithm based on Visual SLAM [D]. Southwest University of Science and Technology, 2023.

[6] Quan M, Piao S, Tan M, et al. Tightly-Coupled Monocular Visual-Odometric SLAM Using Wheels and a MEMS Gyroscope [J]. IEEE Access, 2019, 7:97374-97389.

[7] XU Tong. Simultaneous Localization and Mapping of Mobile Robot Based on Multi-Source Perception Information Fusion [D]. Dalian Maritime University, 2020. (in Chinese)

[8] TANG Yuanwen, LIU Zuoshi. Research on multi-sensor Fusion SLAM of mobile robot in Complex Environment [J]. Manufacturing Automation, 2023, 45(08):108-112+166. (in Chinese)

[9] Zhu J B, Zhao J H, Cui C, et al. Robustness Design of UWB/IMU integrated Navigation System based on EKF [J]. Computer Simulation, 20, 37(12):47-52, 57.

[10] Ye H, Chen Y, Liu M. Tightly Coupled 3D Lidar Inertial Odometry and Mapping [J]. International Conference on Robotics and Automation (ICRA), 2019, 3144-3150.

[11] YUE Shengjie, WANG Hongqi, LIU Qunpo, et al. Robot pose adaptive estimation based on Extended Kalman filter and point-line nearest point iterative scan matching algorithm [J]. Surveying and mapping bulletin, 2022 (7) : 49-53.

[12] Labbé M, Fran çois, Michaud O. RTAB‐Map as an Open‐Source Lidar and Visual Simultaneous Localization and Mapping Library for Large‐Scale and Long‐Term Online Operation [J]. Journal of Field Robotics, 2019:416-446.

[13] ZHANG Wen. Research on Autonomous Navigation Method for Indoor Robots Based on Multisensor Fusion [D]. University of Science and Technology of China, 2017. (in Chinese)

[14] WANG yang. Research on SLAM Mapping Algorithm of Mobile robot based on image optimization [D]. Zhejiang University, 2022. (in Chinese)

[15] Censi A. An ICP variant using a point-to-line metric [C]// IEEE International Conference on Robotics & Automation. IEEE, 2008.

[16] LIU Hewei, QING Zhaobo, HUANG Jiajun. Research on interframe matching algorithm based on improved ICP [J]. Modern Electronic Technique, 2023, 46(08):73-78. (in Chinese)

[17] REN Xisnghua. Lidar Indoor SLAM Method [D]. Harbin Engineering University,2018. (in Chinese)

[18] SUN Jian, LIU Longhui, LI Zhi, et al. Mobile Robot Sensor Data Fusion Algorithm Based on RGB-D Camera and Lidar [J]. Journal of Hunan Institute of Engineering(Natural Science Edition), 2022, 32(01):18-24. (in Chinese)

[19] JIANG Zuopeng, MEI Tiancan. A single-line laser odometer based on PL-ICP and NDT point cloud matching [J]. Laser Journal, 2020,41(03):21-24.

[20] WANG Zirun, YAN Bixi, DONG Mingli et al. Positioning method of wall-climbing Robot based on LiDAR and improved AMCL [J]. Chinese Journal of Scientific Instrumentation, 202, 43(12):220-227. (in Chinese)

[21] FENG Bin. Research on Simultaneous Location and Mapping Algorithm Based on Bayesian Filter [D]. Chang'an University, 2020. (in Chinese)

[22] ZHANG Ke. Research on Mapping and Navigation Algorithm of Indoor Robot Based on Depth Camera[D]. Harbin Institute of Technology, 2020. (in Chinese)

[23] ZHANG Yi, DU Fanyu, LUO Yuan, et al. Map-building Approach Based on Laser and Depth Visual Sensor Fusion SLAM [J]. Application Research of Computers, 2016, 33(10):2970-2972+3006. (in Chinese)

# Face Recognition System Based on Capsule Networks

JiangRong Shi
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 3097078251@qq.com

Li Zhao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 332099732@qq.com

***Abstract*—This study introduces a technique for facial recognition according to capsule networks. The system utilizes the advantages of capsule networks to model the face features in the image hierarchically, and realizes the efficient recognition of faces. First of all, we know the difference between the capsule network and the convolutional neural network through the study of the operating principle and the structure of the capsule network. Secondly, the Capsule Network is realized through deep research on the algorithm for dynamic routing and the internal operating principle of the capsule. Finally, by conducting experiments on the face dataset and optimizing it with the Adam optimization algorithm as well as the boundary loss and reconstruction loss, the capsule network is promoted to learn more robust feature representations to obtain better face recognition results. The experiments show that the face recognition system based on capsule network can reach 93.5% correct rate of evaluation on WebFace dataset, which achieves a high recognition accuracy. The final results demonstrate the feasibility and effectiveness of capsule networks for face recognition.**

***Keywords-Capsule Neural Network; Dynamic Routing; Face Recognition***

## I. INTRODUCTION

In the last few years, face recognition technology has become the most dominant biometric technology today, and its direct, convenient, and contactless features make it easy for users to accept, and it has been extensively used in numerous fields. Deep convolutional neural networks have demonstrated significant potential in a number of fields recently, including image identification. Deep neural networks have, however, been unable to consider the spatial relations of the underlying objects in recent years.

In order to overcome the limitations of deep convolutional neural networks, the capsule network has been widely used in the past. It takes the output of the capsule in the form of vectors, which can not only represent the image according to the intensity of the vectors, i.e., the size, but also describe the direction of the image with vector direction, location, and other information about the image object. Therefore, capsule networks are rapidly developing in deep learning with their unique charm. It is an important topic in image recognition at present and for a long time to come, and many scholars are actively working in this area. The success of capsule networks has many good outcomes in the domain of picture identification, and the technique for recognizing faces based on capsule networks involves a wide range of topics, which is crucial for research given the quick development of IT, the ongoing evolution of society, and the demand for complete recognition [2].

## II. RELATED WORKS

### *A. Face Recognition*

One type of biometric technology is facial recognition that recognizes information about the facial features of a person's face as well as determining the likelihood that it is a particular person. Researchers began to study facial recognition as early as the 1960s, and it was not until the 1990s and a half that the technology entered the stage of real primary application, and the technology is so mature that it hasn't been developed yet. Face recognition has been extensively researched as a type of non-contact

biometric identification method along with the advancements in computer vision and pattern recognition. Examples include authentication, monitoring and security detection. It can be quite convenient and automated. The majority of facial recognition techniques rely on conventional strategies for machine learning such as Principal Component Analysis (PCA) and Linear Discrimination (LDA). With the advent of deep neural networks (CNNs), facial recognition technology has advanced significantly in recent years. Deep learning has demonstrated in recent years that high level abstract features may be extracted from the source image with better robustness and accuracy [3]. Face detection and face feature extraction are the two primary components of most deep learning-based face recognition systems. Face features are extracted to create discriminative features, and face detection is used to find and extract these features. Neural networks with convolutions have been the subject of several proposals in recent years for feature extraction from the feature space. Face recognition is also connected to other technologies including 3D face reconstruction, live detection, and multimodal fusion in order to enhance the system's robustness and performance. However, there are still a number of challenges in the field of face recognition, such as changing lighting conditions, changing pose, blocking, and privacy and security. Traditional machine learning methods have seen a dramatic shift within the domain of face recognition in recent years. The performance and application of the face recognition system will be progressively improved along with the ongoing development and advancement of technology, offering society more practical and secure solutions.

### *B. Convolutional Neural Network*

Convolutional neural networks, or CNNs, were first proposed in the 1980s and early 1990s. Yann LeCun is credited with one of the first successful uses of CNNs [16]. LeNet-5 was really utilized by the United States Postal Service and was initially intended for handwritten digit recognition. This model is a major breakthrough in the application of convolutional neural networks in computer vision. Since the advent of deep learning, Alex Krizhevsky and his colleagues have been able to improve their performance in the 2012 ImageNet Challenge. It introduced innovative designs such as deep structure, massively parallel computation, and ReLU activation function, which has been extensively utilized in the field of computer vision and picture classification [17].

Three layers make up deep convolutional neural networks: the convolutional layer, pooling layer, and fully connected layer. The pooling layer is used to minimize the size of the parameters, while the convolutional layer is used to extract the image's local features. The foundation of convolutional neural networks is a set of filters that can identify characteristics in the input data. The buried topological characteristics are then extracted using pooling and convolution. Convolutional neural networks have been offered as a way to increase system performance and simplify the network's parameters through pooling and weighting procedures.

Deep neural networks have had tremendous success recently in a variety of fields, such as image identification and computer vision. Among the many computer vision tasks that convolutional neural networks have excelled at are picture categorization.

The disadvantages of convolutional neural networks are mainly reflected in two aspects. In the first place, in convolutional neural networks, no attention is paid to the relative positions of different features. Scalar transfers higher level neurons to lower level neurons, but it lacks direction and is unable to convey spatial qualities or the relationship between top and bottom attributes in terms of position. Connection between the objects at the base [1]. Therefore, CNNs have significant limitations in the recognition of spatial relationships. On the other hand, in Convolutional Neural Networks, there is a significant improvement in their robustness, while much valuable information of the model is lost. Convolutional neural network is trained well

when it encounters very similar images in the data set, but when it encounters an image that has flipped, tilted, and other problems related to orientation, whereas convolutional neural network does not work so well. Consider a face, for instance, which is made up of facial contours, eyes, nose, and mouth. When these components are present in convolutional neural networks, there is a strong stimulus, and the components' relative positions and orientations are less significant, so that they can recognize faces from those areas. Except it's not a face for us.

*C. Capsule neural network*

In 2017, Hinton et al. first proposed the concept of capsule networks, which is one of the most cutting-edge techniques in the domain of recognition and classification of images today [12]. Capsule networks are proposed to solve some problems existing in traditional convolutional neural networks, such as translation invariance, insensitivity to scale changes and pose changes. Capsule networks solve these problems by introducing capsule layers. A Feature Map (FM) is produced in traditional CNNs by adding a filter to a convolution process. However, this method is not able to handle spatial relations and pose information efficiently. In contrast, Capsule Networks can better capture the relationship between target instances by representing each target instance as a vector.

Capsule networks have several advantages over conventional CNNs:

- Modeling Spatial Relationships: Capsule Networks can capture spatial relationships between target instances, providing better modeling capabilities for target pose, scale change, and rotation.
- Stronger representation: By using vectors to represent the states of target instances, capsule networks can provide richer and more expressive feature representations, thus improving recognition accuracy.
- Improved robustness: Capsule networks have better robustness to translation invariance and spatial transformations when processing images, and can cope with various changes in complex scenes.
- Strong interpretability: Since the capsule networks represent the state of the target instance with vectors, it is easier for the network to interpret its learned knowledge.

Unlike convolutional neural networks, which are more mature in various fields, the research of capsule networks is still in its infancy, and most of the capsule neural network research stays on the basis of small samples. In 2018, Deng et al. proposed a method to classify hyperspectral images with a small amount of sample data using capsule neural networks [18]. They introduced a new two-layer restricted training paradigm in HSI classification. Two HSI datasets were mainly used in the implementation and the algorithm was mainly used to describe complex and concise data and the algorithm was trained to examine the robustness and representation of individual models or classifiers. Sabour et al. suggested a novel feature network that comprises of a single layer of convolutional neural networks as a preprocessing layer in order to get around the drawbacks of convolutional neural networks when processing images and a layer of advanced capsule as a prediction vector for image classification [12]. Capsule Neural Network is able to recognize all kinds of features, including posture, size, and orientation and so on, but its dynamic routing mechanism is very cumbersome and needs further improvement. To address this problem, Hinton et al. improved the routing algorithm and the capsule structure, and based on this, proposed a capsule network based on the maximum expectation algorithm matrix. This network has better robustness, but large computation and high complexity. Hahn et al. used a simple perceptron model instead of the traditional dynamic algorithm, which improves the system's performance without increasing the parameters and arithmetic cost. Zhang, Y et al. investigated the performance of capsule networks in processing complex data and proposed an improved capsule network structure [13]. Xiang, S et al. introduced a method called Dynamic Capsule Attention (DCA) which was applied to a visual question and answer task [14]. Tang, H et al. proposed a Recurrent Capsule

Network (RCN) for character re-recognition task [15].

Jiang Hong et al used a convolutional layer in front of the first layer of the capsule of CapsNet, and a filter capsule layer at the end of the network. Compared with Capsnet, this method improves the recognition precision of target image and improves the performance of reconstruction [5]. Zhou Qun improved the capsule network and designed a new capsule network JSSA-Caps Net based on spatial-spectral attention module, and used a capsule network to classify hyperspectral remote sensing images, extracting useful characteristics from the combination of spatial and spectral data to enhance classification performance [7]. Yao Yuqian proposed an algorithm for recognizing expressions based on Enhanced Capsule Network (E-CapsNet) and a recognition of an expression algorithm based on Double Enhanced Capsule Network (E2-CapsNet). It was effectively validated on the expression dataset [8]. Hanqing Zhang et al. designed an algorithm for feature extraction and recognition based on Caps-net + SRNN to be able to overcome the inability of traditional CNNs to deal well with image rotation and blurring due to their information loss in the layer of pooling, and experimentally verified the effectiveness of the neural network model suggested in this document [10]. Chen Shan et al. constructed a capsule graph through dot product attention to obtain the dependency relationship between capsules in the same layer. DPA_Caps Graph not only makes up for the lack of ignoring the sibling features in the original routing process, but also achieves to enhance the model's overall performance by adding jump connections for feature extraction in the feature extraction part, which improves the feature expression of the primary capsules by using dot-product attention instead of dynamic routing, which improves the feature selection ability between the capsule layers, and realizes the improvement of the overall performance of the model [11]. Lou Yue made the first attempt to introduce capsule networks and their improved models into the field of plant recognition for applications including plant organ recognition such as flowers and leaves, preserving detailed pose information (e.g., exact position, rotation, thickness, inclination, size of the object, etc.), and achieving generalization using less training data [9]. Yang proposed a cross-domain pedestrian re-recognition method based on deep capsule networks. Through the perspective classification training task, the model can learn the effective features of pedestrians in the image, and these features can be directly migrated to the pedestrian re-recognition task, which alleviates the problem of insufficient pedestrian re-recognition generalization capability [4]. SA-Capsnet gives full play to the feature extraction capability of self-injecting networks as well as the capsule-based neural network's dynamic routing mechanism. Dynamic routing is an attention mechanism, which has greater superiority in image regions. Combining the two can achieve complementary functions and improve the network performance. Tests have been carried out with several samples such as MNIST, Mode MNIST, CIFAR10, etc., and the outcomes demonstrate the model's high prediction accuracy [6].

## III. REQUIREMENT ANALYSIS OF FACE RECOGNITION BASED ON CAPSULE NETWORKS

### *A. System Requirements Analysis*

The system's objectives are to investigate and analyze the capsule network's theory, comprehend its elements, put the network into practice, and accomplish classification training on face datasets, to carry out the design of visualization interface and to test the working efficiency of capsule network in face recognition system. The visual interface is designed with PYQT5, and various operations are carried out by buttons, such as face data set, facial recognition and facial image addition.

### *B. System Main Functions*

This work develops and implements the Capsule Network Face Recognition System, which is based on the theory and architecture of the capsule neural network. The system mainly contains three functions, and the description of each function is specified as follows:

- Train the face dataset. Select the face data to be trained and train the classification of face data.
- Add a face dataset. Choose which category to add to after selecting the face photos you want to add to the file, and then click the Add button. After the addition is completed, you will be prompted to retrain the added face dataset.
- Face Recognition. Select a photo of the face you like to test, display it on the screen, test it by clicking the Detect button, and the person in the picture and their likelihood of being that person will be displayed.

## IV. CAPSULE NEURAL NETWORK

Capsule networks were first discovered in Geoffrey Hinton's academic paper, Transforming Autoencoders. An article titled "Dynamic Routing Between Capsules" was released at the end of 2017 by Geoffrey Hinton and his colleagues [12]. This is a novel neural network model. These days, picture recognition applications are the primary use for this technique.

A component in the brain known as a "capsule" is capable of processing various visual stimuli and encoding information (e.g., position, shape, speed, etc.) very well. In deep learning, a capsule is a structure in the brain that can process different visual stimuli well are structures in the brain that are able to process different visual stimuli well and encode information. In deep learning, a capsule is a set of embedded neurons. Instead of neurons, a network of capsules is comprised of capsules. A capsule is able to represent various features of a particular object in a picture, such as position, size, orientation, texture, etc. A capsule is an independent logical unit. A capsule can produce vectors, the direction of the vector indicates the object's pose, and the length of the vector can represent the degree of similarity of the feature.

### *A. Capsule network Structure*

The encoder in capsule network includes convolutional layer, main capsule layer, and digital capsule layer. The specific process is as follows. The convolution layer is used to detect the features of the 2D image. The 2D image's features are detected using the convolution layer. ReLU is used to activate the convolutional layer, which has 256 9 * 9 * 1 step size 1 cores. The main capsule layer, which receives the data from the convolutional layer, generates a set of features. The 32 primary capsules in this layer resemble those in the convolutional layer. Digit Capsule Layer This layer contains 10 digit capsules, each corresponding to a digit. Each capsule accepts a 6 * 6 * 8 * 32 tensor as input. You could view this as an 8-dimensional vector 6 * 6 * 32, or 1,152 input vectors. Inside the capsule, each input vector maps the 8-dimensional input space to the 16-dimensional capsule output space via an 8 * 16 weight matrix.

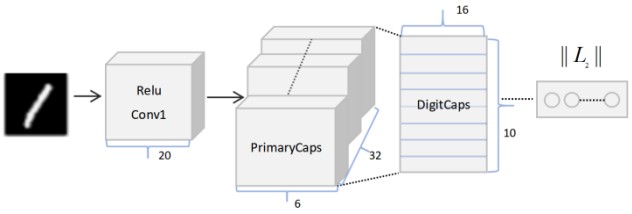



Figure 1. Capsule Network Structure

There are three connecting layers that make up the decoder: the first, second, and third. First, we accept 16 dimensional vectors from the correct digital capsule and learn to translate them into a digital image, Using the loss function — the Euclidean distance between the rebuilt image and the input image—the decoder, a regularizer, learns to reconstruct the 28*28 pixel image after receiving the output of the correct digital capsule. The decoder forces the capsule to learn features that are useful for reconstructing the output image. The ideal reconstruction of an image is one that closely resembles the original. Experiments indicate that the FNN is robust. Through the analysis of the model, we can find out the problem of the model. The rebuilt image goes through three totally connected layers, and the entire decoding process is entirely connected. The rebuilt image goes through three totally connected layers, and the entire decoding process is entirely connected.

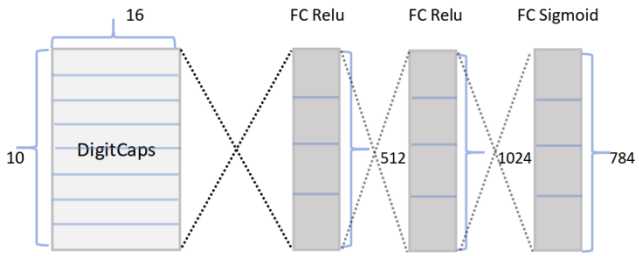



Figure 2. Decoder

*B. Dynamic routing*

In Capsule Network (CN), dynamic routing mechanism is an important technique used to calculate the weights between different capsules to determine their relationship and the degree of interaction. In simple terms, dynamic routing is used to determine how information is passed from one capsule to other capsules in the next layer. Through the dynamic routing mechanism, the capsule network can establish relationships and interactions between objects at different levels and enhances the model's robustness and accuracy by more effectively capturing the structural details and attitude changes of the objects in the recognition task. The following are the specific steps of dynamic routing.

*1)* Initialization: First, initial weights are assigned to each pair of neighboring capsules (corresponding capsules between the previous and next layers). These weights can be initialized randomly.

*2)* Projected Output Vector: Using the weight matrix and current state vector as a basis, an output vector is projected for every capsule. The entities or features that the capsule activates are represented by the output vector. Similarity between vectors. By calculating the dot product of two vectors, this can be accomplished.

*3)* Route Matching: Using the output vector of the current prediction as input, compute the output with the latter layer of capsules.

*4)* Update weights: The weights connecting each capsule are adjusted based on how similar they are to the output vectors of the capsules in the layer below. Capsules with higher similarity will get higher weights to have greater influence.

*5)* Dynamic Routing Generation Selection: The process from step 2 to step 4 is repeated until the specified number of iterations Lou is reached or the convergence condition is satisfied. The weights of the capsule and the anticipated output vector are modified in each cycle.

*6)* Output computation: Ultimately, the output vector obtained after dynamic routing is used as input to class or perform other tasks. To get the final classification result, the output vector can be transformed into a probability distribution using the softmax function.

In a capsule network, the following steps are required to perform the operation of a single capsule:

The input vectors are multiplied by a matrix, where $v_1$ and $v_2$ are generated from the output of the previous capsule, and in a capsule, $W_1$ and $W_2$ are multiplied by $v_1$ and $v_2$, respectively, to obtain new $u_1$ and $u_2$. The formula is as follows:

$$\hat{u}_{j|i} = W_{ij} v_i \tag{1}$$

The input vectors are weighted scalarly, and the weighted vectors are summed.

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{2}$$

Vector-to-vector no linearization, the result is found with the Squash function and used as input for the next capsule. In the Squash function, $v_j$ is the output vector of capsule $j$, and $s_j$ is the input vector of capsule $j$. $s_j$ is also the weighted sum of the output vectors of all capsules in the previous layer, and its value is the capsule $j$ that is currently in. This nonlinear function can be divided into two aspects. The first part represents the scale of the input vector $s_j$, and the second part represents the direction of the input vector, which is also compressed to the interval [0,1). If the $s_j$ vector is 0, then $v_j$ is 0. If $s_j$ is infinity, then $v_j$ can tend to 1. Generally speaking, the Squash function can be utilized to excite a vector

with a vector or as a means of compressing and redistributing the vector length.

$$v_j = \frac{\| s_j \|^2}{1+\| s_j \|^2} \frac{s_j}{\| s_j \|} \tag{3}$$

Through an iterative process of dynamic routing, the capsule network can gradually adjust the weights between capsules to better transfer information and facilitate effective feature learning and pose estimation. This mechanism enables the capsule network to cope with complex spatial relationships and improves the ability to recognize object deformation, rotation, and other situations.

*C. Loss function*

We already know that the length of the digital capsule layer's output vector is some kind of probability based on the preceding section,how should we construct a loss function and then iteratively update the whole network according to this loss function? Dynamic routing is used to update the coupling coefficients. On this basis, it does not need to be updated according to the loss function, but the other convolutional parameters in the ensemble by value must be updated according to the loss function. The loss value for each capsule vector in a training sample is determined using the following formula and the sum of the ten loss values yields the overall loss. This is a supervised learning, so each training sample is properly labeled. Normally, updating these parameters with a standard loss function backpropagated is sufficient, but the original work used edge loss, which is common in SVMs. The expression for this loss function is:

$$\begin{aligned} L_{\mathrm{k}} &= T_k \max(0, m^+ - \| v_k \|)^2 \\ &+\lambda(1-T_k)\max(0,\| v_k \| - m^-)^2 \end{aligned} \tag{4}$$

The $k$ of the formula denotes the classified category, $T_{\mathrm{k}}$ is the classification function (1 for the presence of k and 0 for the absence of k), $m^+$ denotes the upper boundary, and $m^-$ denotes the lower boundary. Here, where the mode of $v_k$ is equal to $L_{\mathrm{k}}$ of the vector.

## V. EXPERIMENT

*A. Selection of face dataset*

Some of the commonly used face recognition datasets are as follows. The roughly 13,000 photos in the LFW (Labeled Faces in the Wild) dataset show a variety of real-life subjects. Every picture is tagged with a person. CelebA is a huge data set of celebrities with more than 200,000 images of celebrities. Each image is labeled with 40 attributes, such as gender, hairstyle, etc. FDDB (Face Detection Data Set and Benchmark) is a dataset dedicated to face detection and contains 5,171 images and 16,419 face annotations. WIDER Face is the most widely used data set for face detection, which consists of 32,203 training images and 40,504 test images. Among them, the training images contain 393,703 face instances. MORPH is a dataset for age evolution studies containing 55,134 images covering 13,618 different individuals. It is mainly used for face detection and eye localization tasks. These are just some of the commonly used datasets, and there are many other face recognition datasets, so you can choose the right one according to your specific needs.

For the training task of face recognition based on capsule networks, the face recognition dataset that I have chosen is is the CASIA-WebFace dataset. One of the most often used data sets from the Chinese Academy of Sciences Institute for Automation (CASIA) is CASIA-WebFace. 494,414 photos with 10,575 identities make up the CASIA-WebFace dataset. These photographs feature faces in a range of expressions, stances, and lighting settings. Each person has several images in the data set, and each person has a unique and permanent identifier. In the CASIA-WebFace dataset, lighting conditions, poses, and other factors make face recognition more difficult, training difficult, and hardware demanding. Therefore, at the beginning of the training, I just selected face images of 44 people without reducing the photos of each person, and increased the dataset to 8,260 face photos of 110 people in

the later training process. Using this data as my final training dataset.

### *B. Building a system to realize face recognition*

According to the needs of the system, three main interfaces are designed in the system, including selecting pictures for face recognition, selecting face datasets for training, and selecting photos and categories to be added to the dataset in the folder. Firstly, we choose to design the interface through PyQt5, the first interface needs to show the picture that needs to be recognized, the model can be chosen to display the identity of the person in the photo and the likelihood that they are that person, if the selected file is not a picture then it will give a prompt to re-select it. The second interface requires a folder of face datasets that can be selected as well as a training log, which shows the results of each epoch of training, and the lower level of the interface will give hints at the beginning and end of training. The third interface allows you to add a face dataset by choosing the face photographs to add, the category to add them to, clicking the add button, and after the addition is successful, receiving a prompt to retrain.

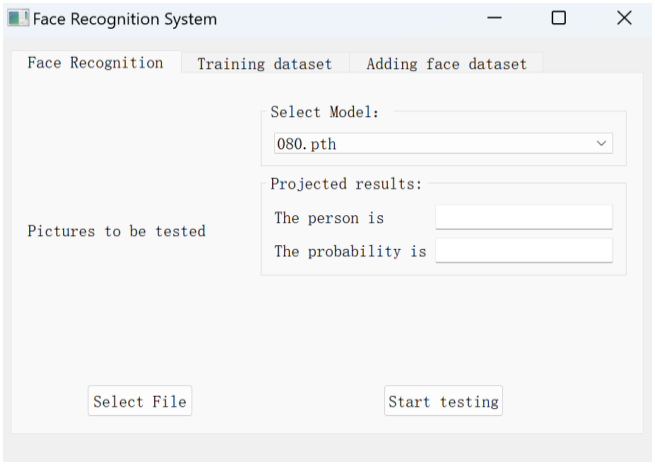



Figure 3. Face recognition

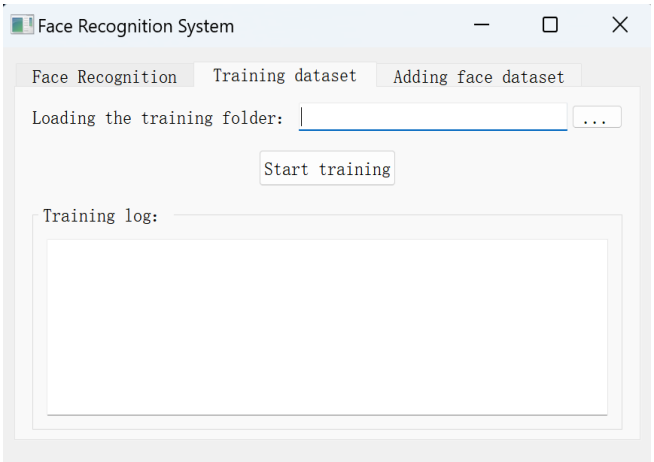



Figure 4. Training dataset

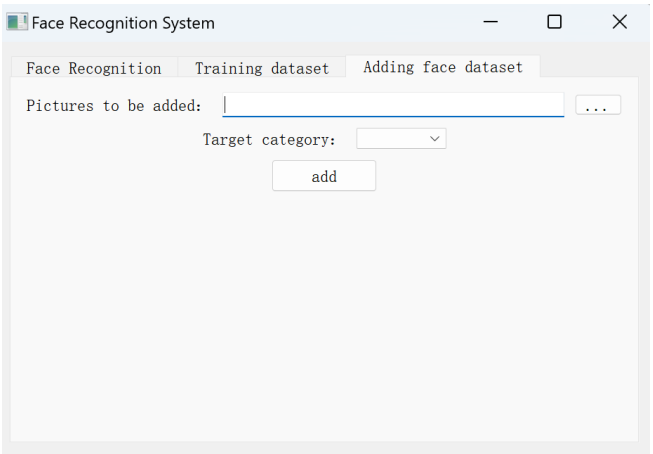



Figure 5. Adding a dataset

### *C. Training Capsule Network*

The deep learning framework Pytorch is used, along with the Adam optimization algorithm for training and the boundary loss and reconstruction loss for optimizing the capsule network. The experimental environment is Python 3.9, the number of iterations is set to 80, the learning rate is 1e-4, and the Batch_size is set to 64. The experimental test part of the photographs can be up to 98.7% accurate, the The correct rate of model evaluation can reach 93.5%. The loss curve and correctness curve are shown below.

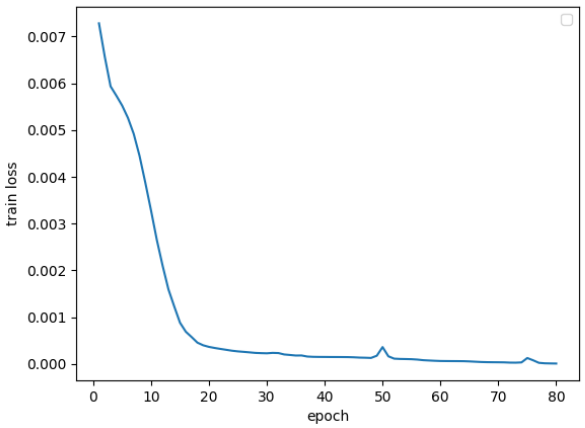



Figure 6. Loss Curve Graph

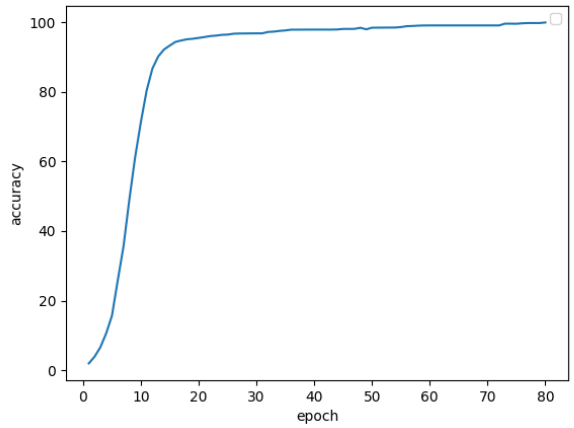



Figure 7. Accuracy Graph

Through this design implementation of capsule network based face recognition system, it is very good to understand the advantages of capsule network image recognition field for image processing. It is better at capturing the spatial relationships found in the picture. The categorization capabilities of the capsule neural network is confirmed by the face recognition results on the CASIA-WebFace face dataset.

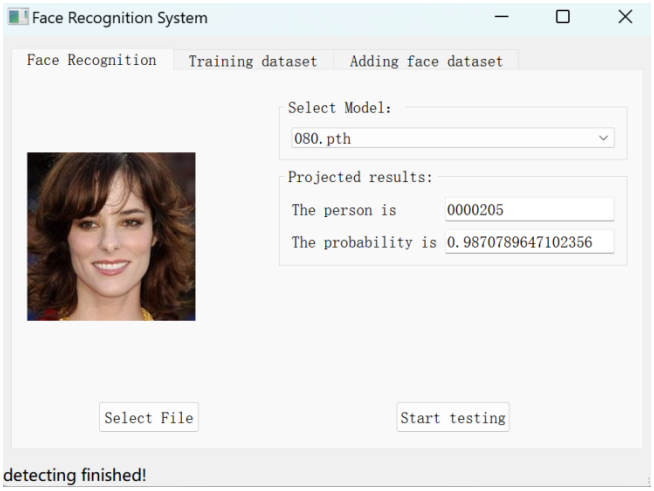



Figure 8. Face test Results

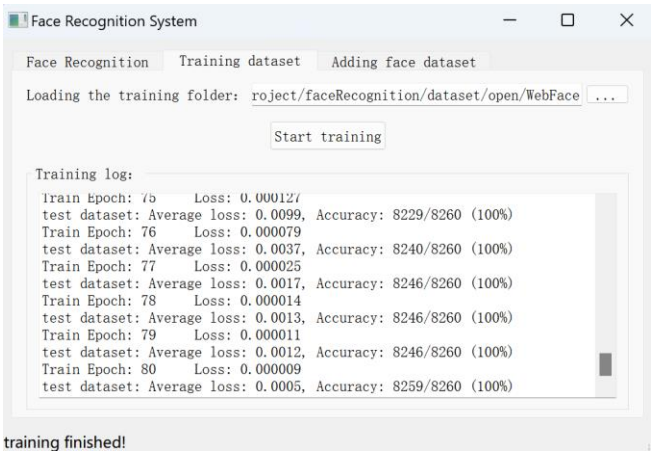



Figure 9. The dataset training process

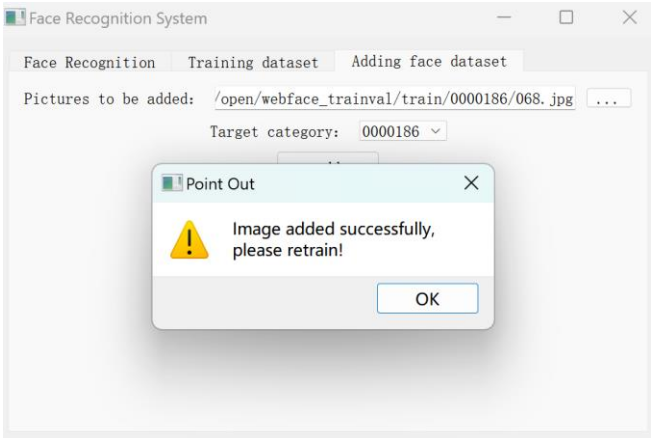



Figure 10. Add photo retraining

## VI. CONCLUSIONS

The capsule network based face recognition system consists of three main functions, which are detection of a single face image, training the face dataset, and adding data to the original face dataset. However, the most important thing to realize these is the need to understand and learn the theoretical basis of capsule network, Early knowledge of convolutional neural networks gave rise to a certain view of neural networks, of which convolution operations are crucial. Convolution operations are also present in the capsule network's back. Secondly, after a more profound understanding, through in-depth study of the working principle of the capsule neural network, determine the advantages of the capsule neural network over the convolutional neural network by understanding its structural makeup, and understand the advantages of the capsule network in the realization process. Learn in-depth about dynamic routing algorithms, how they operate inside individual capsules, and how parameters are updated inside individual capsules. Finally, the development of the capsule network is not perfect, the capsule network still exists some shortcomings, the capsule network has to continue to learn.

## REFERENCES


[1] W. L. He, M.L. Zhu. Current status and future analysis of capsule neural network research [J]. Computer Engineering and Application, 2021, 57(03):33-43.

[2] YANG Jucheng, HAN Shujie, MAO Lei et al. A review of capsule network modeling [J]. Journal of Shandong University (Engineering Edition), 2019, 49(06):1-10.

[3] Zheng Yuanpan, Li Guangyang, Li Ye. A research review of deep learning in image recognition [J]. Computer Engineering and Applications, 2019, 55(12):20-36.

[4] YANG Xiaofeng, ZHANG Laifu, WANG Zhipeng et al. Cross-domain pedestrian re-identification based on capsule networks [J]. Computer Engineering and Science, 2021, 43(09):1591-1599.

[5] JIANG Hong, JIA Shuaiyu, YAO Hongge. Capsule network for object recognition in complex realistic scenes [J]. Journal of Xi'an University of Technology, 2019.39(06):712719.DOI:10.16185/j.jxatu.edu.cn.2019.06.014.

[6] Liu Linsong, Tong Minglei, Wu Dongliang. SA-CapsNet:Self-attentive capsule network[J]. Computer Application Research, 2021, 38(10):3005-3008+3039. DOI:10.19734/j.issn.10013695.2021.03.0092.

[7] Qun Zhou. Research on hyperspectral remote sensing image classification based on capsule neural network [D]. Northern Nationalities University, 2021. DOI:10.27754/d.cnki.gbfmz.2021.000172.

[8] Yao YQ. Research on facial expression feature extraction and recognition algorithm based on capsule network [D]. Beijing Jiaotong University, 2020. DOI: 10.26944/d.cnki.gbfju.2019.000835.

[9] Lou Yue. Research on plant recognition method based on improved capsule neural network [D]. Jilin University,2021.DOI:10.27163/d.cnki.gjlnu.2020.000142.

[10] H.H. Zhang. Research and development of security system based on Caps-Net face recognition [D]. Xinjiang University, 2021. DOI:10.27429/d.cnki.gxjdu.2020.00355.

[11] Shan Chen, Rencheng Sun, Fengjing Shao et al. Research and improvement of dynamic routing based on capsule networks [J]. Computer Engineering, 2022, 48(05):208214.DOI:10.19678/j.issn.1003428.0062928.

[12] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In Advances in Neural Information Processing Systems (pp.3856-3866).

[13] Zhang, Y., Yang, J., & Davis, L. S. (2018). Capsule network performance on complex data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(7), 1552-1566.

[14] Xiang, S., Wang, Y., Liu, Z., & Gilmore, J. H. (2019). Dynamic capsule attention for visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6255-6264).

[15] Tang, H., Yu, N., Wang, R., & Wang, M. (2019). Recurrent capsule network for person re-identification. In Proceedings of the IEEE International Conference on Computer Vision (pp. 7130-7139).

[16] Lecun Y, Bottou L,et al. Gradient-based learning applied to document recognition[J]. Proceedings of IEEE, 1998, 86(11):2278-2324.

[17] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C] // Advances in neural information processing systems. 2012: 1097-1105.

[18] Deng F, Pu S, Chen X, et al. Hyperspectral image classification with capsule network using limited training samples [J]. Sensors, 2018, 18(9):22.

# A Modified Energy Enhancement in WSN Using the Shortest Path Transmission Technique

Ajaegbu Chigozirim
Department of Information Technology
Babcock University
Ogun State, Nigeria
E-mail: ajaegbuc@babcock.edu.ng

Adediran Oluwaseyi
Department of Information Technology
Babcock University
Ogun State, Nigeria
E-mail: seyiaded@gmail.com

***Abstract*—This study introduced a novel energy enhancement approach for Wireless Sensor Networks (WSNs) by leveraging the shortest path transmission technique to minimize energy consumption and extend the network's lifetime. Unlike traditional methods that heavily relied on cluster heads (CHs) for data transmission, our model proposed a non-cluster-based routing algorithm, utilizing Dijkstra's algorithm to identify the most energy-efficient paths for data transmission. Simulation results, based on varying node densities (100, 200, and 300 nodes) within a 200x200 network area, demonstrated the effectiveness of our approach. Our findings indicated a significant reduction in energy consumption, with the network lifetime extending to approximately 100,000 rounds, surpassing traditional LEACH-based and other related protocols. This enhancement not only promised a sustainable WSN deployment but also offered a scalable solution adaptable to different network sizes and configurations.**

***Keywords-Wireless Sensor Network; Network Lifetime; and Cluster Head; Shortest***

## I. INTRODUCTION

In the realm of modern electronics, sensors emerge as pivotal elements, endowed with the dual capabilities of monitoring environmental variables—such as temperature, pressure, and humidity—and communicating this data to base stations or peer devices. This dual functionality has propelled sensor technology to the forefront of advancements in fields ranging from surveillance and monitoring to the Internet of Things (IoT) and telecommunications. The intrinsic small size of sensors mandates an optimization of energy consumption to maximize their utility, a challenge that becomes even more pronounced in the configuration of Wireless Sensor Networks (WSNs).

WSNs, characterized by the interconnectivity of sensor nodes over wireless mediums, have become integral to the infrastructure of telecommunications, notably in cellular network technology. Despite the apparent simplicity of mobile devices to the end-user, these devices are complex assemblies of multiple sensors, the collective functionality of which defines the device's capabilities. This complexity underscores the importance of efficient sensor operation, particularly in terms of energy consumption, for the sustainability of WSNs.

Historical and ongoing research underscores the persistent challenges in deploying WSNs, notably the issue of energy efficiency. Traditional energy-efficient strategies, such as the Low Energy Adaptive Clustering Hierarchy (LEACH) protocol, have laid the groundwork for optimizing energy use in WSNs. LEACH and its derivatives, including concepts like MOD-LEACH and PEGASIS, have significantly advanced the field. However, these models often result in uneven energy consumption among nodes, with nodes serving as cluster heads (CHs) or relay points depleting their energy reserves at a faster rate than their counterparts. This imbalance presents a critical limitation to the operational longevity and reliability of WSNs.

In response to these challenges, our study introduces a novel approach that diverges from the cluster-based paradigms by employing a shortest path transmission technique aimed at equalizing

energy load across the network. Inspired by the energy model of [1], which sought the minimum energy route for data transmission but was limited by its cluster-based operation layout, our methodology explores the synergy between shortest path determination and energy efficiency. Unlike the traditional cluster-based approaches, our model utilizes Dijkstra's algorithm to dynamically identify the most energy-efficient transmission paths, thereby minimizing overall energy consumption and extending the network's operational lifetime.

By examining the correlation between minimum energy routes and shortest paths, this study not only addresses the pressing issue of energy efficiency in WSNs but also proposes a scalable, robust framework for future network deployments. Our approach marks a significant departure from conventional methodologies, offering a pathway to more sustainable, efficient WSN operations.

## II. REVIEW OF RELATED WORKS

The quest for energy efficiency in Wireless Sensor Networks (WSNs) has catalyzed a plethora of research, focusing on innovative routing protocols that promise to extend the operational lifespan of these networks. [2] delved into existing energy-efficient routing mechanisms, offering a comparative analysis between the Modified LEACH protocol and a Mobile sink-oriented improvement over the PEGASIS-based routing protocol. Their study, utilizing MATLAB simulations, demonstrated superior performance of the Mobile sink-enhanced PEGASIS protocol (MIEEPB) over Modified LEACH, signifying the potential of dynamic sink mobility in enhancing energy efficiency.

The paper [3] introduced the "Position Responsive Routing Protocol" for WSNs, which was benchmarked against the established LEACH and CELRP protocols. Their findings underscored significant advancements in energy efficiency and overall network performance, suggesting that positional awareness within routing decisions could substantially benefit WSN sustainability. In a novel approach to routing, [4] explored the deployment of multiple mobile sinks within clustered networks, investigating how the number of mobile sinks influences network lifetime. Their methodology, which involved segmenting the network into clusters, provided insightful data on optimizing mobile sink deployment for extended network durability. Researchers in [5], also explored the use of mobile sinks technique to improve network efficiency in WSN through the Stable Election Protocol. Their simulations outcome showed a significant improvement in network performance and energy consumption of WSN through the SEP based algorithm. In contrast to the conventional MGEAR, [6] proposed LEAG, a hybrid protocol that uses Zigbee techniques for optimized routing and energy reduction. Gateway nodes enable effective data aggregation and transmission to base stations, and their findings showed improvement in network performance and energy efficiency. Also, in [7], authors proposed an enhanced version of the MGEAR protocol intended for homogenous wireless sensor networks. The methodology optimized cluster-head selection based on energy considerations in heterogeneous wireless sensor networks (HWSNs) which increased the throughput as well as network longevity. When compared to different existing protocols, simulation results show significant gains in network lifetime and performance. Researchers in [8], explored MW-LEACH protocol presented a novel clustering hierarchy that chooses cluster leaders according to residual energy, inter-cluster distances, and the ideal member node count. Compared to existing protocols, MW-LEACH exhibited reduced complexity, faster operation, longer network lifetime, and improved fault tolerance by giving priority to nodes with high residual energy and closeness to the network center. The results of experimental evaluations show that MW-LEACH performs better than previous protocols in throughput, energy consumption, packet delivery, network longevity, and latency. In comparison to non-clustering techniques [9] emphasized the significance of clustering algorithms for improving energy efficiency in Wireless Sensor Networks (WSNs). Current approaches suffered from overhead during cluster formation and usually use periodic clustering and cluster head rotation. A unique routing protocol was suggested

to mitigate this challenge. The simulation results showed that this protocol is more energy-efficient than existing ones like LEECH and HEED, as it chooses cluster heads based on residual energy, density, and base station distance. [10] proposed a unique cluster head selection protocol that utilizes the existing cluster head to determine the subsequent leader based on a combination of residual energy and proximity metrics. This method aimed to minimize energy dissipation and enhance the network's longevity by ensuring that the most energetically viable node assumes the cluster head role. [11] study suggested the Optimal Multi-hop Path Finding Method (OMPFM), which finds effective multi-hop paths between cluster heads (CHs) and base stations (BS) to maximize power consumption and network lifetime. Through the use of pre-processing techniques and a genetic algorithm with a novel fitness function, OMPFM outperformed LEACH, GCA, EAERP, GAECH, and HiTSeC by significant margins in terms of first and last node die metrics.

Echoing the sentiment for minimum energy consumption, [1] developed a routing protocol that seeks the least energy-intensive path for data transmission from nodes to the base station. Employing the Hausdorff distance for calculating inter-cluster distances, their protocol optimized energy use across transmissions, offering a fresh perspective on energy-efficient data routing in WSNs. [12] contributed to the dialogue with an energy-balanced routing protocol that leverages the K-means++ algorithm for cluster formation and the Fuzzy Logical System (FLS) for cluster head selection. This approach not only facilitated balanced energy consumption across the network but also introduced a systematic method for cluster formation and leadership assignment, reflecting a growing trend towards algorithmic sophistication in WSN management. Previous study in [13] presented the EE_AC_DR protocol, which uses a Dijkstra Front-Back algorithm for effective data routing and scheduling, to reduce energy consumption in WSNs Large-scale simulations showed that the protocol enhanced network performance and cost-effectiveness by carefully choosing cluster heads and optimizing communication pathways, highlighting its potential to maximize WSN efficiency while consuming less energy. In order to improve security and efficiency in WSNs [14] presented the TBC-DBR method, which uses Dijkstra-based routing and trust-based clustering to provide safe data aggregation. Through simulations, it performed better than the LEACH algorithm, with reduced energy consumption and higher packet delivery rates. In [15], presented a novel method for minimizing latency and increasing network lifetime using Dijkstra's shortest path routing in conjunction with sleep-wake scheduling (DSRSS). Simulation analysis demonstrated this method's superior performance in energy reduction, making it a viable option for optimizing network longevity.

While these studies have collectively advanced our understanding and capabilities in energy-efficient WSN routing, the persistent challenge of equitable energy distribution among nodes remains a critical concern. The reviewed works primarily focus on optimizing routing protocols through cluster head selection, mobile sink deployment, and algorithmic pathfinding. However, most strategies inadvertently impose disproportionate energy burdens on certain nodes, hastening their depletion and, by extension, reducing the network's overall lifespan.

Our current investigation seeks to address this gap by proposing a modified energy enhancement model that diverges from the traditional reliance on cluster heads or fixed routing paths. Instead, it employs a shortest path transmission technique, predicated on the hypothesis that minimizing transmission distance and thereby energy expenditure can achieve a more balanced energy consumption across all nodes. This approach not only promises to extend the operational lifetime of WSNs but also introduces a scalable and flexible framework capable of adapting to varied network topologies and sizes, thus offering a significant contribution to the ongoing efforts in WSN optimization.

## III. METHODOLOGY

We structured our methodological stages as follows:

## IV. Deployment Strategy

The deployment of sensor nodes in our study is conceptualized on a geometrically structured, rectangular grid, mirroring the strategic layout akin to a football pitch. This structured approach facilitates a uniform distribution of nodes across the designated area, with the central base station positioned at the midpoint of the rectangle to ensure optimal accessibility. Such a layout is pivotal for minimizing transmission distances and ensuring uniform energy consumption across the network. Key features of our deployment strategy include:

### *A. Uniform Distribution*

Sensor nodes are evenly spaced within a rectangular grid, ensuring each node is equidistant from its neighbors, akin to the positions of players on a football pitch. This uniformity is crucial for maintaining consistent communication paths.

### *B. Central Base Station*

The base station's central location is strategic, minimizing the maximum distance any node must transmit data, thereby optimizing energy usage.

### *C. Transmission Methodology*

Contrary to traditional cluster-based approaches, our methodology does not rely on the election of cluster heads. Instead, data transmission follows a dynamic leader selection based on proximity and the shortest transmission path, significantly reducing the system's overall energy consumption. The core aspects of our transmission methodology include:

### *D. Dynamic Leader Selection*

Each node, upon its turn to transmit, calculates the shortest path to the base station. The node along this path that receives the data acts as a temporary leader, a role that dynamically shifts as the data hops towards the base station.

### *E. Dijkstra's Algorithm for Pathfinding*

The shortest path for data transmission from any given node to the base station is determined using Dijkstra's algorithm. This algorithm optimizes the route for energy efficiency, dynamically adjusting to the network's current state.

### *F. Operational Premises*

The operational framework of our study is built on several foundational premises that ensure the integrity and efficiency of data transmission within the WSN:

### *G. Fixed and Centralized Base Station*

The base station's position remains unchanged and centrally located to minimize transmission distances across the network.

### *H. Equal Energy Consumption*

A fundamental goal is ensuring that energy consumption for data transmission and reception is equitable across nodes, achievable through adherence to the shortest path strategy.

### *I. Non-Hierarchical Node Status*

All nodes operate on an equal footing, with no distinctions made for cluster heads, ensuring a democratic and energy-efficient data transmission process.

### *J. Timed Transmissions*

Data transmissions are scheduled based on time slots, allowing for organized and predictable network behavior.

### *K. Single and Multiple Hops*

Depending on a node's proximity to the base station, data can be transmitted directly (single hop) or through multiple hops, with the path determined by the shortest route algorithm.

### *L. Static Node Positioning*

Nodes are stationary, simplifying the network topology and making the shortest path calculations more predictable.

### *M. Base Station-Initiated Communication*

All communications are initiated by the base station, centralizing control and simplifying network management.

From our methodology, figure 1 illustrates the flow of data in a wireless sensor network, capturing the essential components and interactions among them. This diagram visually represents how sensor nodes collect and transmit data to the central base station, either directly or through relay nodes along the shortest energy path, based on energy-efficient routing without predetermined cluster heads.

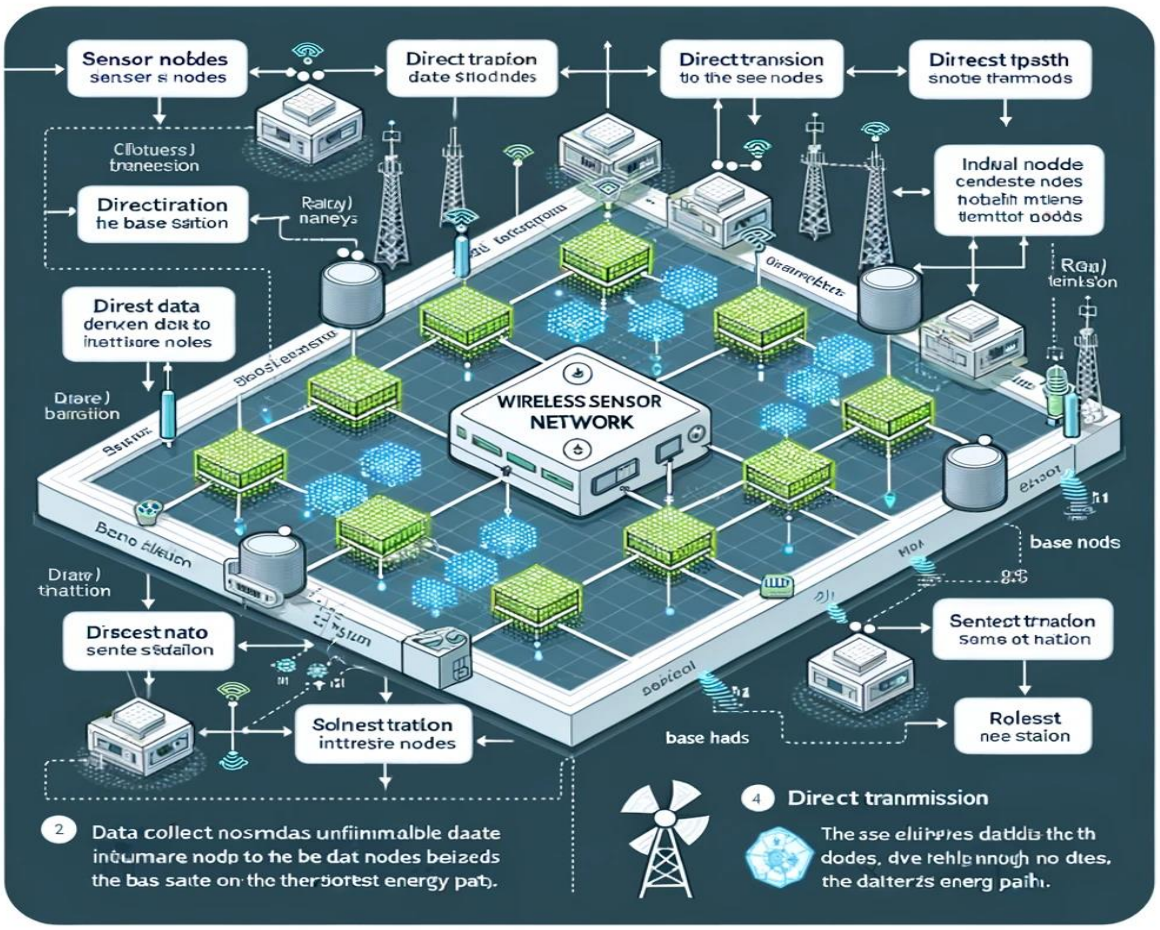



Figure 1. Model of researcher's methodological layout

## V. ENERGY MODEL

Our study advances the energy model by innovating beyond the traditional Low Energy Adaptive Clustering Hierarchy (LEACH) protocol, as outlined by Paul and Dey (2015). The cornerstone of our model is the dynamic selection of transmission paths based on the shortest distance criteria, significantly diverging from LEACH's reliance on static cluster head elections. This approach inherently democratizes the role of cluster heads, distributing energy consumption more evenly across the network and thereby enhancing network longevity.

### *A. Transmission Energy Model*

The transmission energy model is pivotal for calculating the energy expended during data transmission from a node to the base station. Unlike LEACH, where energy expenditure is concentrated around elected cluster heads, our model ensures that any node can assume the role of a temporary relay based on proximity and

optimal path selection. The model is encapsulated by the equation:

$$E_{TX} = E_{elec} + E_{PL} \quad (1)$$

where:

- $E_{TX}$ represents the total energy expended in transmission.

- $E_{elec}$ is the energy dissipated per bit to run the transmitter or receiver circuit.

- $E_{PL}$ denotes the power loss during transmission, calculated as $e_{fs} * (d_1 * d_2)$ , $e_{fs}$ representing the energy dissipated in the free-space model.

*B. Receiver Energy Model*

The receiver energy model calculates the energy consumed during data reception, essential for understanding the total energy dynamics of the network. It is expressed as:

$$E_{RX} = E_{elec} + E_{MP} \quad (2)$$

where, $E_{RX}$ is the total energy consumption for receiving data, $E_{elec}$ signifies the multipath fading channel's energy consumption, calculated as $e_{mp} * d_2^2$ , with $e_{mp}$ indicating the energy dissipated in the multipath model.

Our methodology introduces a pivotal shift by eliminating the need for cluster head elections, thereby reducing the redundancy and inefficiency associated with re-election processes. Every node, positioned within the shortest path to the base station, dynamically becomes a relay, optimizing the energy expenditure across the network. This model not only simplifies the operational mechanics but also ensures a more equitable distribution of energy consumption.

Furthermore, by integrating Dijkstra's algorithm for real-time calculation of the shortest transmission path, our energy model aligns with the operational realities of sensor networks, where maintaining energy efficiency is paramount. This alignment allows for adaptive path selection, ensuring that data transmission always follows the least energy-intensive route.

The proposed energy model underscores the importance of adaptive, path-optimized WSN operation. By systematically calculating energy consumption for potential routes and prioritizing the least costly paths, our model significantly extends the operational lifespan of WSNs. This approach not only demonstrates a considerable improvement over traditional methods but also provides a scalable framework adaptable to diverse network topologies and varying node densities.

In essence, our energy model provides a robust framework for enhancing WSN energy efficiency, highlighting the shift from clustered dependencies to a more fluid, path-optimized network operation. This advancement promises significant implications for the design and deployment of future wireless sensor networks, prioritizing sustainability and operational efficiency.

Assumptions:

- A fixed number of $N$ sensor nodes are uniformly distributed in a rectangular area.
- All nodes have an initial energy $E_{init}$ .
- The base station is centrally located.
- Nodes use a free space or multipath model for energy dissipation during transmission and reception, depending on the distance to the next hop.
- There are no cluster heads; instead, nodes relay data based on the shortest energy path.

*C. Modified Energy Model*

The energy dissipated for a node to transmit a k-bit message over a distance $d$ is given by:

$$\begin{cases} E_{elec} \cdot k + e_{fs} \cdot k \cdot d^2 & if \quad d < d_0 \\ E_{elec} \cdot k + e_{mp} \cdot k \cdot d^4 & otherwise \end{cases} \quad (3)$$

Hence the energy dissipated to receive this message is:

$$E_{RX}(k) = E_{elec} \cdot k \quad (4)$$

Also the Network Lifetime Model is given as $N$ is the total number of sensor nodes in the network, $E_{init}$ is the initial energy of each sensor node, and $E_{consumed}(i)$ is the energy consumed by node $i$ in one round of communication which includes both transmitting and receiving energy.

Thus the network lifetime T can be modelled as the minimum number of communication rounds that any node can participate in before depleting its energy:

$$T = \min_{i \in \{1,\ldots,N\}} \left( \frac{E_{init}}{E_{consumed}(i)} \right) \quad (5)$$

$$E_{consumed}(i) = E_{TX}(i) + \sum_{J \in P_{i \to BS}} E_{RX}(j) \quad (6)$$

Where, $E_{TX}(i)$ is the energy consumed by node $i$ to transmit data, $E_{RX}(j)$ is the energy consumed by node $j$ to receive data, and $P_{i \to BS}$ is the set of nodes along the shortest path from node $i$ to the base station (BS), excluding the transmitting node $i$.

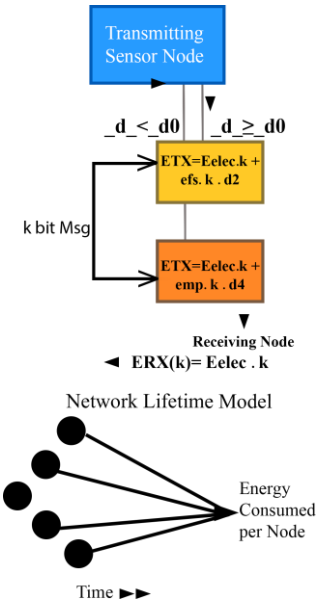



Figure 2. Energy Dissipation Model

## VI. ALGORITHM: ENERGY-EFFICIENT DIJKSTRA'S ALGORITHM

Input: A graph G(V, E) represented by an adjacency matrix where each edge weight E_ij is the energy cost of transmission from node i to node j, and a source node s.

Output: The shortest paths and their energy costs from the source node s to all other nodes in V.

```
Procedure DijkstraEnergyBased(G, s)
  Initialize energyCosts[] := {∞, ..., ∞} with size |V|
  Initialize prevNode[] := {null, ..., null} with size |V|
  Initialize visited[] := {false, ..., false} with size |V|
  Set energyCosts[s] := 0

  While there exists a node u in V that is not visited
    Select u such that energyCosts[u] is minimum and
visited[u] is false
    Set visited[u] := true

    For each neighbor v of u in V
      If visited[v] is false and E_uv > 0
        Set tempCost := energyCosts[u] + E_uv
        If tempCost < energyCosts[v]
          Set energyCosts[v] := tempCost
          Set prevNode[v] := u
        EndIf
      EndIf
    EndFor
  EndWhile

  For each node v in V
    Initialize path[] := empty list
    Set current := v

    While prevNode[current] is not null
      Insert current at the beginning of path[]
      Set current := prevNode[current]
    EndWhile

    If path is not empty
      Insert s at the beginning of path[]
    EndIf

    Output the path from s to v and its total energy cost
energyCosts[v]
  EndFor
EndProcedure
```

## VII. ALGORITHM EXPLANATION

### A. Initialization

`energyCosts[]`: An array holding the cumulative energy cost from the source node `s` to every other node. It is initialized with infinity ($\infty$) to represent that at the start, the cost to reach any node is unknown and assumed to be very high.

`prevNode[]`: An array that tracks the immediate predecessor of each node on the path from the source. It is used to reconstruct the shortest path once the algorithm completes. Initialized with `null` to indicate no predecessors have been determined yet.

`visited[]`: A boolean array indicating whether a node has been visited and its minimum energy cost has been determined. Initially, all nodes are unvisited.

### B. Algorithm Process

*1)* Selecting the Node with Minimum Energy Cost: At each iteration, the algorithm selects the unvisited node `u` with the smallest known energy cost from the source. Initially, this will be the source node itself, as its energy cost is zero.

*2)* Updating Neighbor Costs: For each neighbor `v` of the selected node `u`, if `v` is unvisited and the edge `E_uv` (representing the energy cost from `u` to `v`) is greater than zero, the algorithm calculates a temporary cumulative energy cost (`tempCost`) from the source to `v` via `u`.

- If `tempCost` is less than the current known cost to reach `v`, the algorithm updates `energyCosts[v]` with `tempCost` and records `u` as `v`'s predecessor in `prevNode[v]`.

*3)* Path Reconstruction: Once all nodes have been visited, the algorithm reconstructs the shortest path for each node by tracing back through the `prevNode[]` array. Starting from each node `v` and moving through its recorded predecessors, the path is built in reverse until the source node is reached.

- The path for each node, along with its total energy cost (`energyCosts[v]`), is output by the algorithm.

### C. Relation to Energy Cost Considerations

The edge weights `E_uv` in your network graph represent the energy costs associated with transmitting data from one node to another. In the context of your study, these costs are calculated based on the distance between nodes and the energy dissipation model (free space or multipath depending on the distance).

By selecting paths that minimize these energy costs, the algorithm aligns with the goal of your study to ensure energy-efficient routing in the WSN. It ensures that data is transmitted along the routes that will conserve the most energy, extending the operational lifetime of the network.

Since the algorithm finds the path with the minimum energy cost to each node, the first node to deplete its energy will define the network lifetime `T`, as per the study's scope. This node's energy expenditure rate, dictated by the paths chosen by the algorithm, will directly impact the overall network lifetime.

The pseudocode for the Energy-Efficient Dijkstra's Algorithm is designed to find the most energy-efficient paths in a WSN. This is crucial for your study's objective to maximize the network's operational lifetime while ensuring that data is relayed to the central base station in the most energy-conserving manner possible.

The detailed explanation of the algorithm's steps, with emphasis on energy cost considerations, demonstrates the algorithm's appropriateness for your study and its potential for extending the network's lifetime, which is a key metric for the performance and sustainability of WSNs.

## VIII. SIMULATION PARAMETER

The simulation framework is designed to systematically evaluate the performance of the proposed energy model under varying conditions. Key parameters are defined as follows:

Number of Nodes: Simulations were conducted with node sets of 100, 200, and 300 to assess scalability and performance under different network densities. This variation allows us to explore the model's adaptability to networks of varying sizes.

Network Area: The network is modeled within a 200m x 200m square area. This dimension provides a balanced field for assessing the effectiveness of the shortest path algorithm across different distances and densities.

Channel Type: A wireless channel is utilized, reflecting the operational environment of real-world Wireless Sensor Networks (WSNs). This choice ensures the simulation reflects practical constraints, such as signal attenuation and interference.

Source Node Configuration: Among the nodes, 99 are designated as sensor nodes equipped with a UDP agent, simulating a typical WSN scenario where multiple nodes collect and transmit data to a central node or sink.

Number of Cluster Heads: Although our model moves away from traditional cluster head elections, for comparative analysis, a baseline of 4 cluster heads was established. This parameter is crucial for benchmarking against LEACH-based models.

Antenna Model: An Omni Antenna model is adopted for all nodes, facilitating uniform signal propagation in all directions. This choice simplifies the simulation and aligns with common WSN deployments.

Interface Queue Type: The Queue/Drop Tail/PriQueue mechanism is employed to manage data packets at the node level. This queuing model allows for realistic simulation of packet handling, prioritization, and potential congestion scenarios.

Initial Energy of Sensor Nodes: Each node is initialized with an energy reserve of 5 Joules, setting a uniform starting point for energy depletion studies. This parameter is pivotal for analyzing the network's operational longevity.

Time for Each Round: The simulation progresses in rounds, each lasting 5 seconds. This temporal framing facilitates the observation of energy consumption patterns and network behavior over time.

Transmission Power: Set at 50 picojoules (pj), this parameter dictates the energy cost of data transmission, a critical factor in evaluating the energy efficiency of the proposed model.

Receiving Power: The energy cost for receiving data is set at 10 picojoules (pj), allowing the differentiation between transmission and reception energy expenditures.

Transmission Range: A maximum range of 40 meters for data transmission is established, determining the reach of each node's signal. This range affects the calculation of the shortest path and the selection of relay nodes.

## IX. Analysis and Result

Our simulations of the Wireless Sensor Network (WSN), employing the shortest path technique and Dijkstra’s algorithm with a load-balancing strategy, have provided significant insights into the distribution of energy consumption and workload across the network. The results, illustrated in Figures 1, 2, and 3, underline the efficacy of our proposed model in enhancing network performance and energy efficiency.

Figure 4 showcased a uniform grid layout simulating the strategic distribution of cellular towers or small cell base stations within an urban setup. This configuration optimized coverage with minimal overlap and coverage gaps, ensuring that each node had the opportunity to transmit directly to the centrally located base station or relay through adjacent nodes efficiently. This model demonstrated that optimal placement and routing, based on the shortest path principles, could significantly reduce operational costs and improve sustainability by minimizing transmission power. The findings suggest that such a grid layout, by facilitating strategic infrastructure planning, allows for enhanced coverage and service quality, making it an essential blueprint for the deployment of cellular networks in urban environments.

The simulation graph in Figure 5 depicted a Wireless Sensor Network with a minimum energy path for data transmission from sensor nodes to the central base station. The visualization of the most energy-efficient route, indicated by the green line, highlighted how distance and energy cost minimization are critical for preserving sensor nodes' limited energy resources. This dynamic routing capability is vital for prolonging the

operational life of each sensor node and, consequently, the entire network. The simulation emphasized the importance of a deliberate planning strategy to ensure uniform node distribution, thereby guaranteeing coverage, connectivity, and multiple potential routing paths. This insight is invaluable for real-world WSN deployment, suggesting that a uniform grid layout significantly contributes to consistent connectivity and efficient routing protocol implementation.

Figure 6 presented a plot of the minimum energy level of nodes over time, measured in simulation rounds. The graph illustrated a steady, linear decline in energy levels, culminating in the depletion of energy reserves after approximately 100,000 rounds. This decline indicated a consistent energy consumption rate among the network's most demanding nodes and defined the network's lifetime by the point at which the first node's energy was exhausted. The initial setting of 2 joules for node energy levels and the observed network lifetime underscored the critical nature of managing energy consumption within cellular networks to ensure sustainability and cost-effectiveness. Moreover, the graph advocated for the necessity of load balancing across the network to extend the operational lifetime, suggesting strategic traffic routing through less congested cells as a viable energy conservation strategy.

The simulation results have provided profound insights into the energy dynamics and operational efficiencies achievable within a WSN through the implementation of shortest path routing and load-balancing strategies. These findings not only demonstrate the potential for substantial improvements in network longevity and reliability but also offer a predictive model for energy management and sustainability within cellular network deployments. For cellular operators, these insights are instrumental in ensuring network reliability and optimizing energy usage, contributing significantly to the ongoing efforts in the development and optimization of energy-efficient WSNs for various applications.

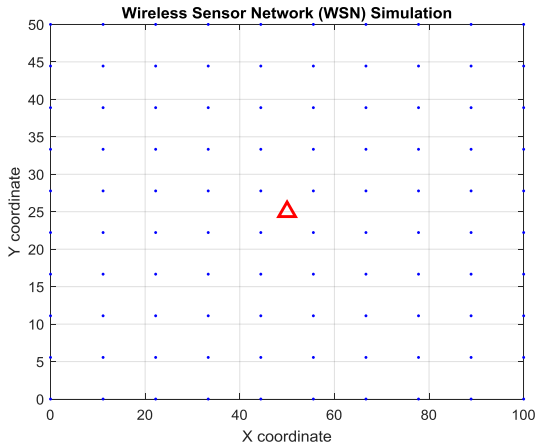



Figure 3. Wireless Sensor Network (WSN) Simulation

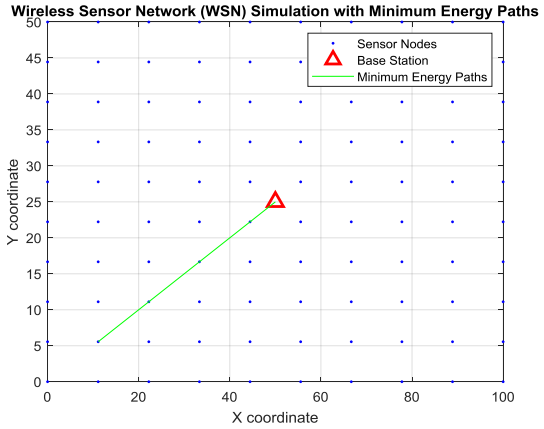



Figure 4. Wireless Sensor Network Simulation with Energy Paths

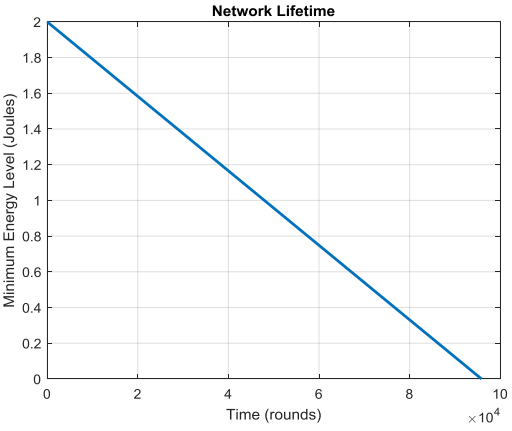



Figure 5. Network Lifetime

## X. COMPARATIVE EVALUATION OF MINIMUM ROUTING ENERGY PATH

Network Topology and Energy Consumption: Our study introduces a paradigm shift in the

design of Wireless Sensor Networks (WSNs), transitioning from the conventional ad-hoc network topology, wherein nodes serve both as data collectors and routers, to a more streamlined model. This novel approach simplifies the network structure, allowing nodes to communicate directly with the base station or via a singular relay. Such a configuration markedly reduces the incidence of node failure, diminishes the potential for signal obstruction, and curtails the overall energy expenditure inherent in multi-hop transmissions.

Advancements over Previous Models: Contrary to the ad-hoc WSN models, where energy consumption patterns were erratic and unpredictable due to the variable number of hops and the dynamic nature of cluster head elections, our model demonstrates a systematic reduction in energy usage. By leveraging a direct or single-relay communication protocol, the network achieves a significant decrease in operational energy demand. This is evidenced by a more gradual decline in energy levels across the network, as depicted in our simulation results.

Sustainability and Network Longevity: The efficiency of our proposed model is further exemplified by the extended network lifetime. Traditional models often reported a steep increase in the number of non-functional (dead) nodes over time, directly impacting network reliability and data integrity. In contrast, our simulations exhibit a controlled, linear reduction in energy levels (Figure 3), indicating a sustainable usage pattern and prolonged network operability.

Predictability and Uniformity in Energy Consumption: A notable advantage of our approach is the predictability and uniformity in energy consumption among the nodes. Previous studies highlighted a rapid depletion of energy reserves in nodes serving as cluster heads, leading to uneven energy distribution and shorter network lifetimes. Our methodology circumvents this issue by distributing the energy load evenly across the network, ensuring that no single node bears a disproportionate burden of the energy expenditure. This not only enhances the network's overall energy efficiency but also contributes to a more balanced and equitable operational framework.

## XI. Implications for WSN Deployment

The implications of our findings are multifaceted, extending beyond theoretical advancements to practical applications in WSN deployment and management. By demonstrating a viable alternative to ad-hoc and cluster-based topologies, our study paves the way for the development of more energy-efficient, reliable, and sustainable WSNs. Such networks are crucial for a wide array of applications, including environmental monitoring, smart cities, healthcare, and industrial automation.

In summary, our comparative evaluation underscores the superiority of our proposed model in terms of energy efficiency, network longevity, and operational reliability. By adopting a simplified communication structure and optimizing the routing mechanism, we present a compelling solution to the perennial challenges of WSN design and implementation. This study not only contributes to the existing body of knowledge but also sets a new benchmark for future research in the domain of wireless sensor networks.

## XII. Conclusions

The study successfully demonstrated the potential of a modified energy enhancement model in WSNs through the implementation of the shortest path transmission technique. By moving away from the conventional cluster-based routing protocols and employing an energy-efficient Dijkstra's algorithm, we observed a substantial improvement in the network's energy conservation and operational longevity. The simulation results clearly indicated a more gradual and uniform energy depletion across the network, thereby ensuring a predictable and extended network lifetime. This approach not only addressed the inherent challenges of energy consumption in WSNs but also provided a scalable and efficient framework for future deployments in various applications. Our study contributed to the ongoing efforts in optimizing WSNs, offering a viable solution that could significantly impact the design

and implementation of energy-conscious wireless sensor networks in real-world scenarios.

## REFERENCES


[1] Paul, S., & Dey, T. (2015). Energy efficient routing in cluster based wireless sensor network. 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Pg. 1-5.

[2] Warrier, M., & Kumar, A. (2016), Energy efficient routing in Wireless Sensor Networks: A survey. International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Pg. 1987-1992.

[3] Zaman, N., Jung, L., & Yasin, M. (2016). Enhancing energy efficiency of wireless sensor network through the design of energy efficient routing protocol. Hindawi Publishing Corporation. Journal of Sensors, Pg. 1-16.

[4] Zhong, P., & Ruan, F. (2018). An energy efficient multiple mobile sinks based routing algorithm for wireless sensor networks. IOP Conference Series: Materials Science and Engineering, 323(1), 012029.

[5] Wang, J., Zhang, Z., Xia, F., Yuan, W., & Lee, S. (2013). An energy efficient stable election-based routing algorithm for wireless sensor networks. Sensors, 13(11), 14301–14320.

[6] Revathi, A., & Santhi, S. (2019). Energy Consumption Based Low Energy Aware Gateway (LEAG) Protocol in Wireless Sensor Networks. International Journal of Engineering and Advanced Technology.

[7] Benelhouri, A., Idrissi-Saba, H., & Antari, J., (2022). An Improved Gateway-Based Energy-Aware Multi-Hop Routing Protocol for Enhancing Lifetime and Throughput in Heterogeneous WSNs. Simulation Modelling Practice and Theory, Volume 116.

[8] Khediri, S., Khan, R., Nasri, N., & Kachouri, A. (2020). MW-LEACH: Low energy adaptive clustering hierarchy approach for WSN. IET Wireless Sensor Systems, 10(3), 126–129.

[9] Haswani, N., & Deore, P. (2018). Cluster Head Selection using Optimised Round Policy in Wireless Sensor Networks. 6th International Conference on Recent Trends in Engineering & Technology.

[10] Rahama, M., Hossen, M. & Rahman, M. (2016). A routing protocol for improving energy efficiency in wireless sensor networks. 3rd International Conference on Electrical Engineering and Information Communication Technology. Pg. 1-6.

[11] Mohammed, A., Mohammed, A., Tat-Chee, W., & Zakaria, A. (2019). Energy efficient multi-hop path in wireless sensor networks using an enhanced genetic algorithm. Information Sciences, Volume 500.

[12] Lin, L., & Donghui, L. (2018). An energy-balanced routing protocol for a wireless sensor network. Journal of Sensors, Pg. 1-12.

[13] Boualem, A., Ayaida, M., Dahmani, Y., de Runz, C., & Maatoug, A. (2019). A New Dijkstra Front-BackAlgorithm for Data Routing-Scheduling via Efficient-Energy Area Coverage in wireless Sensor Network. 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 1971–1976.

[14] Salsabil, A., Raghda, S., Irene, S., & Tawfik I. (2023). Integrated Trust-Clustering and Dijkstra Routing Algorithms for Energy-Efficient WSNs. International Conference on Telecommunications.

[15] Thomas, S., Gayathri, K., & Raj, A. (2017). Joint design of Dijkstra's shortest path routing and sleep-wake scheduling in wireless sensor networks. International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS).

# Personalized Recommendation Multi-Objective Optimization Model Based on Deep Learning

Zepeng Yang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1241169825@qq.com

Ping Lu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1565364293@qq.com

Pingping Liu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1341369601@qq.com

***Abstract*—Recommended in this paper, because the existing single objective experience is poor, and the recommended model in a large difference of targets under the complex relationship of joint optimization and the conflict caused by faults, this paper proposes a personalized recommendation based on the deep learning multi-objective optimization algorithm, the estimated probability of users on the individual behavior as a model to study target, Multiple objectives are integrated into a model for learning. Firstly, the embedding layer is used to change the feature vectors, so that the bottom layer of the model shares the same feature embedding. Secondly, the factorization machine and deep learning are used to construct high-low order feature interaction. Then, the gating network and multilevel expert network constructed by a fully connected neural network are used to learn the characteristic relationship of user behavior. Finally, make connections between goals. Through experiments on public and real datasets, The results show that the multi-objective model proposed in this paper has better co-optimization performance and increases the AUC value by 0.1% compared with advanced personalized recommendation models such as MMoE and ESMM, to achieve the ultimate goal of increasing the prediction accuracy and improving user satisfaction.**

***Keywords-Recommendation Algorithm;Multi-Objective Optimization; Post-Click Conversion Rate; Deep Learning; The Neural Network***

## I. INTRODUCTION

With the arrival of the big data era and the swift advancement of smart devices, personalized recommendations play a significant role in a variety of applications. Recommendation systems often use estimation models that target users' clicks and do not take sufficient account of the behavior generated by users after clicking, thus trapping users in smaller and smaller interest networks, reducing user engagement and satisfaction, resulting in the uneven development of the recommendation ecosystem and declining corporate interests [1]. Therefore, it has become a trend to apply multi-task learning to simulate both user satisfaction and engagement for multi-objective optimization [2].

In recent years, numerous studies have been conducted on recommendation problems that require the simultaneous optimization of multiple objectives but existing recommendation algorithms have the following problems: 1) Sample data is sparse. In general, users rarely rate items, which also lead to overt data being too sparse, so implicit information needs to be used for recommendations. In traditional CVR estimation models, positive and negative samples are usually extremely unbalanced, which increases the difficulty of model training and poses

DOI: 10.2478/ijanmc-2024-0005

generalisation problems [3]. 2) Sample selection bias. Traditional pCVR estimation uses a technique similar to CTR estimation that is, training by clicking on a subset of samples, and inferring the entire display sample space when reasoning. However, this method has the problem of sample selection bias. 3) Multi-objective "seesaw" phenomenon. Some multi-objective models, while improving some objectives, tend to sacrifice the performance of others. One of the main optimization issues in multi-objective learning comes from different target gradients that tend to clash with each other in ways that are not conducive to progress. In some cases, this collision gradient can cause a significant decrease in performance.

Currently, many large-scale recommendation systems both domestically and internationally have implemented multi-task learning with deep neural network models [4]. Researchers pointed out that multi-objective model can use regularisation and transfer learning to improve the model's predictions for all objectives. However, experimental results show that in fact multi-objective Multi-objective models do not consistently exceed their single-objective counterparts across all objectives. Deep learning-based multi-objective models often exhibit high sensitivity to factors such as distribution of data and variations in relationships between targets. The inherent conflicts brought about by target differences can impair the prediction of at least some targets, especially when the parameters in the model are widely shared among all targets.

Therefore, this paper proposes a multi-objective optimization recommendation algorithm that uses deep learning technology to fuse user behavior information, which can better use the prior knowledge in shared network design to capture complex task correlations [5].

This paper presents deep learning-based multi-objective network architecture for personalized recommendation for the sequencing phase of the recommendation system. On the basis of a shared underlying model, the model proposes to use factor decomposer and deep learning to construct higher-order and lower-order feature interactions, and then introduce a separate gating network for each target, and then introduce a multi-level expert network for the model [6]. In addition, it introduces ESMM to optimize how the loss function is constructed, allowing for more accurate fitting of various conversion rates.

This article applies this model architecture to video recommendations as a case study: using the user's past viewing habits as a basis, recommend the videos you want to watch later. The experiments set up two classification tasks and conducted a large number of offline experiments to evaluate the effectiveness of the model, and the results show that it is helpful to evaluate the significant improvement of the index in this prediction task.

## II. RELATED WORK

### *A. Research Overview*

The recommendation model in this paper learns based on feedback from two types of users: (1) participatory behaviors, such as clicking and watching; (2) Satisfying behaviors, such as sharing, commenting, and collecting. Given the historical behavior information of each user, the ranking system takes user characteristics, video content features and historical behavioural features are used as inputs and learn to predict multiple user behaviours. We formulate the sorting problem as a classification task and compute the cross-entropy loss. [7]. Given user characteristics and video characteristics, ranking models predict the probability that users will take actions such as clicks, watch time, shares, and comments.

After identifying multiple ranking objectives and their problem types, multitask ranking models can be trained for these prediction tasks. For each candidate, take these multiple predictions as inputs and output the combined score using a combinatorial function in the form of weighted multiplication to achieve best performance in terms of user engagement and user satisfaction [8].

### *B. Related Knowledge*

Cheng et al. proposed a Wide & Deep Learning model by using multi-source heterogeneous data such as user characteristics, situational characteristics, and project characteristics [9]. As

shown in Figure 1, this model combines the training of a broad linear model (on the left side of the figure) and a deep neural network (left side of the figure) to ensure a balance between the ability of model to memorise and generalize. Guo et al. based on Wide & Deep, combined with factorization machine and deep learning, proposed a factorization-Machine based Neural Network ($Deep_{FM}$) for click-through rate prediction, using factorization machine and deep neural network to model low-level and high-level feature interactions, respectively, compared with Wide & Deep [10]. $Deep_{FM}$ does not require manual feature engineering.

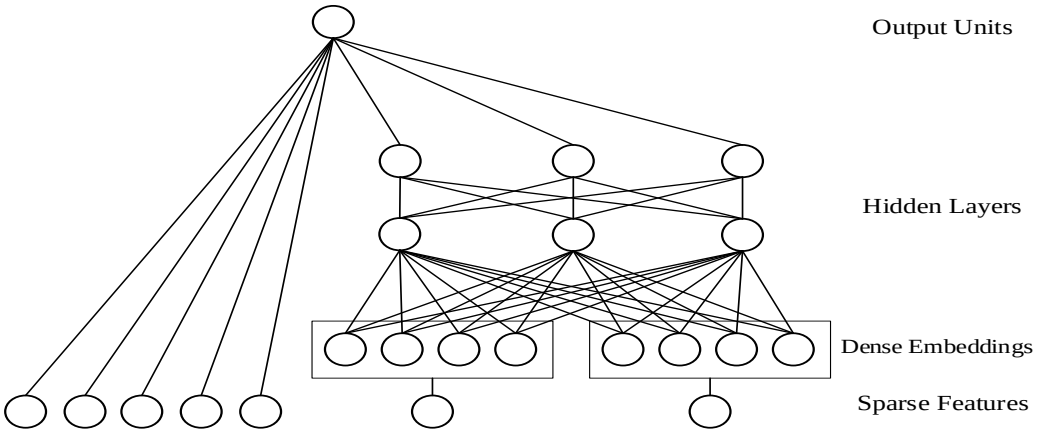



Figure 1. Model structure of the Wide & Deep Learning model.

The shared-substrate multitasking model was proposed in 1998 and is shown in Figure 2. In which the model structure is characterized by the fact that all targets share the same input, and because the underlying parameters are shared by all targets, the risk of overfitting is greatly reduced [11]. At the same time, different goals can also transfer knowledge through these shared parameters when learning, and use the knowledge learned by other goals to help their own goals learn. This model is often regarded as a iconic benchmark approach in multi-objective modelling.

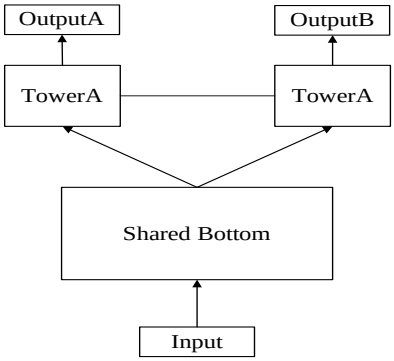



Figure 2. Shared-Bottom model.

Input features in the Web domain are often discrete and sparse, and the interaction between features is critical to effectively model this type of data [12]. In the various multi-objective models of deep learning, it is common to use a model structure that shares underlying parameters. This model structure reduces the risk of overfitting and facilitates the learning of the target because the underlying parameters can be shared by all targets. However, when the correlation between targets is relatively low, this hard parameter sharing will limit the freedom of each target fit, impairing multi-objective learning.

### *C. FM_Shared Expert Multi-Objective Dependency Model（FSMD）*

Figure 3 illustrates the overall modelling framework results, and earning feature relationships for user behaviour using fully connected neural networks, and using noise reduction auto-coding to initialize user behavior information. An efficient multi-objective neural network architecture was designed, which extended the Wide & Deep model and adopted a multi-objective learning model architecture with a

mixture of multiple experts. In addition, a shallow tower was introduced to model and eliminate selection bias, and an ESMM way of constructing loss functions was introduced to establish a connection between targets. In addition, a multi-objective sorting model based on deep learning is formed to integrate user behavior characteristics for interest recommendation.

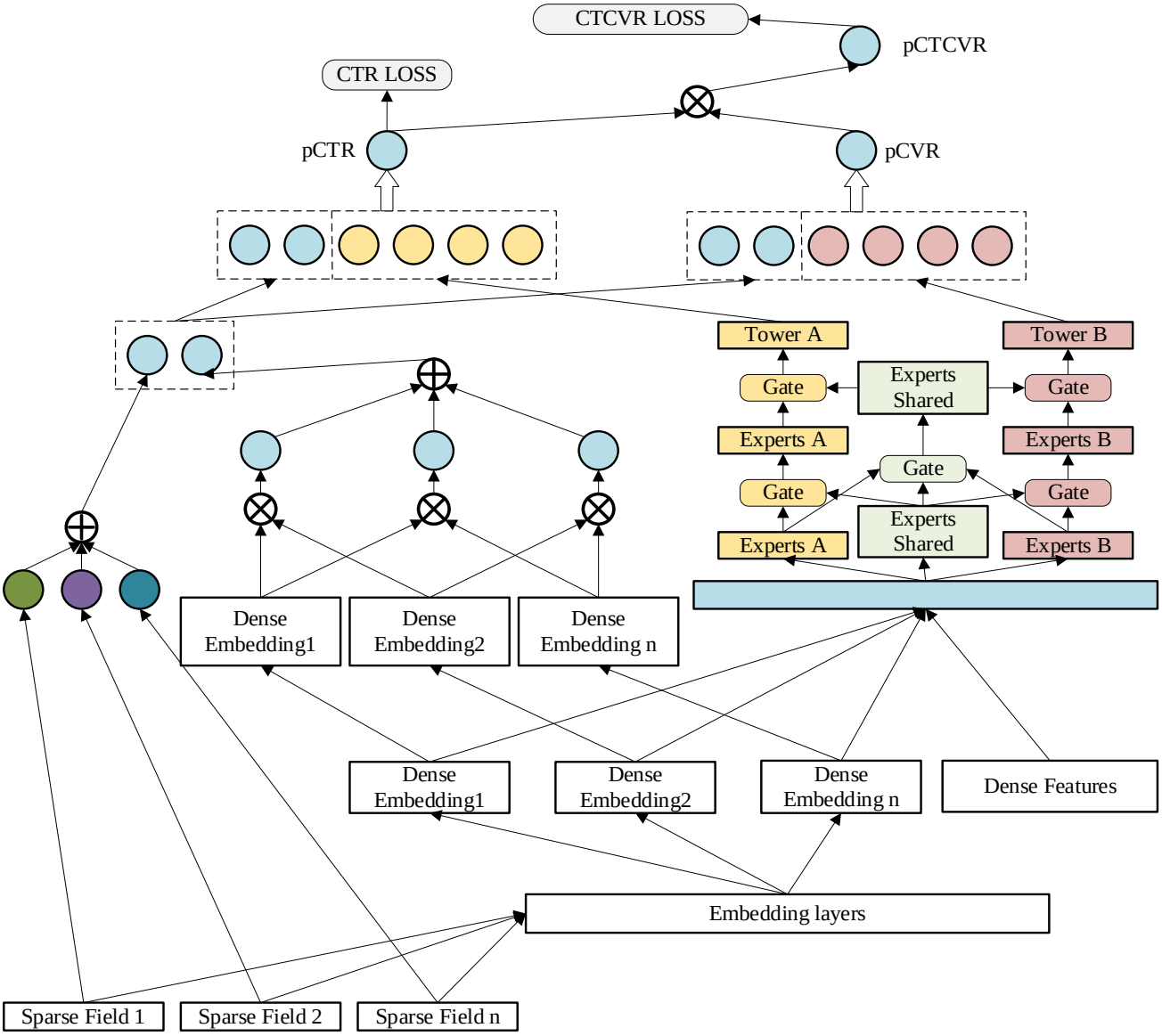



Figure 3. FM_Shared Expert Multi-Objective Dependency (FSMD) Mode

## III. Modeling Methods

### A. Bottom Sharing of High and Low Level Feature Interaction

The idea taken in this paper is to combine the factorization machine MLP [13], first use the factorization machine to model the pairwise interaction between features, and then further model the higher-order feature interactions by adding a fully connected layer. To take full advantage of this technology of $Deep_{FM}$ , the research at this stage chose to build a multi-objective model based on $Deep_{FM}$ , Low-level and high-level feature interactions are modelled using factorial decomposers and deep neural networks, respectively.

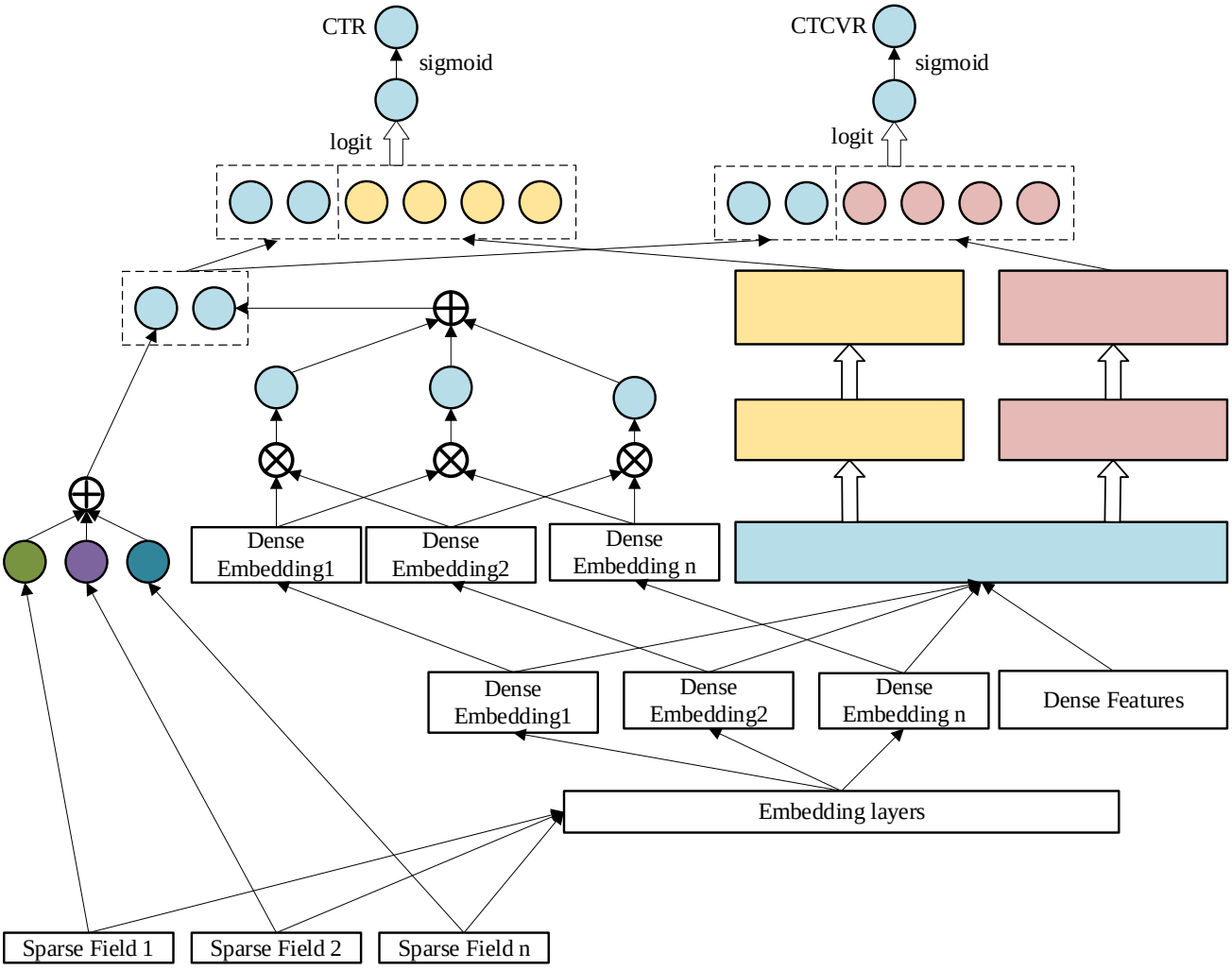



Figure 4. Multi-objective Base Model.

In this phase, the two goals of click and watch time are modeled. Leave the FM part of $Deep_{FM}$ unchanged, replace the DNN part of $Deep_{FM}$ with the Share Bottom structure of hard parameter sharing, and obtain a multi-objective model combining $Deep_{FM}$ and Share Bottom Model as the baseline for the study of the multi-objective model.

As shown in Figure 4, the FM subnetwork on the left calculates the second-order crossover fraction of sparse features and dense features, and the deep subnetwork on the right stitches dense features and continuous features into the network. Finally, the FM first-order, second-order fractions, and the last layer of deep inputs are stitched together, and the estimated value is obtained by sigmoid.

The model predicts the following: (1)

$$\hat{y} = sigmoid(y_{Deep} + y_{FM}) \tag{1}$$

Among them, $y\hat{} \in (0,1)$ is the predicted CTR, $y_{FM}$ and $y_{Deep}$ are the outputs of the FM component and the output of the deep component, respectively.

$$y_{FM} = < w, x > + \sum_{j_1=1}^{d} \sum_{j_2=j_1+1}^{d} < V_i, V_j > x_{j_1}, x_{j_2} \tag{2}$$

Where $x = [x_{field1}, x_{field2}, ..., x_{fieldj}, x_{fieldn}]$ is a vector of d-dimension, $x_{field}$ is the vector

representation of the $j_{th}$ domain of X. For feature i, the importance of the 1st order is measured by a scalar $w_i$, and the impact of its interaction with other features is measured by a latent vector $v_i$. The addition unit <w,x> in the network architecture functions to capture the significance of the 1st-order feature, while the inner product element signifies the impact of the 2nd-order feature interaction. This approach allows for the consideration of both individual feature importance and their interactions, enhancing the model's ability to make personalized and accurate recommendations.

$$y_{deepk} = h^k(f(x)) \tag{3}$$

The output result of the shared hidden layer, f(x), is input to the respective tower network (subnetwork) $h^k$, and finally, each target k gets an output $y_{deepk}$. $\hat{y} = sigmoid(y_{FM} + y_{Deep})$

*B. Gated Network Adaptive Weighting*

Because of the shortcomings of the Share Bottom model structure, Google proposed the Multi-gated Mixture of Experts (MMoE) model in 2018 [14], which introduces multiple expert subnetworks and gating structures, and uses different expert combinations to learn different goals through gating so that each goal can be better learned. As shown in figure 5 is a schematic diagram of the MMoE model structure.

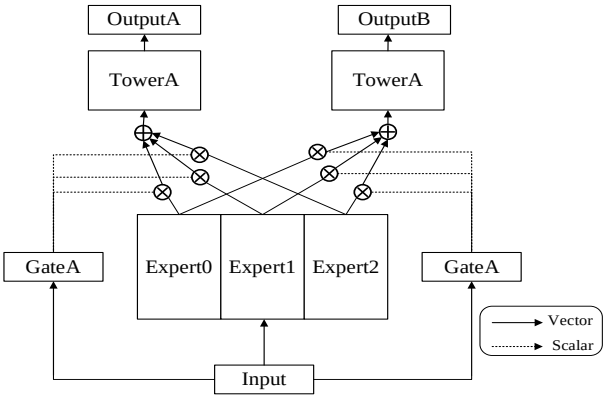



Figure 5. Multi-gate MoE Model.

The design of this stage is influenced by the Multi-Task Mixture of Experts (MMoE) architecture, which involves incorporating a distinct gated network $g^k$ for each target k. These networks are added to the deep part of the model, building upon the framework established in the previous stage. This approach allows for the creation of specialized networks for individual targets, enabling the model to effectively capture the intricacies and nuances specific to each task. g is the gating network that combines the results of the experts, the internal implementation of the gating is composed of the same multilayer perceptron with ReLU activation, and the gating network is a simple linear transformation of the input with the softmax layer, as shown in figure 6:

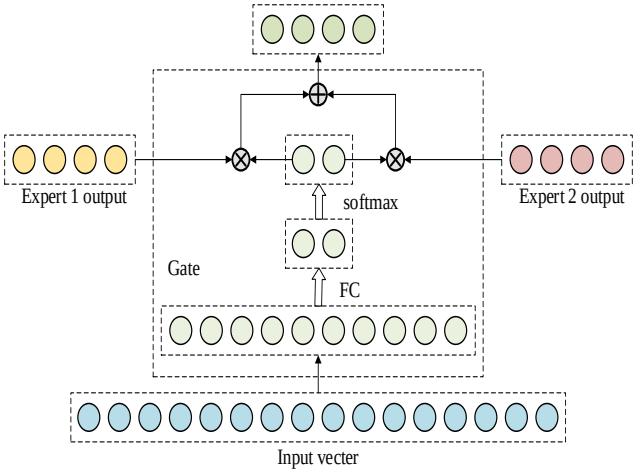



Figure 6. Internal Structure of Gated Network.

The input vector and the output vector of each expert will be passed into the gating network, the input vector will first pass through MLP, and the last layer of softmax will get the weight of each expert, and the output of the gating is the weight on all experts:

$$g^k(x) = soft\max(W_{hk}x) \tag{4}$$

Where $W_{gk} \in R^{nxd}$ is the trainable matrix, n and d denote the number of experts and the edge dimension, respectively.

Multiple expert networks are added at the same time as the introduction of a gating network for

each goal, and the gating network learns different combinations of the expert network for their respective tasks, and the output of the expert network is adaptively weighted. For a task, the output of its corresponding network of experts is:

$$f^k(x) = \sum_{i=1}^{n} g^k(x)_i \, f_i(x) \tag{5}$$

Where $f_i$ (i=1,…,n) is a network of n experts.

The new multi-objective model based on the gated network is shown in Figure 7, and the improved advantage is that each target can train a gated network individually, and the weight of each expert network owned by each target task is adjusted according to the objective adaptively. Get the output of the deep part target k as:

$$y_k = h^k(f^k(x)) \tag{6}$$

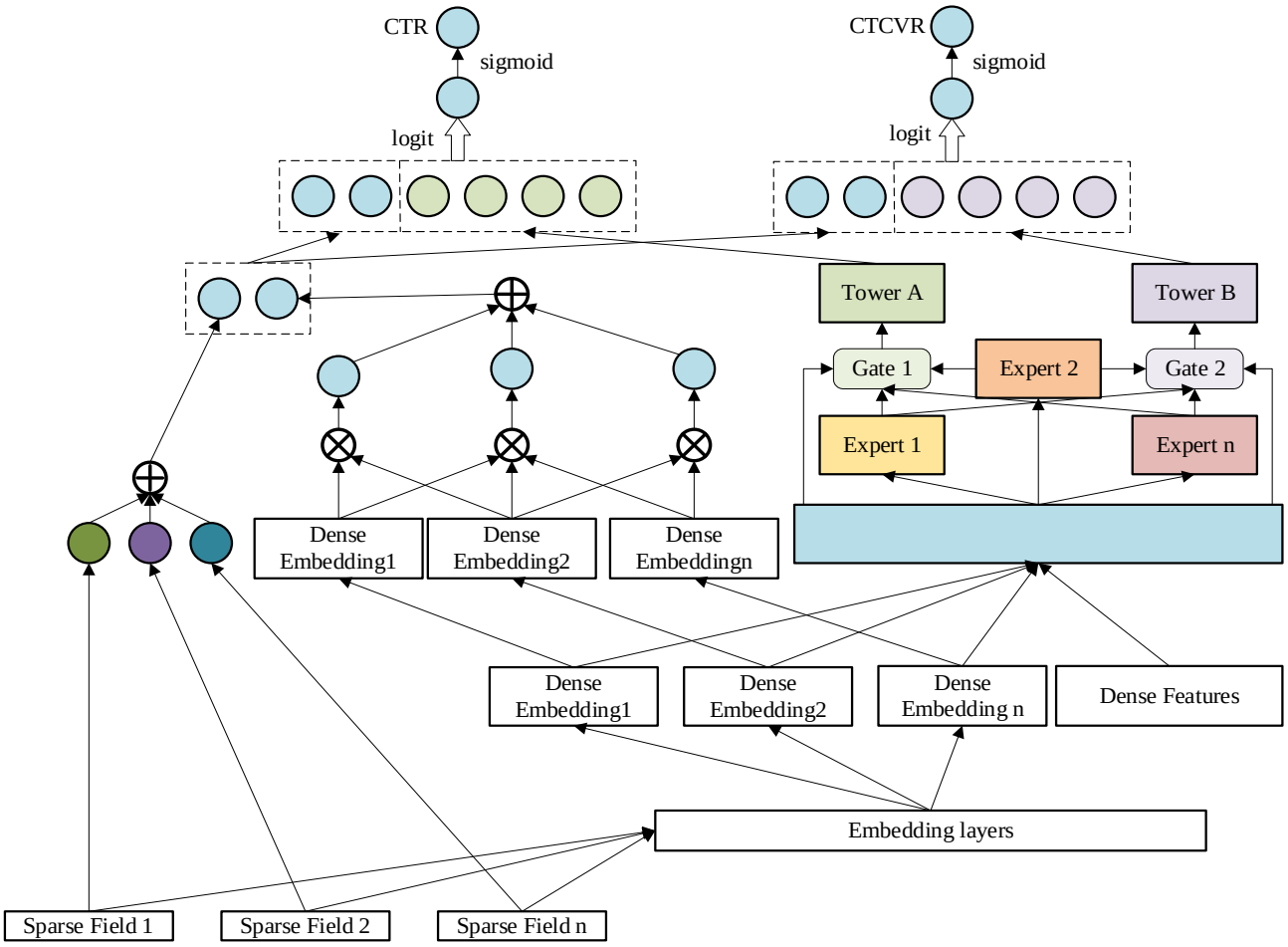



Figure 7. FM_Gate (FG) Model.

In this stage of the model, the target uses the binary classification cross-entropy to make losses, and then the loss weights of the two targets are summed to obtain a total loss function, and the model parameters are solved by optimizing this total loss function. The total loss function is as follows:

$$L = \min_{\theta} w_1 \sum_{i=1}^{N} L_1(y_i, P_{ctr}(x_i, \theta)) + w_2 \sum_{i=1}^{N} L_2(z_i, P_{cvr}(x_i, \theta)) \tag{7}$$

Among them, $L_1$ and $L_2$ are the loss functions of fitting CTR and CTCVR, respectively, and both are binary classification cross-entropy; $x_i$ indicates the input feature; $y_i$ is the click target, click is 1, exposure unclicked is 0; $z_i$ is the conversion goal, converted to 1, click not converted to 0; $P_{ctr}(x_i,\theta)$ is an estimate of CTR, $P_{cvr}(x_i,\theta)$ is an estimate of CVR, θ is the model parameter, $w_1$ and $w_2$ are the weights of the two losses, and N represents the total quantity of samples.

### *C. Multi-level Expert Network*

Multi-objective modeling often has a seesaw phenomenon, usually, multi-objective learning relative to multiple single-objective learning models can improve the effectiveness of a part of the goal, but some multi-objective models often at the expense of the performance of other targets to improve some goals, this problem is called seesaw phenomenon. One of the primary challenges in multi-objective learning arises from the gradients of diverse objectives, which tend to clash with each other in a way that is not conducive to progress, and in some cases can lead to a significant decrease in performance.

To solve the "seesaw" problem that multiple objectives are prone to, a multi-level expert network is introduced based on the previous FG model, and an independent expert network is established for each target while retaining a shared expert network [15]. Introducing multi-level experts, which consist of shared experts and specific target experts, helps mitigate harmful parameter interference and enables the integration of multi-objective features through gated networks. Implementing a novel progressive separation approach facilitates the emulation of interactions among experts, leading to more effective knowledge transfer between intricate and interconnected objectives. Get FM_Sharing Expert (FS) model to solve the seesaw problem, ensure stable optimization, and the deep part of the model is shown in the figure 8.

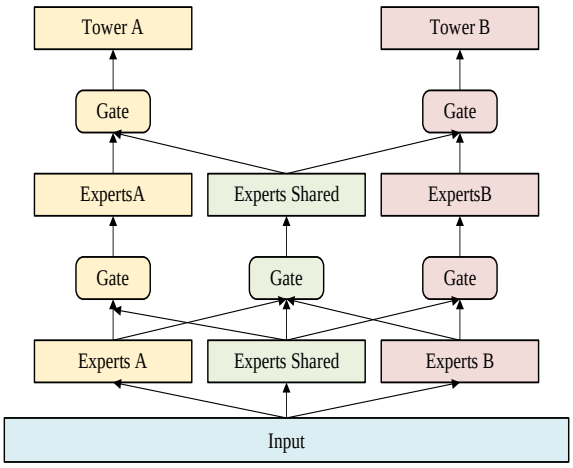



Figure 8. FM_Sharing the deep part of the Expert (FS) Model.

Definition of the gated network in the $j_{th}$ extraction network of the kth sub-target in the FS model deep section:

$$g^{k,j}(x) = w^{k,j}\left(g^{k,j-1}(x))S^{k,j}(x)\right) \qquad (8)$$

Where $w^{k,j}$ the weight is function of the target k as the input to $g^{k,j-1}$, and $S^{k,j}$ is the selection matrix of the $j_{th}$ extraction network for the target k. After calculating all the gating networks and expert networks, the final output of the $k_{th}$ sub target of the deep part of the FS model is:

$$y^k(x) = t^k(g^{k,N}(x)) \qquad (9)$$

### *D. Build Target Dependencies*

In a recommendation scenario, the user's behavior is generally more than one, and the different behaviors occur in order and dependencies. Each of a user's behavior can be a target in a multi-objective model, and there are dependencies between these targets. For this correlated multi-objective model, if each target fits independently, the information about the dependencies between the targets will be lost, and the accuracy of the model will be lost, affecting the sorting effect. Therefore, when doing correlation multi-objective modeling, it is necessary to model every step of the

transformation in user behavior. In the case of video recommendations, for example, in this scenario, the goal we need to model is clicked and playtime and the conversion relationship involved in these two goals can be described as: showing the video to the user——the user clicks on the video——the viewing time exceeds a certain threshold.

Use x to represent the characteristics of the user and the video; y indicates the label of the click, y=1 indicates the click, and y=0 indicates that the exposure is not clicked; z indicates the label of the playback duration, z=1 indicates that the playback time exceeds the threshold, and z=0 indicates that the threshold has not been exceeded, including no clicks. The conversion relationship and probability quantification of exposure, clicks, and duration in user behavior can be expressed as follows [16]:

$$\underbrace{p(y=1,z=1|x)}_{pCTCVR} = \underbrace{p(y=1|x)}_{pCTR} * \underbrace{p(z=1|y=1,x)}_{pCVR} \tag{10}$$

To do this, we combined the loss function of the previous version of the multi-objective model and the ESMM to obtain a multi-objective model as shown in the figure 9.

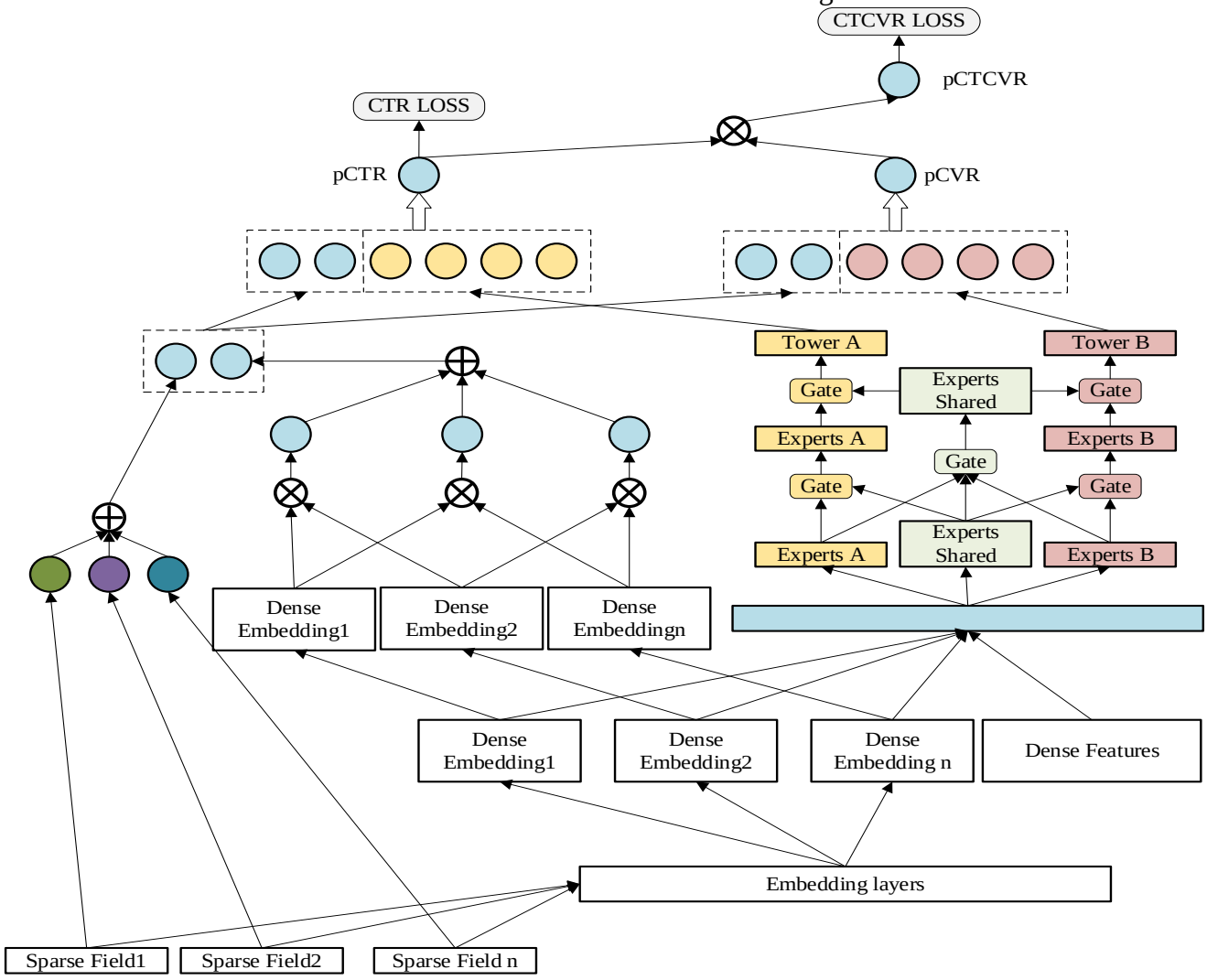



Figure 9. FM_Shared Expert Multi-Objective Dependency(FSD)Model.

Finally, the CTR loss and CTCVR loss weighted sum give a total loss, and the model parameters are solved by minimizing the total loss.

$$L = \min_{\theta} w_1 \sum_{i=1}^{N} L_1(y_i, P_{ctr}(x_i, \theta)) + w_2 \sum_{i=1}^{N} L_2(zi, P_{ctr}(x_i, \theta) * P_{cvr}(x_i, \theta)) \quad (11)$$

Among them, $L_1$ and $L_2$ are the loss functions that fit CTR and CTCVR, respectively, and both are two-classification cross-entropy; $y_i$ is the class label of the click; $z_i$ as a class indicator for the length of playback (1 for longer playback than the threshold, 0 for otherwise), $P_{ctr}(x_i, \theta)$ is an estimate of CTR; $P_{cvr}(x_i, \theta)$ is an estimate of CVR; θ is the model parameter; $w_1$ and $w_2$ are the weights of the two losses respectively, with N being the total number of samples.

## IV. EXPERIMENT

### A. UCI Census Dataset

#### 1) The dataset description

This experiment constructed a multi-task learning problem with multiple features as prediction targets, which used the UCI Census income dataset:

*Goal one:* whether the forecast revenue exceeds $50,000;

*Goal two:* to forecast if the individual is married.

In this data set, there are 42 characteristics, including important information such as age, job type, education, occupation, ethnicity, etc., 199523 training examples, and 49881 test examples.

#### 2) Experimental settings

Since both goals were binary classification problems, the experiment used the AUC score as an evaluation indicator. The income task is the primary task and the marital status task is a secondary task. Each model uses the same hyper parameters, and the parameter settings are shown in the following table. Every model is trained on the training dataset using identical parameter initialization, and the findings are then presented for the test dataset.

TABLE I. PARAMETER SETTINGS

| Parameter Name | Value |
|---|---|
| batch_size | 256 |
| optimizer | adam |
| learning_rate | 0.001 |
| embedding_size | 4 |
| dnn_layers | (512, 256) |
| dnn_use_bn | True |

*Batch_size:* the number of samples used to calculate the gradient, which in this chapter is set to 256;

*Optimizer:* The optimizer for parameter optimization of the constructed network model, in this experiment, the selected optimizer is Adam;

learning_rate: learning rate set to 0.001;

embedding_size: Used for the Dense Embedding layer, combined with the value characteristics of each feature, the corresponding embedding_size is set;

dnn_layers: Represents the number of neurons in the hidden layers of the feed-forward neural network;

dnn_use_bn: is a Boolean value that controls whether to use the BN layer, and its value is set to True;

In the experiment, the task was a binary classification task trained using cross-entropy loss and evaluated with AUC.

#### 3) Experimental Results

TABLE II. RESULTS OF THE UCI CENSUS INCOME DATASET

| Models | AUC/ Income | AUC/ Marital | Mean |
|---|---|---|---|
| Single-Task | 0.9198 | 0.9748 | 0.9473 |
| Shared-Bottom | 0.9148 | 0.9754 | 0.9451 |
| MMoE | 0.9152 | 0.9756 | 0.9454 |
| PLE | 0.9161 | **0.9764** | 0.9463 |
| Base | 0.9134 | 0.9706 | 0.9420 |

| Models | AUC/ Income | AUC/ Marital | Mean |
|---|---|---|---|
| FG | 0.9146 | 0.9693 | 0.9420 |
| FS | **0.9216** | 0.9756 | **0.9486** |

From the respective AUC and average data of the two targets in the results, it can be concluded that the FS model can improve the AUC of the first target by 0.0055 without significantly reducing the AUC of the second target compared with the PLE and optimize the average AUC of the two targets. It can be concluded that the FS model that combines the high order and the low order is better than the PLE model with only the deep part, thus verifying the fusion effect of the interaction between the low order and the high order features. From the comparison of the model base and FS model, the first goal is improved by 0.0082, the second goal is increased by 0.005, and the average AUC of the two goals is increased by 0.0066.

### *B. The Video Site Plays the Dataset*

#### *1) The dataset description*

The dataset used in study is the user log of a video website for 15 consecutive days, including user characteristics, video content characteristics, and user historical behavior data. The data includes user dimensions, video dimensions, and user historical behavior data, which are described separately in these three dimensions.

User-side attributes information: user ID, age range, gender, province or city, city, city level, and device type.

Video side attributes information: video ID, video age, video month, video rating, and video duration.

User behavior information: user ID, video ID, whether to play, whether to share, whether to favorite, whether to comment, watch time, play tag, watch date.

#### *2) Experimental results*

*a)* Comparative experiments

The multi-objective model constructed in this paper is compared with the single-objective model, the classic multi-objective model, and the multi-objective model designed in the previous period, and the model effect is analyzed according to the rating index AUC. Each model in the model comparison experiment uses the same parameters, and the parameter settings are shown in the following table.

TABLE III. PARAMETER SETTINGS

| Parameter Name | Value |
|---|---|
| batch_size | 2048 |
| optimizer | adam |
| learning_rate | 0.001 |
| embedding_size | 4 |
| dnn_layers | (256, 128) |
| dnn_use_bn | True |
| dropout | 0.5 |

The chart below shows the test results after 30 rounds of training.

TABLE IV. PARAMETER SETTINGS

| Models | AUC/ Income | AUC/ Marital | Mean |
|---|---|---|---|
| Single-Task | 0.7192 | 0.6635 | 0.6914 |
| DeepFM | 0.719 | 0.658 | 0.689 |
| Shared-Bottom | 0.7184 | 0.6916 | 0.705 |
| ESMM | 0.7189 | 0.6897 | 0.7043 |
| MMoE | 0.7201 | 0.7057 | 0.7129 |
| Base | 0.7193 | 0.7086 | 0.714 |
| FG | 0.7201 | 0.7084 | 0.7143 |
| FS | 0.7204 | 0.7114 | 0.7159 |
| FSMD | **0.7205** | **0.7117** | **0.7161** |

Table shows the prediction performance of various models on the video dataset. The results show that the model proposed in this paper significantly outperforms all baseline models for the transformation goal. Due to the complex correlation between click goals and duration goals, the seesaw phenomenon can be observed from the results, with some models improving click goals but hurting duration goals, and others improving duration goals but hurting click targets. Specifically, the baseline model that combines FM and deep improves both goals compared to the single-target model, but the improvement is not significant, while the FS model with gating and expert networks but does not establish a connection between the modular targets only improves the click target, but damages the

duration goal. Compared with the typical and widely used multi-objective models MMoE and ESMM, this model has a much greater improvement over the duration target and a small improvement on the click target. Finally, this model converges at a similar rate and achieves significant improvements on the above model with one of the AUCs.

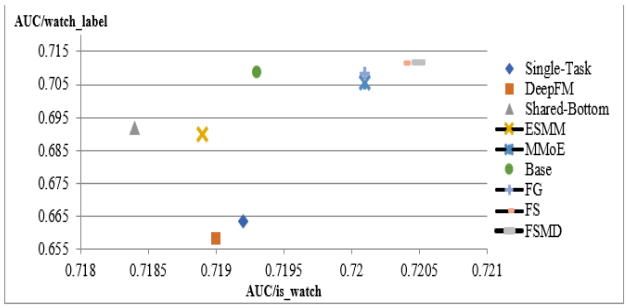



Figure 10. The seesaw phenomenon in each model under complex target association

There is a complex relationship between click targets and duration targets, so modeling two targets at the same time will make the "seesaw phenomenon" more obvious. As can be seen from the figure 10, with the Base model as the baseline zero point, only FS and FSMD are surpassed in the two targets at the same time, and the other models have obvious seesaw phenomenon, only the FSMD model designed in this paper achieves the optimal at the same time.

*b)* Ablation experiment

This experiment compares the AUC of the designed multi-objective model under different network layers. In the experiment, the FSMD model of the two-layer underlying network, the two-layer tower network, the one-layer gating network, and the 8-expert network is selected as the skeleton network and baseline of the ablation experiment, and different network layer combinations are modified on the skeleton network for training and evaluation, and the final ablation results are shown in the table:

In the first ablation experiment , the number of layers of the gated network is set to 2 layers, the number of expert networks is set to 8, and the underlying network is set to (256, 128), comparing the AUC under different tower network shapes.

The experiment tested four structurally different tower network shapes: constant, incremental, decreasing, and diamond. When changing the shape of its network, the number of layers of hidden layers is fixed. For example, when the number of hidden layers is 3, then the four different shapes are constant (128-128-128), increasing (64-128-256), decreasing (256-128-64), and diamond (64-128-64).

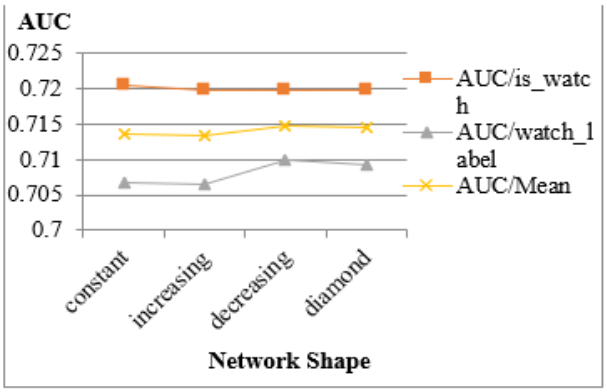



Figure 11. AUC comparison of network shapes.

It can be concluded from the results, as shown in figure 11, that the diamond network is higher than other network shapes in both the conversion target and the average AUC of both goals based on not significantly reducing the AUC of the click target. It can therefore be concluded that the diamond is the optimal choice in the choice of tower network shape.

In the ablation experiment, set Tower_mlp_dims to (256,128) and Bottom_mlp_dims to (256,128), the number of experts is 8, comparing AUC under different gating network layers.

It can be concluded from the result, as shown in figure 12, that when the number of layers of the gating network is 3, the model achieves the highest number of layers compared to the duration target and the average AUC of the two targets. It can be concluded that the number of layers in the gating network should not be too small, nor too much, the effect of 3 or 4 layers is better, considering the complexity of the model, it is considered that 3 layers are the best choice.

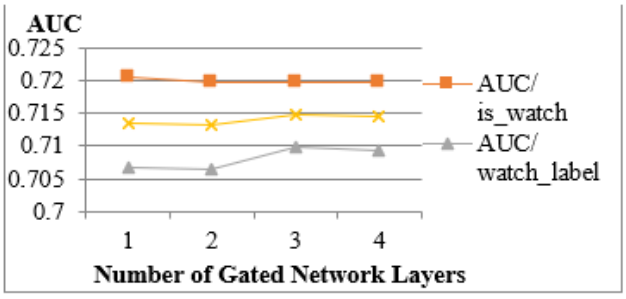



Figure 12. AUC comparison of the number of layers in a gated network.

The third ablation experiment, set Tower_mlp_dims to (256,128), Bottom_mlp_dims (256,128), and the number of gated network layers is 1, comparing AUC under different numbers of expert networks.

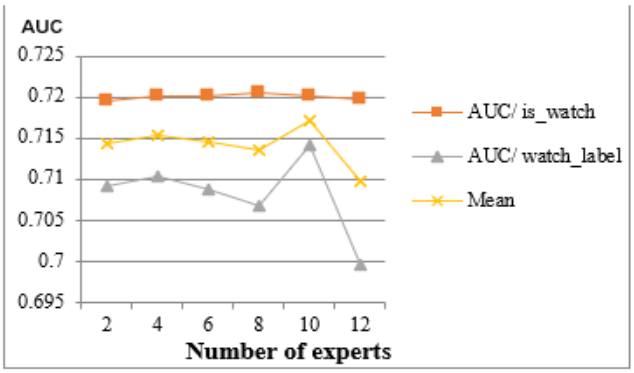



Figure 13. AUC comparison of the number of experts.

As can be seen from the figure 13, a change in the number of expert networks does not have much impact on the click target, but it changes a lot on the duration goal. When the number of expert networks rose from 2 to 10, the conversion goal improved significantly, and the model reached the highest number of experts compared to the number of other experts in terms of duration target and average AUC of both goals. As the number of expert networks continues to increase, it can be seen that there is a significant decrease in the AUC of the duration target, so the more expert networks the better.

## V. Conclusions

In this article, some of the real-world challenges of current recommendation systems are first described, including sparse sample data, selection bias implicit in user feedback, and the phenomenon of "seesaws." To address these challenges, a multi-objective optimization ranking model based on deep learning is proposed and applied to the question of recommending what videos to watch next. To effectively optimize multiple ranking targets, the multi-expert hybrid model architecture is extended, and an effective method is built to reduce and model the bias of selecting multi-objective models by using soft parameter sharing and combining high- and low-level feature interactions and multi-level expert networks. In addition, through experiments on different datasets, it can be concluded that the model proposed in this paper is a significant improvement over the existing single-objective model for all objectives in all target groups, concluding that the model in the multi-objective case shows the benefits of facilitating goal co-operation and preventing negative migration and see-saw phenomena. Therefore, it is confirmed that the multi-objective model based on deep learning designed in this paper shows greater advantages in improving the shared learning efficiency of different scale target groups, and the technology we propose has achieved substantial improvement in participation and satisfaction indicators.

## References

[1] Zhou G, Mou N, Fan Y, et al. Deep interest evolution network for click-through rate prediction. AAAI 2019, 33: 5941-5948.

[2] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In Computer Vision and Pattern Recognition, 2018.

[3] Chen C,Meng X,Xu Z,et al.Location-aware personalized news recommendation with deep semantic analysis.IEEE Access,2017:173-182.

[4] Wang R, Fu B, Fu G, et al. Deep & cross network for ad click predictions. ADKDD 2017: 1-7.

[5] LeCun Y, Bengio Y, Hinton G. Deep Learning. Nature, 2015, 521(7553): 436-444.

[6] Song W, Shi C, Xiao Z, et al. Autoint: Automatic feature interaction learning via self-attentive neural networks. CIKM 2019: 1161-1170.

[7] Chen Q, Zhao H, Li W, et al. Behavior sequence transformer for e-commerce recommendation in Alibaba. Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data. 2019: 1-4.

[8] Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1871–1880.

[9] Cheng H T,Koc L,Harmsen J,et al.Wide & deep learning for recommender systems//Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.Boston,USA,2016:7-10.

[10] Guo H,Tang R,Ye Y,et al.DeepFM:A factorization-machine based neural network for CTR prediction//Proceedings of the 26th International Joint Conference on Artificial Intelligence.Melbourne,Australia,2017:1725-1731.

[11] Rich Caruana. 1998. Multitask learning. In Learning to learn. Springer, 95–133.

[12] Chen C,Meng X,Xu Z,et al.Location-aware personalized news recommendation with deep semantic analysis.IEEE Access,2017:173-182.

[13] Rendle S.Factorization machines//Proceedings of the 2010 IEEE 10th International Conference on Data Mining.Sydney,Australia,2010:995-1000.

[14] Jiaqi Ma,Zhe Zhao,Xinyang Yi,Jilin Chen,Lichan Hong,Ed H. Chi. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts[P]. Knowledge Discovery & Data Mining,2018: 1930-1939.

[15] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In Fourteenth ACM Conference on Recommender Systems (RecSys '20). Association for Computing Machinery, New York, NY, USA, 269–278.

[16] Xiao Ma et al. “Entire Space Multi-Task Model: An Effective Approach for Estimating Post-Click Conversion Rate” International ACM SIGIR Conference on Research and Development in Information Retrieval (2018).

# Research on Simulation Approximate Solution Strategy for Complex Kinematic Models

WenJing Qu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: quwenjing170@163.com

Zhongsheng Wang
State and Provincial Joint Engineering Lab. of Advanced Network, Monitoring and Control
Xi'an Technological University
Xi'an, 710021, Shaanxi, China
E-mail: wzhsh1681@163.com

***Abstract*—In order to meet the needs of military, road construction, multimedia industry and other aspects, UAVs are gradually given more functions. As the basic function of UAV applications, the fixed-point delivery problem model has higher and higher accuracy requirements. However, in the actual scene, the UAV delivery problem is affected by the interaction of various factors such as flight height, air resistance, and dive angle, which makes it difficult to achieve high stability and high hit accuracy. This paper will analyze the complex motion model based on the fixed-point delivery of explosives by UAV, study the relationship between the stability of UAV delivery and the hit accuracy, and analyze the influence of relevant parameters on the problem by using modeling. In this paper, a multivariate nonlinear continuous time change model is proposed, and a continuous time slice discretization idea operation model is introduced to approximate the time slice splitting inside the UAV launch motion. Secondly, the design quantified evaluation index reaction the initial velocity of the explosive, the launch Angle, the height off the ground and other parameters to analyze the model. Finally, the best scheduling strategy is obtained and verified by using the idea of variable traversal and trial-and-error simulation. The experimental results show that the variation of UAV flying height, speed, depression and other interference factors is consistent with the prediction of score and hit accuracy, according to the environment setting of this model, when the UAV is 300 meters above the ground and 290 meters away from the target horizontal position, the delivery speed is 250m/s, and the pitch angle is about 27°, the fixed-point delivery of explosives is the best strategy.**

***Keywords-Nonlinear Model; Traversal Search; Trial And Error Simulation; Continuous Time Discretization***

## I. INTRODUCTION

Thanks to the rapid development of science and technology, the functions of all kinds of unmanned intelligent machines are increasingly rich, and more and more industries have put unmanned intelligent machines into social production and daily use. Among them, UAVs (Unmanned Aerial Vehicle) are widely used in military, special industries, consumer and other fields to achieve fixed-point delivery tasks due to their strong performance and minimal restrictions on terrain roads [1]. Because the UAV is affected by gravity and external environmental factors during flight, and the artificial control method is not easy to maintain the stability of the fuselage for a long time, resulting in the failure to accurately achieve the fixed-point delivery task [2]. Therefore, it is urgent to study the UAV fixed-point delivery task and improve its precision.

The stable delivery of UAVs is a combination problem [3], which needs to consider the flight characteristics and requirements of UAVs, the demand for materials and the environmental factors on site [4]. The operational stability of the UAV will also affect the completion of the mission. In this paper, the discretization execution and ergodic analysis model of continuous time nonlinear problems with multivariate variables are constructed, and the discretization simulation model of continuous system is designed to solve the continuous time state problem. Then a trial-and-error simulation model is built, and all possible values are traversed through the trial-and-error mechanism, and the relationship between

DOI: 10.2478/ijanmc-2024-0006

variables is analyzed [5]. Finally, a multivariate weighted evaluation model and a random disturbance reaction simulation test model are proposed to evaluate the relationship between stability and hit conditions by using scores and Gaussian random probability number perturbations, and to design corresponding scheduling strategies.

## II. Relevant Research

UAVs have shown significant strategic and practical value in the field of geological exploration and mining as well as the analysis of blasting movement in the military field. In geological exploration and mining, combined with the application of machine vision, intelligent algorithms and sensor technology, drones can achieve a high degree of monitoring and evaluation of blasting operations, improve the safety and efficiency of operations, and promote the digital and intelligent transformation of mining blasting operations. This provides a sustainable solution for environmental protection and resource utilization. In the military field, the motion analysis of the targeted delivery of explosives by UAS achieves highly intelligent and accurate strikes by integrating technology such as machine learning, visual navigation and intelligent countermeasures system, which improves the adaptability and survival ability of UAS in complex battlefield environments [6]. At the same time, the research on path planning algorithm, military ethics and compliance with international regulations also provides scientific support for the safe and efficient operation of UAV targeting explosives.

In the aspect of motion analysis of UAV targeted explosive delivery, the literature review deeply discusses the flight trajectory and motion characteristics of UAV in dynamic environment, as well as the important influence of external factors on stability and accuracy [7]. Through the comparison and evaluation of different technologies, the key factors to improve the hit accuracy are revealed, which provides an important theoretical basis and technical support for improving the combat effectiveness. In the future, with the continuous development of technology, UAV blasting motion analysis will be more combined with artificial intelligence, big data and other technologies to achieve more accurate and efficient management and control of blasting operations [8], and promote continuous innovation and development in the field of military science and technology and geological exploration.

## III. Experimental model and design

The delivery task of the UAV is set as "delivering explosives", that is, launching spherical explosives into the corresponding target through the launching cylinder installed at the front end of the UAV. Since the launch speed of the explosives is very large, even larger than the flight speed of the UAV, when measuring the distance, the actual initial speed of the explosive should be calculated by adding the speed of the UAV and the launching speed of the explosive relative to the UAV. For the explosive, these two do not need to be split. Since the explosion of explosives has a certain spread range, the UAV needs to maintain a certain distance from the target point [9], but it can not be too far away and lead to launch deviation [10].

When the explosive is launched, the launch port carried by the drone directly powers the explosive to obtain a large initial speed. After leaving the launch port, the drone device no longer exerts any force on the explosives, so the explosives are only subjected to the effect of vertical downward gravity, vertical upward air buoyancy, and air resistance in the opposite direction of movement [11].

### *A. Discretization strategy*

In calculus theory, integrating a continuous curve of change can be divided into several small pieces, each of which is treated as a rectangle for discrete computation. When the width of each piece is small enough, the discrete rectangular synthesis is transitioned into a continuous curve integral.

In the computer control algorithm, time slice rotation is a classic scheduling strategy, which controls the scheduling process according to the unit time. Each time slice either repeats a certain

operation or executes the corresponding operation for an individual process.

According to the idea of calculus and computer time slice rotation scheduling algorithm, the continuous physical process of variable force and variable velocity is decomposed into several discrete time slices [12], each time slice is regarded as a simple constant force kinematic process. An appropriate and sufficiently small time slice length is selected, and all processes are connected in series and accumulated to obtain the final approximate result.

### *B. Monte Carlo replacement tree search*

In the calculation of the model, the drop distance *L*, flight height *H*, flight speed $v_0$ and air resistance *Fz* of the UAV can be calculated by the accumulation of the above modeling process, in which the air resistance changes with time and speed in a fixed scene, one of the other three variables is set as a variable according to the form of Monte Carlo tree search, and the other two are related to each other. As shown in Figure 1. The fixed and constant parameters are called quantification, the first variable is called the independent variable, and the second variable is called the dependent variable. Air resistance is named as a variable.

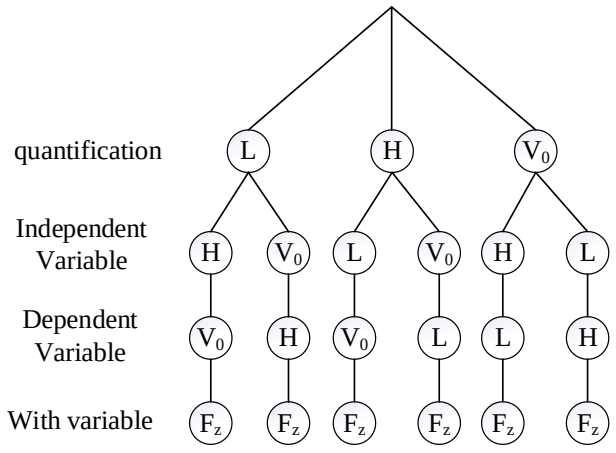



Figure 1. Monte Carlo tree search strategy model

The whole process of the model is based on the mechanical analysis of the interaction and the motion analysis [13]. According to the setting of the time slice, the motion state in each time slice is approximately regarded as the uniformly variable motion, which is convenient to calculate the relevant variables. The process in the time slice is shown in the following formula, and the flow chart is shown in Figure 2.

$$\begin{Bmatrix} v \\ L \end{Bmatrix} \rightarrow \theta \rightarrow F \rightarrow a \rightarrow \begin{Bmatrix} v' \\ L' \end{Bmatrix} \tag{1}$$

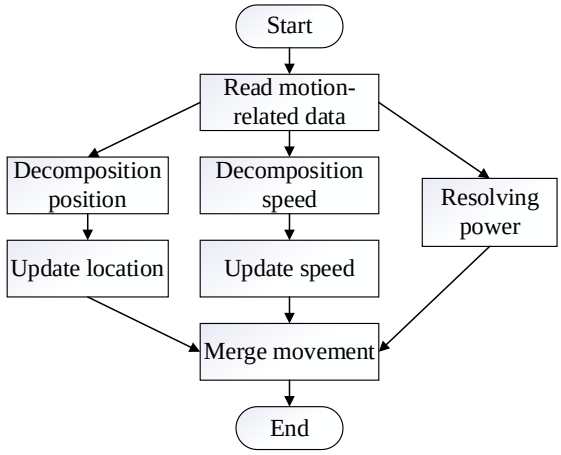



Figure 2. Program flow chart in time slice

### *C. Height-distance model*

Based on the analysis of the basic physical model, it is found that the UAV delivery task is a multi-variable nonlinear model, not a simple univariate relationship, and it is difficult for strategy analysis to directly compress the global variables into one relationship. Therefore, the model designed with fixed parameter traversal is a nonlinear model that analyzes multiple variables.

Based on the analysis requirements of UAV launch distance, flight height, dive Angle and initial velocity of explosives (UAV flight speed plus explosive launch speed), this paper makes a comparative analysis according to the idea of control variable experiment and the traversal idea of search tree.

Due to the shock wave range of the blasting explosive and considering the UAV mission efficiency, distance-related constraints should be set into the model, as shown in the following formula (2) and (3).

$$H \in \left[H_{\min}, H_{\max}\right] \tag{2}$$

$$L \in \left[ L_{\min}, L_{\max} \right] \tag{3}$$

The minimum height $H_{min}$ is the minimum height of the UAV, the maximum height $H_{max}$ is the starting height of the UAV, and the range of $L$ is set as the distance range limit between the UAV and the target.

The length $L_{plat}$ of the level flight stage of the UAV is designed as the difference between the horizontal distance $L$ of the take-off point and the horizontal distance $L$ of the drop point, as shown in the following formula (4).

$$L_{plat} = L_0 - L \tag{4}$$

*D. Objective function*

In the case of known wind speed $v_{wind}$, altitude range $H$, distance range $L$, projectile muzzle velocity $v_0$ and initial departure position $L_0$ of the UAV, the acquisition function of the flight scheduling strategy of the UAV is represented by the following formula (5), where $p$ () represents the implementation and acquisition strategy process of the above model.

$$P = P\left( H, L, L_0, v_{wind}, v_0 \right) \tag{5}$$

Among them, the initial speed of the blasting explosive is the UAV flight speed $v_{fly}$ plus the launch speed $v_{shoot}$, as shown in the following formula (6).

$$v_0 = v_{fly} + v_{shoot} \tag{6}$$

*E. Traverse the simulation strategy*

The background setting of this problem involves two indeterminate variables, the height range and the distance range of UAV delivery. In the multivariate nonlinear analysis, the analysis form is complex, and because the internal forces change with time, or even with the different time slice width Settings, so it is difficult to directly analyze the results from a mathematical analysis Angle.

The traversal simulation trial-and-error strategy is adopted. The process is shown in Figure 3. One of the variables is traversed through the value, corresponding to another quantity is obtained, and then the conditions that meet the conditions are screened in the results.

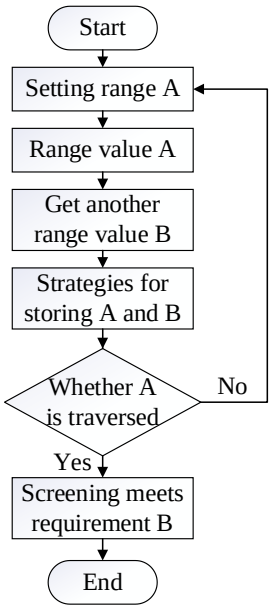



Figure 3. Traversal trial-and-error flow chart of multi-range variables

*F. Quantitative analysis of stability*

In the stability analysis of UAV, the influence of manufacturing technology and operating technology of UAV is excluded, and the environmental influence factors are mainly considered [14]. Therefore, the factors affecting the stability of the UAV include wind speed, wind direction, UAV flight height and depression Angle. Wherein, the wind direction design is integrated into the wind speed design, and the wind speed is represented as $v_{wind}$, the flight height of the UAV is $H_0$, the flight speed of the UAV is $v_{fly}$, and the depression Angle of the UAV is β.

Stability directly affects the accuracy of the hit. The main principle is that stability will cause the UAV to shake in position, affecting the accurate grasp of height, the accurate grasp of horizontal distance, the control of tilt Angle, and the size of flight and launch speed. This affects the state of the explosive launched, and ultimately affects the accuracy of the detonation point.

The quantitative score index of stability is used to react on the initial velocity, launch Angle, ground height and other indicators of explosives, and simulation tests are carried out to design the evaluation system of positioning accuracy, and the relationship between stability and positioning accuracy is analyzed.

*1) Multivariate weighting*

The relevant values of wind speed, UAV flight height and flight speed are evaluated in turn according to the 100 score standard, and the values are deducted from the 100 score according to the multi-weighted design strategy, and the evaluation results are finally obtained. The expression is as follows formula (7).

$$Score = 100 - f_1(v_{wind}) - f_2(H_0) - f_3(v_{fly}) - f_4(\beta) \quad (7)$$

Considering that the wind speed has a relatively large impact on the stability of the UAV, the weight is set to 1, that is, the function is expressed as follows formula (8).

$$f_1(v_{wind}) = v_{wind} \quad (8)$$

Consider the design in a relatively balanced state, the higher the height, the greater the impact, the weight is set to 1/50, The expression is as shown in the following formula (9).

$$f_2(H_0) = \frac{H_0}{50} \quad (9)$$

The speed of the UAV itself also has a greater impact on the stability. The greater the speed, the greater the impact. The weight is set to 1/20, then the function block is as shown in the following formula (10).

$$f_3(v_{fly}) = \frac{v_{fly}}{20} \quad (10)$$

The impact of depression Angle on the stability of UAV is also the greater the Angle, the greater the impact, The expression is as follows formula (11).

$$f_4(\beta) = 20(1 - \cos\beta) \quad (11)$$

*2) Hit accuracy setting*

There are two strategies for the design of hit accuracy, absolute accuracy and relative accuracy. Absolute precision indicates the distance between the landing point and the planned point. In general, the longer the distance you need to travel in the launch process, the lower the probability of your own hit. That is, the accuracy of the hit should be evaluated by the deviation relative to the firing distance. Therefore, the relative accuracy has more evaluation significance.

The horizontal distance between the drop point and the target point is $X_0$, and the horizontal distance between the actual landing position of the explosive and the original drop point is $X'$, then the expression of absolute accuracy u is as shown in the following formula (12).

$$u = \left| X_0 - X' \right| \quad (12)$$

Relative accuracy $u\%$ is as follows formula (13).

$$u\% = \frac{u}{X_0} \times 100\% = \frac{\left| X_0 - X' \right|}{X_0} \times 100\% \quad (13)$$

*G. Random disturbance strategy*

In the modeling of numerical simulation, the core of simulation stability and hit accuracy is analyzed, and the random changes of initial parameters under the influence of stability factors are analyzed. Considering that certain Gauss disturbance *Gauss* is generated for the height *H*, flight speed *v* and depression Angle $\beta$ of the UAV, the actual initial test parameters of blasting explosive delivery during launch state are as follows:

$$v_0 = v_{shoot} + v_{fly} + Gauss_v(score) \quad (14)$$

$$H_0 = H_0 + Gauss_H\left(score\right) \quad (15)$$

$$\beta = \beta + Gauss_\beta\left(score\right) \quad (16)$$

The Gaussian perturbation is a random Gaussian distribution related to the stability score. The velocity, height and depression Angle have corresponding perturbation values to adapt to the parameter values. The Gaussian perturbation is set to produce a mean of 0, and the variance is determined by the object and score of the corresponding perturbation. Where, the disturbance variance of depression Angle and velocity is regarded as acting on the horizontal component velocity at the same time, and is set as (100-*score*)/10, and the disturbance variance of height is (100-*score*).

In the overall simulation model, the hits before and after the impact of stability score will be calculated respectively, and the results before and after will be input into the accuracy measurement model to evaluate the hit accuracy, so as to link the score with the hit accuracy. The process is shown in Figure 4.

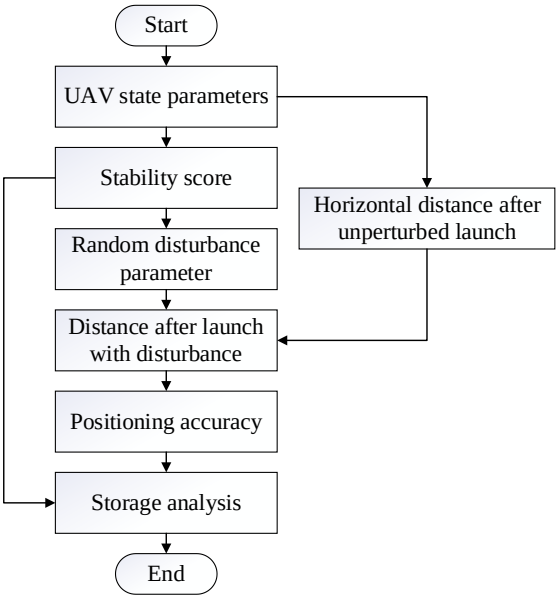



Figure 4. Simulation strategy of stability score and hit accuracy

When the stability score is associated with the hit accuracy, then the analytical solution is carried out, and the objective function is to obtain the maximum stability score under the condition of meeting the requirements of the scene, The expression is as follows formula (17).

$$Goal \rightarrow \left(Score_{\max} \middle| H, L, L_0, v_{wind}, v_{fly}, v_{shoot}, \beta\right) \quad (17)$$

Because the strategy of this model is a multi-variable nonlinear model, it is difficult to obtain direct analytical solutions, so the simulation strategy modeling is a more practical and operational strategy. Under the limitation of preset conditions, it is necessary to find the best strategy, still adopt the idea of traversal, and try all possible values to find the best scheduling scheme. The flow chart is shown in Figure 5.

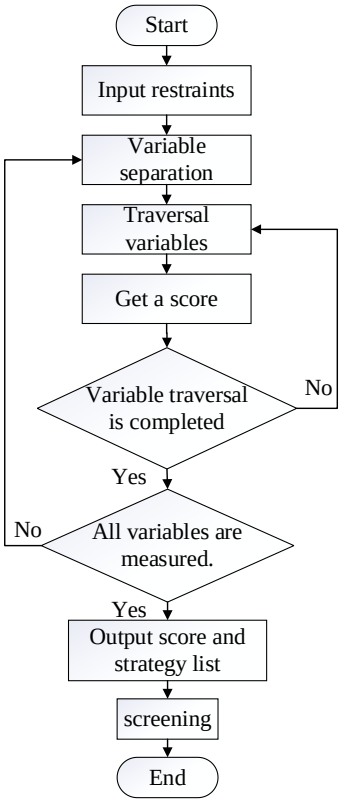



Figure 5. Traversal variable simulation strategy of multi-variable nonlinear model

## IV. EXPERIMENTAL TEST AND RESULT ANALYSIS

### *A. Experimental platform*

The model designed in this paper is based on python programming ideas and is programmed and implemented in jupyter platform.

*B. Experimental test*

*1) Generate the initial throw state in order*

To generate parameters that can be calculated and simulated for testing, set the range of parameters as shown in Table 1.

TABLE I. VARIABLE SETTINGS

| Variable name | Range of variable |
|---|---|
| Launch height | 100~1000 |
| Flight speed | 100~300 |
| Launching velocity | 250 |
| Horizontal component velocity (Represents the change in depression Angle) | Half of the initial speed - initial speed (depression Angle 60 degrees -0 degrees) |
| Wind speed effect | -6~6 |

*2) Stability score calculation*

According to the calculation strategy of stability, the influence factors of height, speed, wind speed and depression Angle are subtracted from 100 points. Finally, the stability score of each set of parameters is obtained. According to the scoring rules, the higher the height, the higher the speed, the greater the wind speed, and the greater the depression Angle, the score will also decrease.

*3) Planning point*

Without the influence of stability score, according to the original emission parameters, the horizontal distance from the launching point to the landing point is 185 meters when there is no interference.

*4) Random disturbance*

The stability score is converted into the influence on the velocity, height and depression Angle of the launch in the form of random disturbance, and the test is tried several times. The random disturbance takes the form of generating random numbers with a mean of 0 as interference.

*5) Accuracy measurement*

After ten random perturbations and fall point tests on the parameters that measure the stability score, the results shown in table 2 are obtained. Because of the great chance of random disturbance, the hit accuracy is different among different simulations.

100 simulated disturbance positioning tests for the above parameters were performed and the results were plotted in Figure 6. Although there are still great accidental factors in the simulation of 100 times, it can be preliminarily seen that the hit error revolves around the overall trend of a value fluctuation.

TABLE II. THE SIMULATION RESULTS OF A CERTAIN PARAMETER AFTER 10 DISTURBANCES

| Time | Height | Horizontal initial velocity | Landing distance | Absolute accuracy | Relative accuracy |
|---|---|---|---|---|---|
| 1 | 459.5609 | 193.9832 | 178.7461 | 5.741393 | 3.112076 |
| 2 | 565.9514 | 202.8413 | 192.8809 | 8.393398 | 4.543575 |
| 3 | 484.5841 | 203.5067 | 188.4999 | 4.012370 | 2.174874 |
| 4 | 478.3433 | 191.5439 | 178.4841 | 6.003414 | 3.254103 |
| 5 | 481.6434 | 201.0121 | 186.3224 | 1.834916 | 0.994602 |
| 6 | 519.0507 | 199.4992 | 187.8286 | 3.341048 | 1.810989 |
| 7 | 510.4967 | 207.1368 | 193.2061 | 8.718591 | 4.725843 |
| 8 | 491.1663 | 203.0071 | 188.5453 | 4.057764 | 2.199479 |
| 9 | 477.7795 | 198.9608 | 184.3180 | 0.169518 | 0.091886 |
| 10 | 561.1234 | 198.3617 | 189.0825 | 4.594944 | 2.490653 |

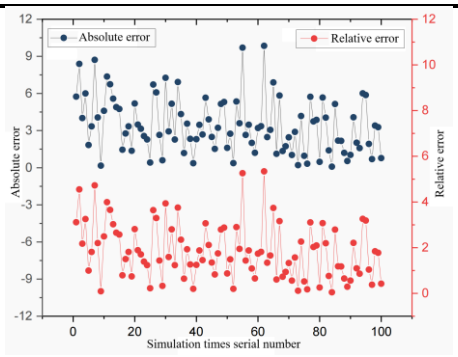



Figure 6. Absolute and relative hit errors of 100 simulations of a certain parameter

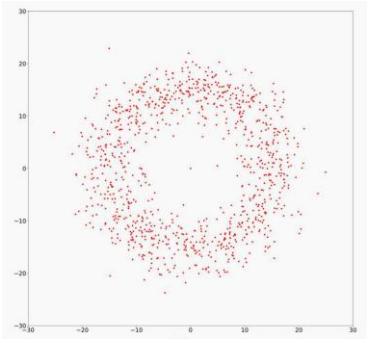


Figure 7. Distribution of drop points for 1000 simulations

Figure 7 shows the simulated distribution of 1000 perturbations under fixed emission parameters, where the center point is the exact landing point without disturbance. The simulation

results show that a single point cannot directly explain the global situation. Among all the simulation points, some are close to the theoretical drop point, and some deviate greatly, but the overall point of fall can form a roughly circular distribution around the argument.

### *C. Analysis of experimental results*

#### *1) Simulation verification*

The higher the stability score, the less interference to the initial state of the projectile, so the theoretical hit accuracy is smaller. In the set launch parameters scenario, the score is from 50 to 100, each score is traversed 100 times, and its average hit error is calculated, as shown in Figure 8.

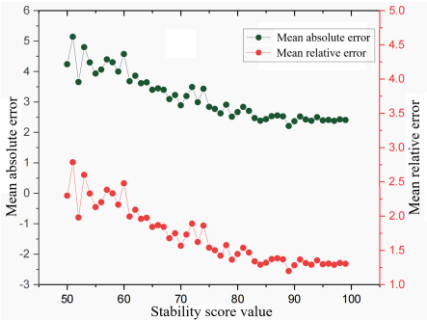
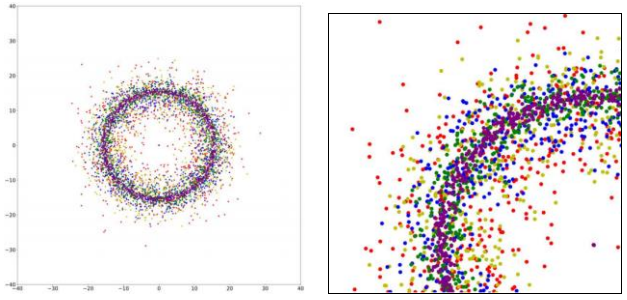



Figure 8. Average hit error of 100 simulations with different scores

The relationship between the score and the hit error showed a downward trend. With the increase of stability score, the hit error gradually decreases, that is, the hit accuracy gradually increases, which accords with the prediction of stability and hit accuracy in the modeling stage.



a) Total distribution b) Detail distribution

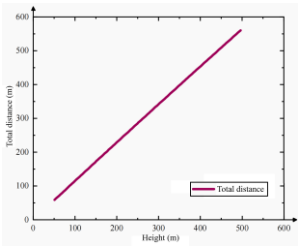
Figure 9. Comprehensive analysis of the landing points of 1000 simulations with different scores

Figure 9 shows the hit position of 1000 tests with different scores under fixed launch parameters. The red dot represents the score of 55 points, the yellow dot represents the score of 65 points, the blue dot represents the score of 75 points, the green dot represents the score of 85 points, the purple dot represents the score of 95 points, and the middle point of the ring is the original falling point.

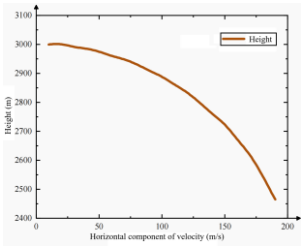
With the increase of the score, due to the decrease of the disturbance, the distribution range of the falling point is gradually accurate and gradually indents to the center of the circle. Therefore, it can be obtained that the higher the stability score, the smaller the disturbance, and the higher the hit accuracy; the lower the stability, the higher the disturbance, the wider the range of the landing point, the more inaccurate.

#### *2) Model strategy simulation test*

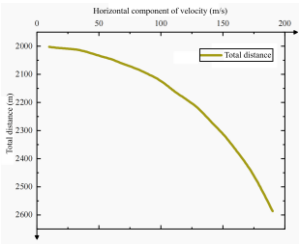
Taking the speed after the combination of flight speed and launch speed as the initial speed of the explosive launch time, the quantitative variable grouping analysis is adopted, the simulation test of the model is carried out, and the drawing analysis is carried out. The meaning of each diagram is presented in the name of the small diagram in the table, as shown in Figure 10.



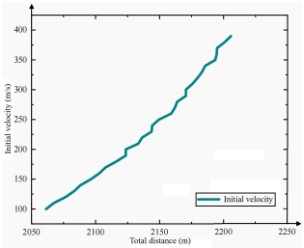

(a) The relationship between total distance and height when velocity and depression are fixed



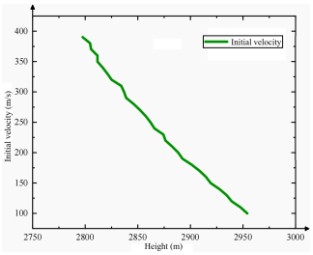

(b) Relationship between height and horizontal component of velocity when total delivery distance and initial velocity are fixed



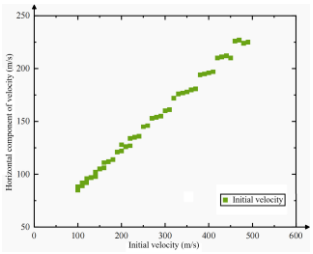

(c) The relationship between the total distance and the horizontal velocity component when the initial velocity and the drop height are fixed



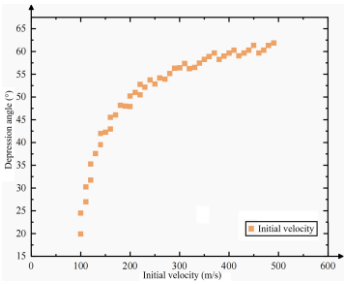

(d) The relationship between the total distance and the initial velocity when the depression Angle and the drop height are fixed




(e) The relationship between the initial velocity and the drop height when the dip Angle and the total drop distance are fixed




(f) The relationship between velocity and horizontal sub-velocity when the height and distance are fixed




(g) The relationship between velocity and depression Angle when the height and distance are fixed

Figure 10. Variable relationship analysis of delivery distance, delivery height, delivery speed and dive Angle

It can be concluded that when the speed and dive Angle are fixed, the delivery height and delivery distance are positively correlated, and close to the linear positive correlation trend.

When the delivery distance and initial velocity are fixed, with the increase of the horizontal velocity component (that is, the dive Angle decreases), the horizontal distance of the delivery object in the air will also increase, so the delivery height will decrease. When the initial speed and drop height are fixed, as the horizontal velocity component increases, the horizontal drop distance also increases, so the total distance also increases.

When the dive Angle and delivery height are fixed, the total drop distance increases with the increase of the initial speed. When the depression Angle and the total delivery distance are fixed, the initial velocity is negatively correlated with the delivery height. With the increase of the required delivery height, the time in the air of the explosive increases, but the horizontal distance should decrease, so the initial velocity should decrease.

When the delivery height and delivery distance are fixed, the initial velocity and the horizontal initial velocity component almost show a standard positive correlation, but from the Angle of depression, the dive Angle should gradually increase, and then tend to balance.

Obviously, the smaller the dive Angle, the larger the horizontal component of the initial velocity, the smaller the vertical component, when the height is determined, the more time can be moved in the air, the longer the horizontal movement time is relatively longer, the greater the horizontal movement distance, the greater the delivery distance. The other variables showed a roughly positive correlation.

*3) Adjustment strategy*

According to the stability scoring strategy and the relationship between various factors affecting the movement of the UAV, the height is required to be as low as possible, the depression Angle and the flight speed of the UAV are as small as possible, and the wind speed is 6m/s, the flight speed of the UAV is 300km/s-400km/s, and the launch speed is 500km/s. That is, the blasting speed range is 222m/s-250m/s. Therefore, the best delivery parameters are as follows.

The height is 300 meters, the delivery speed is 250m/s(900km/h), the horizontal distance is 290 meters, the horizontal component speed is set to 225m/s, and the cosine value of the depression Angle is 8/9.

According to the strategic planning, the drone from the height of 800 meters, first dropped to 300 meters, and then in the horizontal position of 290 meters away from the target, the height of 300 meters, the delivery distance of about 417 meters, the maximum flight speed of the drone, and the horizontal direction into the cosine value of 8/9 Angle, about 27 degrees, the launch of explosives is the best.

## V. CONCLUSIONS

This model has shown many advantages in the process of design and implementation. First, the model analyzes the relevant factors affecting the UAV mission from the perspective of kinematics. Secondly, the continuous system is discretized, the continuous time system is processed in slices, and

the calculation results are simulated step by step through calculus thinking, so that the model can accurately reflect the continuous changes of the system. Thirdly, the approximate strategy is adopted inside the time slice, which enables the complex continuous system to be processed. In addition, the model designs a multivariate analysis strategy based on Monte Carlo replacement tree search. Through traversing all cases and conducting simulation tests, convincing quantitative analysis results are obtained.

The innovative trial-and-error simulation strategy enables the model to find the required strategy in the actual scene and discover the law between the variables. Finally, the model uses multivariate weighted scoring mechanism to quantitatively analyze the stability, which provides convenience for the simulation comparison between the stability parameters and the precision parameters. In summary, this model not only has high theoretical accuracy and engineering practicability, but also has certain innovation and flexibility, which can provide important reference and guidance for the research and application in related fields.

## REFERENCES

[1] Chen. Research on path planning for UAV fixed-point mission [D]. Nanjing University of Posts and Telecommunications, 2022.

[2] Guo Daotong, Ke Hongfa, Feng Jianfeng. Static stability analysis of longitudinal motion of a UAV [J]. Tech Wind, 2018 (30): 249 + 254.

[3] Wu Yongxin. Research on the application of logistics drones in China 's rural e-commerce logistics market [D]. Shenzhen University, 2017.

[4] Liu Wenfeng. Small UAV fixed-point cruise multi-point material delivery system [J]. Research.

[5] Li Jianren, Xu Chuang, Hong Yongxin. A material delivery system based on UAV [P]. Guangdong Province: CN109987227A, 2019-07-09.

[6] Wang Jing, Liu Jiakai, Wang Ya. Research on anti-UAV fiber projectile based on fixed-point air explosion control [J]. According to Journal of Weapon Equipment Engineering, 2022, 43 (05): 22-25.

[7] Wei Chaoyou. Mobile landing of UAV [D]. Guangdong University of Technology, 2021.

[8] Qin Juan. Research on UAV target positioning and location deployment algorithm [D]. Beijing University of Posts and Telecommunications, 2020.

[9] Chen Tao. Research on fixed-point landing control technology of high aspect ratio UAV [D]. Nanjing University of Aeronautics and Astronautics, 2019.

[10] Li Yifan. Research on attitude control and precise fixed-point hovering of four-rotor UAV in complex environment [D]. Shanghai Jiaotong University, 2018.

[11] Pan Junjie, Guo Zhen, Zhao Junyuan. Motion and force analysis of transmission tower bar with UAV [J]. Zhejiang Electric Power, 2021, 40 (06): 54-63.

[12] Sun Zongyan, Wang Qiang, Zheng Yafei. Research on flight dynamics model of flying wing UAV [C]. Chinese Aviation Society. The 10th Youth Science and Technology Forum of China Aeronautical Society. Science Popularization Press, 2022: 861-867.

[13] Yu Shilong, Hu Jiajie. Overview of UAV autonomous flight control system [J]. China Science and Technology Information, 2022, No.685 (20): 41-43.

[14] Li Yao. Structural design and stability analysis of heavy-duty four-rotor UAV frame [D]. Shenyang University of Technology, 2022.

# Automatic Landing Control of Aircraft Based on Cognitive Load Theory and DDPG

Chao Wang
School of Armament Science and Technology
Xi'an Technological University
Xi'an, China
E-mail: 1017344731@qq.com

Changyuan Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: Cyw901@163.com

***Abstract*—The keypoint of autonomous driving technology is the accurate instructions maked by desicision-makers based on the perception information. Human plays an important role in the decision-makers. The cognitive load is usually used to quantify the impact of human-computer interaction during flighting. In this paper, we proposed a innovate automatic landing control method based on the cognitive load theory and Deep Deterministic Policy Gradient. Different to the traditional algorithm which heavily relays on an accurate model, the reinforcement learning algorithm is used to design the control strategy in the proposed method. And an improved DDPG algorithm is proposed based on the impact of cognitive load, to improve the training efficiency of the DDPG algorithm and reduce the correlation between data. And construct a human-machine reinforcement learning model. The final position, mean square error of pitch angle, and standard deviation of the aircraft gradually decrease with the number of iterations and tend to 0, indicating that the aircraft is gradually stabilizing its landing. The experimental results demonstrate that the proposed model can greatly improve the longitudinal stability of the aircraft.**

***Keywords-Component; Cognitive Load; Human Factors; Longitudinal Stability Control; Reinforcement Learning; Deep Deterministic Policy Gradient***

## I. INTRODUCTION

The research on pilot cognitive load can be traced back to the early 20th century [1-3]. Cognitive load refers to an individual's psychological resource used to solve problems or complete tasks within a certain period of time [4,5]. When a person's working memory capacity is overloaded with new information received directly or indirectly, the burden on the cognitive system increases, forming a cognitive load [6]. Nowadays, artificial intelligence technology has been widely applied in many fields, and its future development cannot be estimated. These developments will profoundly transform related fields. But if there is only a single application of artificial intelligence, although machines driven by AI technology and automation technology, such as cars, have both automation capabilities, that is, multiple mechanical/control systems that enable cars to travel according to instructions, and autonomous capabilities such as AI driven environmental perception and path planning, machines can already complete more and more tasks without human participation. However, as artificial intelligence is increasingly applied in various fields, especially in the field of automation control, human participation will become increasingly indispensable. The machine intelligence driven by AI has brought enormous imaginative space for automation applications in various fields and also requires the influence of human factors. The process of pilot information processing includes: firstly, the pilot obtains sensory and tactile information; Secondly, the pilot's ability to make decisions based on past experience when making plans; Finally, the level of pilot's operational behavior [7]. After information processing, the amount of cognitive resources consumed by pilots is called cognitive load. The autonomous willingness of pilots to react and operate runs through the process of human machine environment interaction, while the subjective driving intention of pilots is rarely considered in current research results. The reflection of actual effectiveness in flight control processes under different cognitive loads needs to be emphasized.

## II. RELATED THEORIES

### A. Cognitive load theory

Early measurement methods often relied on methods such as action response detection, pen and paper filling, form testing, and intelligence testing, due to limitations in experimental equipment and environment. The validity of various subjective scales has been validated and modified by many researchers in order to achieve more accurate evaluation results. Due to subjective evaluation methods such as paper, pen, and questionnaire surveys, which require participants to fill in based on their own perception, the results are often subjective.

Objective measurement methods have been applied to most human-computer interaction studies due to the inability to directly observe and measure cognitive load, such as eye tracking [8-9], Index of Cognitive Activity (ICA) [10], and other technologies. In addition to objective methods, some scholars in the field of multimedia learning research use subjective tools such as the Paas Psychological Effort Scale [11] and the NASA-TLX (National Aeronautics and Space Administration Task Load Index) scale [12] to evaluate cognitive load. The main methods currently used to measure cognitive load in human-computer interaction research include physiologytask [13] performance based [14][13] , and subjective self-assessment [15].

TABLE I. COGNITIVE COMPLIANCE

| Types | Indirect mode | Specific mode |
|---|---|---|
| subjective measure | Subjective assessment scale | NASA-TLX, WP scales, etc |
| | Task performance | Dual task measurement |
| objective measurement | Physiological measurement data | Electrocardiogram, oculomotor, electroencephalogram, etc |

Through literature analysis, it was found that in human-computer interaction research, objective methods tend to be used to obtain relatively reliable and effective data, and there are few studies that use a single subjective method to measure cognitive load (such as Clarke, Schuetzler, and Windle et al.), in order to avoid the influence of personal characteristics of participants on experimental results [16]. The stimuli experienced by individuals indirectly affect the changes in physiological data and represent the level of psychological processing. The hypothesis that human physiological changes to some extent reflect an individual's psychological state establishes a physiological method for measuring cognitive load [17]. Indirect objective measurement methods such as eye tracking technology[18], functional near-infrared spectroscopy (fNIRS) [19], skin electric response, electroencephalogram (EEG) [20]have been used to measure cognitive load in human-computer interaction research.

Heart rate variability is an indicator of electrocardiogram signals, referring to the irregularity of differences between consecutive heart beat cycles. Physiological functions that are not subjectively controlled by humans, including heartbeat, respiration, blood pressure fluctuations, and digestion, are all regulated by the autonomic nervous system of the human body. Due to the influence of multiple factors such as hormones, staying up late, and diet, there is no optimal standard interval for heart rate variability. However, the time-frequency indicators and other characteristic information of heart rate variability can provide non-invasive and quantitative evaluation of the autonomic nervous system, so electrocardiogram signals are selected as measurement data for human factors.

TABLE II. TIME DOMAIN INDICATORS

| Name | unit | illustrate | Formula |
|---|---|---|---|
| *MEAN* | ms | Mean RR interval | $MEAN = \frac{\sum_{i=1}^{N} RR_i}{N}$ |
| *SDNN* | ms | Normal RR interval standard deviation | $SDNN = \sqrt{\sum_{i=1}^{N} \frac{(RR_i - RR)^2}{N}}$ |
| *rMSDD* | ms | Root mean square of RR interval difference between neighbors | $rMSSD = \sqrt{\sum_{i=1}^{N} \frac{(RR_{i+1} - RR_i)^2}{N}}$ |
| $pNN_{50}$ | % | The proportion of RR interval difference greater than 50ms | $pNN_{50} = \frac{NN_{50}}{NN} \times 100\%$ |

Time domain analysis is the simplest and most intuitive way to study HRV, and its analysis principle is based on quantitative exploration of statistical indicators such as MEAN and SDNN in RR interval sequences. HRV time-domain indicators commonly used in analysis.

The area enclosed by the power spectrum curve and coordinates on each frequency band is numerically the power of the signal in that frequency band. Therefore, the energy characteristics of each frequency band are extracted based on the power spectrum to quantitatively analyze the frequency domain characteristics of HRV，as shown in the figure.

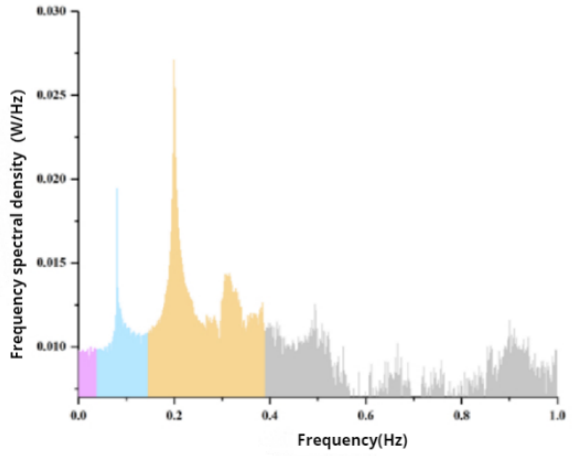



Figure 1. Power Spectrum

TABLE III. MEANING OF EACH FREQUENCY

| Name | Abbreviation | Meaning | Frequency range |
|---|---|---|---|
| Very low frequency | *VLF* | - | <0.04 Hz |
| low frequency | *LF* | Reflecting sympathetic nervous activity | 0.04～0.15 Hz |
| High frequency | *HF* | Reflecting parasympathetic nerve activity | 0.15～0.4 Hz |

$$LF_{norm} = \frac{LF}{TP - VLF} \times 100\% \quad (1)$$

$$HF_{norm} = \frac{HF}{TP - VLF} \times 100\% \quad (2)$$

Obtain the HRV frequency domain indicators on each side of the pentagonal flight through frequency domain analysis, (including the standardized low-frequency power $LF_{norm}$, $HF_{norm}$ standardized high-frequency power , and the ratio of low-frequency to high-frequency power result *LF/HF*.)

## *B. Reinforcement Learning Theory*

Reinforcement learning consists of three parts: intelligent agent, reward function, and environment. As shown in the figure, the initial state of the environment is inputted to the intelligent agent. The intelligent agent selects appropriate actions based on the state, and the actions are inputted to the environment. The environment obtains the reward value generated by the action and the new state. The two are inputted to the intelligent agent. The intelligent agent corrects the strategy based on the reward value, outputs new actions based on the new state, and thus repeats the cycle. The goal of reinforcement learning is to learn a strategy function $\pi$ ($x$), which is a mapping from state space $x$ to action space $a$. Reinforcement learning algorithms can be divided into three categories structurally: actor critic (A-C) structure, value function based reinforcement learning, and policy based reinforcement learning.

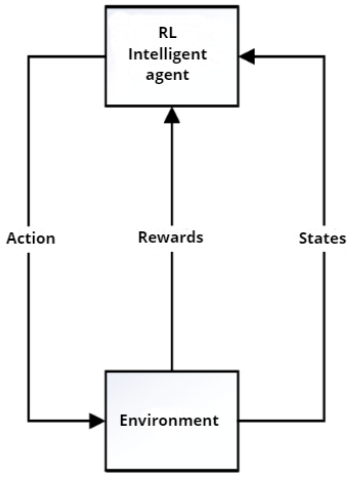



Figure 2. Meaning of Each Frequency

The actor and critic represent the policy $\pi$ and the value function V(s), respectively, and are approximated using a neural network. The input of

the actuator is the current state of the aircraft, the output is the changes in speed, pitch angle, and altitude, the input of the evaluator is the state of the aircraft, including the cognitive load level, and the output is a state value function. After inputting the action into the actor, a new state variable is obtained and the real-time reward value is obtained based on the reward function formula. The critic iteratively updates based on the direction that minimizes the time difference error, while the actor also updates based on the weighted gradient of the time difference error. In an iterative update, update the critic first and then update the actor. When the cumulative reward value reaches the target requirement or the training reaches the set number of times, the training stops.

The core of DDPG is to split the actor and evaluator critic into two networks: the current network and the target network. After the actor generates an action for the environment, samples $(s_i, a_i, s_{i+1}, r_i)$ are generated and placed in the experience replay pool, as shown in the figure 3.

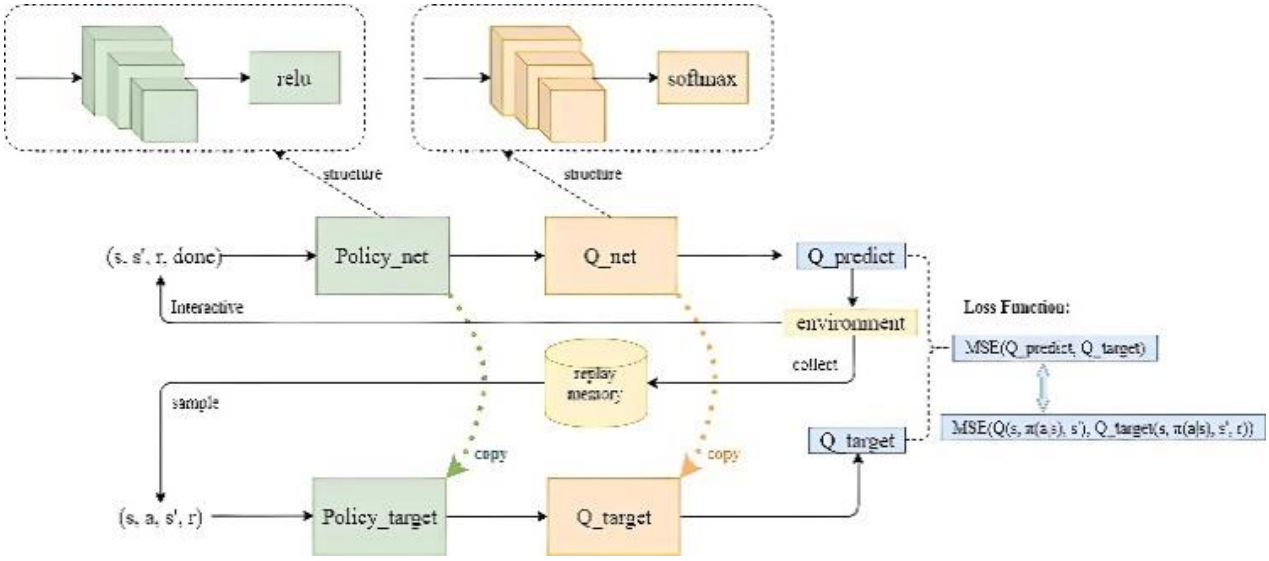



Figure 3. Schematic diagram of DDPG

The function of the current network's critic is to update parameter $\theta^Q$ and calculate the current state-action value $Q(s_i, a_i)$, while the target network of critic calculates the value $Q'(s_i, a_i)$. Afterwards, update the critical current network based on the loss function.

$$Loss = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i \mid \theta^Q))^2 \tag{3}$$

The updated current network will periodically copy the weights $\theta^{Q'}$ to the target network. Actor's current network acceptance status $s_i$, select the optimal action $a_i$ based on weight $\theta^\pi$. And update the weights according to the gradient formula.

$$\nabla \theta\pi J \sim \frac{1}{N}\sum_i \nabla a_i Q(s_i, a_i \mid \theta^Q) \mid \nabla \theta^\pi \pi(s_i \mid \theta^\pi) \tag{4}$$

The target network selects the optimal action $a_{i+1}$ based on the state $s_{i+1}$ and weight $\theta^{\pi'}$ in the experience replay pool. The current network will periodically copy weights to the target network.

$$y_i = r_i + \gamma * '(s_{i+1}, \pi'(s_{i+1} \mid \theta^{\pi'}) \mid \theta^{Q'}) \tag{5}$$

$$\theta^{Q'} = \tau * \theta^Q + (1-\tau)\theta^{Q'} \tag{6}$$

$$\theta^{\pi'} = \tau * \theta^\pi + (1-\tau)\theta^{\pi'} \tag{7}$$

## III. Design of DDPG model improved based on cognitive load theory

### A. Selection of Quantitative Indicators for Cognitive Load

Due to the need to characterize short-term cognitive load, electrocardiogram time-frequency indicators *MEAN*, $LF_{norm}$, $HF_{norm}$ were selected to calculate the cognitive load of pilots.

$$U_{rzfh} = c_1 \frac{mean}{MEAN} + c_2 \frac{lf_{norm}}{LF_{norm}} + c_3 \frac{hf_{norm}}{HF_{norm}} + c_4 \quad (8)$$

$MEAN$, $LF_{norm}$, $HF_{norm}$ is the constant value of human physiological indicators in a sedentary state. To solve for each weight value and constant term, define $d_{ij}$ is the relative value of the $j^{th}$ electrocardiogram indicator in the $j^{th}$ measurement window, based on the comparison between the measured value and the reference value. Matrix $W = (d_{ij})_m * 4$ is The relative value matrix of the norm indicator $MEAN$, $LF_{norm}$, $HF_{norm}$, where $b = (c_1, c_2, c_3, c_4)^T$ is the vector composed of the corresponding weight values and constant terms for each indicator $\omega = (U_{rzfh1}, U_{rzfh2}, U_{rzfh3} \ldots U_{rzfhm})^T$ is the cognitive load vector, where the total score of the NASA-TLX scale is used as the training value. Therefore, the determination of weight values is transformed into finding the optimal solution $e^0$ for the equation $W \cdot e = \omega$, such that for all $\forall e \in R$, $||W \cdot e^0 - \omega|| \leq ||W \cdot e - \omega||$ holds true. According to the generalized inverse matrix theorem and its existence conditions $e = W^{-1}\omega$, Using 20 sets of experimental measurements and a set of equations listed with 4 measurement windows in each group, substitute $e = W^{-1}\omega$ the weight matrix of each electrocardiogram indicator obtained is $e = (73.76, -28.53, 17.96, -116.44)^T$, Substitution $\omega = W \cdot e$, we obtain

$$U_{rzfh} = 73.76 \frac{mean}{MEAN} - 28.53 \frac{lf_{norm}}{LF_{norm}} + 17.96 \frac{hf_{norm}}{HF_{norm}} - 116.44 \quad (9)$$

Compared to calculating the average cognitive load using time-domain and frequency-domain indicators over a time period, using the above formula to fit short-term cognitive loads is more accurate and real-time This can then serve as one of the input data for the reinforcement learning model.

### B. Reinforcement Learning A-C Structure Neural Network Design

The neural network structure of the critic and actuator is shown in the figure. The hidden layer size is 100, where the input layer of the critic inputs the aircraft's cognitive load state $U_{rzfh}$ and Aircraft status $x = (z, \omega, \theta, v)^T$ .The hidden layer consists of five fully connected layers and three ReLU activation functions. The output layer outputs the value of the state action function with a learning rate of 0.001; The input layer of the actuator inputs the state of the aircraft, the hidden layer consists of four fully connected layers and three relu activation functions, and the output layer outputs the deflection angle and acceleration of the controller. The learning rate of the actuator is 0.0001, and the gradient thresholds of the actuator and critic are both 1.

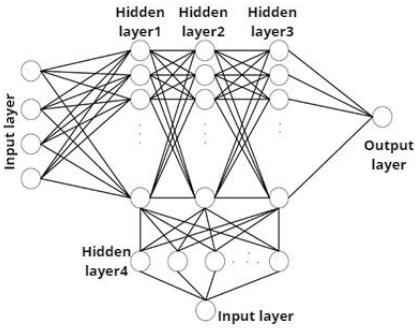



Figure 4. Critic Network Structure

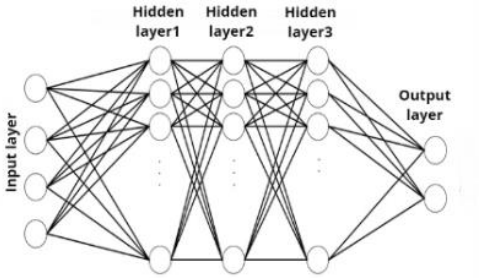



Figure 5. Actuator Network Structure

### C. Reward Function Design

Assuming that the observation input of the reinforcement learning agent is 4 dimensions, the state input of the aircraft is $x=(z,\omega,\theta,v)^T$ , the output of the agent is used as the control action of the aircraft, and the action signal is saturated before being used as the input of the aircraft, with a maximum deflection angle not exceeding 0.6 rad. Based on the above standards, the training conditions for aircraft are:

$$\begin{cases} 0 \le z \le 600m \\ -1 \le \theta \le 1rad \end{cases} \tag{10}$$

When the range set is exceeded, the training is terminated. If the range is too small, the exploration space of the action during the training process is limited, and it takes a lot of time for the reward function to converge. If the range is too large, it does not conform to the actual operation situation, and it is easy to control the situation when the aircraft pitch angle reaches 80 °. When the aircraft pitch angle exceeds a certain range, it is impossible to adjust back to a stable state during actual operation, Therefore, by setting the training range, useless training sample data is screened out. The design of a reward function guides the aircraft to approach the expected operating state during the training process, and the design of the reward function directly affects the control accuracy and robustness of the final controller. Based on experience and experimentation, Setting the coefficients for the relatively direct state variables z and θ as 0.3 and 0.5, and the coefficients for 0.04 and 0.03, respectively. and reward the previous control action with a coefficient of 0.005. Overall, the final reward function is:

$$R=-0.3\Delta z^2-0.04\Delta\omega^2-0.5\Delta v^2-0.05\delta_c \tag{11}$$

## IV. DATA COLLECTION AND ANALYSIS

### A. Collection equipment

The human factor wireless physiological acquisition platform includes experimental equipment such as v1.0 ArgoLAB signal acquisition device, laptop, high-definition camera, etc. The experimental equipment is located in a space with artificial low light and maintained at a comfortable temperature. The electrocardiogram collection device is a wireless optical capacitive pulse sensor with a sampling frequency of 512 Hz. The specific technical parameters are described in TableIV . Then, the v1.0 ErgoLAB wireless receiver is connected to a laptop to transmit the subject's electrocardiogram collection signal in real-time through a local area network, with a transmission frequency of 2.4 GHz.

TABLE IV. TECHNICAL PARAMETERS OF SIGNAL ACQUISITION EQUIPMENT

| Name | Value range |
|---|---|
| resolution ratio | ≥16Bit |
| ECG measurement range | -1500 $\mu V$ ～1500 $\mu V$ |
| Adjustable magnification | 1,2,3,4,5,6,7 |
| accuracy | 0.183 $\mu V$ , 0.0915 $\mu V$ , 0.061 $\mu V$ , 0.046 $\mu V$ , 0.037 $\mu V$ , 0.026 $\mu V$ |
| Number of wireless sensor channels | ≥1 |
| Wireless transmission frequency | 2.4GHz |
| Distance | 10m～100m |
| Battery operating time | ≥4h |

### B. Data collection process

The flight simulation adopts DCS World Steam Edition. Conduct experiments with 8 skilled flight trainees and apprentices. The debugging content of experimental equipment mainly includes model, airport, weather, date, and aircraft location， as shown in the figure. After entering the experimental course software, open the instructor console and select "Five sided Flight" experiment in "Create Task", then click the start button. Enter the scene settings interface again, change the aircraft model to su-25T, set the takeoff runway to Senaki, select the current experimental date and time, and ensure suitable meteorological conditions.

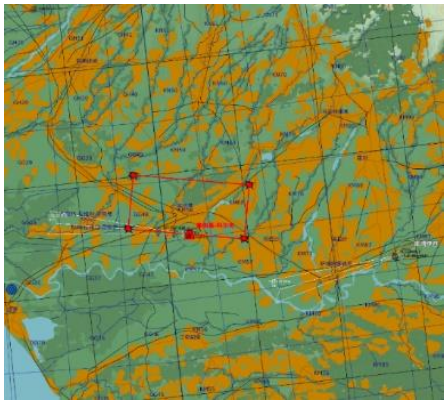


Figure 6. Experimental setup map runway

The specific experimental operation process is as follows:

(1) Set the weather of the flight simulator to clear, with a temperature of 20 degrees Celsius and a cloud layer of 2500m. Set the initial position of the aircraft at the runway entrance, align it with the centerline of the runway, and the subjects begin to perform pre takeoff checks.

(2) After completing the pre takeoff checklist, the subjects fully push the accelerator and maintain stable acceleration until the airspeed gauge shows more than 55 knots. Then, pull the lever to lift the wheels of the aircraft backwards and take off at a climbing rate of 500 feet per minute.

(3) Turn right 90 ° to both sides at a turning landmark, with a maximum turning angle of 20 °, heading from 50 ° to 150 °, and maintaining a climbing speed of 70 knots. Then turn to the third side at the second turning point.

(4) The aircraft has reached the altitude of the takeoff and landing route, with a stable airspeed of around 80 knots, maintaining the altitude heading airspeed.

(5) Reduce speed in advance on the short three sides, perform a pre landing checklist, check that the throttle valve is open, check the engine parameter table, check the engine temperature, the remaining fuel level on the fuel gauge, check that the mixing ratio lever is in the rich oil position, check the effectiveness of the braking device, lightly retract the throttle, start descent, maintain a descent rate of 500 feet per minute and an airspeed of around 70 knots.

(6) Turn to the fifth side at the four turning points, with a maximum turning angle of 30 °. Check that there are no obstacles on the runway, control the throttle as needed, release the throttle before touchdown, gently level the aircraft and wait for touchdown. After touchdown, gently brake to a stop.

### *C. Experimental Results and Analysis*

For baseline drift and other noise mixed in electrocardiogram signals, a low-pass filter is set to remove them; Then utilize a notch filter to eliminate power frequency interference mixed into it, apply threshold method to extract features of R waves in the electrocardiogram waveform, and quantitatively analyze the time-domain indicators of HRV. The main idea of the threshold method is to utilize the characteristic of QRS characteristic waves being the most oscillatory band within the electrocardiogram waveform. By setting different threshold ranges, the starting point of QRS main waves is obtained, and then the position of the R-wave vertex is determined using window and amplitude thresholds.

#### *1) Quantitative results of cognitive load*

The original physiological signal obtained is shown in the figure. The horizontal axis represents the number of samples, in units of $10^4$, and the vertical axis represents the amplitude, in units of $\mu V$ :

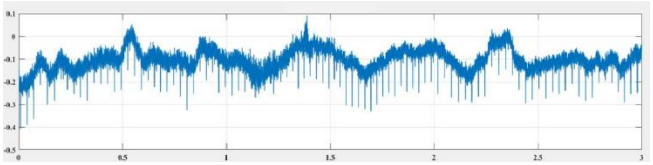


Figure 7. Original electrocardiogram signal map

The image after denoising is shown in the figure:

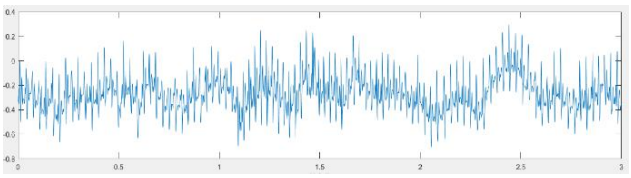


Figure 8. Electrocardiogram image after denoising

Using threshold method to extract R-waves from denoised electrocardiogram data, the results of R-wave extraction are shown in the figure:

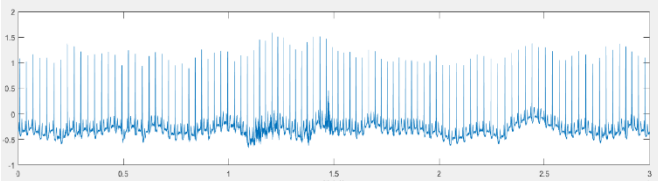


Figure 9. ECG image after R-wave extraction

Cognitive load $U_{rzfh}$ obtained through time-domain and frequency-domain analysis and processing, The curve is shown in the figure, and the trend of change roughly fits the subjective measurement, with a value range of (15,35).

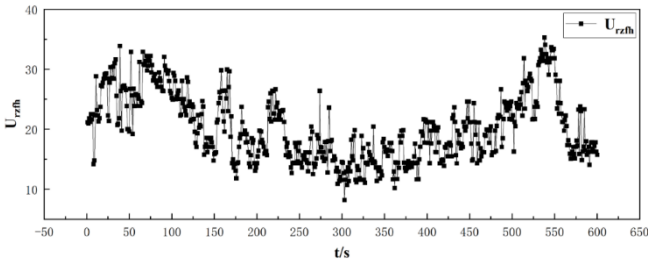



Figure 10. Cognitive load value

As shown in the figure, in the Five sided Flight, cognitive load follows a trend of first rising and then falling in the image after quantification. This is because cognitive load is more significantly influenced by psychological factors during takeoff and landing stages.

*2) Experiment environment and parameters*

The CPU of the server used in the experiment is a 256 core AMD EPYC 9654 with a frequency of 2.4GHz, with 128GB of memory and two A800 graphics cards. Each graphics card has 80GB of memory, and the operating system is Ubuntu. The framework is the TensorFlow platform of Python 3.6. The number of samples per batch is set to 512, the number of iterations is set to 1000, the learning rate is set to 0.01, the delay steps are 2, the experience pool size is 1000, the Actor network learning rate is 0.0001, the Critical network learning rate is 0.0002, and the exploration rate is 0.9.The closer the aircraft is to the expected state, the greater the reward value, Set the training objective to achieve an average reward greater than 200 over five consecutive episodes.

In subsequent testing, it was found that due to the setting of the training range, the aircraft exceeded the training range within 1 second, resulting in the termination of this training set. However, the cumulative reward value of this set exceeded 200 due to the small number of samples. The controller obtained after terminating the training for 5 consecutive sets cannot complete the aircraft stability control task. Change the training completion conditions to meet the target requirement when the sampling reaches 400 times in each set and the average value of the reward function in 5 consecutive sets is greater than 200. By establishing a simulation environment for training, terminate the training when the reward function is received and meets the requirements. The process of obtaining a controller through reinforcement learning is a continuous process of adjustment and improvement, and there is no optimal result. Through the simulation results of this training, the reward function and training requirements can be further adjusted to gradually reach the expected state of aircraft operation.

*3) Evaluation of landing experiment results*

In Figures 11 to 14, during the initial training stage, the aircraft is in the exploration and learning experience stage, so the learning effect is not ideal. However, the gradual increase in training frequency makes the aircraft's experience more and more rich. After the initial trial and error learning, the cumulative return of the algorithm increases rapidly, and the reward value increases and stabilizes in time step, quickly reaching convergence. In addition, the return value gradually increases, with an exploration variable enhancement value of 0.8 and a decrease in model entropy below -2, indicating a good training effect of the model.

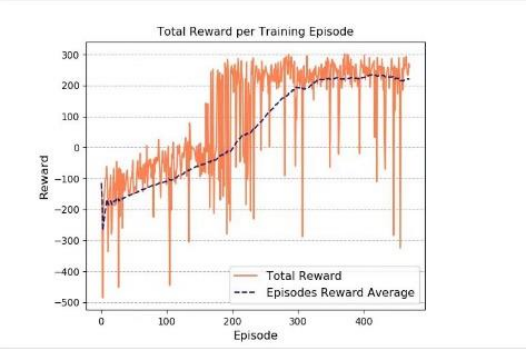



Figure 11. Reward Value Turn Curve

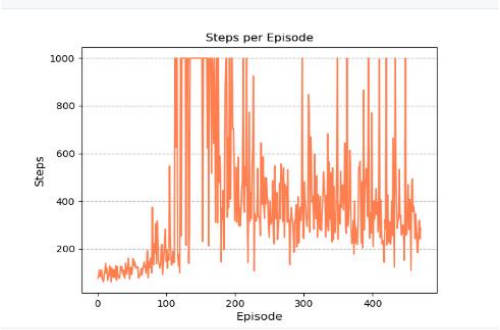



Figure 12. Time Step Turn Curve

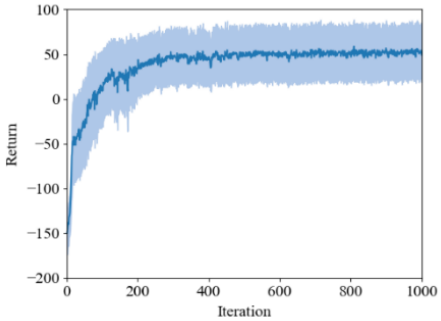



Figure 13. Return Value Change Curve Figure

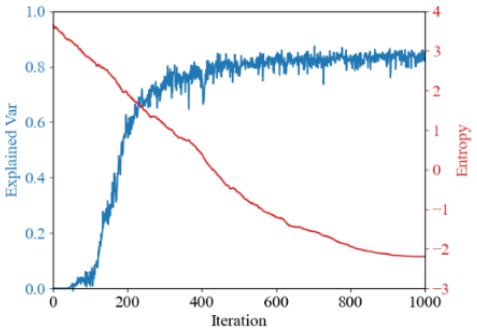



Figure 14. Exploring Variables and Entropy Change Curve

In Figures 15 and 16, the velocities and angular velocities in all directions decrease in fluctuation, and the acceleration in the z-direction tends to 0. The angular velocities in all directions also tend to 0, indicating that the aircraft has landed from the air and entered the ground sliding phase.

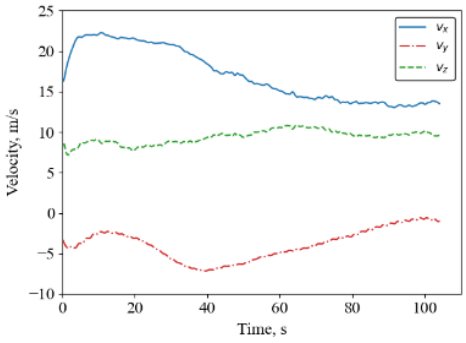



Figure 15. Speed time curve

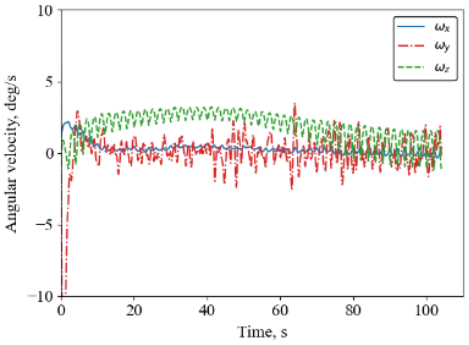



Figure 16. Angular velocity time curve

Figures 17 and 18 show that the final position, mean square error of pitch angle, and standard deviation of the aircraft gradually decrease with the number of iterations and tend to 0, indicating that the aircraft is gradually stabilizing its landing.

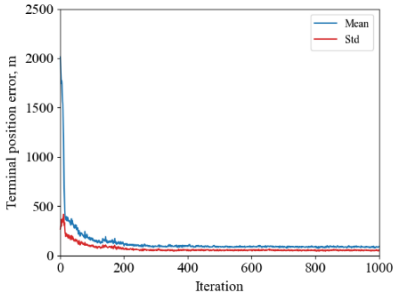



Figure 17. Final position mean square deviation and standard deviation Figure

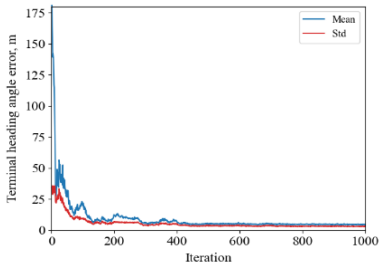



Figure 18. Final pitch angle mean square deviation and standard deviation

## V. CONCLUSION

This article uses linear fitting method to obtain cognitive load curve based on time-domain and frequency-domain analysis data of heart rate specificity test, and obtains cognitive load data that is consistent with the frequency of flight data. Based on cognitive load theory, the DDPG algorithm has been improved by incorporating human factors into the closed-loop of reinforcement learning. By using the improved DDPG algorithm for training, the number of ineffective explorations in the early stage was effectively reduced, and the impact of physiological level changes was considered in the field of human-computer interaction, achieving good control effects.

## REFERENCES

[1] Tesch M. Air Disaster[M]. Fyshwick: Aerospace Publication, 1994.

[2] Shappell S A, Wiegmann D A. Applying Reason: The human factors analysis and classification system [J]. Human Factors and Aerospace Safety, 2001, 1:59-86.

[3] Ullsperger P, Freude G, Erdmann U. Auditory probe sensitivity to mental workload changes--an event-related potential study[J]. International Journal of Psychophysiology, 2001, 40(3): 201-209.

[4] SWELLER J. Cognitive load theory [J]. Psychology of learning and motivation, 2011,55:37-76.

[5] BLESSINGER K, COMEAUX D. User experience with a new public interface for an integrated library system [J]. Information Technology and Libraries, 2020, 39(1).

[6] GONG Deying. Optimization management of cognitive load in multimedia learning [D]. Chongqing: Southwest University, 2009 (in Chinese).

[7] CHEN S, EPPS J. Using task-induced pupil diameter and blink rate to infer cognitive load [J]. Human–Computer Interaction, 2014, 29(4): 390-413.

[8] BRAUN M, BROY N, PFLEGING B, et al. Visualizing natural language interaction for conversational in-vehicle information systems to minimize driver distraction [J]. Journal on Multimodal User Interfaces, 2019, 13(2): 71-88.

[9] WU Lei, SU Yao, SHENG Qianqian, et al. Influence of Augmented Reality Assembly Indicators Symbol Based on Eye Tracking [J]. Packaging Engineering, 2022, 43(04):45-51+70 (in Chinese).

[10] BISWAS P, DUTT V, LANGDON P. Comparing ocular parameters for cognitive load measurement in eye-gaze-controlled interfaces for automotive and desktop computing environments [J]. International Journal of Human-Computer Interaction, 2016, 32(1): 23-38.

[11] BAUMEISTER J, SSIN S Y, ELSAYED N A M, et al. Cognitive cost of using augmented reality displays [J]. IEEE transactions on visualization and computer graphics, 2017, 23(11): 2378-2388.

[12] TERVONEN J, PETTERSSON K, MäNTYJäRVI J. Ultra-short window length and feature importance analysis for cognitive load detection from wearable sensors [J]. Electronics, 2021, 10(5): 613.

[13] Wang Di Research on Pilot Psychological State Evaluation Method Based on Physiological Signals [D], [Master's Thesis] Harbin: Harbin Institute of Technology, 2018 (in Chinese).

[14] SCHEWE F, VOLLRATH M. Ecological interface design effectively reduces cognitive workload–The example of HMIs for speed control [J]. Transportation research part F: traffic psychology and behaviour, 2020, 72:155-170.

[15] HWANG G-J, HSU T-C, HSIEH Y-H. Impacts of different smartphone caption/subtitle mechanisms on English listening performance and perceptions of students with different learning styles [J]. International Journal of Human–Computer Interaction, 2019, 35(4-5):333-344.

[16] YAN S, TRAN C C, CHEN Y, et al. Effect of user interface layout on the operators' mental workload in emergency operating procedures in nuclear power plants [J]. Nuclear Engineering and Design, 2017, 322:266-276.

[17] KRAMER A F. Physiological metrics of mental workload: A review of recent progress [M]. London: Multiple-task performance, 2020:279-328.

[18] Liu Xin. Measuring cognitive load levels based on eye movement data [D], [Master's thesis]. Chongqing: Southwest University, 2017 (in Chinese).

[19] FAN Lin, WANG Shuyi, WANG Yuqi, et al. Ergonomics and Cognitive Load of AR Guided Puncture Training System Based on fNIRS [J]. Packaging Engineering, 2021,42(20):146-151 (in Chinese).

[20] Mickael C, Fabre E , Giraudet L, et al. EEG/ERP as a Measure of Mental Workload in a Simple Piloting Task [J]. Procedia Manufacturing, 2015, 3(7): 5230-5236.

# The Time-Sensitive Networking Scheduling Algorithm Based on Q-learning

Jiayi Zhao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 1766614602@qq.com

Jing Cheng
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: chengjing@xatu.edu.cn

***Abstract*—Time-Sensitive Networking (TSN) occupies a vital position in modern communication domains, with the 802.1Qbv standard being an important network technology designed to meet real-time requirements. This standard requires network traffic to be transmitted within strict time windows, presenting challenges in network planning, necessitating efficient resource allocation and scheduling strategies. This paper addresses the 802.1Qbv planning problem through the introduction of reinforcement learning algorithms, offering an automated and intelligent solution. We have designed a reinforcement learning agent capable of perceiving network status, learning optimal resource allocation strategies, and dynamically adjusting in real-time environments. Through simulation and experimentation, we have validated the effectiveness of our proposed method, comparing it with traditional planning approaches. The contribution of this study lies in introducing a novel solution to the 802.1Qbv planning problem for time-sensitive networks, enhancing network resource utilization and performance. This approach offers strong support for the development and enhancement of TSN-like networks, holding significant importance for meeting the growing demands of real-time applications.**

***Keywords-Time-Sensitive Networking; Reinforcement Learning; Network Planning; IEEE 802.1Qbv***

## I. INTRODUCTION

### *A. Research Motivation and Significance*

Time-Sensitive Networking (TSN) plays a pivotal role in modern society, supporting a multitude of critical applications including industrial automation, intelligent transportation systems, real-time multimedia transmission in [7]. With the increasing demands for real-time performance, the planning and management of time-sensitive networks have become more complex and challenging. In particular, the introduction of the 802.1Qbv standard complicates network planning due to its requirement for network traffic to be transmitted within strict time windows, necessitating efficient resource allocation and scheduling strategies.

At present, the primary strategies for solving the 802.1Qbv scheduling issue include Satisfiability Modulo Theories (SMT) and linear programming. However, these methods have issues with planning time and adaptability to complex requirements.

This study introduces reinforcement learning as a means to enhance the adaptability of planning under complex conditions, optimize planning efficiency, and improve network performance.

### *B. Research Status at Home and Abroad*

International scholars have extensively researched the planning and management of Time-Sensitive Networking (TSN). The scheduling issues of TSN were first influentially addressed in 2016, primarily using the Satisfiability Modulo Theories (SMT) method, which initially solved the planning problem. Reference [1] provides an introduction to TSN.

Farzaneh and colleagues developed an automated scheduling synthesis tool for supporting TSN using graphical modeling in [2]. They also optimized the solution algorithms for planning based on previous studies. Reference [3] use array theory encoding to solve the 802.1Qbv problem. M. Pahlevan and others proposed a heuristic scheduling algorithm based on genetic algorithms in [4]. Gavriluţ and colleagues introduced a

scheduling algorithm based on Greedy Randomized Adaptive Search Procedures (GRASP) in [5]. In 2019, the TSNSCHED tool was introduced in [6]. It takes topology as input, utilizes SMT for solving, and supports stream planning for TSN unicast and multicast. Li C and colleagues proposed an integer linear programming based joint routing and scheduling method for multicast time-sensitive traffic in [13]. They reduced the scale of the scheduling problem by pruning the topology and grouping the traffic.

Mai T L and others proposed the use of machine learning methods in [8], including supervised and unsupervised learning, to explore the solution space of well-constructed TSN. Zhong C and colleagues were the first to propose using deep reinforcement learning to address the dynamic scheduling problem of TT streams in [9]. Their solution can promptly recover and reschedule the affected TT streams when network topology changes occur. Jia H and others proposed a deep reinforcement learning scheduling framework in [10], for incremental scheduling of TT streams. Based on this, they designed a three-step selection paradigm to mitigate the issue of a vast action search space. In [11], Jin X and colleagues proposed a method for scheduling large-scale data under the condition of limited Schedule Entries.

In 2022, Daniel B and others introduced a heuristic scheduling algorithm called HERMES in [12], which utilizes multiple scheduling queues to enhance the schedule ability of time-sensitive traffic.

Overall, researchers both internationally and domestically have recognized the importance of TSN and have achieved a series of results using different research methods and technologies. However, due to the complexity and real-time demands of TSN, this field remains challenging and requires further in-depth study and innovation. This research aims to contribute to the study of TSN planning issues in the domestic context by introducing reinforcement learning methods to improve network performance and resource utilization.

## II. Design of Algorithm

*Network Mathematical Model*

Time-Sensitive Networking is a network architecture made up of end systems (ES), switches (SW), and full-duplex physical links. This paper abstracts the time-sensitive network as a directed graph $G = (N\ L)$.

N symbolizes all device nodes within TSN, encompassing both switches and end systems. End systems are responsible for sending and receiving traffic that carries data within the network. Switches comprise multiple input and output ports and are responsible for forwarding frames from input ports to the corresponding output ports based on the destination node.

L denotes the set of physical connections within TSN, with each element signifying a unidirectional, simplex physical link. These links are the physical medium connecting the output ports of network nodes and are responsible for the actual transmission of traffic.

Fig. 1 illustrates a basic model of TSN topology., including 2 switches and 4 end systems, interconnected by a total of 10 unidirectional links.

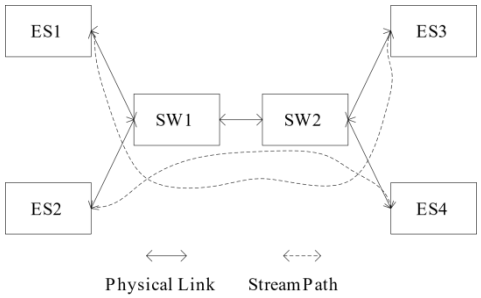



Figure 1. Simple TSN Network Model

In the model, a physical link is represented as a unidirectional link $[n_a\ n_b]$, defined by a triplet $\langle [n_a, n_b]\ speed, [n_a\ n_b]\ count, [n_a\ n_b]\ delay \rangle$. Here, $[n_a, n_b]\ speed$ denotes the transmission rate, $[n_a, n_b]\ count$ indicates the number of output port queues connected to node $n_a$ on the link, and $[n_a, n_b]\ delay$ represents the propagation delay.

The traffic model of TSN mainly includes Scheduled Traffic (ST) streams, Audio Video

Bridging (AVB) streams, and Best Effort (BE) streams in [15], with the priority decreasing in that order. The scheduling only considers the ST stream problem. The entirety of ST streams within the network is represented by the symbol S. For an ST stream $s_i \in S$ originating from node $n_1$ and destined for node $n_{n_i}$, passing through intermediate nodes $n_2,\ldots,n_{n_i-1}$, the transmission path of $s_i$ is represented as $s_i = \left\{[n_1,n_2],\ldots,[n_{n_i-1},n_{n_i}]\right\}$. The flow instance of $s_i$ on the link $[n_a, n_b]$ is represented as $s_i^{[n_a,n_b]}$, and each $s_i$ is defined by a triplet $\langle s_i.\text{len}, s_i.\text{period}, s_i.\text{e2e}\rangle$. Here, $s_i$.len indicates the size of stream $s_i$, the quantity of data transmitted within each period; $s_i$.period denotes the period length of stream $s_i$; $s_i$.e2e represents the upper limit of tolerable end-to-end delay(ED) for a stream $s_i$.

The hyper period of the link $[n_a, n_b]$ is defined as the least common multiple of all scheduled stream's periods passing through the link. The scheduling algorithm is only required to organize the streams within a single hyper period and repeat this schedule across multiple hyper periods.

All traffic is transmitted in units of frames in [14]. $F_i^{[n_a,n_b]}$ denotes all frame instances of stream $s_i$ transmitted on the link $[n_a, n_b]$, where $f_{i,k}^{[n_a,n_b]}$ represents the kth frame instance in the set. Each frame instance is defined by a triplet $\left\langle f_{i,k}^{[n_a,n_b]}.offset,\ f_{i,k}^{[n_a,n_b]}.trans\_dur,\ f_{i,k}^{[n_a,n_b]}.period\right\rangle$. Here, $f_{i,k}^{[n_a,n_b]}.offset$ represents the offset of the instance's transmission time relative to the start of the hyper period, satisfying Equation (1):

$$f_{i,k}^{[n_a,n_b]}.offset \in \left[0,\ f_{i,k}^{[n_a,n_b]}.\text{period}\right] \tag{1}$$

$f_{i,k}^{[n_a,n_b]}.trans\_dur$ represents the transmission delay of the instance, determined by its size and the link's bandwidth. $f_{i,k}^{[n_a,n_b]}$.period represents the duration of the instance, equivalent to the stream's cycle.

*Scheduling Constraints*

Scheduling constraints are essential conditions for determining the correctness and effectiveness of scheduling results. The traffic scheduling problem in TSN can be defined as an optimization problem: assigning appropriate transmission slots to all planned streams in the network, with the objective of maximizing optimization indicators while satisfying all scheduling constraints. The specific constraints are as follows:

Frame Periodicity Constraint: Planned streams are periodic, and it must be ensured that each frame completes transmission before the end of its period. This can be expressed as for $\forall s_i \in S, [n_a, n_b] \in s_i$ $\forall f_{i,k}^{[n_a,n_b]} \in F_i^{[n_a,n_b]}$, satisfying Equation (2):

$$\begin{cases} f_{i,k}^{[n_a,n_b]}.offset \geq 0, \\ f_{i,k}^{[n_a,n_b]}.offset \leq \\ f_{i,k}^{[n_a,n_b]}.period - f_{i,k}^{[n_a,n_b]}.trans\_dur. \end{cases} \tag{2}$$

Link Conflict-Free Constraint: To ensure the isolation and correctness of the flow's transmission, it is necessary to guarantee that each frame exclusively occupies the queue of the output port and the corresponding physical link during the same transmission slot. This can be represented by Equation (3):

$$\begin{aligned} &\forall [n_a,n_b] \in L\ \forall f_{i,k}^{[n_a,n_b]} \in F_i^{[n_a,n_b]}, \\ &f_{j,l}^{[n_a,n_b]} \in F_j^{[n_a,n_b]}, i \neq j: \\ &f_{i,k}^{[n_a,n_b]}.offset + \alpha \times f_{i,k}^{[n_a,n_b]}.period \geq \\ &f_{j,l}^{[n_a,n_b]}.offset + \beta \times f_{j,l}^{[n_a,n_b]}.period + \\ &f_{j,l}^{[n_a,n_b]}.trans\_dur \vee \\ &f_{j,l}^{[n_a,n_b]}.offset + \beta \times f_{j,l}^{[n_a,n_b]}.period \geq \\ &f_{i,k}^{[n_a,n_b]}.offset + \alpha \times f_{i,k}^{[n_a,n_b]}.period + \\ &f_{i,k}^{[n_a,n_b]}.trans\_dur. \end{aligned} \tag{3}$$

Frame Transmission Constraint: The transmission start time for a given frame on the subsequent link must not precede the completion time of its transmission on the prior link. This can be represented by Equation (4):

$$\begin{aligned}&\forall s_i \in S, [n_a, n_x] \in s_i \\ &f_{i,k}^{[n_a,n_x]} \in F_i^{[n_a,n_x]}, f_{i,k}^{[n_x,n_b]} \in F_i^{[n_x,n_b]}: \\ &f_{i,k}^{[n_a,n_x]}.offset + f_{i,k}^{[n_a,n_x]}.trans\_dur + \\ &[n_a,n_x]\, delay + \delta \le f_{i,k}^{[n_x,n_b]}.offset\end{aligned} \tag{4}$$

Flow Isolation Constraint: To prevent the interleaving of frame order in the scheduling queue, restrictions are imposed on any two Scheduled Traffic (ST) streams arriving at the same switch. If a frame from one ST stream has already arrived at the switch, frames from another ST stream cannot reach the same output port until all frames of the first stream have been completely sent to the output port. This can be represented by Equation (5):

$$\begin{aligned}&\forall [n_a,n_b] \in L\ s_i^{[n_a,n_b]}, s_j^{[n_a,n_b]} \in S, i \ne j: \\ &f_{i,ni}^{[n_a,n_b]}.offset + \alpha \times s_i.period + \delta \le \\ &f_{j,1}^{[n_y,n_a]}.offset + \beta \times s_j.period + \\ &[n_y,n_a]\, delay \vee f_{j,nj}^{[n_a,n_b]}.offset + \\ &\beta \times s_j.period + \delta \le f_{i,1}^{[n_x,n_a]}.offset + \\ &\alpha \times s_i.period + [n_x,n_a]\, delay.\end{aligned} \tag{5}$$

End-to-End Delay Constraint: The total delay experienced by each stream from source to destination must not exceed its specified maximum allowable delay. This can be represented by Equation (6):

$$\begin{aligned}&\forall s_i \in S: f_{i,ni}^{[n_y,n_b]}.offset + \\ &f_{i,ni}^{[n_y,n_b]}.trans\_dur - \\ &f_{i,1}^{[n_a,n_x]}.offset \le s_i.e2e\end{aligned} \tag{6}$$

*Scheduling Optimization Indicators*

In the TSN scheduling process, it is first necessary to satisfy the basic scheduling constraints to ensure the correctness of the scheduling results. Subsequently, the aim is to provide low-latency transmission services and enhance network transmission performance to achieve higher quality scheduling results. Therefore, two optimization indicators will be set to improve the quality of scheduling from the perspectives of slot utilization balance and ED.

A classic optimization goal in a TSN scheduling problem is the minimization of ED. To measure the quality of scheduling results, this paper establishes an optimization indicator for the ED of ST streams, aimed at achieving scheduling results that meet the scheduling constraints while minimizing the ED of ST streams. The ED denotes the duration from when the initial frame of the traffic begins transmission at the source node to the completion of the last frame's transmission. In conjunction with the TSN scheduling model, the calculation of ED for stream $s_i$ is as Equation (7):

$$\begin{aligned}e2e_i = &f_{i,ni}^{[n_{i(n_i-1)},n_{n_i}]}.offset + \\ &f_{i,ni}^{[n_{n_i-1},n_{n_i}]}.trans\_dur - f_{i,1}^{[n_{i1},n_{i2}]}.offset\end{aligned} \tag{7}$$

Define the ED delay vector for $s_i$ as $ED_i$, which can be expressed as Equation (8):

$$ED_i = 1 - \frac{e2e_i}{s_i.e2e} \tag{8}$$

It can be concluded that the smaller $e2e_i$ is, the larger $ED_i$ will be, indicating better network performance. This paper calculates the overall ED of all ST streams using the Average End-to-end Delay (AED) method. This approach is used to measure the network's end-to-end delay indicator:

$$AED = \sum_{i=1}^{N} \frac{ED_i}{N} \tag{9}$$

Another optimization goal is the uniformity of time slots. More uniform time slots can reduce the average delay of messages other than Scheduled Traffic (ST) messages. As long as the delay of ST messages is kept within the required limits, this is acceptable. The planned time slots are shown in Fig.2 as follows:

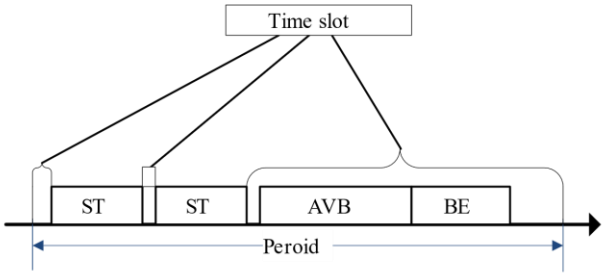



Figure 2. TSN time slot

The network's slot balance indicator primarily assesses the balance of idle slots on each link. The balance of each link is abstracted into a Link Balance Vector (LB), which is described using the standard deviation method. For link $[n_a, \eth_b]$, the calculation of the Link Balance Vector (LB) is as per Equation (10):

$$LB_{[n_a,\eth_b]} = 1 - \sqrt{\frac{\sum_{i=1}^{n^{[n_a,\eth_b]}} \left( slot_i^{[n_a,\eth_b]} - \overline{slot} \right)^2}{n^{[n_a,\eth_b]}}} \quad (10)$$

In this context, $n^{[n_a,\eth_b]}$ represents the number of idle slots on the $[n_a, \eth_b]$ link. $slot_i^{[n_a,\eth_b]}$ denotes the length of the i-th idle slot on $[n_a, \eth_b]$. $\overline{slot}$ is the average length of idle slots on the $[n_a, \eth_b]$link.

The overall Network Link Balance (NLB) for the network can be described by the average of all Link Balance Vectors across the links, as shown in Equation (11):

$$NLB = \sum_{i=1}^{N} \frac{LB_i}{N} \quad (11)$$

N signifies the total count of links within the network.

*Design of the Scheduling Algorithm*

The Algorithm introduces the concept of time discretization, converting the continuous time interval into a set of transmission moments. This transforms the problem of choosing transmission moments for traffic into a multi-treasure hunt in a three-dimensional space, with each planned stream acting as a searcher. The global link slot allocation in the network is designed as the environment. The scheduling of traffic at each hop along its transmission path is designed as the state space, where the current state fully characterizes the process, conforming to Markov properties. The action of choosing a transmission moment for the traffic on that link in the current state is designed as the action space, where the action space is the set of transmission moments obtained by discretizing the traffic's period. Regarding exploration strategy, this paper improves upon the ε-greedy strategy, adaptively adjusting the exploration probability ε based on iteration count, allowing the scheduling results to better converge to the optimal state. Combined with the optimization indicators mentioned in the previous section, a reward function is designed to evaluate scheduling actions and update the Q-table, until the algorithm reaches the maximum number of iterations, thus obtaining the final scheduling results.

The task of the TSN scheduling algorithm is to calculate the transmission moments for frames of all planned streams on the transmission links. The TSN scheduling problem has Markov properties; by continuously applying a one-hop scheduling policy to the network, observing changes in the state of the network environment, and obtaining immediate reward values, the algorithm trains an adaptive traffic scheduling model. After numerous iterations of training, an optimal behavior strategy is obtained, guiding the agent to execute the optimal scheduling actions. The process of the algorithm is as follows:

*1)* Parse configuration files, initialize network topology information, traffic collection, etc.

*2)* Initialize the scheduling model, including the environment, Q-table collection, learning rate α, discount factor γ, maximum number of

iterations (episodes), exploration probability ε, state space, and action space.

*3)* Use an adaptive exploration strategy to select transmission moments, calculate the ε exploration factor, randomly select from the action space with a probability of ε, and select the transmission moment with the highest Q value from the Q-table with a probability of 1-ε.

*4)* Execute scheduling actions, move to a new state, and check if scheduling constraints are met. If not, provide negative feedback; if constraints are met, mark in the environment.

*5)* Update the Q-table according to the action record, regardless of whether the scheduling is successful or not.

*6)* If all ST streams are scheduled, indicating a successful scheduling, calculate the ED and NLB indicators of the scheduling results, put the weighted R value into the experience pool, and give positive reward feedback according to the reward function, indicating the end of this iteration round, then proceed to Step 7. If not all scheduling is completed and still in an intermediate state, go back to Step 3 and proceed with the scheduling process.

*7)* Ascertain if the count of iterations (episodes) has hit its upper limit. If not, go back to Step 3 and continue iterative training until the maximum number of iterations is reached, concluding the training.

After the algorithm training is completed, the final Q-table collection is obtained. The Q value indicates the superiority of choosing different transmission moments for traffic in each state; the larger the Q value of a transmission moment, the closer it can approach the scheduling goal. According to the final Q-table, the optimal transmission slots are planned for each planned stream on each link, serving as the basis for generating the gate control list.

The environment (Env) represents the overall slot allocation situation of the links in the current network, which can be seen as the mapping information between the traffic and the transmission moments on the transmission link. The definition of Env is shown in Equation (12):

$$Env = \left\{ s_i \rightarrow [n_a, n_b] \rightarrow f_{i,k}^{[n_a, n_b]}.offset \right\} \quad (12)$$

The scheduling space is divided into three types of states: effective, intermediate, and failure. A successful scheduling state is represented by all planned streams having generated scheduling schemes that meet the scheduling constraints. An intermediate state occurs when all streams scheduled so far meet the constraints, while a failure state arises when the current scheduling includes scenarios that do not meet the constraints. As shown in Fig.3 below, the diagram depicts the scheduling of three streams, with the top showing a successful state, the middle an intermediate state, and the bottom displaying a failed state due to a conflict.

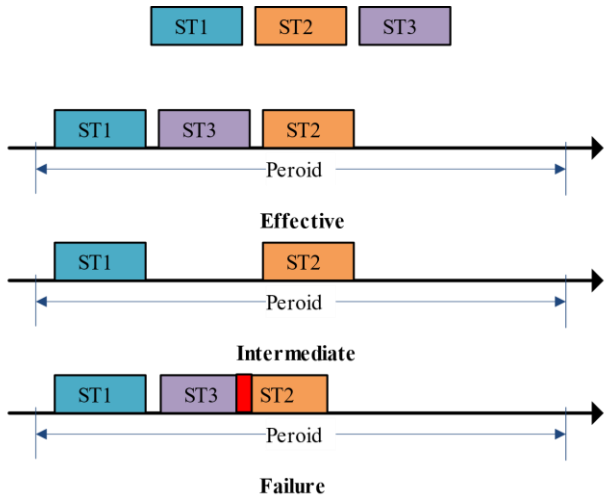



Figure 3. TSN Scheduling space status example

Let *State* represent the state space, where $State_t \in State$ denotes the scheduling state at moment t. The definition of $State_t$ is as shown in Equation (13):

$$State_t = \langle S, L, Env \rangle \quad (13)$$

The action space is a set of time points obtained by discretizing the size of the traffic cycle. Let A represent the action space, and $a_t \in A$ denote the action of the agent at moment t, which is the transmission time chosen for the traffic. Thus, the definition of $a_t$ is as per Equation (14):

$$a_t = sche\left\langle s_i, [n_a, n_b] \; f_{i,k}^{[n_a, n_b]}.offset \right\rangle \quad (14)$$

Where *sche* — on the $[n_a, \eth_b]$ link, the allocated transmission time for the frame instance of stream $s_i$ is $f_{i,k}^{[n_a,\eth_b]}.offset$.

During the scheduling process in TSN, agents need to continuously select scheduling actions from

the action space for learning. This paper adopts an improved ε-greedy strategy for action selection, exploring with a probability of ε, and exploiting with a probability of 1-ε.

The calculation method for the exploration probability ε is as per Equation (15):

$$\varepsilon = \cos\left(\frac{\pi}{2} * \frac{cur\_epi}{episodes}\right) \tag{15}$$

In the learning process of TSN scheduling, each time a scheduling action is applied to the environment, a new scheduling state is generated. The reward function can provide a corresponding feedback evaluation for the new scheduling state.

When the new state $s_{i+1}$ does not meet the scheduling constraints, it indicates that $s_{i+1}$ cannot ensure that all planned streams in the current network can be correctly scheduled, and is considered a failure scheduling state. In this case, the reward function returns a negative reward value. When the new state $s_{i+1}$ meets the scheduling constraints, and all ST streams in the network have been scheduled, a positive reward value is returned. When the new scheduling state $s_{i+1}$ meets the scheduling constraints, but there are still unscheduled planned streams in the network, this scheduling state is an intermediate state, and a reward value of 0 is returned. In summary, the definition of the reward is as shown in Equation (16):

$$R(s_i, a_i) = \begin{cases} -1, \ failure \\ 0, \ intermediate \\ \lambda_1 * (\exp\left(1 - \dfrac{e2e}{CC}\right) - 1) + \\ \lambda_2 * (\exp\left(1 - \dfrac{NLB}{NLB_{MAX}}\right) - 1) \\ , \ effective \end{cases} \tag{16}$$

In the Q-learning-based TSN scheduling algorithm, during the process of generating scheduling results, it is necessary to design a Q-table to store reward values. After executing a scheduling action $a_i$ in any state $s_i$, the reward function returns a reward value, denoted as $Q(s_i, \eth_i)$. In the scheduling algorithm designed in this paper, each planned stream corresponds to a Q-table. Each Q-table is an n*m experience matrix, where n is the number of links that the traffic route passes through, and each row of the Q-table, from top to bottom, corresponds to each transmission link of the traffic in sequence; m is the number of time points obtained by discretizing the period of the traffic. The Q-table is initially set to all zeros and is continuously updated with values through learning.

## III. RESULTS

In this paper, experiments will be conducted based on the optimization indicators mentioned earlier, to compare and analyze the Q-learning-based TSN scheduling algorithm with the SMT scheduling algorithm, thereby verifying the advantages and disadvantages of the scheduling algorithms.

The subject of the TSN scheduling algorithm is the planned streams. The experimental data mainly includes the collection of planned streams awaiting scheduling in the network and the topology information. The TSN topology used in the experiment is illustrated in Fig.4 below. This network topology comprises 16 network end systems, 8 TSN network switches, and 31 full-duplex physical links. The assumption is made

that every network device and physical connection within the network shares the same model, with identical bandwidth, and the lengths of the links are equal.

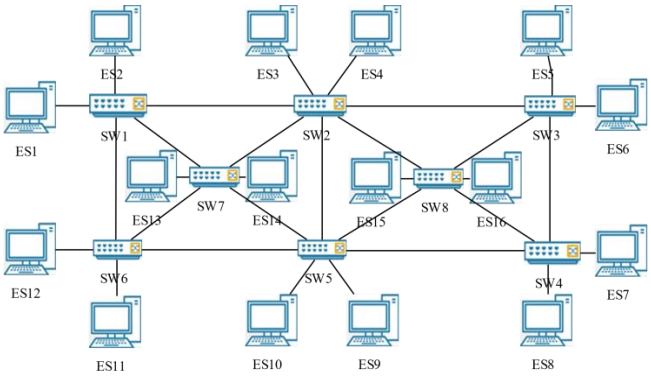



Figure 4. The experimental network topology

In the course of the experiment, the size, period, source, and destination of the ST streams are generated in random. The period of ST streams is chosen at random from the options of 1ms, 2ms, 4ms, 8ms, and 16ms. If the size of an ST stream exceeds the MTU, it can be divided into a collection of frames. According to standard Ethernet limitations, each frame is between 64B and 1518B. Given that the bandwidth of all network links is configured to 1000Mbps, the transmission delay for a frame at the output port varies between 5.12 microseconds and 121.44 microseconds.

The experiment injected 80 planned streams into the network for scheduling, with the size, period, source, and destination systems of each stream randomly generated. Assuming that the transmission paths of the flows have been determined by a specific routing algorithm, there are a total of 302 hops, and the hyper period for scheduling is 4000 us. The parameters for the scheduling algorithm are configured with a maximum of 50,000 iterations, a learning rate of 0.6, a discount factor of 0.9, an end-to-end delay index weight $\lambda_1$ of 0.5, and a NLB weight $\lambda_2$ of 0.5.

After obtaining the experimental results and analyzing the data, we listed the maximum and minimum values of the NLB for the scheduling results during the 50,000 iterations of training for these two scheduling algorithms, as shown in Fig.5:

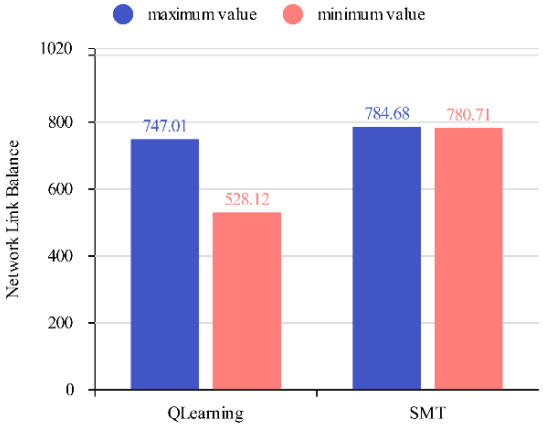



Figure 5. Comparison of NLB

As depicted in Fig.6 below, the change curve of the optimal value of the NLB of the scheduling results is given for these two algorithms under successful scheduling conditions across 50,000 iterations of training. In the graph, the horizontal axis denotes the algorithm's iteration count, whereas the vertical axis reflects the NLB value of the scheduling results. It can be observed that NLB of the scheduling results obtained by the SMT scheduling algorithm remains unchanged during the iterations. The reason is that the SMT solver lacks exploration capability, and the solution results are only related to the solution space of the scheduling problem. The NLB of the Q-learning scheduling algorithm shows a significant decline. During the training process, the algorithm continuously adjusts the scheduling strategy using an adaptive exploration strategy and gradually converges to higher-quality scheduling results under the guidance of the reward function.

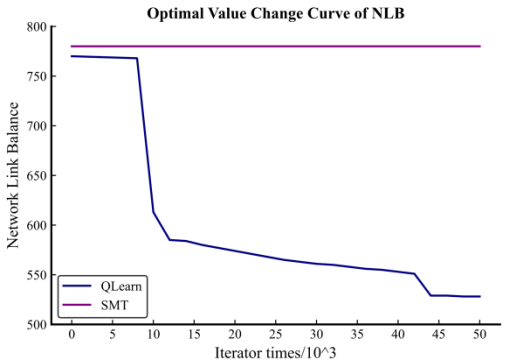



Figure 6. Optimal Value Change Curve of NLB

It can be deduced that the Q-learning based TSN network scheduling algorithm considerably surpasses the SMT algorithm in achieving a higher NLB. Moreover, the Q-learning-based approach, as compared to the SMT algorithm, possesses the capability to continuously optimize based on experience during iterations, converging towards higher-quality scheduling results in terms of the index.

## IV. Conclusion

This paper revolves around the planning problem of TSN, involving mathematical analysis and modeling, and defines optimization indicators. Subsequently, a TSN planning and scheduling method based on Q-learning is proposed. Employing a discretization approach, the scheduling problem parameters and functions in Q-learning are defined and trained. Finally, experiments are conducted to compare the results of this method with traditional approaches. It is evident that the TSN scheduling method based on Q-learning shows significant superiority over traditional algorithms in terms of indicators such as NLB.

This paper proposes a TSN scheduling method based on Q-learning, which also provides insights for subsequent TSN planning problems. For example, it raises questions such as whether there are faster training-based methods that can be applied to TSN planning issues, or whether it's possible to combine current popular approaches, such as large models, for planning purposes.

## References


[1] Messenger J L, Time-Sensitive Networking: An Introduction [J]. IEEE Communications Standards Magazine, 2018, 2(2): 2933. DOI:10.1109/MCOMSTD.2018.1700047.

[2] Farzaneh M H, Kugele S, Knoll A. A graphical modeling tool supporting automated schedule synthesis for time-sensitive networking [C]. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2017: 1-8.

[3] Oliver R S, Craciunas S S, Steiner W. IEEE 802.1Qbv Gate Control List Synthesis using Array Theory Encoding [C]//IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE, 2018. DOI:10.1109/RTAS.2018.00008.

[4] Pahlevan M, Obermaisser R. Genetic Algorithm for Scheduling Time-Triggered Traffic in Time-Sensitive Networks[C]//2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation. IEEE, 2018. DOI:10.1109/ETFA.2018.8502515.

[5] Gavrilut V, Pop P. Scheduling in time sensitive networks (TSN) for mixed-criticality industrial applications [C]//14 Th IEEE International Workshop on Factory Communication Systems. IEEE, 2018:1-4. DOI:10.1109/WFCS.2018.8402374.

[6] Santos A C T D, Schneider B, Nigam V, TSNSCHED: Automated Schedule Generation for Time Sensitive Networking[C]//2019 Formal Methods in Computer Aided Design (FMCAD). 2019. DOI:10.23919/FMCAD.2019.8894249.

[7] Bello L L, Steiner W. A perspective on IEEE time-sensitive networking for industrial communication and automation systems [J]. Proceedings of the IEEE, 2019, 107(6): 1094-1120.

[8] Mai T L, Navet N, Migge J. A hybrid machine learning and schedulability analysis method for the verification of TSN networks [C]. In: 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), 2019: 1-8.

[9] Zhong C, Jia H, Wan H, et al. DRLS: A Deep Reinforcement Learning Based Scheduler for Time-Triggered Ethernet[C]//2021 International Conference on Computer Communications and Networks (ICCCN). IEEE, 2021: 1-11.

[10] Jia H, Jiang Y, Zhong C, et al. TTDeep: Time-Triggered Scheduling for Real-Time Ethernet via Deep Reinforcement Learning [C]. In: 2021 IEEE Global Communications Conference (GLOBECOM), 2021: 1-6.

[11] Jin X, Xia C, Guan N. Real-time Scheduling of Massive Data in Time Sensitive Networks with a Limited Number of Schedule Entries [J]. IEEE Access, 2020, PP (99): 1-1. DOI:10.1109/ACCESS.2020.2964690.

[12] Bujosa D, Ashjaei M, Papadopoulos A V, et al. HERMES: Heuristic multi-queue scheduler for TSN time-triggered traffic with zero reception jitter capabilities [C]. In: Proceedings of the 30th International Conference on Real-Time Networks and Systems, 2022: 70-80.

[13] Li C, Zhang C, Zheng W. Joint Routing and Scheduling for Dynamic Applications in Multicast Time-Sensitive Networks [C]//IEEE International Conference on Communications Workshops. IEEE, 2021. DOI:10.1109/ICCWorkshops50388.2021.9473540.

[14] Nasrallah A, Thyagaturu A S, Alharbi Z, et al. Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research [J]. IEEE Communications Surveys & Tutorials, 2018, 21(1): 88-145.

[15] Reusch N, Zhao L, Craciunas S S, et al. Window-Based Schedule Synthesis for Industrial IEEE 802.1Qbv TSN Networks [C]// 2020 16th IEEE International Conference on Factory Communication Systems (WFCS). IEEE, 2020.

# Lightweight Low-Altitude UAV Object Detection Based on Improved YOLOv5s

Haokai Zeng
College of Ordnance Science and Technology
Xi'an Technological University
Xi'an, 710021, China
E-mail: zhk790743337@outlook.com

Jing Li
College of Electronic Information Engineering,
College of Ordnance Science and Technology
Xi'an Technological University
Xi'an, 710021, China
E-mail: gem99li@163.com

Liping Qu
College of Ordnance Science and Technology
Xi'an Technological University
Xi'an, 710021, China
E-mail: quliping@qq.com

***Abstract*—In the context of rapid developments in drone technology, the significance of recognizing and detecting low-altitude unmanned aerial vehicles (UAVs) has grown. Although conventional algorithmic enhancements have increased the detection rate of low-altitude UAV targets, they tend to neglect the intricate nature and computational demands of the algorithms. This paper introduces ATD-YOLO, an enhanced target detection model based on the YOLOv5s architecture, aimed at tackling this issue. Firstly, a realistic low-altitude UAV dataset is fashioned by amalgamating various publicly available datasets. Secondly, a C3F module grounded in FasterNet, incorporating Partial Convolution (PConv), is introduced to decrease model parameters while upholding detection accuracy. Furthermore, the backbone network incorporates an Efficient Multi-Scale Attention (EMA) module to extract essential image information while filtering out irrelevant details, facilitating adaptive feature fusion. Additionally, the universal upsampling operator CARAFE (Content-aware reassembly of features) is utilized instead of nearest-neighbor upsampling. This enhancement boosts the performance of the feature pyramid network by expanding the receptive field for data feature fusion. Lastly, the Slim-Neck network is introduced to fine-tune the feature fusion network, thereby reducing the model's floating-point calculations and parameters. Experimental findings demonstrate that the improved ATD-YOLO model achieves an accuracy of 92.8%, with a 31.4% decrease in parameters and a 28.7% decrease in floating-point calculations compared to the original model. The detection speed reaches 75.37 frames per second (FPS). These experiments affirm that the proposed enhancement method meets the deployment requirements for low computational power while maintaining high precision.**

***Keywords-Lightweight; Small Object; UAV Detection***

## I. INTRODUCTION

Small UAVs are aircraft known for their diminutive size, cost-effectiveness, and ability to fly at low altitudes. Due to their compactness and ease of operation, small UAVs have emerged prominently on modern military battlegrounds, garnering significant favor. They demonstrate exceptional performance in tasks such as reconnaissance, surveillance, communication, and target identification. However, the low-altitude, slow-flight attributes of small UAVs render them challenging to effectively counter using conventional detection methods, thus presenting novel challenges for military defense. Consequently, detecting UAV targets in flight has become a pivotal approach to addressing this issue[1].

Utilizing image and video analysis, employing computer vision algorithms for real-time UAV detection and tracking stands as the most promising method for UAV detection. In comparison to standard radar-based methodologies, this system offers a myriad of advantages, encompassing enhanced accuracy and reduced costs [2].

To date, the majority of detection tasks have been predominantly conducted through the utilization of deep learning methodologies for

DOI: 10.2478/ijanmc-2024-0009

feature extraction. This method mainly consists of two types of algorithms: two-stage and single-stage object detection. The former involves initially delineating the regions of interest before determining target position and class information. Representative algorithms include R-CNN [3], Fast R-CNN [4], Faster R-CNN [5], and Mask R-CNN [6]. The latter directly ascertain target position and class information without the need for separately identifying regions of interest. Typical algorithms include YOLO [7-9] and SSD [10]. Given the high maneuverability of low-altitude UAVs, capable of swiftly altering direction, moving at high velocities, and executing diverse flight maneuvers, numerous scholars opt for the YOLOv5 algorithm and its enhancements to execute UAV target detection tasks.

Lu et al. [11] introduced an improved YOLOv5s-based algorithm for small rotary-wing UAV target detection, demonstrating enhanced accuracy and feature extraction capabilities, it experiences a certain decrease in detection speed. Yang et al. [12] developed a real-time detection algorithm, named GCB-YOLOv5s, for low-altitude UAVs using machine vision detection techniques. While this algorithm boosts detection speed, it also leads to a slight decline in detection accuracy. Bao et al. [13] presented a real-time detection method for micro UAVs based on YOLOv5. Although the algorithm demonstrates commendable real-time performance for UAV targets at low altitudes, its effectiveness in detecting UAV targets in distant scenes is limited, and its robustness is relatively poor.

In summary, existing algorithms have improved the detection accuracy of low-altitude UAV targets but have overlooked the complexity and computational burden of the algorithms. Hence, the engineering challenge at hand is: how to enhance the algorithm's detection efficiency for UAV targets while preserving detection accuracy, employing lightweight design principles. Consequently, this paper suggests a lightweight detection algorithm for low-altitude UAVs, coined ATD-YOLO, and based on an enhanced version of YOLOv5s. The key enhancements of this algorithm are as follows:

1. By merging multiple publicly available datasets, a relatively comprehensive UAV target dataset is constructed.2. Based on the lightweight model FasterNet [14], the paper proposes a lightweight module called C3F to substitute the C3 module in the input feature extraction network. This substitution substantially reduces the number of parameters and floating-point calculations, thereby achieving lightweight effects on the overall network. 2. A more optimal upsampling method, CARAFE [15] is employed to increase the receptive field, enhancing feature sharpness post traditional upsampling. 3. EMA [16] is integrated into the backbone network to extract vital image information while filtering out irrelevant details, thus enabling adaptive feature fusion and enhancing detection accuracy. 4. Slim-Neck [17] is introduced into the neck part of the network, replacing Conv layers and C3 layers with lightweight convolutional neural networks GSConv and VOVGSCSP. This further reduces the computational workload and parameter complexity of the model, thereby improving its inference speed without sacrificing detection accuracy.

The improved ATD-YOLO network structure, shown in Figure 1, maintains detection accuracy while adopting a lightweight design, meeting the demands for real-time operation for UAV target detection. The enhanced detection model better accommodates the computational constraints of UAV detection devices, providing a research solution for lightweight improvements in UAV target detection.

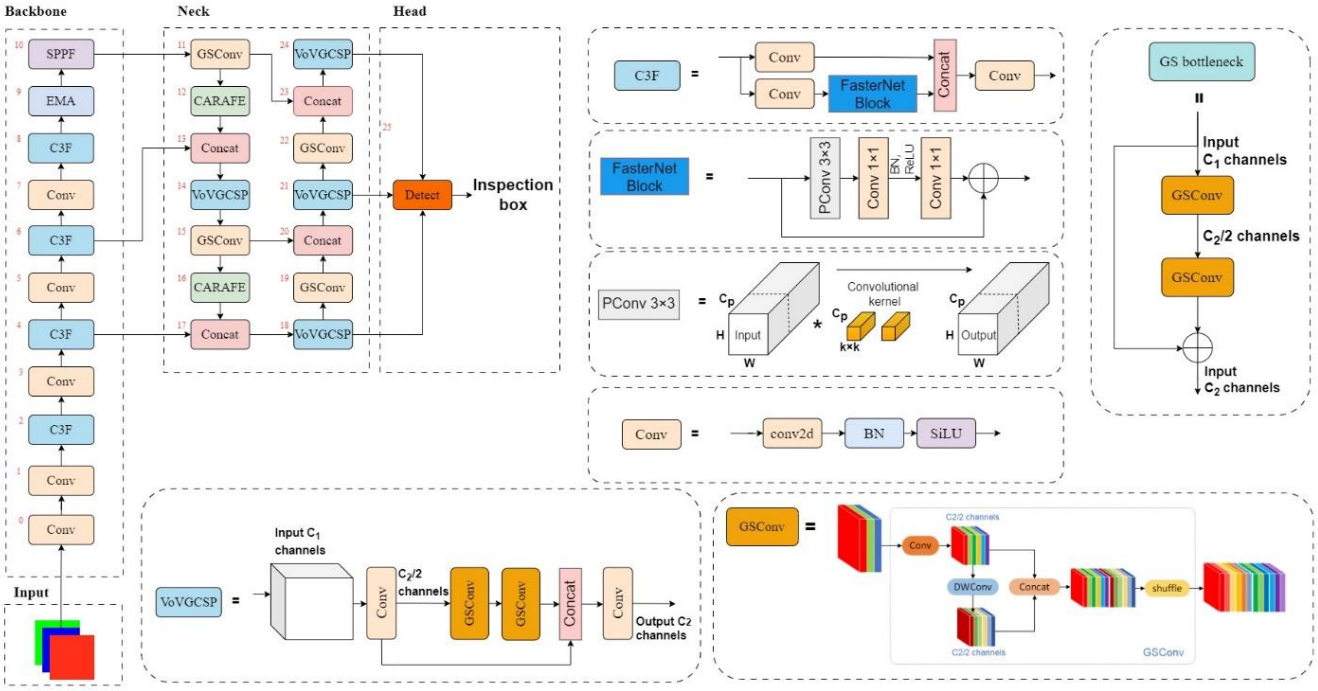



Figure 1. ATD-YOLO Network Structure

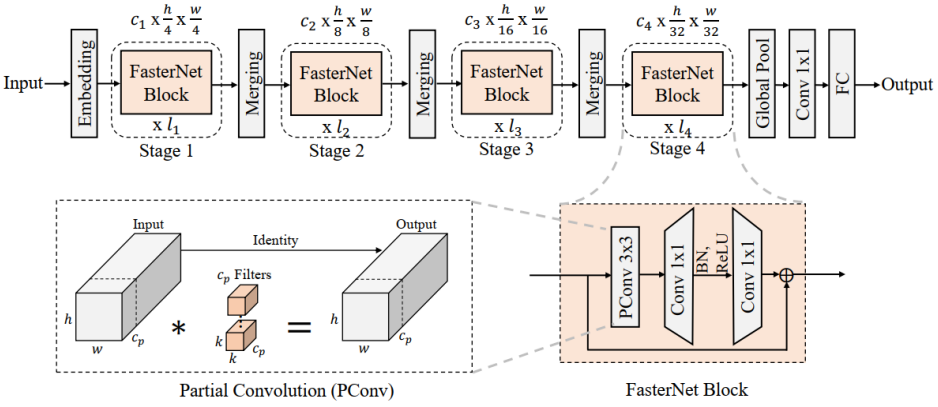



Figure 2. Framework of image measuring system [14]

## II. Algorithm Improvement And Optimization

### A. C3F Lightweight Feature Extraction Module

Addressing redundant computation, Chen et al. [14] introduced Partial Convolution (PConv), which reduces memory access while optimizing the parameter problem caused by redundant computation, greatly improving the ability to capture spatial features. FasterNet is constructed with PConv and 1x1 convolutional structures. In Figure 2, h, w, and k represent the height, width, and kernel size of the feature map, respectively, while Cp indicates the number of channels in conventional convolution.

PConv only uses general Conv to achieve spatial feature acquisition on some input channels, while maintaining the remaining channels unchanged. Calculate by considering the first or last consecutive Cp channel as a representation of the entire feature map. Ensure its generality while maintaining the same number of input and output feature map channels. The FLOPs of PConv are h × w × k2 × C2p, which only accounts for 1/16 of the general Conv.

The C3 module in the YOLOv5 network is pivotal for increasing network depth and receptive field, enhancing feature extraction. Initially, it included Conv1, Conv2, Conv3 modules, and one or more Bottleneck modules. While this design enriches the learning capabilities of the C3 module, it also adds to the computational load and model complexity. Thus, this paper introduces a lightweight feature extraction module, C3F, inspired by the FasterNet module concept.

Figure 3 illustrates the structural diagram of the C3F feature extraction module. Here, h, w, and k signify the height, width, and convolution kernel size of the feature map, while Cp represents the number of conventional convolution channels.This module introduces partial convolution and replaces BattleNet with FasterNet Block in the C3 module, reducing computational redundancy and memory access while maintaining the speed and efficiency of feature extraction.

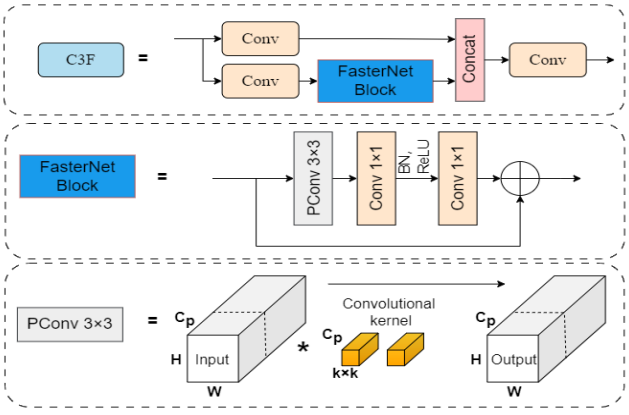



Figure 3. C3F structural schematic diagram

## B. CARAFE upsampling module

Object detection models often employ nearest neighbor or bilinear interpolation for feature map upsampling. While adaptive upsampling uses methods such as deconvolution. However, these traditional methods have certain shortcomings in accurately reconstructing target detail information, which can easily lead to partial information loss of small targets, thereby affecting detection accuracy. In contrast, CARAFE improves the quality of upsampled features by recombining content aware features to make the upsampling kernel semantically relevant to the feature map. The CARAFE operator can better preserve and recover feature information details, so this article chooses to use the CARAFE operator for upsampling to enhance regional sensitivity and generate more accurate high-resolution feature maps.

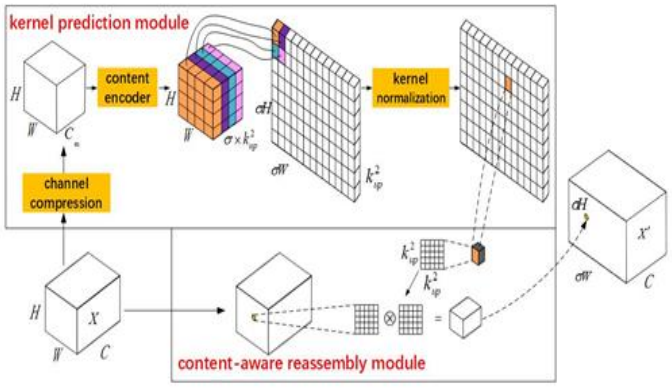



Figure 4. CARAFE upsampling calculation flowchart

## C. Multi scale attention mechanism module

In various computer vision tasks, the significant effectiveness of channel or spatial attention mechanisms in generating clearer feature representations has been demonstrated. However, modeling cross channel relationships through channel dimensionality reduction may have side effects on extracting deep visual representations.

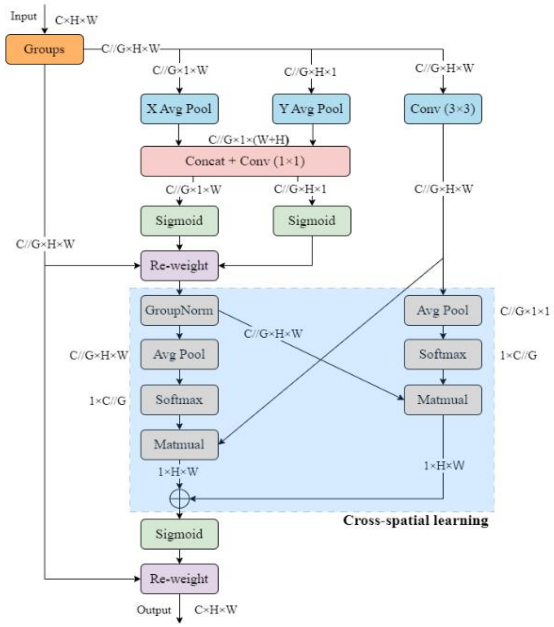



Figure 5. EMA Attention Mechanism

EMA without dimensionality reduction [16] aims to preserve channel information while minimizing computational overhead. This is achieved by reshaping some channels into batch dimensions and grouping channel dimensions into multiple sub-features to evenly distribute spatial semantic features. EMA learns effective channel descriptions in convolutional operations without reducing channel dimensionality, enhancing pixel-level attention for advanced feature maps. This lightweight and flexible EMA attention model serves as a core module applicable to lightweight networks.

Figure 5 outlines the EMA attention module, comprising Feature Grouping, Parallel Subnetworks, and Cross spatial learning. EMA divides the input feature map into multiple sub-features based on channel dimensions. It employs two parallel subnetworks: one with 1x1 branches and the other with 3x3 branches. The 1x1 branch encodes channel information using global average pooling operations, while the 3x3 branch captures local cross-channel interaction. EMA then fuses the output feature maps of the two subnetworks using cross-space learning, resulting in an attention weight map of the same size as the input feature map to enhance its expressive power.

### *D. Lightweight fusion stage*

Large deep learning models are difficult to deploy on industrial embedded devices. Many lightweight networks use a large number of depthwise separable convolutions, and even if the C3 of the backbone network is replaced with lighter modules, there are still a large number of 1x1 convolution operations, making it difficult to achieve sufficient accuracy. This dense convolution operation actually consumes more resources, and even with channel shuffling, the effect is still poor. Therefore, this article embeds the Slim Neck [17] network is integrated into the feature fusion stage, incorporating the GSConv module and the VOVGSCSP module.

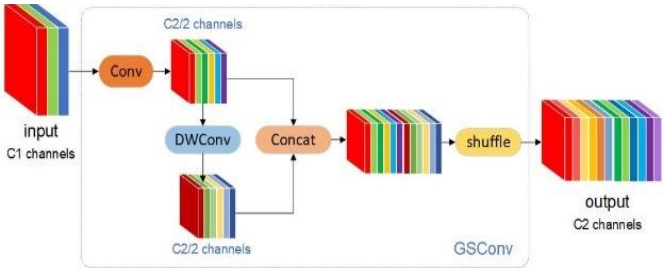



Figure 6. GSConv Module

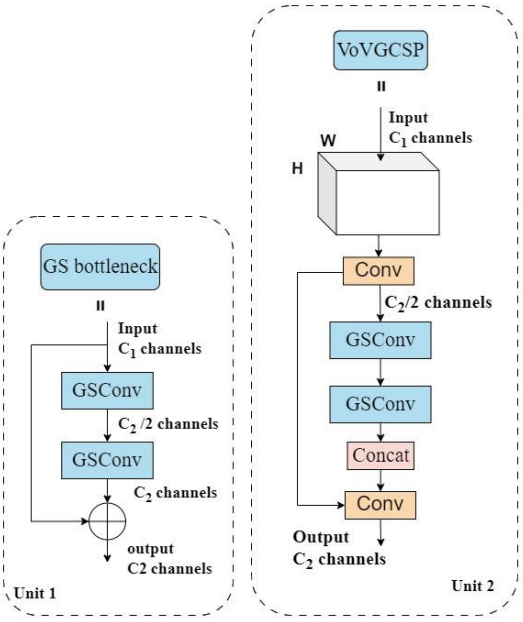



Figure 7. VoVGCSP Module

The GSConv module consists of the Conv, DWConv, Concat, and Channel Mixing sub-modules, illustrated in Figure 6. Its operational procedure is as follows: Initially, the input feature map, with C1 channels, undergoes standard convolution to produce a feature map with C2/2 channels. Subsequently, depthwise separable convolution generates another feature map with C2/2 channels. These two feature maps are concatenated to form a unified feature map with C2 channels. Finally, the channel mixing operation adjusts the output characteristics to the desired channel count. Through this approach, the GSConv module combines depthwise separable convolution and standard convolution to reduce computational complexity and improve overall recognition accuracy by addressing limitations in feature extraction and fusion.

The integration of GSConv convolution aims to simplify model complexity. To enhance model inference speed without sacrificing accuracy, we introduced the VOVGSCSP module, as depicted in Figure 7. In Figure 7, Unit 1 illustrates the bottleneck unit structure of VOVGSCSP, while Unit 2 showcases a cross-stage VOVGSCSP module employing a single aggregation method.

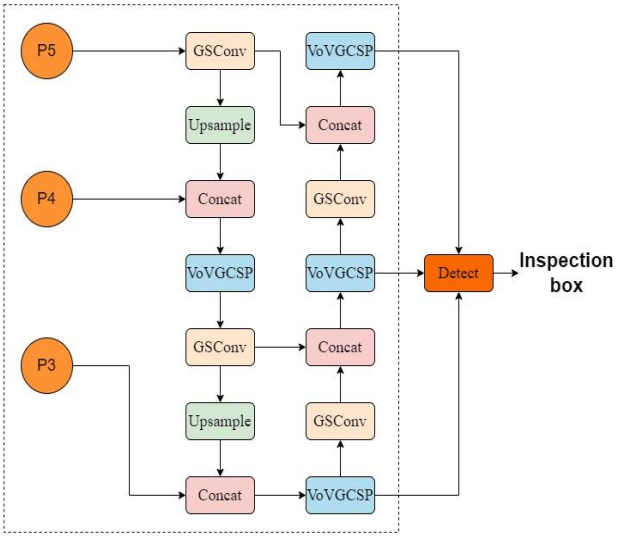



Figure 8. The positions of GSConv and VOVGCSP modules

## III. Experimental Testing and Result Analysis

### A. Experimental Design and Parameter Setting

In the identical operational setting and employing the UAV dataset, experimental tests are conducted to compare the enhanced ATD-YOLO algorithm model with other algorithm models. The experimental environment configuration is outlined in Table 1. This comparison seeks to validate the improved algorithm's efficacy in detecting low-altitude UAV targets.

Throughout the training phase, a combination of the cosine annealing learning rate decay method and the SGD algorithm is utilized. The training regimen spans 600 epochs with a batch size of 16 and a momentum of 0.937. Mosaic data augmentation is employed to enrich the backgrounds of images by randomly scaling, cropping, and arranging four training images in a mosaic pattern. This augmentation technique enhances the accuracy and robustness of small object detection.

TABLE I. Experimental Setup Configuration

| Name | Environment Configuration |
|---|---|
| System Environment | Ubuntu 22.04 |
| CPU | AMD Ryzen 9 5950X |
| GPU | RTX 4060 Ti 16GB |
| Deep Learning Framework | Pytorch 1.13.1 |
| IDE | CUDA 11.7 |

### B. Building a Relatively Comprehensive Dataset

In this study, extensive data from diverse sources and papers was reviewed and collected. Utilizing this data, we constructed a tailored dataset called "Anti-Mini Drone" for our research purposes. The Det-Fly dataset [18] addresses the lack of drone data from a single perspective by directly collecting images of target drones in the air, including various postures such as upward, downward, and forward views. However, this dataset only contains one type of drone, limiting its generality in detecting other types of drones. The Drone-vs-Bird dataset [19] not only covers rich drone and environmental data but also includes some bird data. This poses challenges when drones resemble birds in appearance, especially during long-distance observations. However, the drawback of this dataset is its inability to meet the detection requirements of other types of drones. The Real World dataset [20] contains various types of drones and environments sourced from YouTube videos, but the image resolution is low. Most of the data is captured from a forward and upward perspective, which implies certain limitations in drone detection from a downward view. The Multi-view drone tracking dataset [21] records drone flight trajectories from different angles using multiple consumer-grade cameras, but the environmental capture is relatively homogeneous. The DUT anti-UAV dataset [22] consists of both a detection dataset comprising 10,000 images and a tracking dataset containing 20 videos. However, it's worth noting that the distribution of target dimensions within the dataset is uneven. The Anti-UAV dataset [23] includes visible light and infrared data, but the problem is that the shooting environment is singular, suitable only for research on multi-modal

fusion tracking and the alignment between infrared and visible light cameras is not perfect in time and space.

Overall, the above drone datasets each have their own advantages and disadvantages, often overcoming only one or two difficulties in constructing drone datasets. By merging these six different datasets, a low-altitude drone dataset that meets the requirements of this study was constructed, making it more reflective of real outdoor flight scenarios for low-altitude drones. Table 2 illustrates the distribution of images.

TABLE II. ORIGIN OF THE DATASET AND QUANTITY OF IMAGES

| Dataset | Number of Images |
|---|---|
| Det-Fly | 3893 |
| Drone-vs-Bird | 3959 |
| Real World | 1525 |
| Multi-view drone tracking | 3447 |
| DUT anti-UAV | 3639 |
| Anti-UAV | 2767 |

In the Anti-Mini Drone dataset, Figure 9 demonstrates a notable prevalence of small targets. The majority of targets in the dataset exhibit aspect ratios less than 0.1 times the original image dimensions. This distribution aligns with the relative sizes of objects commonly encountered in real outdoor scenarios during low-altitude unmanned aerial vehicle (UAV) flights.

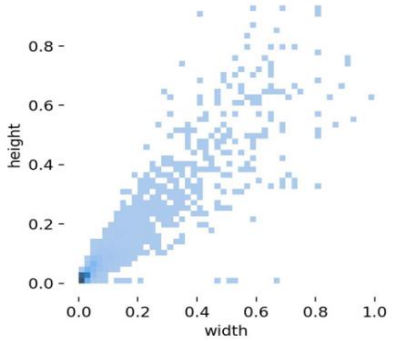



Figure 9. Length and Width Distribution Chart of the Anti-Mini Drone Dataset

*C. Experimental evaluation metrics*

To evaluate the enhanced ATD-YOLO algorithm, metrics such as parameter count, floating-point operations (GFLOPs), average precision (AP), and frames per second (FPS) are selected. Since the study focuses on detecting drones across different categories, mean average precision (mAP) and AP values are considered equivalent. Given the predominance of small objects, the mAP.5 criterion is adopted for evaluation to reflect the model's performance and speed accurately. Precision measures the proportion of correctly detected objects, while recall assesses the proportion of correctly predicted objects among all true objects:

$$\text{Precision} = \frac{\text{TP}}{TP + FP} \tag{1}$$

$$\text{Re}call = \frac{TP}{TP + FN} \tag{2}$$

FN represents false positive samples predicted by the model, TP denotes true positive samples predicted correctly, and FP stands for false negative samples. The average precision (AP) reflects the detection accuracy for a single class of targets, usually calculated by integrating the Precision-Recall (P-R) graph:

$$AP = \int_0^1 P(R)dR \tag{3}$$

Detection speed is frequently measured in FPS (Frames Per Second), indicating the number of images processed by the object detection network per second. A higher FPS value signifies faster processing speed. The expression for FPS is given by Formula 4:

$$FPS = \frac{FrameNum}{ElapsedTime} \tag{4}$$

(a) Sky Background (b) Treetops Background (c) Dusk Background (d) Evening Background

Figure 10. Samples of simple background from Anti-Mini Drone

(a) Mountainous background (b) Wooded background (c) Similar colors (d) Semi-obscured

Figure 11. Samples of complex background from Anti-Mini Drone

### *D. Experimental Results Analysis*

*a)Comparative Experiment on Lightweight Modules:* Experimental analysis confirmed the effectiveness of the C3F module. Replacing the C3 module in YOLOv5s with C3F increased mAP.5 by 0.1%, while reducing parameters and floating-point operations by 9.7% and 12.6%, respectively, and improving FPS by 6.26. Substituting with C2f increased mAP.5 accuracy by 0.9%, but raised parameters and floating-point operations by 17.6% and 22.7%, respectively, while decreasing FPS by 13.41. Replacing with C2f-Faster decreased mAP.5 by 0.3%, with parameters and floating-point operations decreasing by 8.2% and 6.1%, respectively, and FPS increasing by 8.78. These results validate the effectiveness of the C3F module in achieving high accuracy with minimal algorithmic overhead.

TABLE III. CONTRAST EXPERIMENT OF ATTENTION MODULE

| Module | mAP.5/% | GFLOP /G | Params/106 | FPS |
|---|---|---|---|---|
| C3 | 92.2 | 15.8 | 7.01 | 68.79 |
| C3F | 92.3 | 13.8 | 6.33 | 75.05 |
| C2f | 93.1 | 19.4 | 8.25 | 55.38 |
| C2f-Faster | 91.9 | 14.5 | 6.58 | 77.57 |

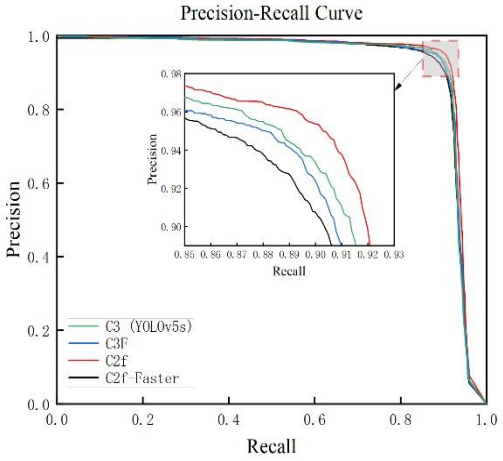



Figure 12. PR curves for various feature extraction modules (IOU=0.5)

Figure 12 illustrates the PR curves of various feature extraction modules at IOU = 0.5. It also demonstrates that the improved algorithm's Precision and Recall with the C3F module are slightly lower than those with the original C3 module. However, considering the improvement in detection accuracy, parameter count, floating-point operations, and frame rate, the improved model still exhibits superiority in target localization regression.

*b)Comparative Experiment on Attention Mechanism Modules:* In response to the observed increase in model parameter count, floating-point operations, and FPS after adding EMA, various attention mechanisms were replaced at the

original position for comparative experiments. Table 4 presents the experimental results. The improved algorithm with EMA sacrifices some performance compared to other attention mechanism algorithms but achieves better detection accuracy.

TABLE IV. CONTRAST EXPERIMENT OF ATTENTION MODULE

| Module | mAP.5/% | GFLOP /G | Params/106 | FPS |
|---|---|---|---|---|
| SE[24] | 91.8 | 13.8 | 6.37 | 74.93 |
| ECA[25] | 92.2 | 13.8 | 6.34 | 74.37 |
| CBAM[26] | 92.3 | 13.8 | 6.37 | 72.63 |
| CA[27] | 91.2 | 13.8 | 6.36 | 72.87 |
| EMA | 92.7 | 14.1 | 6.38 | 69.83 |

Figure 13 illustrates the PR curves comparing various attention mechanisms at IOU = 0.5. It also demonstrates that the EMA attention mechanism outperforms other attention mechanisms in both Precision and Recall.

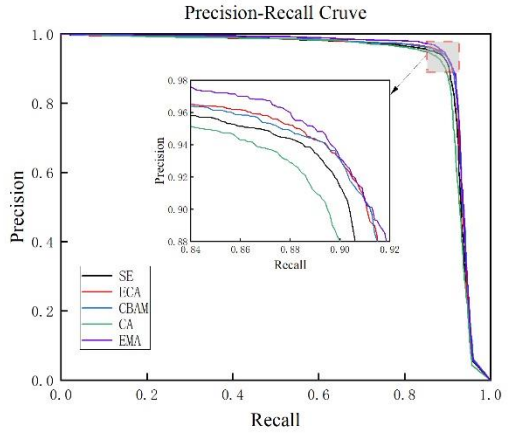



Figure 13. PR curves for various feature extraction modules (IOU=0.5)

*c)Ablation Experiment:* Ablation experiments were conducted on the Anti-Mini Drone dataset to test different model configurations, including variations in the C3F lightweight feature extraction module, EMA attention mechanism module, CARAFE upsampling module, and Slim-Neck module. Results from these experiments are summarized in Table 5.In the first row, the initial YOLOv5s model achieved a detection accuracy of 92.2%. By replacing the C3 module with the C3F module, the parameter count decreased by about 17.2%, while floating-point operations increased by 5.6%. The mAP.5 increased by 0.1 percentage points, and the FPS improved by 6.26, indicating that adopting FasterNet to enhance the C3 module effectively reduces model parameters while maintaining detection capabilities.In the third row, adding the EMA attention mechanism module increased mAP.5 by 0.4% compared to the second row. However, parameter count and floating-point operations increased by 1.1% and 2.1%, respectively, while FPS decreased by 5.22.In the fourth row, after introducing the CARAFE upsampling module, parameter count increased by about 0.8%, while floating-point operations increased by 0.7%. The mAP.5 increased by 0.4 percentage points, while FPS decreased by 1.98.In the fifth row, embedding the Slim-Neck network resulted in a decrease of 18.28% in parameter count and 21.98% in floating-point operations compared to the fourth row. Despite a slight decrease of 0.3% in mAP.5, FPS increased by 7.5. Compared to the initial model in the first row, parameter count and floating-point operations decreased by 25.39% and 30.37%, respectively, while mAP.5 and FPS increased by 0.5% and 6.56%, respectively.

TABLE V. RESULTS OF ABLATION EXPERIMENTS

| YOLOv5s | C3F | EMA | CARFE | Slim-Neck | Params/106 | GFLOP/G | mAP.5/% | FPS |
|---|---|---|---|---|---|---|---|---|
| √ | | | | | 7.01 | 15.8 | 92.2 | 68.79 |
| √ | √ | | | | 6.33 | 13.8 | 92.3 | 75.05 |
| √ | √ | √ | | | 6.38 | 14.1 | 92.7 | 69.83 |
| √ | √ | √ | √ | | 6.40 | 14.1 | 93.1 | 67.85 |
| √ | √ | √ | √ | √ | 5.23 | 11.0 | 92.8 | 75.35 |

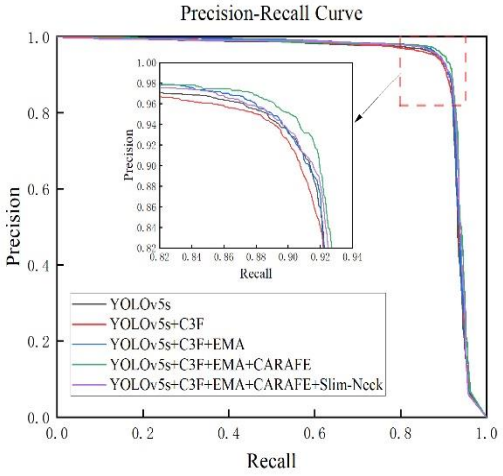



Figure 14. Model PR curve (IOU=0.5)

The findings indicate improved detection accuracy with reduced parameter count and floating-point operations, confirming the effectiveness of the enhancement strategies. Figure 14 displays PR curves of the target detection algorithm at an IOU of 0.5, showing that the improved algorithm outperforms the original in both Precision and Recall, highlighting its superior performance in target localization regression.

*d)Comparative Experiment on Mainstream Algorithms:* Experimental analysis compared the effectiveness of the C3F, C2f, and C2f-Faster modules in feature extraction networks. Results in Table 3 show that replacing C3 with C3F increased mAP.5 by 0.1%, with reduced parameters and floating-point operations and increased FPS. Substituting with C2f improved mAP.5 by 0.9% but increased parameters and floating-point operations while decreasing FPS. Replacing with C2f-Faster resulted in a 0.3% decrease in mAP.5, with reduced parameters and floating-point operations and increased FPS. These results confirm the effectiveness of the C3F module in achieving high accuracy with minimal overhead, demonstrating its superiority in lightweight feature extraction.

TABLE VI. MAINSTREAM ALGORITHM COMPARATIVE EXPERIMENT RESULTS

| Module | Params/106 | GFLOP/G | AP.5/% | FPS |
|---|---|---|---|---|
| YOLOv3 Tiny | 8.66 | 12.9 | 79.1 | 166.67 |
| YOLOv5s | 7.01 | 15.9 | 92.2 | 68.79 |
| YOLOv7 Tiny | 6.01 | 13.2 | 88.4 | 63.30 |
| YOLOv8s | 11.12 | 28.4 | 89.0 | 109.89 |
| ATD-YOLO | 5.23 | 11.0 | 92.8 | 75.35 |

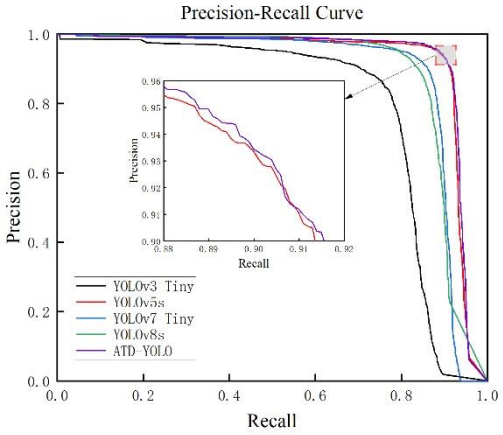


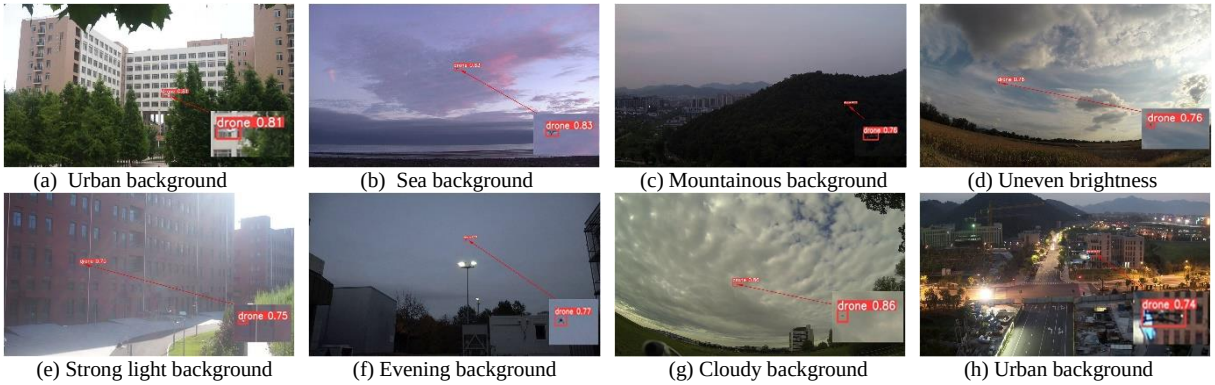

Figure 15. PR curves of mainstream algorithms on the test set (IOU=0.5)

Figure 15 displays PR curves for various algorithms on the test set at IOU = 0.5, indicating the improved algorithm's superiority in both Precision and Recall.

## *E. Analysis of Comparative Experiment Results*

*a) Comparative Experiment:* The improved algorithm's detection results in various scenarios are intuitively and clearly demonstrated in Figure 12, indicating its superiority over the original model in drone object detection across different scenes, primarily reflected in confidence and detection outcomes. Specific scenarios include background with buildings (Figure 16 (a)), flying over the sea (Figure 16 (b)), flying in mountainous areas (Figure 16 (c)), flying under uneven brightness conditions (Figure 16 (d)), flying in strong sunlight conditions (Figure 16 (e)), flying in the evening (Figure 16 (f)), flying with cloud backgrounds (Figure 16 (g)), and flying in urban backgrounds (Figure 16 (h)).In the background with buildings, even with interference

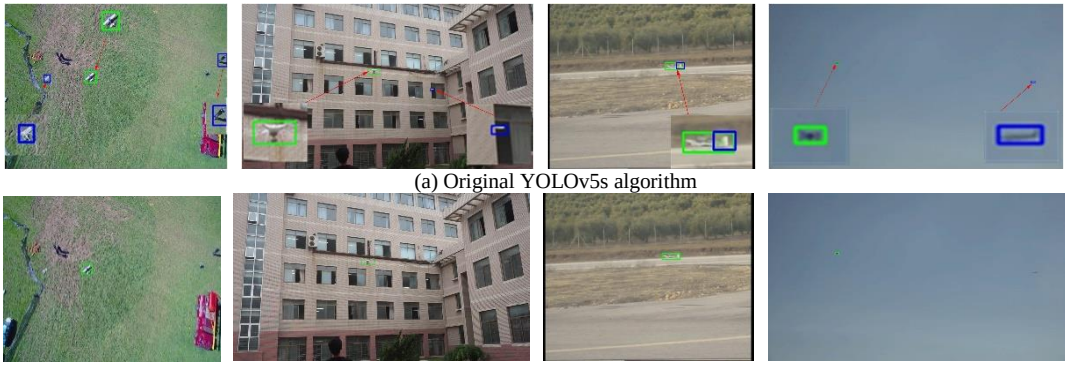
such as trees and buildings, where the background is complex and the drone is similar in height to the background, the improved algorithm can successfully detect the target. In other scenarios such as flying over the sea, in mountainous areas, under uneven lighting, in strong sunlight, in the evening, with cloud backgrounds, and in urban backgrounds, the improved algorithm also performs admirably, successfully detecting drone targets without missing or false detections. This firmly establishes the effectiveness of the improved algorithm in complex scenarios.




(a) Urban background (b) Sea background (c) Mountainous background (d) Uneven brightness

(e) Strong light background (f) Evening background (g) Cloudy background (h) Urban background

Figure 16. Object detection outcomes in diverse scenarios



(a) Original YOLOv5s algorithm

(b) ATD-YOLO algorithm

Figure 17. Object detection outcomes in a consistent scenario

*b) Comparative Study:* We compared the original YOLOv5s algorithm with the enhanced ATD-YOLO algorithm on a test dataset. In Figure 17, green boxes show correct identifications, while blue boxes indicate misidentifications. The original algorithm sometimes misidentified pedestrians and debris as targets when a drone was on the lawn. However, the improved algorithm performed better, correctly identifying targets and avoiding mistakes like thinking reflections were drones or confusing a drone's tail with the whole drone. Overall, the improved algorithm is better at detecting drones, making fewer mistakes while still being efficient.

## IV. Conclusions

In addressing the challenge of improving the simultaneous accuracy and detection efficiency of low-altitude UAV target detection algorithms, this paper presents an enhanced algorithm, ATD-YLO, built upon YOLOv5s. This algorithm successfully achieves lightweight target detection, aiming to maintain detection precision and efficiency in low-altitude UAV detection tasks under limited hardware resource platforms.

ATD-YOLO introduces several innovations to improve performance. It includes PConv, a new convolutional layer, and C3F, a lightweight feature

extraction module inspired by FasterNet. C3F replaces the original C3 module, reducing parameters and computations while maintaining recognition accuracy. EMA, an attention mechanism module, is also integrated to extract key information from images while ignoring irrelevant data, enhancing detection accuracy. Furthermore, the introduction of CARAFE, a generic upsampling module, increases the receptive field for feature fusion, and Slim-Neck, a lightweight network, further promotes network efficiency.

The effectiveness of the proposed approach was validated by training and validating the improved ATD-YOLO algorithm on the Anti Mini Drone dataset. Experimental results revealed an accuracy increase from 92.2% to 92.8% compared to the initial algorithm. Furthermore, the improved algorithm reduced parameter count and floating-point computations by 31.4% and 28.9%, respectively, while achieving a detection speed of 75.35 FPS. The improved algorithm outperforms YOLOv3 Tiny, YOLOv7 Tiny, and YOLOv8s in recognition accuracy by 13.7%, 4.4%, and 3.8%, respectively, with model parameter counts of 60.39%, 87.02%, and 47.03% of theirs, and floating-point computations of 85.27%, 83.33%, and 38.73% of theirs, respectively. The FPS is 12.35 higher than that of YOLOv7 Tiny, but only 45.21% and 68.56% of YOLOv3 Tiny and YOLOv8s, respectively. Therefore, ATD-YOLO exhibits promising performance and meets the lightweight detection requirements for UAVs. In the next phase of research, efforts will focus on dataset expansion to include more categories such as birds in flight and other airborne objects, as well as improving network detection speed.

## REFERENCES


[1] Zhao F, Zhao C, Guo J.Visual perception-based anti-drone technology: Development dynamics and trend [J]. National Defense Technology, 44(05), 35-45. DOI: 10.13943/j.issn1671-4547.2023.05.05.

[2] Oh H M, Lee H, Kim M Y .Comparing Convolutional Neural Network(CNN) models for machine learning-based drone and bird classification of anti-drone system [C]//2019 19th International Conference on Control, Automation and Systems (ICCAS).2019. DOI:10.23919/ICCAS47443. 2019. 8971699.

[3] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[4] Girshick R. Fast r-cnn [C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[5] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks [J]. Advances in neural information processing systems, 2015, 28.

[6] He K, Gkioxari G, Dollár P, et al. Mask r-cnn [C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.

[7] 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.

[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

[9] Redmon J, Farhadi A. Yolov3: An incremental improvement [J]. arXiv preprint arXiv:1804.02767, 2018.

[10] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector [C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[11] LU Q, YU Y Q, XU D M, et al. Improved YOLOv5 Small Drones Target [J]. Computer Science, 2023, 50(S2): 212-219.

[12] YANG H Y, RONG Y S, JIAN Y H, et al. GCB-YOLOv5s algorithm for real-time detection for a low altitude UAV [J]. Journal of Ordnance Equipment Engineering, 2023, 44(07): 1-8.

[13] BAO W Q, XIE L Q, XU C, et al. A Real-time detection method of micro UAV based on YOLOv5 [J]. Journal of Ordnance Equipment Engineering, 2022, 43(05): 232-237.

[14] Chen J, Kao S, He H, et al. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 12021-12031.

[15] Wang J, Chen K, Xu R, et al. Carafe: Content-aware reassembly of features [C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3007-3016.

[16] Ouyang D, He S, Zhang G, et al. Efficient Multi-Scale Attention Module with Cross-Spatial Learning [C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.

[17] Li H, Li J, Wei H, et al. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles [J]. arXiv preprint arXiv: 2206.02424, 2022.

[18] Zheng Y, Chen Z, Lv D, et al. Air-to-Air Visual Detection of Micro-UAVs: An Experimental Evaluation of Deep Learning [J].IEEE Robotics and Automation Letters, 2021, PP(99): 1-1. DOI: 10.1109/LRA. 2021. 3056059.

[19] Coluccia A, Fascista A, Schumann A, et al. Drone vs. Bird Detection: Deep Learning Algorithms and Results from a Grand Challenge [J]. Sensors, 2021, 21(8): 2824. DOI:10.3390/s21082824.

[20] Pawelczyk M L, Wojtyra M .Real World Object Detection Dataset for Quadcopter Unmanned Aerial Vehicle Detection [J]. IEEE Access, 8:174394-174409 [2023-10-12]. DOI: 10.1109/ACCESS.2020. 3026192.

[21] Li J, Murray J, Ismaili D, et al. Reconstruction of 3D flight trajectories from ad-hoc camera networks [C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020: 1621-1628.

[22] J. Zhao, J. Zhang, D. Li and D. Wang, "Vision-Based Anti-UAV Detection and Tracking [J]. IEEE Transactions on Intelligent Transportation Systems, Dec. 2022, DOI: 10.1109/TITS. 2022.3177627.

[23] Jiang Nan, Wang Kuiran, Peng Xiaoke, et al. Anti-UAV: A large multi-modal benchmark for UAV tracking [J]. arXiv preprint arXiv, 2021. 2101(2), 1-13.

[24] Hu J, Shen L, Sun G. Squeeze-and-excitation networks [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 7132-7141.

[25] Wang Q L, Wu B G, Zhu P F, et al. ECA-net: efficient channel attention for deep convolutional neural networks [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020.

[26] Woo S, Park J, Lee J Y, et al. CBAM: convolutional block attention module [M]//Ferrari V, Hebert M, Sminchisescu C, et al. Computer vision-ECCV 2018. Lecture notes in computer science. Cham: Springer, 2018, 11211: 3-19.

[27] Hou Q B, Zhou D Q, Feng J S. Coordinate attention for efficient mobile network design [C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 20-25, 2021, Nashville, TN, USA. New York: IEEE Press, 2021: 13708-13717.

# Securing Operating Systems (OS): A Comprehensive Approach to Security with Best Practices and Techniques

Zarif Bin Akhtar
MPhil Research Postgraduate Student, Master of Philosophy (MPhil) in Machine Learning and Machine Intelligence, Department of Engineering, University of Cambridge, United Kingdom
E-mail: zarifbinakhtarg@gmail.com; zarifbinakhtar@ieee.org

***Abstract*—Operating system (OS) security is paramount in ensuring the integrity, confidentiality, and availability of computer systems and data. This research manuscript presents a comprehensive investigation into the multifaceted domain of OS security, aiming to enhance understanding, identify challenges, and propose effective solutions. The research methodology integrates diverse approaches, including an extensive exploration for available knowledge process mechanics, empirical data collection, case studies investigations, experimental analysis, comparative studies, qualitative analysis, synthesis, and interpretation. Through various experimental perspectives, theoretical foundations, historical developments, and current trends in OS security are also explored. Empirical data collection involves gathering insights from publicly available reports, security advisories, case studies, and expert interviews to capture real-world perspectives and experiences. Case studies illustrate practical implications of security strategies, while experimental analysis evaluates the efficacy of security measures in controlled environments. Comparative studies and qualitative analysis provide insights into strengths, limitations, and emerging trends in OS security. The synthesis and interpretation of the findings offer actionable insights for improving OS security practices, policy recommendations, and providing towards future research directions. This research contributes to advancing knowledge in OS security and informs the development of effective strategies to safeguard computer systems against evolving threats and vulnerabilities.**

***Keywords-Computing; Cryptography; Data Security; Network Security; Operating Systems (OS); OS Security; Privacy; Security.***

## I. INTRODUCTION

Operating systems (OS) serve as the backbone of modern computing, facilitating the management of hardware resources and enabling users to interact with software applications. However, the increasing complexity and interconnectedness of computer systems have made OS security a critical concern. Ensuring the integrity, confidentiality, and availability of operating systems is essential for safeguarding sensitive data, protecting against malicious attacks, and maintaining system functionality. This research manuscript delves into the multifaceted domain of operating system security, exploring various strategies, threats, and solutions aimed at enhancing the security posture of modern computing environments. In today's interconnected world, where cyber threats loom large, understanding the intricacies of OS security is paramount for organizations and individuals alike.

The manuscript begins by delineating the fundamental concepts of OS security, illuminating the importance of protection mechanisms in controlling access to system resources. It examines the distinction between security and protection, emphasizing the role of security measures in guarding against external threats and internal vulnerabilities. Passwords, encryption, and access control mechanisms emerge as foundational pillars of OS security, ensuring that data and programs are utilized only by authorized users in a prescribed manner.

Subsequently, the manuscript delves into the myriad threats that pose a risk to operating systems, ranging from malware and network intrusions to buffer overflow techniques. Malicious software, including viruses, worms, and Trojan horses, presents a pervasive threat to system integrity, capable of compromising data,

disrupting operations, and facilitating unauthorized access. Network intrusions and buffer overflow techniques exploit vulnerabilities in system architecture, underscoring the need for robust security measures to mitigate these risks.

Against this backdrop of evolving threats, the manuscript explores strategies and solutions for enhancing operating system security. Authorization, authentication, and access control mechanisms emerge as pivotal tools for verifying user identities and regulating resource access. Furthermore, the manuscript delves into advanced security measures such as encryption techniques, intrusion detection systems, and firewall configurations, aimed at fortifying system defenses and thwarting malicious activities.

This research manuscript offers a comprehensive examination of operating system security, delving into the underlying principles, emerging threats, and proactive measures for safeguarding modern computing environments. By expounding the intricacies of OS security, this manuscript aims to empower readers with the knowledge and tools needed to bolster the security posture of their operating systems and mitigate potential risks effectively.

## II. Methods and Experimental Analysis

This research adopts a comprehensive approach to investigate operating system (OS) security, encompassing various research methods to provide a thorough understanding of the subject matter. The methodology commences with a rigorous background research, which involves inspecting scholarly articles, research papers, textbooks, and reputable online resources to gain insights into the theoretical underpinnings and historical evolution of OS security. By synthesizing existing knowledge, this exploration lays the foundation for the subsequent phases of the research. Building upon the nonfiction evaluation, empirical data is collected from diverse sources to enrich the understanding of OS security practices and challenges. This data collection process includes accessing publicly available reports on cyber threats and vulnerabilities, analyzing security advisories from software vendors, studying case studies of security breaches, and examining empirical studies surrounding OS security implementations. Additionally, insights are gathered from security forums, online communities, and expert interviews to capture real-world perspectives and experiences.

The methodology employs case studies to provide concrete illustrations of OS security strategies and their practical implications. These case studies encompass real-world scenarios of security incidents, successful security implementations, and the ramifications of security lapses. Through in-depth analysis of specific cases across various industries and organizational contexts, this research aims to explain the effectiveness of different security measures and their impact on system resilience. Furthermore, experimental analysis is conducted in controlled environments to complement theoretical insights and empirical observations.

This experimental phase involves deploying testbeds comprising different operating systems and security configurations. Various security tools, techniques, and countermeasures are evaluated for their efficacy in mitigating common threats such as malware, network intrusions, and buffer overflow attacks. Performance metrics are measured to assess the effectiveness of security solutions and their implications for system performance. Additionally, comparative studies are conducted to evaluate the strengths and limitations of different OS security approaches. Comparative analyses involve benchmarking security features, performance metrics, and usability aspects across multiple operating systems, security products, and architectures.

By comparing diverse security solutions and their implementations, this research aims to identify best practices, emerging trends, and areas for improvement in OS security. Qualitative analysis techniques, such as content analysis and thematic coding, are employed to analyze textual data gathered from literature reviews, case studies, and expert interviews. Qualitative analysis aims to identify recurring themes, patterns, and insights related to OS security practices, challenges, and emerging trends. The findings from qualitative analysis are integrated with quantitative data to

provide a comprehensive understanding of OS security dynamics retrospective.

Finally, the research synthesizes and interprets findings derived from works examinations, data collection, case studies, experimental analysis, comparative studies, and qualitative analysis. Through this synthesis and interpretation, the research aims to develop coherent narratives, theoretical frameworks, and actionable insights that contribute to the advancement of OS security knowledge and practice.

## III. Background Research and Iterative Exploration for associated Available Knowledge

Operating system security (OS security) involves implementing measures to protect the integrity, confidentiality, and availability of an operating system (OS). It encompasses various techniques and methods aimed at safeguarding the OS from threats such as viruses, malware, unauthorized access, and remote intrusions by hackers. These measures include regularly updating the OS with patches, installing and updating antivirus software, monitoring network traffic with firewalls, and managing user accounts to ensure they have only the necessary privileges. By implementing these preventive-control techniques, OS security aims to prevent unauthorized access, data breaches, and other security incidents that could compromise the functioning and security of the operating system and the data it handles. Operating system security encompasses a range of measures and techniques aimed at safeguarding the integrity, confidentiality, and availability of an operating system (OS). It involves preventing unauthorized access to system resources and ensuring that data and programs are used only by authorized users and in desired manners. Protection mechanisms are implemented to control access to resources by programs, processes, or users, thereby enabling safe sharing of common namespaces like directories or files in multiprogramming operating systems. Passwords serve as the primary security tool, ensuring that only authorized users can access the system. Encryption techniques are used to maintain the confidentiality of passwords and other sensitive information. Additionally, OS protection measures come into play when determining access privileges for files shared among users, with the OS enforcing strict adherence to specified access privileges [1-11]. The primary goals of an OS security system are to ensure integrity, secrecy, and availability. Integrity involves preventing unauthorized users from altering vital system files and resources, while secrecy ensures that only authorized users can access system objects, with restricted access to system files. Availability ensures that system resources are not monopolized by a single user or process, preventing service denial situations. OS security measures are designed to protect against various threats, including malware, network intrusions, and buffer overflow attacks. Malware refers to malicious software designed to harm computer systems or users, while network intrusion detection systems (IDS) monitor network traffic for malicious transactions and alert administrators to potential threats. Buffer overflow attacks exploit vulnerabilities in systems by overwriting adjoining memory areas with malicious code disguised as data, potentially leading to security breaches [12-21].

To ensure OS security, various preventive measures are implemented. Authorization and authentication mechanisms verify access to system resources and authenticate users' identities, respectively. Access controls prevent unauthorized browsing of system files and trapdoors, while invalid parameters and line tapping can lead to security violations if not properly managed. Additionally, electronic data capture techniques and rogue software pose threats to system security if not adequately addressed. Proper access controls and waste recovery mechanisms are essential to mitigate these risks and ensure the overall security of the operating system [22-26].

Operating system security involves implementing measures to protect system integrity, confidentiality, and availability while preventing unauthorized access and ensuring the safe sharing of resources among users. By employing authentication, access controls, and encryption techniques, OS security aims to mitigate various threats such as malware, network intrusions, and

buffer overflow attacks, thereby safeguarding the overall functionality and security of the operating system.

## IV. The Security Problem

The prospect of security addresses the protection of systems from deliberate attacks, whether internal or external, aimed at stealing information, damaging data, or causing disruption. It distinguishes between accidental misuse and intentional attacks. There are many common types of security violations.

***Breach of Confidentiality:*** Involves theft of private or confidential information like credit card numbers, trade secrets, or financial data.

***Breach of Integrity:*** Unauthorized modification of data, which can have serious consequences such as opening security holes or altering program source code.

***Breach of Availability:*** Involves unauthorized destruction of data, often for the purpose of causing havoc or vandalism.

***Theft of Service:*** Unauthorized use of resources like CPU cycles or network services.

***Denial of Service (DoS):*** Preventing legitimate users from using the system by overwhelming it with excessive requests.

It terms of the security problem identification aspect, mainly four levels of protection that a system must have to ensure apex mobility.

***Physical:*** Protecting physical access to resources, including preventing theft of backup tapes and controlling access to the root console.

***Human:*** Ensuring that humans with access to the system are trustworthy and cannot be coerced into breaching security, while also addressing vulnerabilities like social engineering, phishing, dumpster diving, and password cracking.

***Operating System:*** Protecting the operating system from security breaches such as denial of service, memory-access violations, and excessive privilege execution.

***Network:*** Protecting both the network itself and the local system from attacks, particularly important as network communications and portable devices become more prevalent.

The interval position levels emphasize the importance of understanding and implementing security measures to protect systems from deliberate attacks and maintain confidentiality, integrity, and availability of data and resources. To better understand figure 1 provides a visualization in terms of standard security attacks.

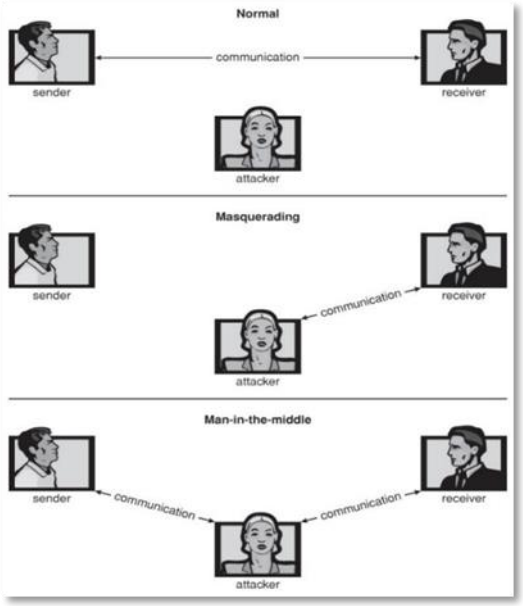



Figure 1. A Visualization of Standard security attacks

## V. The Program Threats

Program threats are a significant concern for modern systems, and several common threats which usually takes place and are incurred are mentioned and explored with associated cases.

***Trojan Horse:*** A Trojan Horse is a program that performs malicious actions while appearing to perform legitimate functions. It can be intentionally designed or result from legitimate programs being infected with viruses. Classic examples include login emulators that steal account credentials and spyware that gathers user information covertly.

***Trap Door:*** A Trap Door is a deliberate security hole inserted by a designer or programmer for future access to the system. Once a system has been compromised by a trap door, it

can never be fully trusted again, even if restored from backup tapes.

***Logic Bomb:*** Logic Bombs are code designed to execute malicious actions only under specific conditions, such as a particular date or event. An example is the Dead-Man Switch, which triggers when a designated user fails to log in regularly.

***Stack and Buffer Overflow:*** Exploiting bugs in system code, this attack occurs when buffers overflow, allowing the attacker to overwrite adjacent memory areas, including the return address. By overflowing the buffer with malicious code and altering the return address, attackers can execute their code and potentially gain unauthorized access to the system.

***Viruses:*** Viruses are code fragments embedded in legitimate programs, designed to replicate and cause harm. Various types include file viruses, boot viruses, macro viruses, and polymorphic viruses, each with unique characteristics and methods of spreading. Viruses often spread through Trojan Horses, email attachments, or unsafe downloads. Some viruses, like the 2004 virus targeting Microsoft products, exploit vulnerabilities to infect systems and propagate rapidly. The existence of monocultures, where most systems run the same software, can increase the vulnerability and potential harm caused by viruses.

Understanding and mitigating program threats is crucial for maintaining the security and integrity of modern systems. Measures such as robust security protocols, regular software updates, and user education are essential in combating these threats and protecting sensitive data and resources.

In order to provide a better understanding on the perspective of the matter, figure 2 provides the necessary illustration of the technical computing in line with program threats with their associate layout frame configuration process functionalities involved through the cycle of the frameworks.

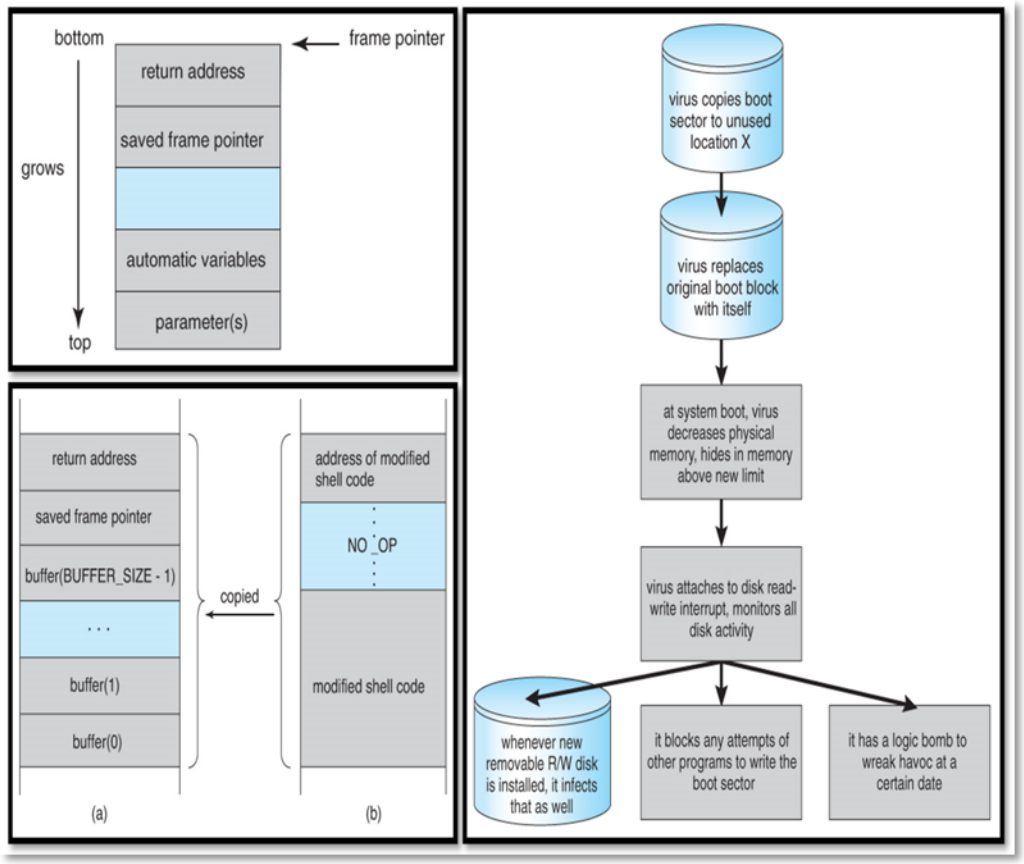



Figure 2. An illustration of Program Threats (On the left with the layout for a typical stack frame, Hypothetical stack frame for (a) before then (b) after, on the right A boot-sector computer virus)

## VI. The System and Network Threats

System and network threats pose significant risks to the security and functionality of modern computing environments. This segment explores various threats targeting operating systems and networks, or leveraging these systems to launch attacks.

***Worms:*** Worms are processes that replicate themselves to consume system resources and wreak havoc. The Morris Internet worm, launched in 1988, rapidly spread across the early Internet, exploiting vulnerabilities in common utilities like rsh, finger, and sendmail. Once on a system, the worm systematically attempted to discover user passwords and propagate to other systems. Rapid network connectivity led to the worm's quick demise, but it raised concerns about the potential for widespread damage from such attacks.

***Port Scanning:*** Port scanning involves systematically attempting to connect to every known or possible network port on a remote machine to identify vulnerabilities. It is often conducted from compromised systems (zombies) and can lead to the exploitation of security flaws. Port scanning tools like nmap and nessus are also used by administrators to identify weaknesses in their own systems without exploiting them.

***Denial of Service (DoS):*** DoS attacks aim to overwhelm systems with excessive requests, rendering them unusable for legitimate users. Attack methods include tight loops requesting system services, social engineering tactics like chain letters, and locking accounts after failed login attempts. While some DoS attacks are deliberate, others may occur unintentionally due to legitimate factors like sudden traffic spikes or inexperienced users.

These threats highlight the importance of robust security measures, regular system updates, and user education to mitigate risks and protect against potential damage or disruption to systems and networks. Additionally, the use of defensive tools and proactive monitoring can help identify and address vulnerabilities before they are exploited by attackers. Concerning the Morris internet worm an illustration of it is provided within figure 3 in terms of the technicality of the matter.

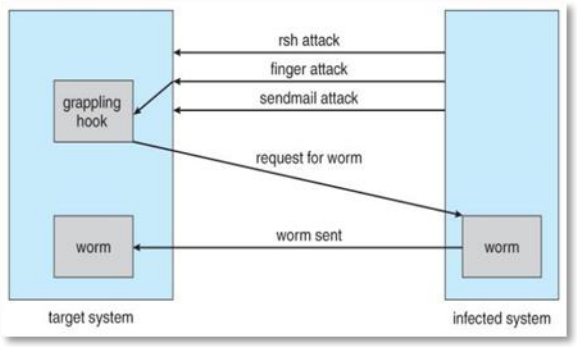



Figure 3. The Morris Internet worm an illustration

## VII. Cryptography as a Security Tool

Cryptography serves as a vital tool in ensuring the security of communications, particularly in the context of network transmissions where messages can be intercepted or altered by malicious actors. Two primary concerns in network security are trust and confidentiality, both of which cryptography addresses through the use of keys and encryption algorithms.

***Encryption:*** Encryption transforms a plaintext message into ciphertext using an encryption algorithm and a secret key, ensuring that only the intended recipient with the corresponding decryption key can decipher the message. Symmetric encryption uses the same key for both encryption and decryption, while asymmetric encryption employs separate keys for encryption (public key) and decryption (private key). Common symmetric encryption algorithms include DES, Triple DES, AES, Twofish, RC5, and RC4. Asymmetric encryption algorithms include RSA. Encryption ensures confidentiality by preventing unauthorized access to sensitive information during transmission over insecure networks.

***Authentication:*** Authentication verifies the identity of message senders and ensures message integrity. Hash functions generate fixed-size message digests from input data, providing a compact representation of the original message. Message-authentication codes (MACs) use symmetric encryption to authenticate message integrity. Digital signatures, part of asymmetric

encryption, provide authentication and non-repudiation, ensuring that the sender cannot deny sending a message.

***Key Distribution:*** Symmetric key distribution is challenging due to the need to securely transmit keys, but asymmetric encryption simplifies this process by allowing the public key to be freely shared while keeping the private key secret. Digital certificates, signed by trusted third parties, validate the authenticity of public keys, mitigating the risk of man-in-the-middle attacks.

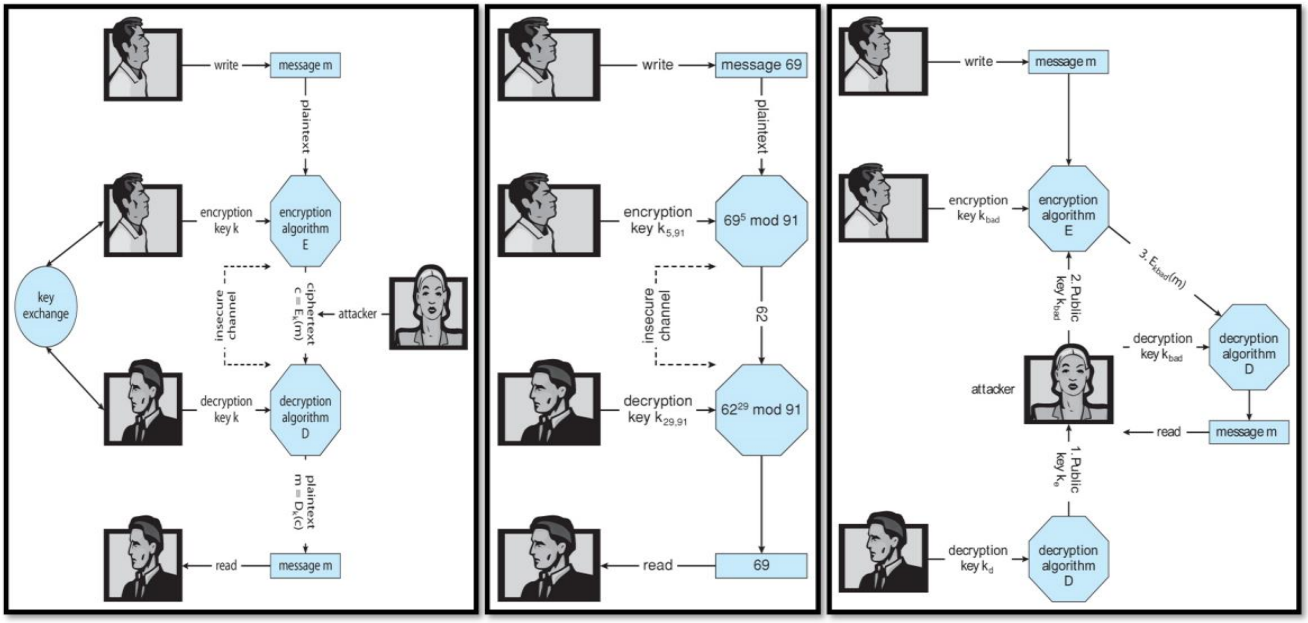



Figure 4. Cryptography Security Tool in action (on the left A secure communication over an insecure medium, in the middle Encryption and decryption using RSA asymmetric cryptography, on the right A man-in-the-middle attack on asymmetric cryptography)

***Implementation of Cryptography:*** Cryptography can be implemented at various network layers, each with its advantages and limitations. IPSec secures network-layer communications, while SSL/TLS (Secure Sockets Layer/Transport Layer Security) secures transport-layer communications, commonly used in web browsers for secure communication with web servers. SSL/TLS employs session keys for symmetric encryption, ensuring secure communication between clients and servers.

Cryptography, through encryption, authentication, and key distribution mechanisms, plays a critical role in securing network communications, safeguarding data confidentiality, authenticity, and integrity in the face of potential threats and vulnerabilities. Its implementation at different layers of the network stack ensures comprehensive protection against various security risks. To provide the mechanics and functionalities of cryptography as a security tool figure 4 provides an illustration in action in terms of network security.

## VIII. The User Authentication Perspective

User authentication is a critical aspect of computer security, ensuring that only authorized individuals can access resources and perform specific tasks. The most common form of user authentication is through passwords, although various vulnerabilities exist with this method.

***Passwords:*** Passwords are widely used for user authentication, where possession of the correct password confirms the user's identity. Vulnerabilities associated with passwords include guess ability, shoulder surfing, packet sniffing and potential for being written down or shared with others. Systems often have configurable parameters for password generation and

enforcement, such as minimum length, frequency of change, and history checks.

***Encrypted Passwords:*** Modern systems encrypt passwords before storing them, ensuring they are not stored in clear text form. Encrypted passwords are stored in files with restricted access, typically readable only by the superuser. Random seeds are included in the encryption process to prevent identical plaintext passwords from generating the same encrypted password.

***One-Time Passwords:*** One-time passwords enhance security by resisting attacks like shoulder surfing. They are often based on challenges and responses or electronic cards with constantly changing numbers. Two-factor authentication may be used with one-time passwords, requiring an additional traditional password for added security.

***Biometrics:*** Biometric authentication relies on physical characteristics of users that are difficult to forge or duplicate. Examples include fingerprint scanners, palm readers, retinal scanners, voiceprint analyzers, etc. Biometrics provide high security but may face challenges in cases of physiological changes or injuries.

User authentication methods aim to strike a balance between security and convenience, with each method having its own advantages and vulnerabilities. While passwords remain the most common form of authentication, newer methods like one-time passwords and biometrics offer additional layers of security, albeit with their own considerations and challenges. Effective user authentication is crucial for protecting sensitive data and ensuring system integrity in computing environments.

## IX. The Implementation of Security Defenses

Implementing security defenses is crucial for protecting computer systems and networks from various threats and vulnerabilities. This involves establishing security policies, conducting vulnerability assessments, implementing intrusion detection measures, ensuring virus protection, and utilizing auditing, accounting, and logging mechanisms.

***Security Policy:*** A well-defined security policy serves as a guideline for all stakeholders and is regularly updated to address evolving security needs. It covers various aspects such as password requirements, port scanning frequency, virus detection protocols, etc.

***Vulnerability Assessment:*** Periodic assessments are conducted to detect vulnerabilities in the system. Assessments include port scanning, checking for weak passwords, examining permission settings, monitoring system files for changes, etc. Systems connected to the Internet are inherently less secure and require extra precautions.

***Intrusion Detection:*** Intrusion detection systems (IDS) aim to detect and respond to attacks, whether successful or unsuccessful. Techniques include signature-based detection and anomaly detection. IDS can alert administrators, automatically block suspicious traffic, or divert attackers to honeypots for monitoring and analysis.

***Virus Protection:*** Anti-virus programs employ signature-based detection to identify known viruses and may also detect anomalies in program behavior. Best practices include avoiding suspicious software sources and periodically verifying the integrity of known safe programs.

***Auditing, Accounting, and Logging:*** Logging systems record various system activities like authentication attempts, file changes, network accesses, etc. Detailed logs can help detect anomalous behavior and provide insights into system performance. Logging also poses performance overheads, and careful configuration is required to balance security needs with system performance.

***Tripwire Filesystem (New Sidebar):*** The Tripwire filesystem monitors files and directories for changes, assuming most intrusions involve some form of file modification. It records file properties in a database and uses hash codes to monitor changes in file contents. Protecting the Tripwire system itself, especially the database, is crucial for maintaining its integrity.

Implementing a comprehensive security defense strategy involves a combination of

proactive measures like vulnerability assessments and intrusion detection, reactive measures like virus protection, and continuous monitoring and analysis through auditing, accounting, and logging mechanisms.

## X. The Firewalling to Protect Systems and Networks

Firewalls are essential components of network security infrastructure that act as barriers between different security domains, monitoring and controlling traffic flow based on predefined criteria. They can be hardware devices or software applications deployed at the boundary between internal networks and external entities, such as the internet.

***Functionality:*** Firewalls monitor and log activity between different security domains, restricting traffic based on specified rules and criteria. They can allow or block traffic types like HTTP, Telnet, SSH, etc., based on organizational policies.

***De-Militarized Zone (DMZ):*** A common firewall architecture involves setting up a DMZ between the internal network and the outside world. The DMZ allows outside computers to reach designated services like web servers but prevents access to the internal network. Even if the DMZ is breached, the attacker cannot access the internal network.

***Firewall Vulnerabilities:*** Firewalls themselves are susceptible to attacks, including tunneling (encapsulating forbidden traffic), denial of service attacks, and spoofing. Ensuring firewall resilience against such attacks is crucial for maintaining network security.

In terms of specialized forms of firewalls there are various types associated. The distinctive ones that play main roles are usually of four types.

***Personal Firewalls:*** Software layers that protect individual computers, either as part of the operating system or as separate software packages.

***Application Proxy Firewalls:*** Understand specific protocols and act as intermediaries for services like SMTP, examining and filtering incoming requests.

***XML Firewalls:*** Specialized in examining and rejecting ill-formed XML packets, providing security for XML-based communication.

***System Call Firewalls:*** Guard the boundary between user mode and system mode, rejecting system calls that violate security policies.

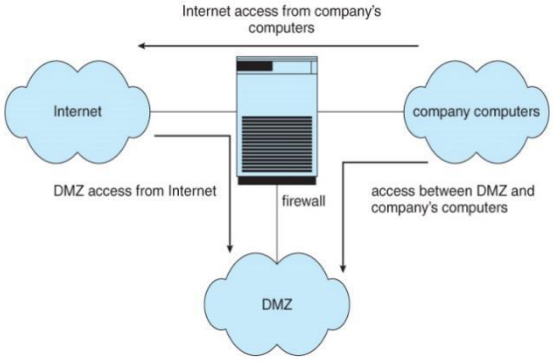



Figure 5. An illustration of Domain separation via firewall

Firewalls play a vital role in protecting systems and networks from unauthorized access and malicious activities. They are deployed strategically to enforce security policies and safeguard sensitive data, but they also require careful management and regular updates to address emerging threats and vulnerabilities in the cybersecurity landscape. To provide an idea figure 5 provides an illustration to better understand the matter. An overall visualization of the findings is provided in figure 6 for better understanding.

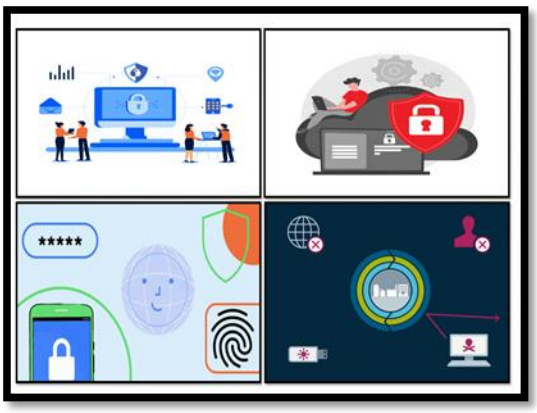


Figure 6. An overall visualization of the findings

## XI. The Computer-Security Classifications

The U.S. Department of Defense's "Trusted Computer System Evaluation Criteria" outlines a

classification system for computer security, ranging from the least trustworthy (Level D) to the highest level of security (Class A). These classifications are based on the system's ability to enforce security measures, control access, and protect sensitive information.

***Level D:*** Systems at this level lack user identification and authorization. Examples include DOS and early versions of Windows. Users have full access and control over the system without any restrictions.

***Level C1:*** Introduces user identification and authorization. Provides some means of controlling user access to files. Suitable for use by a group of cooperating users. Common UNIX systems fall into this category.

***Level C2:*** Adds individual-level control and monitoring. Allows file access control on a per-individual basis. Supports monitoring and logging of specific user activities. Special secure versions of UNIX, like SCO, have been certified for C2 security levels.

***Level B:*** Introduces sensitivity labels on system objects (e.g., "secret", "top secret"). Users have different clearance levels, controlling their access to objects. Human-readable documents are labeled with sensitivity levels.

***Level B2:*** Extends sensitivity labels to all system resources, including devices. Supports covert channels and auditing of events that could exploit covert channels.

***Level B3:*** Allows the creation of access-control lists denying access to specific objects.

***Class A:*** The highest level of security. Architecturally similar to B3 but developed using formal methods to prove system integrity. Developed by trusted personnel in secure facilities.

These classifications dictate the security features a system must implement, but the specific implementation is determined by security policies. Systems and policies can be reviewed and certified by trusted organizations, such as the National Computer Security Center, and may also adhere to other standards governing physical protections and other security measures.

## XII. DISCUSSIONS

Operating system (OS) security stands as a cornerstone in contemporary computing environments, ensuring the integrity, confidentiality, and availability of data and resources. This manuscript delved into the multifaceted domain of OS security, aiming to provide a comprehensive exploration of its theoretical underpinnings, practical implications, and emerging trends. As technology progresses and cyber threats become more sophisticated, understanding the principles and challenges of OS security is paramount for ensuring the robustness and resilience of computer systems.

At the heart of OS security lie foundational principles such as the confidentiality, integrity, and availability (CIA) triad, access control mechanisms, authentication protocols, encryption techniques, and secure coding practices. By delving into these theoretical foundations, we gained insights into the fundamental principles that underpin secure operating environments. Furthermore, tracing the historical evolution of OS security from early mainframe systems to contemporary multi-user, networked environments provided a very valuable context for understanding its development and current state.

The landscape of OS security is fraught with challenges stemming from vulnerabilities in system architecture, software flaws, insider threats, social engineering attacks, and the proliferation of malware. This manuscript endeavors to dissect the diverse nature of security threats faced by modern operating systems through real-world case studies and empirical data analysis. By explaining these challenges, we aim to equip readers with a nuanced understanding of the evolving threat landscape and its implications for OS security management.

To mitigate the risks posed by security threats, organizations should employ an array of security strategies and best practices. These encompass access control mechanisms, encryption technologies, intrusion detection systems (IDS), security patches and updates, network firewalls, and user authentication protocols. By evaluating the effectiveness of these strategies in mitigating

common threats, we hoped to provide insights into their practical implications for OS security management and implementation.

The manuscript also hopes that emerging trends and future directions in OS security, including the adoption of cloud computing, virtualization, containerization, the Internet of Things (IoT), and artificial intelligence (AI) in security applications is paramount. Additionally, delving into emerging threats such as ransomware, supply chain attacks, and zero-day vulnerabilities, discussing proactive measures to address these challenges. By examining these emerging trends, the aim was to anticipate future developments in OS security and provide recommendations for proactive security measures.

Throughout the manuscript, the presentations of a wide series of case studies and experimental analyses to illustrate the practical implications of security strategies in real-world scenarios. These case studies highlight successful security implementations, security breaches, incident response strategies, and lessons learned from security incidents. Experimental analyses evaluate the effectiveness of security measures through controlled experiments, vulnerability assessments, and penetration testing, providing empirical insights into their efficacy.

Drawing from the findings and insights garnered through the research, it also offers policy recommendations and best practices for enhancing OS security. These recommendations encompass regulatory compliance, security awareness training, incident response planning, data protection strategies, and collaboration among stakeholders to address common security challenges. By providing actionable recommendations, the aim was to guide policymakers and practitioners in enhancing the security posture of computer systems and networks.

This research manuscript presents a comprehensive examination of operating system security, encompassing theoretical foundations, practical considerations, emerging trends, and policy implications. By integrating diverse research methodologies and empirical insights, the manuscript contributes to advancing knowledge in OS security and provides actionable recommendations for enhancing the security posture of computer systems and networks in the face of evolving cyber threats.

## XIII. Conclusions

This research manuscript has provided a thorough exploration of operating system security, encompassing theoretical foundations, practical considerations, emerging trends, and policy implications. Through a comprehensive analysis of the theoretical underpinnings of OS security, including the CIA triad, access control mechanisms, authentication protocols, and encryption techniques, the investigations illuminated the fundamental principles that underpin secure operating environments. Moreover, by delving into the challenges and threats faced by modern operating systems, including vulnerabilities in system architecture, software flaws, insider threats, social engineering attacks, and the proliferation of malware, this manuscript has shed light on the complex threat landscape confronting organizations and individuals in today's interconnected world. Through real-world case studies and empirical data analysis, it has highlighted the multifaceted nature of security threats and their implications for OS security management. Furthermore, this manuscript has explored a range of security strategies and best practices employed by organizations to mitigate the risks posed by security threats, including access control mechanisms, encryption technologies, intrusion detection systems, security patches and updates, network firewalls, and user authentication protocols. By evaluating the effectiveness of these strategies in mitigating common threats, it has also provided insights into their practical implications for OS security management and implementation.

Additionally, the exploration examined emerging trends and future directions in OS security, such as the adoption of cloud computing, virtualization, containerization, the Internet of Things, and artificial intelligence in security applications. By anticipating future developments in OS security and discussing proactive measures to address emerging threats, this manuscript aims to guide policymakers and practitioners in

enhancing the security posture of computer systems and networks.

Through a series of case studies and experimental analyses, the research illustrated the practical implications of security strategies in real-world scenarios and evaluated their efficacy through controlled experiments, vulnerability assessments, and penetration testing. By providing actionable recommendations for enhancing OS security, including regulatory compliance, security awareness training, incident response planning, and data protection strategies, this manuscript seeks to empower stakeholders to bolster the security posture of computer systems and networks.

This research manuscript contributes to advancing knowledge in OS security by integrating diverse research methodologies and empirical insights. By synthesizing theoretical foundations with practical considerations and policy implications, this manuscript provides a comprehensive understanding of OS security and offers actionable recommendations for enhancing the security posture of computer systems and networks in the face of evolving cyber threats.

## ACKNOWLEDGMENT

The idea representation with the research focusses along with the context concerning the investigative exploration and manuscript writing was done by the author himself. All the datasets, data models, data materials, data information, computing toolsets used and retrieved for the conduction concerning this research are mentioned within the manuscript and acknowledged with its associated references where appropriate.

## REFERENCES

[1] "About The Calyx Institute - Calyx Institute". calyxinstitute.org. Retrieved 2 November 2021.
[2] "Kali NetHunter Documentation". Kali Linux Documentation. Retrieved 5 April 2020.
[3] "Kali Linux 1.0 review". LinuxBSDos.com. 14 March 2013. Retrieved 26 November 2019.
[4] Simionato, Lorenzo (24 April 2007). "Review: BackTrack 2 security live CD". Linux.com. Retrieved 10 April 2019.
[5] Barr, Joe (13 June 2008). "Test your environment's security with BackTrack". Linux.com. Retrieved 10 April 2019.
[6] "BackTrack 4 - Hacking galore". Dedoimedo.com. 15 May 2009. Retrieved 10 April 2019.
[7] "BackTrack 5 R3 review". LinuxBSDos.com. 17 August 2012. Retrieved 10 April 2019.
[8] "Parrot Security Could Be Your Next Security Tool". Linux.com | the source for Linux information. 2 December 2016. Retrieved 9 March 2018.
[9] Vervloesem, Koen (27 April 2011). "The Amnesic Incognito Live System: A live CD for anonymity [LWN.net]". lwn.net. Archived from the original on 21 August 2017. Retrieved 14 June 2017.
[10] "Devs cook up 'leakproof' all-Tor untrackable platform". The Register. 13 November 2012. Retrieved 10 July 2014.
[11] Greenburg, Andy (17 June 2014). "How to Anonymize Everything You Do Online". Wired. Retrieved 10 July 2014.
[12] "Whonix adds a layer of anonymity to your business tasks". TechRepublic. 4 January 2013. Retrieved 10 July 2014.
[13] Pentoo (Gentoo) Based Linux Review, Features and Screenshot Tour, TecMint.
[14] KITE Introduces a New Secured FOSS Based Operating System.
[15] A Look at Pentoo Linux and Its Security Analysis Tools, eWeek.
[16] 12 Best Operating Systems For Ethical Hacking And Penetration Testing | 2018 Edition
[17] "about | Alpine Linux". alpinelinux.org.
[18] says, GigaTux (24 August 2010). "Alpine Linux 2 review | LinuxBSDos.com".
[19] "Fedora Silverblue User Guide: Fedora Docs". docs.fedoraproject.org. Archived from the original on 11 October 2021. Retrieved 11 October 2021.
[20] OpenBSD Project (19 May 2020). "OpenBSD". OpenBSD.org. Retrieved 12 October 2020.
[21] "Qubes OS bakes in virty system-level security". The Register. 5 September 2012.
[22] Stallings (2005). Operating Systems, Internals and Design Principles. Pearson: Prentice Hall. p.6.
[23] "Desktop Operating System Market Share Worldwide". StatCounter Global Stats. Archived from the original on 2 October 2023. Retrieved 3 October 2023.
[24] "Mobile & Tablet Operating System Market Share Worldwide". StatCounter Global Stats. Retrieved 2 October 2023.
[25] "Twenty Years of Linux according to Linus Torvalds". ZDNet. April 13, 2011. Archived from the original on September 19, 2016. Retrieved September 19, 2016.
[26] "What Is Linux: An Overview of the Linux Operating System". Medium. 11 April 2020. Retrieved 16 July 2023.