Personalized Self-Guided Tour Strategy by Integrating Random Forest Preference Model and Attractions Association Model

Huili Duan
Modern Service College, Zhengzhou Tourism College, Zhengzhou, 451464, China
E-mail: dhuili2022@163.com

Keywords: random forest algorithm, frequent pattern growth algorithm, personalization, self-guided tour, preference selection

Received: April 10, 2024

With the rapid development of tourism, personalized self-guided tours have become an important trend in the tourism market. Since traditional personalized recommendation methods often ignore elements such as users' personal preferences and characteristics of travel destinations, in order to improve the reliability of personalized self-guided tour strategy research. The study fuses the random forest algorithm preference model with the frequent pattern growth algorithm association model to provide personalized self-guided tour strategies. The study firstly utilizes the random forest algorithm to predict the attraction preference selection, and then utilizes the frequent pattern growth algorithm to mine the association relationship among the attractions, so as to provide effective data for strategy formulation. The results indicated that the prediction accuracy, recall, and data processing precision of the strategy model for attraction data were 92.07%, 93.07%, and 81.06%, respectively. The coverage of the strategy model was, however, much higher than that of the comparison model for regional attractions, featured attractions, popular attractions, and specialized attractions, at 78.09%, 85.61%, and 63.26%, respectively. This suggests that the model developed in the study can help meet the goal of the trip and greatly increase the accuracy of attractions in the process of developing a personalized self-guided tour strategy. The study intends to increase user happiness and travel suggestion accuracy in order to strongly promote the growth of the tourism industry.

Povzetek: Raziskava združuje model naklonjenosti z algoritmom naključnega gozda (RFA) in model povezav atrakcij s FP-Growth za personalizirane strategije samostojnih potovanj. Rezultati izboljšajo kvaliteto priporočil in povečajo zadovoljstvo turistov.

1 Introduction

Personalized self-guided tours (PSGT) strategy refers to customizing the most suitable travel solutions for users based on their personal preferences and preferences to enhance user satisfaction and experience. Traditional personalized recommendation algorithms mainly make recommendations based on the user's historical behavioral data or interest labels, but they often ignore the user's personal preferences and the characteristics of travel destinations [1-2]. Random forest algorithm (RFA) is an integrated learning method that can effectively solve the overfitting problem and has high prediction accuracy by combining multiple decision tree (DT) models for prediction. However, the application of RFA preference models in PSGT strategies often focuses only on the user's behavioral data and characteristics, ignoring the user's personal preferences and destination characteristics [3-4]. In addition, frequent pattern growth algorithm (FP-Growth) is an association rule mining algorithm (ARMA) that can be used to discover frequent item sets and association rules in a dataset. In the field of tourism, FP-Growth can be used to mine association rules between tourists' consumption behaviors and preferences to discover common characteristics and behavioral patterns of different types of tourists [5-6]. By mining and analyzing these association rules, useful insights and guidance can be provided for the formulation of PSGT strategies. Therefore, the study fuses the RFA preference model with the FP-Growth association model to enhance the effectiveness of PSGT strategies. RFA is first utilized to construct a prediction model of users' preference for attraction selection. Then FP-Growth is utilized to mine association rules of tourist destinations to provide more accurate recommendations. The study aims to use the fusion of RFA preference model and FP-Growth association model to enhance the recommendation effect of PSGT strategy, to provide tourists with a more personalized and high-quality tourism experience, and to promote the development and progress of the tourism industry.

The study's first section provides an overview of the state of the art in terms of machine learning (ML) research in the tourist industry. The second section of the study uses FP-Growth to mine the correlation relationship between attractions based on attraction PS after first using RFA to forecast the attraction preference selection (PS). Through simulation studies, the application

performance of the developed PSGT-strategic model (SM) is confirmed in the third section. The experimental results are compiled in the fourth section, which also examines the benefits and drawbacks of the study's methodology.

2 Related works

PSGT strategy research is particularly important in the context of the current increasingly diverse tourism market. To achieve this objective, a large number of specialists and academics have focused on the fields of ML and data mining. Among them, RFA and FP-Growth, as two representative technical tools, have shown great potential and value in the research of PSGT strategies. Deng et al. proposed an edge computing-based travel destination selection preference prediction method to improve the accuracy and prediction duration of travel destination selection preference prediction. The study utilized edge computing to select preference features, and then utilized the RFA to sort and get the weight value, and then performs preference prediction based on the weight value. The results showed that the method can significantly improve the prediction accuracy and also shorten the time-consuming process of selecting tourist destinations [7]. Vieira et al. analyzed the techniques of acceptance extension model, perceived usefulness, and perceived ease of use in social networks in order to assess the behavior of travelers in choosing tourist attractions. The study was conducted by analyzing social data from different social software and considering the above three factors. The study of tourists' behavioral intentions was greatly improved by this strategy, according to the results [8]. Huda et al. conducted a study on sustainable digital strategies for hotels in order to enhance the market expansion of budget hotels. The study was conducted to enhance the operational practices of hotels by restoring financial sustainability through surveys of different customers and enhancing digital market strategies in resilient sectors of the tourism industry. The results showed that the study was able to consistently gain a small advantage in the sustainability of frequent guest maintenance [9]. Yen et al. proposed a new approach based on network data envelopment analysis and social network analysis to realize the potential research on the accessibility of different tourist destinations. The study realized the sharing of network data by classifying and statistically analyzing the data of tourism intermediaries. The results showed that the method can effectively help destination managers and governments to identify tourism opportunities and implement appropriate strategies [10].

Liang used RFA to build a regression model based on data mining and data learning to investigate the growth of rural tourism. RFA was used to analyze and summarize the tourism service strategies of rural revitalization regions. The results indicated that RFA was able to weigh the importance of the variables in the service strategy, and thus could be applied to the development of the service strategy [11]. Zhang et al. applied ML methods such as RFA to the strategy research of customized tourism through the study of big data customized tourism. This suggested that users could receive customized travel search services through data mining and classification utilizing several big data elements related to tourism [12]. To provide tourists with effective visit information, the frequent item set method was employed to mine the points of interest in different attractions, as proposed by Höpken et al. The study analyzed tens of thousands of tourist image information to cluster the attractions that are of interest to potential tourists, and then used association rules for data mining. The results showed that the method is able to increase the support level to 4.6%. [13]. Shabtay et al. constructed a new bootstrap frequent pattern for the mining of different program interests through a multi-objective mining algorithm. The study utilized the bootstrap frequent pattern to be able to find the interest relationships of different item sets quickly while using a small amount of memory. The results indicated that the bootstrap frequent pattern can significantly improve the problem of counting class rules among different items [14]. In order to conduct an effective analysis of the relevant research, the study analyzed and summarized the research conducted by the experts and scholars mentioned above, as shown in Table 1, which is the result analysis table obtained by citing the methods used in the literature.

Table 1: Analysis of results obtained from the method of using citations

Literature	Main methods	Main findings	Performance index
Deng et al	Edge algorithm	Significantly improve prediction accuracy and shorten time consumption	Improved prediction accuracy and reduced time consumption
Vieira et al	The acceptance expansion model in social networks	Analysis of significantly enhancing tourist behavioral intentions	Improving the accuracy of behavioral intention analysis
Huda et al	Research on sustainable digital strategy for hotels	Continuously gaining small advantages in sustainable development through frequent customer maintenance	Financial sustainability recovery and improved customer maintenance effectiveness

Yen et al	Based on network data envelopment analysis	Effectively help identify tourism opportunities and execute strategies	Enhancing the ability to discover tourism opportunities
Liang	Data mining and random forest algorithm	The random forest algorithm can balance the importance of variables in service strategies	Improving the accuracy of service strategy formulation
Zhang et al	Research on big data customized tourism	79.84% of customers are willing to purchase customized big data services again	Users have a high willingness to purchase again, and personalized search services have good results
Höpken et al	Identifying frequent item sets and mining interest points	Support increased to 4.6%	Improving the accuracy of interest point mining
Shabtay et al	Constructing guided frequent patterns using multi-objective mining algorithms	Significantly improve the numerical rule problems between different projects	Improved speed in mining interest relationships between projects and reduced memory usage

Through the analysis of the above research, it is found that the random forest preference model and the scenic spot association model play a very important role in the development of PSGTs. Using the RFA can effectively improve the accuracy in the prediction of attraction PS, and the FP-Growth algorithm can significantly improve the mining of the association relationship between different attractions. Additionally, the study will provide a clear comparison of the performance metrics of each reference [6], [8], and [12] with the model approach in terms of performance. The approach of Deng et al. [6] is particularly effective in real-time and prediction accuracy. The approach of Vieira et al. [8] is highly specialized in understanding travelers' social behaviors and psychological preferences. Finally, the approach of Zhang et al. [12] is particularly effective in providing personalized services. In contrast, the RFPAP model is particularly adept at the scientific assessment and analysis of tourism resources, while the FP-Growth model is highly effective in mining association rules from large amounts of data. The integration of these methods may result in more comprehensive and accurate solutions in the field of personalized travel recommendation. In summary, the use of RFA can effectively improve the accuracy in the prediction of attraction PS, and the FP-Growth algorithm can significantly improve the mining of correlations between different attractions. Based on this study, the two algorithms are integrated and used to construct a PSGT strategy model. The study aims to provide more and more favorable data support for the development of tourism.

3 Self-guided tour strategy by integrating RFPAP model and FP-growth model

To meet the personalized needs of tourists, the study designs a self-guided tour strategy based on data mining by fusing the RFPAP model with the FP-Growth model. This strategy will utilize advanced data analysis methods to deeply mine tourists' tourist attraction preferences (PTA) and the correlations between attractions, so as to provide tourists with more accurate and personalized travel route suggestions. The study utilizes RFA to predict PTA and FP-Growth algorithm to mine the correlations among attractions on the basis of prediction. The design of PSGT strategy is accomplished through the fusion of the two models.

3.1 RFA-based attraction preference prediction model construction

To analyze the attraction selection of PSGT, the study utilizes RFA for predictive analysis of PTA. In the prediction of tourism PTA using RFA, the data of attraction selection by itself is of little value, but it is effectively categorized and processed by RFA. This is used as a basis for obtaining information about attraction PS from a large amount of tourism data, and it also improves the prediction accuracy when multivariate prediction is performed. Through extensive analysis of RFA, the algorithm is composed of a large number of DTs [15-16]. This DT is the predictive classifier in the algorithm, and by effectively analyzing the DT, the sensitivity and accuracy of data classification in self-guided tour attractions can be improved. The flow chart of DT construction based on attraction selection is shown in Fig. 1.

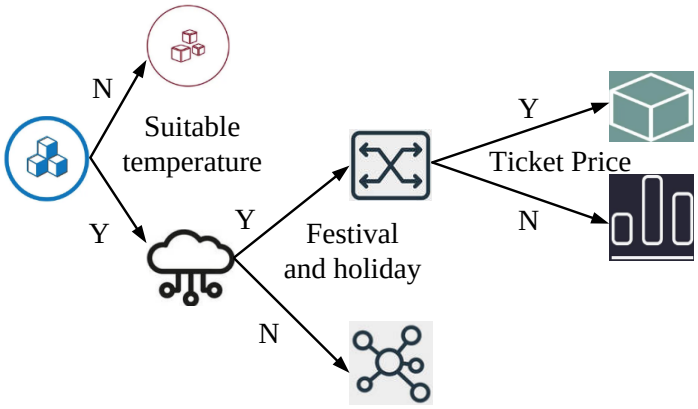


Figure 1: Process diagram for constructing a decision tree based on scenic spot selection

Fig. 1 illustrates the decision-making process employed by the research team. It can be observed that the suitability of weather conditions is initially assessed, and subsequent decisions are then made based on ticket prices. If the ticket price is deemed appropriate, further consideration will be given to whether the scenic spot has any festivals or activities. Conversely, if the price is deemed unsuitable, or for other reasons, users will not choose the attraction. The decision-making process of scenic spot selection is clearly displayed by the clear definition of nodes and branches. This enables users to make more suitable choices. In the process of predicting tourism PTA using RFA, the specific operation to improve the accuracy of classification is divided into 3 steps. The first step is to use the sampling technique to extract the original data, the extraction process is a put-back operation, and the corresponding training set can be obtained through random extraction. These datasets are classified, and each extraction will have about 36.8% of the data not sampled, and these unsampled data forms the out-of-bag (OOB) [17-18]. The OOB error can be calculated to evaluate the performance of the model. Therefore, OOB can be used for performance analysis of RFA. Then on the basis of obtaining the OOB, the study sets the features in each sample to M. At this point, a variable is extracted for each node in each tree. This variable will be analyzed to find the strongest variable of classification ability, and combined with the threshold value is used for the determination and analysis of classification nodes. Finally, the study utilizes the generated random forest to classify the tourist attraction data and the classification result is based on the number of votes of the classifier. A schematic diagram of attraction prediction by RFA is shown in Fig. 2.

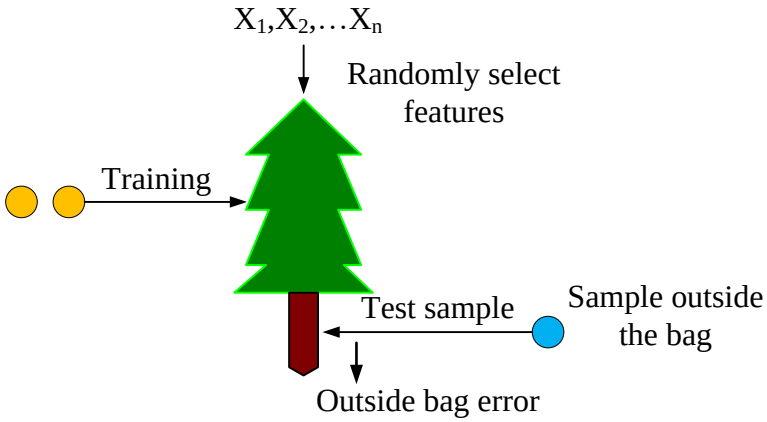


Figure 2: Schematic diagram of scenic spot prediction using random forest algorithm

Fig. 2 illustrates the sample data points utilized for training and testing. It can be observed that two-thirds of the data points are employed for training, while the remaining one-third are utilized for testing. The green tree icon represents the remaining samples that have not yet been included in the current training or testing process. The random selection of features serves to mitigate the risk of overfitting and enhance the model's capacity for generalization. The yellow circle represents "out-of-set samples," which, along with their corresponding "out-of-set errors," are employed to assess the efficacy of the model on previously unseen data, thereby providing an additional metric for evaluating the model's generalizability. The study must obtain a classification

matrix in order to examine the expected values of attraction selection and guarantee the accuracy of the classification results. The classification matrix constructed by the study can be represented by Equation (1).

$$A=\begin{Bmatrix} a_1 f_1 & a_1 f_2 & \cdots & a_1 f_p \\ a_2 f_1 & a_2 f_2 & \cdots & a_2 f_p \\ \vdots & \vdots & \ddots & \vdots \\ a_n f_1 & a_n f_2 & \cdots & a_n f_p \end{Bmatrix} \quad (1)$$

In Equation (1), A denotes the classification matrix and f_p is the feature p value corresponding to the classification process. $a_n f_p$ is the feature p value corresponding to the n th tourist. After calculating the obtained classification matrix, the eigenvector values of tourists in attraction selection can be expressed by Equation (2).

$$X=\left(X_1, X_2, \cdots, X_n\right) \qquad X \in A \quad (2)$$

In Equation (2), X_n denotes the feature vector of the n th visitor at the time of selection. The predicted value of attraction selection at this time can be expressed by Equation (3).

$$Y=f(X)=\left(y_1, y_2, \cdots, y_n\right) \quad (3)$$

In Equation (3), $f(X)$ denotes the classification function of RFA, and y_n denotes the predicted value of PTA for the n th tourist. It is discovered that the sorting of DT would significantly affect the prediction outcomes in the PTA prediction process. The sorting judgment according to the majority principle can be employed to obtain the importance sorting results of the selection variables. Furthermore, the credibility of the sorting results of the majority principle is considerably higher than that of the sorting results of the selection of single factors. In the process of feature sorting, it is necessary to judge the metric importance of feature variables to determine the sorting principle [19-20]. The study uses the OOB data as a test set based on the previous OOB data as a way to judge the generalization ability when DT classification. The corresponding sorting principle at this point can be expressed by Equation (4).

$$I_k(i)=OOB(E_k)_i-OOB(E_k) \quad (4)$$

In Equation (4), $OOB(E_k)_i$ denotes the i th incorrectly assessed eigenvalue in the incorrect assessment result. $OOB(E_k)$ denotes the reordered misassessment value. Equation (5) can be used to calculate the DT importance measure after obtaining the ranking principle.

$$I(i)=\frac{\sum_{i=1}^{N} I_k(i)}{N} \quad (5)$$

In Equation (5), N denotes the number of DTs. Combining the above studies, the study constructed the random forest preferred attraction prediction (RFPAP) model on the basis of RFA. The prediction flowchart of this model is shown in Fig. 3.

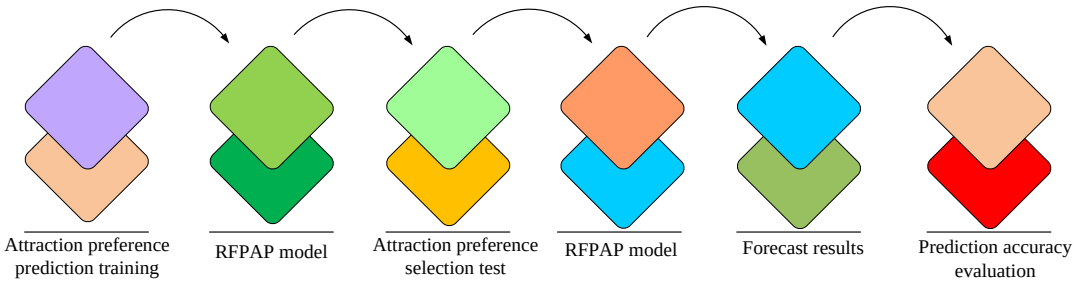


Figure 3: Process diagram of preference prediction model for random forest scenic spot selection

Fig. 3 illustrates the two-stage structure of the scenic spot selection preference prediction model developed in the study. The model comprises a training stage and a testing stage. Firstly, during the training phase, the study employs the RFA to classify and process the raw data, with the objective of selecting representative training datasets and identifying the optimal parameter configuration through algorithm iteration. Subsequently, during the testing phase, the selected and optimized dataset will be applied to the RFPAP model for the actual prediction of attraction selection preferences, and the prediction results will be recorded. Finally, in order to evaluate the performance of the model, the predicted results are compared in detail with the actual selection values of tourists. This comparison allows for the calculation of the prediction accuracy of the model, thereby ensuring its accuracy and reliability.

3.2 Design of attraction association model based on FP-Growth algorithm

Through the study of PTA prediction, it is found that PSGT tourists do not choose a single attraction in the process of attraction selection. Instead, they will choose

based on the association between attractions and attractions. To better analyze the self-guided tour strategy, the study utilizes ARMA to mine the association relationship between different attractions, so as to get more accurate PS prediction results [21-22]. The study chooses FP-Growth as the ARMA between attractions. This algorithm is a class of mining algorithms obtained by improving on the basis of Apriori algorithm. The effectiveness of correlation mining between attractions can be greatly increased by reducing the need for repetitively scanning the attraction database while also preventing the production of candidate datasets throughout the mining process. In the process of attraction correlation mining using the FP-Growth algorithm, the attraction database needs to be compressed first, and then the compressed database is divided into a complete set of correlation database [23-24]. Through the compression of the database, it can effectively reduce the amount of computation when attraction data relevance mining. The flowchart of attraction relevance data mining based on FP-Growth algorithm is shown in Fig. 4.

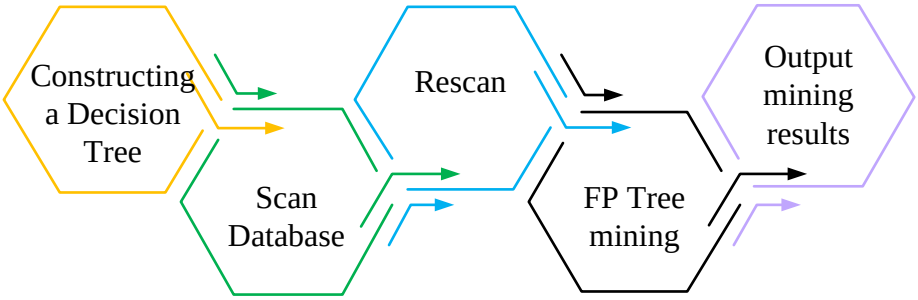


Figure 4: Flowchart of attraction association data mining based on FP-Growth algorithm

In Fig. 4, the FP-Growth algorithm mines the attraction association data in the process of traversing the database twice to search for the data information it needs. This can significantly reduce the database scanning time and improve the efficiency of the relationship mining. The quantity of attraction data information will have a big influence on how accurate associative mining is in the attraction associative relationship mining process. Therefore, the study uses web crawler technology to obtain the attraction data in a tourism website under authorized circumstances. By obtaining the data of tourists in selecting multiple attractions to visit, the FP-Growth algorithm is able to obtain the likelihood of tourists selecting the next tourist attraction through data mining [25-26]. The research of PSGT attraction recommendation strategy is carried out on this basis. When organizing and correlation mining the data of tourist attractions, it is necessary to design the corresponding correlation mining table according to the goal of mining. As shown in Table 2 is the research based on a city's tourist attractions designed inter-attraction association rule mining information table.

Table 2: Information on mining association rules between scenic spots

Attraction and tourist selection association number	Attraction ID	Attraction associated markers
G1	J1	Y
G2	J2	Y
G3	J3	Y
G4	J4	Y
G5	J5	Y
G6	J6	Y
G7	J7	N
G8	J8	Y
G9	J9	N
G10	J10	N

Equation (6) illustrates how the study created an evaluation matrix utilizing the association rule mining data in Table 1 in order to assess the correctness of

association connection mining through the collection of mining information.

$$V=(V_{ik})_{m\times n}=\begin{bmatrix} V_1 & V_{11} & V_{1n} \\ V_2 & V_{22} & V_{2n} \\ \vdots & \vdots & \vdots \\ V_m & V_{m2} & V_{mn} \end{bmatrix} \quad (6)$$

In Equation (6), V indicates each indicator value, $i=1,2,\cdots,m$, $k=1,2,\cdots,n$, m =4, n =17. After the indicators are constructed, they must be normalized in order to be compared and analyzed. Equation (7) can be used to calculate the standardized values.

$$X_{ik}=\frac{V_{ik}-\min\limits_{k}V_{ik}}{\max\limits_{i}-\min\limits_{i}V_{ik}} \quad (7)$$

In Equation (7), V_{ik} is the normalized k value corresponding to the i th indicator. After normalizing the data, the correlation of the data can be calculated and analyzed on the basis of the normalization process, and the formula for the correlation can be expressed in Equation (8).

$$\zeta_{ik}=\frac{\min\limits_{i}\min\limits_{k}\left|X_{0k}-X_{ik}\right|+\rho\max\limits_{i}\max\limits_{k}\left|X_{0k}-X_{ik}\right|}{\rho\max\limits_{i}\max\limits_{k}\left|X_{0k}-X_{ik}\right|+\left|X_{0k}-X_{ik}\right|} \quad (8)$$

In Equation (8), $\left|X_0k-X_ik\right|$ denotes the absolute value corresponding to the sequences X_0 and X_i at point k. $\min\limits_{i}\min\limits_{k}\left|X_0k-X_ik\right|$ denotes the minimum value of the bipolar absolute value of the two sequences. $\max\limits_{i}\max\limits_{k}\left|X_0k-X_ik\right|$ denotes the maximum value of the bipolar absolute value of the two sequences. ρ denotes the coefficient. At this point, it is necessary to confirm the weight value of each evaluation index, which can be expressed by Equation (9).

$$\begin{cases} W1=(\omega_{11},\omega_{12},\omega_{13}) \\ W2=(\omega_{21},\omega_{22},\omega_{23}) \\ W3=(\omega_{31},\omega_{32},\omega_{33}) \end{cases} \quad (9)$$

After obtaining the weight values of the assessment indicators at all levels, the attraction association model can be constructed by utilizing the mining results of the correlation rules between attractions. The purpose of constructing the attraction association model is to improve the accuracy of the correlation judgment between attractions and attractions, and the attraction association process of the model is shown in Fig. 5.

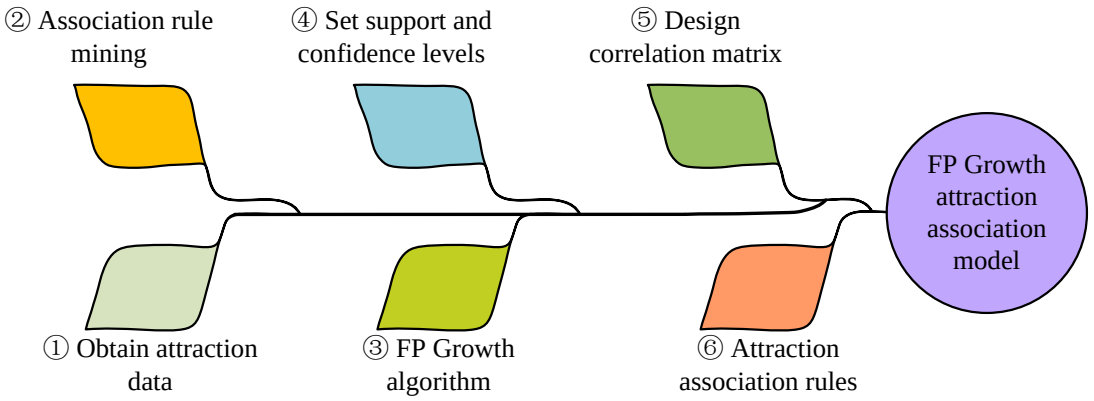


Figure 5: Flow chart of attraction association based on FP-Growth algorithm

Fig. 5 illustrates the process of constructing a scenic spot association model using the FP-Growth. The data processing and setting stage involves obtaining scenic spot data, analyzing the correlation degree betweenscenic spots, using the algorithm to mine the correlation degree, setting confidence and support, designing the correlation coefficient matrix, mining scenic spot rules, and finally outputting the correlation degree model.

3.3 SM design of personalized self-guided tours based on RFPAP and FP-Growth

By utilizing the RFPAP model for the preference study of attraction selection and the FP-Growth model for the mining of attraction association relationships, the study completes the design of a tourism strategy for PSGT. The goal of the plan is to offer travelers a more precise and customized travel experience to cater to their individual needs and tastes. First, the RFPAP model is used to study tourists' preferences in the attraction selection process. The model mines the characteristics and influencing factors of tourists' preferences for attractions by analyzing tourists' historical behavioral data. When using the model to analyze personalized self-help tours, it is necessary to set the relevant parameters, and the steps of parameter setting are as follows: ① Data preprocessing: cleaning and integrating tourism data, and conducting necessary transformation and standardization. ② Determine the FP-Growth model parameters: Set the minimum support to filter out frequent and complex items. ③ Set RFPAP model parameters: Set relevant parameters based on model definition and business requirements. ④ Fusion strategy parameter setting: Determine the fusion method and set weight or priority parameters. ⑤ Model training and validation: Train and validate models using historical data, evaluate parameter settings. ⑥ Parameter optimization: Utilize automation tools to optimize parameters and improve model performance. ⑦ Record: Record parameter settings and optimization processes, and write documentation to explain. ⑧ Continuous adjustment: Monitor model performance after going live and adjust parameters based on actual conditions [27-28]. By understanding tourists' preferences, attraction recommendations and travel route suggestions can be provided to tourists that better meet their needs. Second, the FP-Growth model is used to mine the associative relationships among attractions. The model analyzes a large amount of attraction data to discover potential connections and common features between attractions. By understanding the associative relationships among attractions, it can provide tourists with more systematic and comprehensive travel route suggestions, and promote tourists' in-depth visit and experience of attractions [29-30]. Finally, based on the output of the RFPAP model and FP-Growth model, the study designed a PSGT strategy. The strategy takes into account tourists' preferences and the correlation between attractions to provide tourists with personalized tour itinerary suggestions. By optimizing the tour sequence, attraction recommendations and activity arrangements, it improves tourists' travel experience and satisfaction. The specific strategy content is: according to the tourists' personalized needs and time schedule, develop a self-guided tour strategy that meets the needs of tourists. The strategies include tour order, attraction recommendations, and activity arrangements to improve tourists' travel experience and satisfaction. In the practical application, feedback and evaluation of tourists' self-guided tour strategies are collected. The preference model and association model are updated and adjusted according to the feedback data to adapt to the changing market demand and tourists' preferences. Regularly evaluate the performance of the models and make necessary optimizations and improvements to maintain the accuracy and effectiveness of the models.

In the PSGT strategy formulation process, in order to evaluate the strategies effectively, the study used strategy support, strategy confidence and strategy enhancement as measures for the performance evaluation of the formulated strategies. Strategy support is the percentage of the formulated strategy that can satisfy the sample, and the support can be calculated using Equation (10).

$$\sup port(C \to D) = \frac{q(C \cup D)}{N} \tag{10}$$

In Equation (10), C and D denote data that are not intersected in the strategy. $q(C \cup D)$ denotes the data that contain the same attractions in the strategy at the same time. Strategy confidence refers to the proportion of different attractions appearing in the strategy. The higher the confidence rating, the more likely it is that the same attractions would recur, and hence the more reliable the approach that is developed. The strategy confidence can be expressed by Equation (11).

$$confidence(C \to D) = \frac{q(C \cup D)}{q(C)} \tag{11}$$

In Equation (11), $q(C)$ denotes the minimum confidence level. The strategy enhancement degree can be expressed in Equation (12).

$$life(C \to D) = \frac{q(C \cup D)}{q(D)} \tag{12}$$

In Equation (12), $q(D)$ denotes the probability value of the degree of enhancement. By analyzing and summarizing the design process of PSGT strategy, the study successfully completed the design of PSGT strategy. The flowchart of SM design of PSGT based on RFPAP and FP-Growth is shown in Fig. 6.

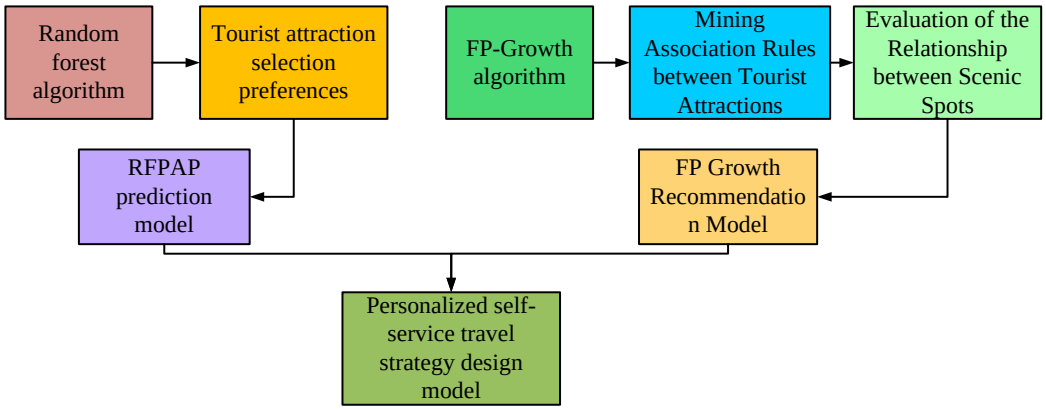


Figure 6: Design flowchart of personalized self-service travel strategy model based on RFPAP and FP-Growth

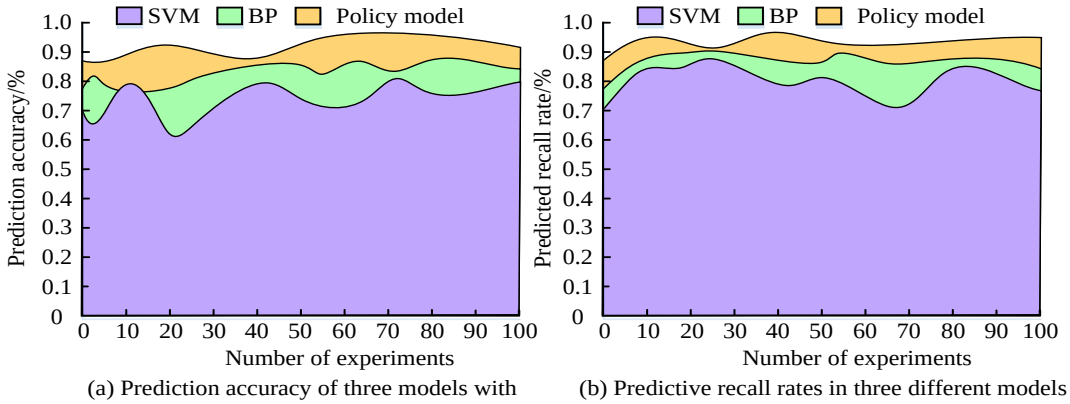
Combined with the analysis of Fig. 6, it can be noticed that the research utilizes the crawler technology to obtain the data information of tourist attractions, and uses the RFPAP model to complete the PS of attraction selection. The attraction association model is then created by mining the association relationship between attractions using the FP-Growth algorithm. Finally, the two models are fused and used to construct the design of PSGT strategy and complete the self-guided tour strategy output.

4 Performance analysis of self-guided tour strategic model integrating RFPAP model and FP-Growth model

To validate the performance of the PSGT strategy design that incorporates the RFPAP model and the FP-Growth model, the study compares the support vector machine (SVM), back propagation neutral network (BPNN) with the RFPAP+FP-Growth model. The comparison of the three models is used to validate the specific performance of the SM for PSGT constructed by the study. To design a reliable, PSGT strategy, the study obtains data from tourism websites and official government websites. The content of the tourism website reflects the preferences of tourists, while the official government website provided accurate basic information. To ensure the accuracy of the data, measures are taken to remove duplicates and erroneous data, as well as to deal with missing values. Finally, a dataset comprising 100 data points on attractions in the same province is constructed for the purpose of evaluating the performance of the PSGT strategy model. These measures ensure the accuracy, consistency, and professionalism of the data, thereby providing a solid foundation for the design of PSGT strategies.

4.1 Performance analysis of RFPAP model for attraction preference prediction

The study compares SVM, BP, and SM in order to validate the prediction models' performance in the PTA prediction process. As demonstrated by the comparison findings of predictive accuracy and recall in various extracted data ratios for the three models in Fig. 7, these metrics are employed as validation measures for performance testing.



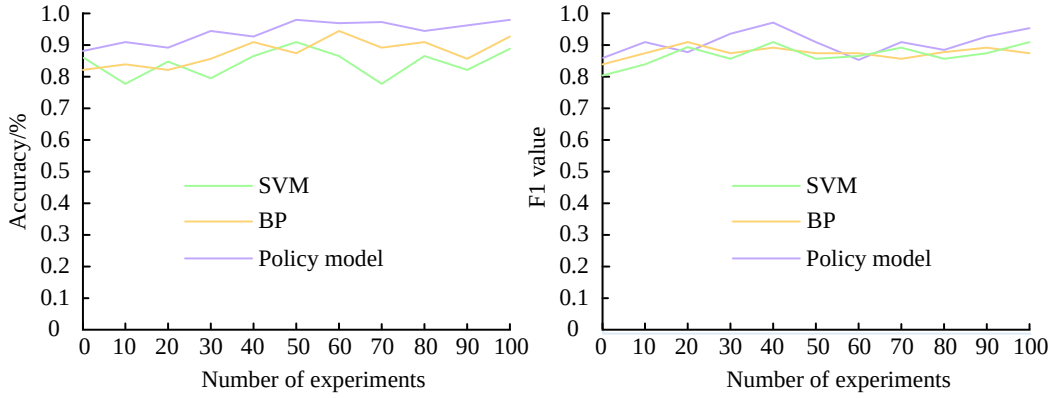
(a) Prediction accuracy of three models with different data extraction ratios

(b) Predictive recall rates in three different models with different ratios of extracted data

Figure 7: Comparison results of prediction accuracy and recall in three different models with different data extraction

ratios

In Fig. 7(a), the SM has the highest prediction accuracy of 92.07% for attraction data among different attraction extracted data. In addition, the prediction accuracy of SVM and BP model for attraction data is 78.95% and 83.76% respectively. In Fig. 7(b), in the comparison of the recall of data extracted from different attractions, the recall of attraction data of SM is 93.07%. The attraction data recall of SVM and BP models are 81.26% and 88.61%, respectively. This suggests that the prediction model is more resilient and reliable in terms of recall rate and accuracy of data prediction. In order to further validate the processing ability of SM in tourist attraction data, the study takes the precision rate and F1 value of PTA prediction as the validation indexes. Fig. 8 displays the comparative findings of the three models' F1 value and precision rate in PTA prediction data processing.



(a) The accuracy of three models in processing tourist attraction preference prediction data

(b) F1 values of three models in processing tourist attraction preference prediction data

Figure 8: Comparison of accuracy and F1 value of three models in processing tourist attraction preference prediction data

There are some variations in the three models' capacities to handle PTA data when processing PTA projection data, as shown in Fig. 8(a). The SVM model has the lowest accuracy, while the SM model has the best accuracy. SVM, BP, and SM have data processing accuracy of 81.06%, 89.73%, and 93.58%, in that order. The F1 values of SVM, BP, and SM are 0.81, 0.86, and 0.92 in Fig. 8(b), in that order. This suggests that the study's SM may provide more useful data and has a greater accuracy in PTA data processing. Additionally, the model's performance is noticeably superior to that of the comparative model. To verify the attraction coverage ability of the SM in the attraction PS process, the study used the attraction coverage as a validation index and selected regional attractions, specialty attractions, popular attractions and niche attractions. In addition, it is used to verify the coverage ability of SM during PS. The comparison results of attraction coverage of the three models in attraction PS are shown in Fig. 9.

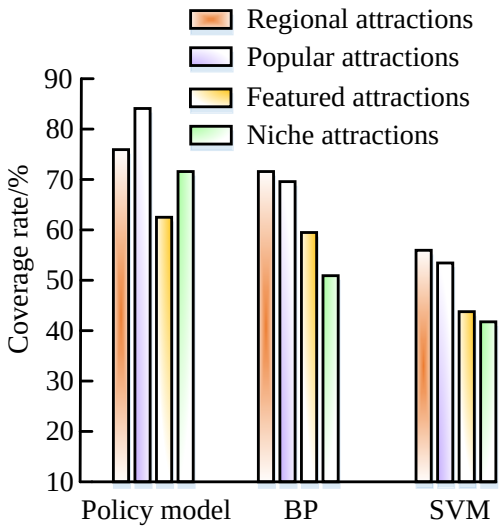
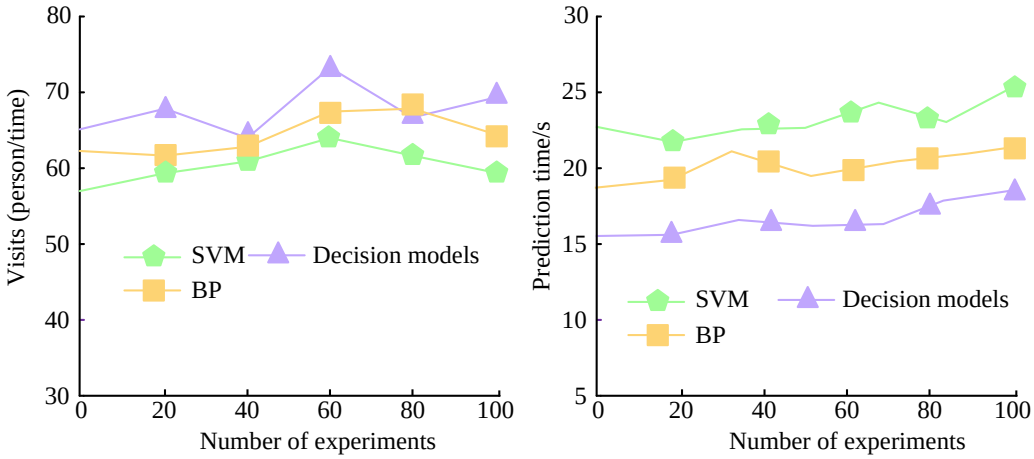


Figure 9: Comparison of attraction coverage rates among three models in attraction preference selection

Attraction coverage plays an important role in forecasting preferences for tourist attractions. It can represent the diversity of attractions and shows how many tourist attractions there are in an area relative to all the attractions. Attraction coverage can influence the attraction recommendation results, help synthesize user needs, and plan travel routes. By considering attraction coverage, user preferences for tourist attractions can be better met. This indicates that the higher the attraction coverage, the greater the tourists' preference for choosing attractions may be when predicting preferences for tourist attractions. In Fig. 9, the attraction coverage of SM in the prediction process of PTA is significantly higher than that of BP model and SVM model. The coverage rates of SM, BP model, and SVM model in regional attractions, specialty attractions, popular attractions, and niche attractions are 78.09%, 85.61%, 63.26%, and 72.05%, 72.55%, 69.83%, 58.66%, and 47.81%, 57.0%, 54.22%, 43.93%, and 40.15%, respectively. This suggests that the study's constructed SM performs more steadily in PTA prediction.

4.2 Performance analysis of attraction association for FP-Growth modeling

To validate the performance of SM in attraction association mining, the study uses the visitor's visit and stay time as validation metrics for exploring the performance of association mining. The comparative findings of the three models in the attraction association process are displayed in Fig. 10 with regard to visit volume and association prediction time.



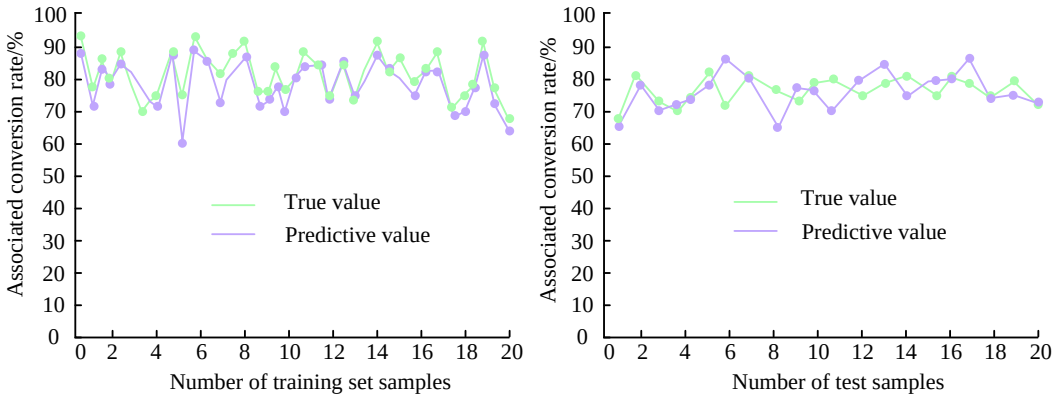
(a) Comparison of Visits Related to Scenic Spots in Three Models

(b) Comparison results of three models for predicting time related to scenic spots

Figure 10: Comparison results of visit volume and association prediction time of three models in the process of scenic spot association

In Fig. 10(a), there is still a gap between the mining ability of the three models in the process of mining the

visits to the attractions' associative relationships. The average visits through the correlation mining SVM model is 62 people/time, the average number of visits of BP network is 68 people/time, and the average visits of SM is 73 people/time. In Fig. 10(b), in the process of attraction correlation mining, the three models' correlation prediction time shows an increasing trend of time consuming with the increase of the experiments. The average of prediction time consuming of SM in the process of attraction correlation prediction is 15.8 s. The average of prediction time consuming of BP model is 18.9 s. The average of prediction time consuming of SVM model is 22.9 s. This shows that the SM designed by the research is capable of finding the correlation between attractions in correlation mining to find the association between attractions, thus increasing the number of attraction visits and reducing the prediction time consumed. To verify the application effect of the SM, the real value of the attraction association conversion rate is compared with the predicted value of the SM. As shown in Fig. 11, the results of the comparison between the real and predicted values of the association conversion rate are shown.



(a) Prediction results of training samples (b) Prediction results of test samples

Figure 11: Comparison results between actual and predicted values of association conversion rate

In Fig. 11(a), in the training set, the average of the true association conversion rate is 88.32% and the average of the SM predicted association conversion rate is 80.05%. In Fig. 11(b), in the test set, the average value of the true association conversion rate is 77.29% and the average value of the SM predicted association conversion rate is 73.61%. The comparison reveals that although the performance evaluation index value of the modeling approach in the test samples is not as good as that in the training samples, it has some performance advantages. In addition, the difference with the real value is not big, 8.27% in the training set and 3.68% in the test set. This suggests that SM is highly applicable. SVM, BP, and SM are used for comparison in order to confirm the accuracy of SM in the attraction association relationship mining process. Fig. 12 displays the findings of a comparison of the three models' associative relationship accuracy.

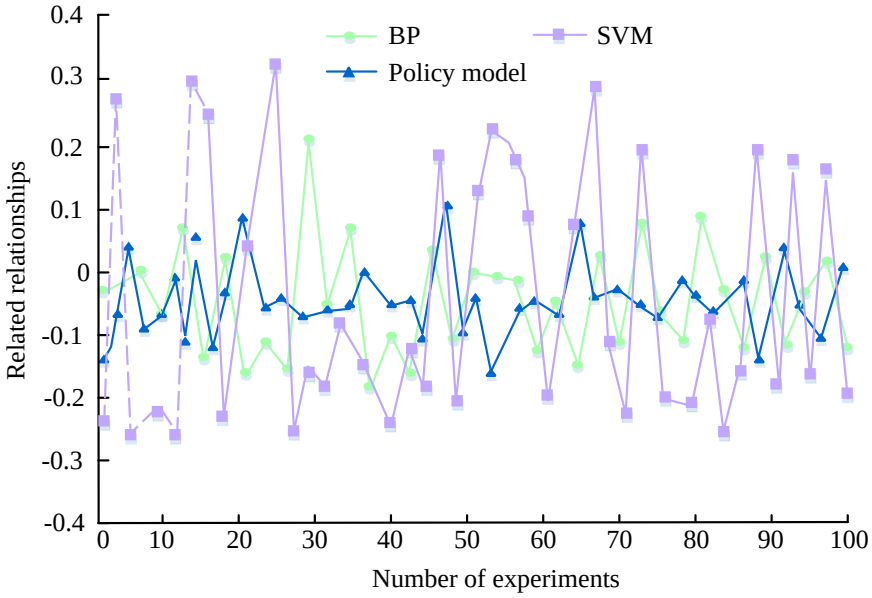


Figure 12: Comparison of accuracy of correlation relationships among three models

The errors of all three models in the correlation prediction process in Fig. 12 exhibit notable changes as the number of iterations rises. The SVM method has a -0.29 negative prediction error and a 0.34 positive prediction error. The BP model's negative prediction error is -0.21 and its positive prediction error is 0.24. The SM has a -0.15 negative prediction error and a 0.08 positive prediction error. This suggests that the SM has the smallest error range of the three models in performing correlation mining has the smallest error range, followed by BP, and the worst is SVM. The positive and negative errors of SM are 0.26 and 0.14 lower than those of SVM. This indicates that SM has a stronger mining ability in the correlation mining process, which can provide richer data results for attraction strategy development.

4.3 Strategic model application performance analysis of personalized self-guided tours

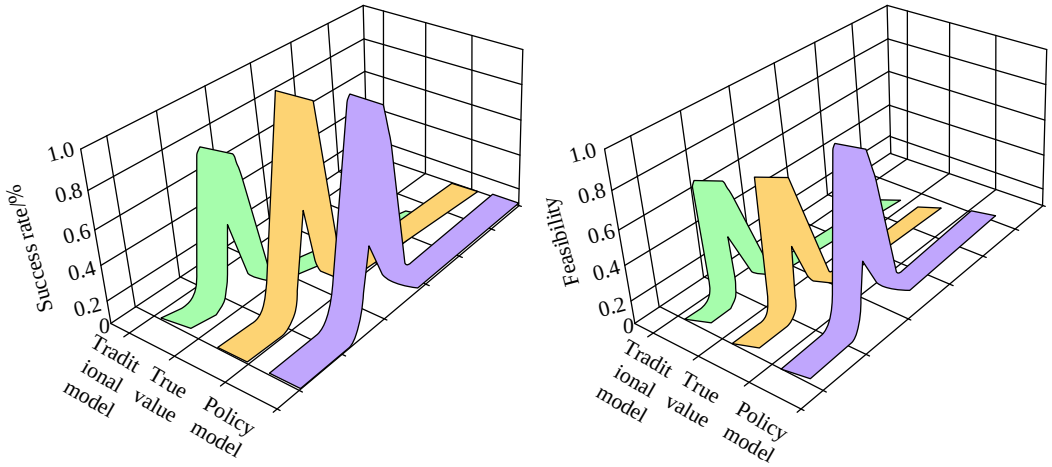
To verify the application performance of the SM of PSGT, the study takes the policy support, policy confidence, and policy enhancement as the verification indexes, and compares the analyzed values of the SM with the real values. Additionally, Table 3 displays the comparison results.

Table 3: Comparison of support, confidence, and improvement results

Attraction and tourist selection association number	Strategy model analysis value			True value		
	Support	Confidence level	Enhancement degree	Support	Confidence level	Enhancement degree
1	0.18	0.65	1.08	0.16	0.71	1.37
2	0.16	0.61	1.13	0.4	0.68	1.26
3	0.21	0.6	1.09	0.13	0.69	1.29
4	0.19	0.61	1.07	0.12	0.66	1.4
5	0.21	0.59	1.08	0.15	0.63	1.35
6	0.16	0.57	1.09	0.12	0.67	1.28
7	0.17	0.60	1.12	0.13	0.65	1.31
8	0.18	0.57	1.05	0.14	0.62	1.28
9	0.21	0.59	1.07	0.13	0.70	1.33
10	0.16	0.58	1.15	0.11	0.65	1.29

In Table 3, the mean support, confidence and enhancement of the analytical value of the attraction-tourist choice association of the SM are 0.183, 0.597 and 1.093, respectively. The mean support,

confidence and enhancement of the real value of the attraction-tourist choice association of the SM are 0.159, 0.666 and 1.316. The difference between the analytical value and the real value of the support, confidence and enhancement is 0.024, 0.069 and 0.223. The differences are 0.024, 0.069 and 0.223. This indicates that the gap between the SM of PSGT designed by the study using the RFPAP model and the FP-Growth model and the tourists' actual appearance of attraction choices is not large, and has a strong feasibility. The study contrasts the traditional model, the SM with the real value, and uses the success rate of attraction strategy suggestion and the feasibility of excursion strategy recommendation as the validation indexes in order to further validate the practical application effect of the SM of PSGT. Fig. 13 shows the results of attraction strategy recommendation success rate and feasibility comparison.



(a) Success rate of attraction recommendation strategy (b) Feasibility of attraction recommendation strategy

Figure 13: Successful recommendation of tourist attraction strategies and feasibility comparison results

In Fig. 13(a), during self-guided tour, the success rate of real recommendation is 79.26%, the predicted value of SM recommendation is 82.17%, and the predicted value of recommendation of traditional model is 73.68%. In Fig. 13(b), in the feasibility of attraction tour, the real feasibility is 63.86%%, the predicted value of SM feasibility is 65.94%, and the predicted value of feasibility of traditional model is 59.87%. The difference between the predicted value and the feasibility prediction value of SM and traditional model is smaller compared with the true value, which indicates that SM has some enhancement performance and can improve the success probability of attraction selection. The comparison results of adaptability and robustness validation for different scenic spots are shown in Table 4.

Table 4: Comparison results of adaptability and robustness validation for different scenic spots

Types of attractions	Sample quantity	RFPAP+FP-Growth model prediction accuracy/%
Cultural attractions	100	93.15
Natural attractions	100	92.63
Urban attractions	100	89.94

Table 4 analysis indicates that the prediction accuracy of the strategy model for cultural, natural, and urban attractions is 93.15%, 92.63%, and 89.94%, respectively, for the same number of attractions. With regard to the adaptability of the RFPAP+FP-Growth model, it is evident that the model demonstrates excellent adaptability, enabling it to adjust prediction strategies in accordance with the distinctive characteristics of different types of attractions. This allows for the provision of personalized self-service travel suggestions for tourists. In terms of robustness, the RFPAP+FP-Growth model is capable of maintaining stable predictive performance in a variety of scenarios. The results of the RFPAP+FP-Growth model in the personalized self-service travel process are presented in Table 5.

Table 5: Feedback results of RFPAP+FP-Growth model in personalized self-service travel process

Feedback metrics	Prediction accuracy/%	User Satisfaction/Score	Average feedback time/day	Repeated booking rate/%	Complaint rate/%	Recommendation rate/%
RFPAP+FP-Growth model	90.67	8.7	2.8	75.92	2.08	86.15

In Table 5, the model exhibits an accuracy of 90.67% in predicting user preferences and travel needs, while concurrently achieving a high user satisfaction score of 8.7. Users provided feedback on the model's recommendations with a median time of 2.8 days. Conversely, the model's complaint rate is a mere 2.08%, indicating its high reliability and stability. In conclusion, the RFPAP+FP-Growth model demonstrates efficacy in PSGT planning, consistently providing users with accurate and satisfactory personalized recommendations.

5 Discussion

The preceding research indicates that the RFPAP model prioritizes user behavior analysis and personalized recommendations, whereas the FP-Growth model emphasizes the extraction of frequent item sets and association rules. The integration of the two models allows for the analysis of the correlation between services while considering the user's personalized needs, thereby enhancing the comprehensiveness of the recommendation system. In self-guided tours, RFPAP is based on users' historical recommendations, while FP-Growth analyzes destination and activity associations to discover users' potential points of interest and provide accurate recommendations. The discrepancies in data can influence the outcomes of the models, including the user behavior, destination, and activity data. When integrating models, it is essential to ensure the effective integration of different data sources and features to guarantee optimal performance. In the field of research pertaining to the development of personalized self-service travel strategy models, the innovative integration of the RFPAP model and the FP-Growth model has the potential to bring about revolutionary progress in the realm of personalized services within the tourism industry. The RFPAP model provides tourists with comprehensive resource information by employing scientific analysis of the mechanical properties and evolutionary laws of tourism resources. The FP-Growth model employs big data mining techniques to accurately capture the travel preferences and consumption habits of tourists. The combination of the two not only addresses the shortcomings of traditional methods in terms of the provision of personalized services, but also offers tourists a more accurate and efficient travel experience through the precise evaluation of tourism resources, the provision of personalized recommendations regarding tourism plans, and the optimization of tourism services. This integration method not only reflects the scientific and progressive nature of the approach, but also meets the needs of the personalized tourism era, injecting new vitality into the development of the tourism industry.

6 Conclusion

The study aims at the problem of ignoring the user's personal preference and the low accuracy of the feature selection of tourist destinations in traditional personalized recommendation algorithms. The study constructed the SM for PSGT by integrating the RFA preference model and FP-Growth association model. Through the construction of the SM, it is expected to formulate a PSGT recommendation strategy for tourists that is more in line with their needs, and to improve the travel experience and satisfaction. The results indicated that the mean value of support, mean value of confidence and mean value of enhancement of SM in attraction associative relationship research were 0.183, 0.597 and 1.093, respectively. The recommended predictive value was 82.17% and the feasibility predictive value was 65.94%, which were all small gaps from the true values, which indicated that all the SM constructed in the study had a strong practicality. The method not only significantly improves the PS capability of attractions, but also enhances the mining capability of correlations, thus improving the formulation and achievement of PSGT strategies. Although the study achieved good results, there are still some shortcomings. The dataset used in the study is small, covering only a limited sample of self-guided tour destinations and users. The next step of the study can consider expanding the size of the dataset to improve the accuracy and generalization ability of the model.

References

[1] Ravinder Jangra, S. P. Kaushik, and Surjit Singh Saini. An analysis of tourist’s perceptions toward tourism development: study of cold desert destination. Geography and Sustainability. 2(1):48-58, 2021. https://doi.org/10.1016/j.geosus.2021.02.004.

[2] I Gusti Bagus Rai Utama, Christimulia Purnama Trimurti, Jaya Pramono, and Ni Putu Dyah Krismawintari. Tourist behavior destination selection model. International Journal of Advanced Science and Technology. 29(5):7352-7375, 2020. https://doi.org/orcid.org/0000-0003-0517-5355.

[3] Miriam Edith Pérez-Romero, Martha Beatriz Flores-Romero, and Víctor G. Alfaro-García. Tourism and destination competitiveness: an exploratory analysis applying the forgotten effects theory. Journal of Intelligent & Fuzzy Systems.

40(2):1795-1804, 2021. https://doi.org/10.3233/JIFS-189186.x.

[4] Hamam Mokayed, Tee Zhen Quan, Lama Alkhaled, and V. Sivakumar. Real-time human detection and counting system using deep learning computer vision techniques. Artificial Intelligence and Applications. 1(4):221-229, 2023. https://doi.org/10.47852/bonviewAIA2202391.

[5] Weiguang Xie, and Yunchi Ma. Tourism resource evaluation and countermeasures based on network communication and topsis algorithm. Wireless Communications & Mobile Computing. 21(5):1-13, 2021. https://doi.org/10.1155/2021/8172282.

[6] Bin Deng, Jun Xu, and Xin Wei. Tourism destination preference prediction based on edge computing. Mobile Information System. 21(2):1-11, 2021. https://doi.org/10.1155/2021/5512008.

[7] Shengsha Xu. An apriori algorithm to improve teaching effectiveness. International Journal of Performability Engineering. 16(5):792-799, 2020. https://doi.org/10.23940/ijpe.20.05.p13.792799.

[8] Bruno Miguel Vieira, Ana Pinto Borges, and Elvira Pacheco Vieira. The role of social networks for decision-making about tourism destinations. International Journal of Internet Marketing and Advertising. 18(1):1-27, 2023. https://doi.org/10.1504/ijima.2023.10053065.

[9] Miftachul Huda. Digital marketplace for tourism resilience in the pandemic age: voices from budget hotel customers. International Journal of Organizational Analysis. 31(1):149-167, 2022. https://doi.org/10.1108/IJOA-10-2021-2987.

[10] Huang-Ping Yen, Po-Chi Chen, and Kung-Cheng Ho. Analyzing destination accessibility from the perspective of efficiency among tourism origin countries. SAGE Open. 11(2):1-14, 2021. https://doi.org/10.1177/21582440211005752.

[11] Jianhui Liang. Problems and solutions of art professional service rural revitalization strategy based on random forest algorithm. Wireless Communications and Mobile Computing. 22(1):1-11, 2022. https://doi.org/10.1155/2022/9752512.

[12] Heqing Zhang, Tingting Guo, and Xiaobo Su. Application of big data technology in the impact of tourism E-commerce on tourism planning. Complexity. 21(13):1-10, 2021. https://doi.org/10.1155/2021/9925260.

[13] Wolfram Höpken, Marcel Müller, Matthias Fuchs, and Maria Lexhagen. Flickr data for analysing tourists' spatial behaviour and movement patterns: a comparison of clustering techniques. Journal of Hospitality and Tourism Technology. 11(1):69-82, 2020. https://doi.org/10.1108/JHTT-08-2017-0059.

[14] Lior Shabtay, Philippe Fournier-Viger, Rami Yaari, and Itai Dattner. A guided FP-Growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data. Information Sciences. 553(2):353-375, 2021. https://doi.org/10.1016/j.ins.2020.10.020.

[15] Yanling Dong, Shufen Zhang, Jingcheng Xu, Haoshi Wang, and Jiqiang Liu. Random forest algorithm based on linear privacy budget allocation. Journal of Database Management. 33(2):1-19, 2022. https://doi.org/ 10.4018/JDM.309413.

[16] Krit Sriporn, Cheng-Fa Tsai, and Paohsi Wang. Analyzing tourist behavior in virtual museums using intelligent approach with feature selection. Journal of Information Science and Engineering. 38(3):681-695, 2022. https://doi.org/10.6688/JISE.202205_38(3).0011.

[17] Sérgio Moro, and Joaquim Esmerado. An integrated model to explain online review helpfulness in hospitality. Journal of Hospitality and Tourism Technology. 12(2):239-253, 2021. https://doi.org/10.1108/JHTT-01-2020-0026.

[18] Jiaman Ding, Cuihua Liu, Runxin Li, Jinguo You, and Lianyin Jia. An improved deep forest classification algorithm. International Journal of Modeling Identification and Control. 40(4):305-314, 2022. https://doi.org/10.1504/IJMIC.2022.125548.

[19] Hong Li, Man Qiao, and Shuai Peng. Research on the recommendation algorithm of rural tourism routes based on the fusion model of multiple data sources. Discrete Dynamics in Nature and Society. 22(6):1-10, 2022. https://doi.org/10.1155/2022/2262148.

[20] Jiaqi Zhang. A novel recommendation algorithm based on multi-objective evolution. Applied and Computational Engineering. 22(6):29-34, 2023. https://doi.org/10.54254/2755-2721/2/20220524.

[21] Duaa Hamed AlSaeed. LOCUS: a mobile tourism application and recommender system for personalized places and activities. Informatica. 47(2), 2023. https://doi.org/10.31449/inf.v47i2.4351.

[22] Zakria Mahrousa, Dima Mufti Alchawafa, and Hasan Kazzaz. Frequent itemset mining based on development of fp-growth algorithm and use mapreduce technique. Association of Arab Universities Journal of Engineering Sciences. 28(1):83-98, 2021. https://doi.org/10.33261/JAARU.2021.28.1.008.

[23] Indra Griha Tofik Isa. Sales strategy by determining tenant position at Sriwijaya state polytechnic through association rules analysis. International Journal of Entrepreneurship and Business Development. 5(3):563-573, 2022. https://doi.org/10.29138/ijebd.v5i3.1855.

[24] Jianzhou Cui. Tourism English based on association rules mining algorithm. ICAIIS 2021: 2021 2nd International Conference on Artificial Intelligence and Information Systems. 28(6):1-5, 2021. https://doi.org/10.1145/3469213.3472785.

[25] Houria Laaroussi, Fatima Guerouate, and Mohamed Sbihi. Deep learning approach and topic modelling

for forecasting tourist arrivals. International Journal of Electrical and Computer Engineering Systems. 14(4):411-419, 2023. https://doi.org/10.32985/ijeces.14.4.5.

[26] Liyan Shen. Analysis of immersive virtual reality tourism resources based on navier-stokes equations. Informatica. 48(6), 2024. https://doi.org/1031449/inf.v48i6.5514.

[27] Zhaoyu Wang. Ecological protection design strategy for mountain tourism highway slopes. Journal of World Architecture. 5(3):9-12, 2021. https://doi.org/10.26689/jwa. v5i3.2176.

[28] U. Umbetov, A. B. Shinykulova, and G. S. Morokina. Algorithm of solving goal in tourism for hierarchical management with limited necessary resources. Herald of Science of S Seifullin Kazakh Agro Technical Research University. 1(108):190-201, 2021. https://doi.org/10.51452/kazatu.2021.1(108).155.

[29] Purwanto Purwanto, Retnowati Retnowati, and Hari Suryanto. Strategy for enhancing community economy through optimization of tourism areas (a study on rawa pening lakes in central java province-indonesia). Journal of International Conference Proceedings. 3(1):183-193, 2020. https://doi.org/10.32535/jicp.v2i4.794.

[30] Córdoba Mariano, and Balzarini Mónica. A random forest-based algorithm for data-intensive spatial interpolation in crop yield map. Computers and Electronics in Agriculture. 184(6):1-9, 2021. https://doi.org/10.1016/j.compag.2021.106094.

Integrating ARIMA Model for Enhanced Financial and Tax Data Management and Accurate Departmental Budget Prediction

Mingyue Lv
School of Accounting, Changchun Finance College, Changchun 130000, China
E-mail: lvmingyue1990@126.com

Keywords: time series, financial and tax data, departmental budgets, leadership decision-making, data management

Received: July 4, 2024

Aiming at the problems of low timeliness and inaccurate budgets in the financial sector, the study constructs a model for fiscal and tax data management and departmental budget forecasting based on the time series model, using the autocorrelation function and partial correlation function to identify the time series characteristics. At the same time, the architecture of the fiscal and tax data management system is designed to realize data fusion and sharing and improve forecasting accuracy. The experimental results indicated that the departmental budget data of a province from 2015 to 2022 was a non-stationary series, which was converted to a stationary series after the smoothing process. The parameters of the time series model were determined by combining auto correlation function, partial auto correlation function and Akaike information criterion. The determined time series model was used to predict the departmental budget and the average error between the TV and PV of the departmental budget for the years 2020 to 2022 were 7.3%, 7.4% and 12.1%, respectively. The prediction errors of the research-designed model, the generalized autoregressive conditional heteroskedasticity model, the long and short-term memory network model and the generative adversarial network model in the static prediction were 12.1%, 10.9%, 28.1% and 28.9%, respectively. In the dynamic prediction, the average prediction errors of the four models were 11.3%, 15.4%, 6.3%, and 10.1%, with the accuracy of 88.7%, 84.6%, 93.7%, and 89.9%, respectively. Therefore, the prediction precision of the research prediction method is high and the error rate is low, which can effectively realize departmental budget prediction analysis and is of practical significance in guiding financial practice.

Povzetek: Raziskava uporablja model ARIMA za integracijo upravljanja finančnih in davčnih podatkov ter kvalitetno napovedovanje proračunov oddelkov, kar izboljša točnost in podpira odločanje vodilnih.

1 Introduction

Economic and social data are being collected due to the rapid development of computer science and computing, which makes it very easy for government agencies to implement intelligent management and services [1-2]. Financial work is related to all aspects of the economy and society, the use of economic and social data can improve the level of financial work. The traditional financial data processing method has problems such as slow processing speed and low efficiency. The use of Chinese customized reports to analyze data can neither meet the demand for data analysis nor provide effective assistance to leadership decision-making [3-5]. Effective analysis and processing of data in the financial business system is an important means to assist leadership decision-making and guide financial practice. Data mining technology is currently an important technology for processing big data and performing knowledge discovery [6]. The application of data mining technology to data warehouses represents a sophisticated process, entailing the selection of optimal analysis algorithms and the extraction or mining of pertinent information [7-8]. Analytical algorithms for data mining approaches that are frequently utilized include neural networks, fuzzy sets, rule-based reasoning, decision trees, example reasoning, applied statistical methods, and genetic algorithms [9]. Times series (TS) algorithm is one of the applied statistical methods to make predictions based on the temporal nature of data. TS algorithms are widely used in medical diagnosis, stock prediction, image analysis and diagnosis of electromechanical systems [10]. Peng et al. proposed a classification algorithm based on TS images to address the difficulty of mapping wetland degradation. The method utilized TS images to synthesize phenological features, combined with random forest algorithm and hierarchical decision tree to classify wetland types. Experimental results indicated that the algorithm has good precision [11]. To estimate the worldwide solar radiation, Mughal and colleagues suggested a prediction time model based on neural networks. To anticipate an appropriate choice of global solar radiation, the strategy integrated a nonlinear autoregressive model with statistical analysis to assess the outcomes. The experimental results indicated that the model has a high precision with a root mean square error of 1.28% [12]. Garai et al. proposed a hybrid and its learning algorithm based on empirical modal decomposition of adaptive noise-complete sets to address the problem of irregularities present in noisy TS data. The

method utilized a combination of cryptographic algorithm and particle swarm algorithm to predict noisy TS data. The experimental results demonstrated that the algorithm was able to effectively predict the price series [13]. Li proposed dynamic time warping based on time weighting to address the problem of the influence of different time points. The method obtained matching time weights by measuring the distance between historical time points and the latest time points, combined with dynamic time warping. Experimental results demonstrated that the method was effectively used for similarity measurement in TS data mining [14]. Tan et al. proposed TS external regression algorithm in order to analyze the relationship between learning TS and continuous scalar variables. The method predicted the future values of TS through a temporal classification algorithm, combined with benchmarking. The experimental results indicated that the algorithm had high prediction precision [15].

Financial data management is the use of data warehousing techniques to enable intelligent management of the process of collecting, transforming, analyzing and distributing data. Shahid et al. proposed data warehouse for healthcare sensitive data applications to address the difficulty of monitoring changes in childhood obesity. The method collected children's data through different sensors and created an obesity prevalence model to monitor and store personal data on related behaviors. The outcomes demonstrated that this data warehouse could effectively enhance data precision and privacy [16]. Amo et al. constructed an educational data warehouse for educational data storage, analysis and access in order to improve educational data management performance. The approach utilized modular and extensible data systems to address data management and access complexity. The experimental results demonstrated the usefulness of this educational warehouse in assisting educational decision making and improving data security [17]. An IoT based real-time warehouse management system, suggested by Khan et al., separated the warehouse into several domains and combined architectural views to display the model in order to automate and digitize the administration of the manufacturing business. The approach collected and transmitted data through a generic prototype system. Experimental results indicated that the system was efficient in data transmission with low latency [18]. Kahn et al. migrated a research data warehouse to a public cloud in order to achieve innovation-driven clinical research data, thematically categorizing internal and public presentations and analysis documents before and after the transition. The experimental results showed that this migration process minimized the waste of resources and costs [19]. Tufano et al. proposed data warehouse management system to track data in order to mine logistics data to monitor its business processes, smooth warehousing processes and support strategic decision making. The method predicted benchmark metrics of the data storage system by training classifiers. Experimental results indicated that the warehouse system was effective in realizing data-driven and improving data prediction precision [20]. The results of the related research work are summarized in Table 1.

The application of data mining techniques can facilitate the classification and prediction of financial data, while data warehouse technology can be employed to transform financial data into knowledge. The combination of the aforementioned needs and technologies facilitates the acceleration of data management and mining in the financial sector. The study proposes a financial and tax data management and departmental budget prediction method that integrates the auto-regressive integrated moving average model (ARIMA), with the objective of assisting leaders in decision-making, improving financial sector management, and promoting financial innovation.

Table 1: Summary of related work

Literature number	Author(s)	Methodology	Data set	Results	Limitations
[11]	Peng K, Jiang W, Hou P, Wu Z, Ling Z, Wang X, Mao, D	A new algorithm for detailed wetland type classification combining k-fold random forests and hierarchical decision trees	North and South-Central Asia	Continental-scale wetland maps were obtained with an overall accuracy of 90.0 ± 0.5%.	Classification rules are not well set up, and there are some classification errors.
[12]	Mughal S N, Sood Y R, Jarial R K	Neural networks based on time series	/	The root-mean-square error was 1.28, which was lower than that of other time-series models; the hourly prediction model was more accurate than the daily prediction	/

				model.	
[13]	Garai S, Paul R K, Yeasin M, Paul A K	Hybrid and its learning algorithm based on adaptive noise-complete ensemble empirical modal decomposition; prediction of noisy time series data using encryption algorithm with particle swarm algorithm	Daily Wholesale Prices of Potatoes in Six Major Indian Markets	Optimization-based model combinations outperformed their individual counterparts	Insufficient prediction performance for data with different volatility levels
[14]	Li H	Dynamic time warping extension method based on time weights analysis	UCI data and financial stock exchange data	Different historical time points have different effects on the contribution to the minimum distance between two time series; the method facilitates similarity measures in time series data mining compared to existing methods	Model complexity needs to be simplified
[15]	Tan C W, Bergmeir C, Petitjean F, Webb G	Time series external regression algorithm	Time series external regression dataset	Rocket achieves the highest overall accuracy when adapting regression	Accuracy needs to be improved
[16]	Shahid A, Nguyen T A N, Kechadi M T	Three-tier flexible BigO data warehouse architecture, including backend, access control and controller layers	/	The data warehouse effectively improves data accuracy and privacy	Performance on distributed data needs to be improved
[17]	Amo D, Gómez P, Hernández-Ibáñez L，Fonseca D	Modular and scalable data system architecture	/	Addresses the complexity of data management and access	Model complexity needs to be simplified
[18]	Khan M G, Huda N U, Zaman U K U	An IoT-based real-time warehouse management architecture	A textile mill warehouse	Key performance parameters such as system resiliency, efficiency, and latency are effectively improved.	Architecture scalability, system security are not enough

[19]	Kahn M G, Mui J Y, Ames M J， Yamsani A K, Pozdeyev N, Rafaels N, Brooks I M	A thematic classification model for analyzing documents	/	Network, compute, and storage architectures that realize performance and cost advantages	Internal deployment network and system touch points add complexity
[20]	Tufano A, Accorsi R, Manzini R	Classifier-based warehousing system	Data from 16 different real companies in different industrial sectors and with different participants	Warehouse system is data-driven and improves data prediction accuracy	Insufficient research on predictive design of storage systems

2 Methods and materials

In light of the accelerated advancement of financial informatization technology, the accumulation of financial and tax data is becoming increasingly prevalent. To fully realize the value of these data and to conduct departmental budgeting in a scientific and rational manner, the study proposes the integration of the TS model for financial and tax data management and departmental budget prediction. This approach aims to anticipate financial and tax data trends and to inform the subsequent analysis and processing of these data. The method begins with the construction of the TS model, which is then utilized to analyze and process financial and tax data. This is followed by the mining of departmental historical budget data, which is then used to predict the departmental budget. The TS model serves as a guiding force throughout this process.

2.1 Construction of the times series model

TS data are correlated data that are arranged in chronological order and dynamically change over time, which is a special kind of stochastic process [21-22]. TS algorithms are useful for analyzing and forecasting future data as well as for identifying patterns in previous data. The auto-regressive and moving average model (ARMA) is the most commonly used TS prediction method [23-24]. In ARMA (p, q), AR is autoregressive and p denotes the autoregressive terms. Q indicates the quantity of moving average terms, while MA stands for moving average. ARMA is commonly used to deal with smooth data series [25-26]. The AR model is predicted by linearly combining the current moment perturbation value with the previous moment observation. The calculation is shown in Equation (1).

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} \quad (1)$$

In Equation (1), X_t denotes the observed value at moment t. $\phi_i \left(i = 1, 2, \cdots, p\right)$ denotes the autoregressive model coefficients. $X_i \left(i = t-1, t-2, \cdots, t-q\right)$ is the observed values at different moments. The perturbation values of the present and preceding moments are combined linearly to predict the MA model. Equation (2) shows the result of the calculation.

$$X_t = \mu_t + \theta_1 \mu_{t-1} + \theta_2 \mu_{t-2} + \cdots + \theta_q \mu_{t-q} \quad (2)$$

In Equation (2), μ_t denotes the residuals and $\theta_i \left(i = 1, 2, \cdots, q\right)$ denotes the moving average model coefficients. The calculation of ARMA model is shown in Equation (3).

$$\begin{aligned} X_t &= \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} \\ &+ \mu_t + \theta_1 \mu_{t-1} + \theta_2 \mu_{t-2} + \cdots + \theta_q \mu_{t-q} \end{aligned} \quad (3)$$

When TS is non-smooth, it needs to be converted to a smooth series after finite difference operation. The ARIMA is developed on the basis of ARMA and can be used for modeling non-smooth TS [27-28]. The TS model prediction testing process is shown in Figure 1.

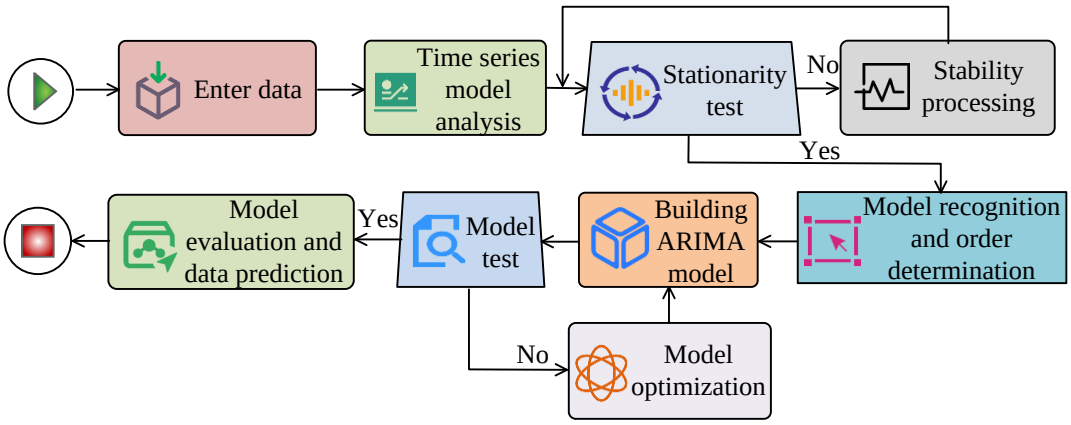


Figure 1: Time series model prediction flowchart

The objective of the stability test is to analyze and preprocess the raw data in order to obtain the model required for the study. This is followed by the preliminary stability test on the series, which is conducted in accordance with the time series diagram. After the stability test, if the data series is stationary, it is directly analyzed and processed. To acquire the differential order, differential processing must be performed if the data series is non-stationary. The data series after differential processing to get a differential series term, specifically as shown in Equation (4).

$$\nabla x_t = x_t - x_{t-1} \tag{4}$$

In Equation (4), ∇x_t denotes the first-order difference series term at moment t. x_t is the data series at moment t. x_{t-1} is the data series at the moment of $t-1$. Similarly, the second-order differential series term is obtained after the second-order difference (SOD) processing of the data series, as shown in Equation (5).

$$\nabla^2 x_t = \nabla x_t - \nabla x_{t-1} \tag{5}$$

In Equation (5), $\nabla^2 x_t$ denotes the second-order differential series term at moment t. ∇x_{t-1} denotes the first-order differential series term at moment $t-1$. Therefore, the higher-order differential series is calculated as shown in Equation (6).

$$\nabla^d x_t = \nabla^{d-1} x_t - \nabla^{d-1} x_{t-1} \tag{6}$$

In Equation (6), d denotes the difference order. $\nabla^d x_t$ and $\nabla^{d-1} x_t$ denote the d th and $d-1$ th differential series terms at the t moment, respectively. $\nabla^{d-1} x_{t-1}$ denotes the $d-1$ times difference series term of $t-1$ moment. If a non-stationary data series has a linear trend, the difference can be used to remove its linear trend and obtain a stationary series. The augmented Dickey-Fuller test (ADF) is used to further determine the stability of the differential TS. If the statistical P-value of the series test is less than 0.05, the differential series is determined to be stationary [29-30]. According to Equation (1), the nonzero characteristic root of the characteristic equation is obtained as x_1, x_2,..., x_p. A stationary series is indicated when the characteristic root's absolute value is smaller than 1. The characteristic root is equal to 1, which indicates that the series is nonstationary series. Model identification is based on the data auto correlation function (ACF) and partial auto correlation function (PACF), to identify the features of TS and construct the corresponding TS model. The data auto correlation is calculated as shown in Equation (7).

$$\hat{\gamma}_k = \frac{1}{n}\sum_{t=k+1}^{n} x_t x_{t-k}, k = 0,1,2,\cdots,n-1 \tag{7}$$

In Equation (7), $\hat{\gamma}_k$ denotes the data self-covariance and $x_i\left(i=1,2,\cdots,t\right)$ denotes the data series. Equation (8) can be used to compute the self-covariance of the starting data.

$$\hat{\gamma}_0 = \frac{\sum_{t=1}^{n} x_t^2}{n} \tag{8}$$

In Equation (8), $\hat{\gamma}_0$ represents the initial data self-covariance. Therefore, the data ACF is shown in Equation (9).

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} = \frac{\sum_{t=k+1}^{n} x_t x_{t-k}}{\sum_{t=1}^{n} x_t^2} \tag{9}$$

In Equation (9), $\hat{\rho}_k$ denotes ACF. The selection of ARIMA model is determined based on the data ACF and PACF. Table 2 displays the assessment criteria for the ARIMA model.

Table 2: Judgment criteria of ARIMA model

Model	ACF	PACF
$AR(p)$	Trailing	p-order truncation
$MA(q)$	q-order truncation	Trailing
$ARMA(p,q)$	Trailing	Trailing
Model not suitable	Truncation	Truncation

Trailing tails indicate that there are always non-zero values that vary randomly around zero after being greater than a certain order. Truncated tails indicate that the value tends to zero quickly after being greater than a certain order. The ARIMA model can be expressed as ARMA (p, d, q). Once the difference order d is determined, the ARIMA model becomes ARMA (p, q). A deformation of Equation (3) is performed as shown in Equation (10).

$$\begin{aligned}&\left(1-\phi_1 L-\phi_2 L^2-\cdots-\phi_p L^p\right) X_t \\ &=\left(1+\theta_1 L+\theta_2 L^2+\cdots+\theta_q L^q\right) \mu_t\end{aligned} \tag{10}$$

In Equation (10), $1-\phi_1 L-\phi_2 L^2-\cdots-\phi_p L^p$ and $1+\theta_1 L+\theta_2 L^2+\cdots+\theta_q L^q$ denote polynomials of order p and q about L, respectively. Using the delay operator, then the decentralized model is represented as shown in Equation (11).

$$\Phi(L) X_t=\Theta(L) \mu_t \tag{11}$$

The ARIMA model uses Akaike information criterion (AIC) for ordinalization. The AIC function is shown in Equation (12).

$$AIC=n \log \sigma^2+2(p+q) \tag{12}$$

In Equation (12), AIC denotes the function and σ^2 denotes the residual sum of squares. n denotes stationary series sample size. The AIC criterion fixed order calculation is shown in Equation (13).

$$\begin{aligned}&AIC(p,q)= \\ &\min_{k,l} AIC(k,l) \quad 0 \leq k \leq M, 0 \leq l \leq H\end{aligned} \tag{13}$$

In Equation (13), $AIC(p,q)$ denotes the AIC criterion fixed order function. M denotes the ARIMA model order maximum. H denotes the ARIMA model order maximum. In ARMA (p, d, q), $AR(p)$ model indicates the relationship between the current data and the data of the previous p period, $p \in \{0,1,2,3\}$. d indicates the number of differences of non-smooth data into smooth data series. The $MA(q)$ model represents the relationship between the current data and the error of the data in the previous q period, $q \in \{0,1,2,3\}$. White noise (WN) detection checks to see if all of the series' relevant data has been fully retrieved, or if the residual term is a WN series. If the series is WN, it indicates that the important data has been taken out of the series. The zero mean homoskedasticity of the WN series is calculated as shown in Equation (14).

$$E(\mu_t)=0 \tag{14}$$

In Equation (14), $E(\mu_t)$ denotes the WN series error. The WN series independent homogeneous distribution is calculated as shown in Equation (15).

$$E(\mu_t, \mu_s)=\begin{cases}\lambda^2, t=s \\ 0 \quad , t \neq s\end{cases} \tag{15}$$

In Equation (15), $E(\mu_t, \mu_s)$ denotes the error of series μ_t and series μ_s. The model test is to detect whether the residual series of ARIMA model is WN. If it is not WN, then the residuals still contain valuable information, and the model needs to be optimized or more valuable information needs to be extracted. Model prediction is to predict the model that passes the test. The overall construction process of the ARIMA model is shown in Figure 2.

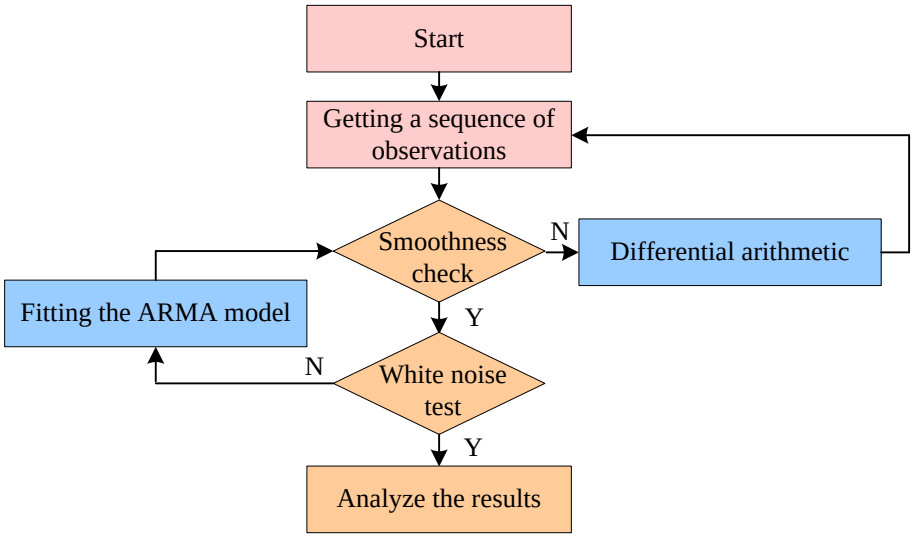


Figure 2: Flowchart of the overall construction of the ARIMA model

As illustrated in Figure 2, the ARIMA1 construction process initially entails the aggregation of time series data from disparate sources, including financial statements, sales records, market research, and so forth. This is followed by the application of linear interpolation to address any gaps in the data set and the utilization of a clustering algorithm to handle discrete values and anomalous data, thereby guaranteeing the completeness and precision of the data. The ADF smoothness test is performed on the processed data, and the results that do not pass the smoothness test must be differentiated until they pass the white noise test. If the sequence does not pass the white noise test, it is necessary to enter the model fitting. If the model passes all tests, the model is used to predict future values.

In addition, to ensure the robustness of the model, the study adopts the K-fold cross-validation method. The dataset is divided into K subsets, each of which is rotated as the test set, and the remaining subsets are used as the training set. The ARIMA model is trained on the training set, and the parameters of the ARIMA model are adjusted according to the characteristics of the data. Moreover, the error metric is used to measure the difference between the predicted and actual values of the model.

2.2 Design of a method for forecasting financial and tax departmental budget data based on ARIMA modeling

The effective management of financial and tax data facilitates the improvement of the precision and scientificity of the prediction of financial departmental budget data. The effective prediction of departmental budget data can be achieved by the comprehensive mining of departmental historical data through data mining technology, which is the financial and tax data analysis of the core content [31-33]. Therefore, the study proposes a prediction method for financial departmental budget data based on the ARIMA model to predict the financial departmental budget data and guide the analysis and processing of financial and tax data. At present, finance-related departments have generally established corresponding business application systems for daily departmental business processing. These systems do not realize data integration and sharing, resulting in lagging financial and tax basic data and business data, reducing the precision and timeliness of data analysis. The study suggests a financial and tax data management system to increase the efficiency and precision of data processing.

The architecture of the financial and tax data management system is displayed in Figure 3. The architecture consists of data collection, storage, application and release. The system integrates the fundamental financial and tax data within the business application framework, facilitating the integration of financial data and providing robust data support for informed decision-making at the executive level.

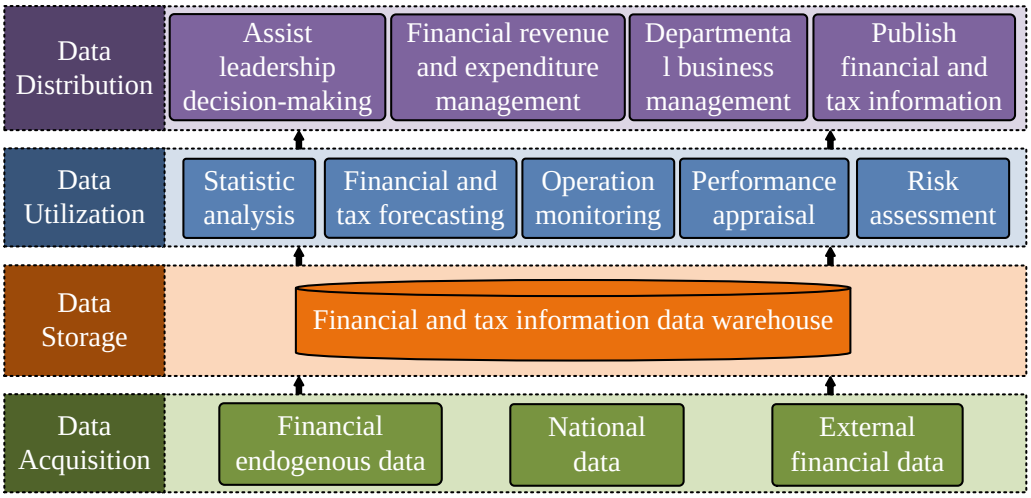


Figure 3: Architecture of financial and tax data management system

The business essential data is extracted, transformed, and loaded by the data collection layer before being uploaded to the data storage layer. The data storage layer analyzes and summarizes the data through the financial and tax information warehouse and prepares for data application. The data application layer performs statistical analysis, financial and tax prediction, and operation monitoring on the data in the warehouse. The data dissemination layer utilizes the results of the application layer for decision-making, management, and information dissemination. The financial and tax data warehouse adopts ERwin data modeling tools to construct the logical subject model of the financial and tax data warehouse and optimize the complex environment of financial and tax data management. The financial and tax data warehouse system divides financial and tax data into key taxpayer models, financial and tax revenue flow models and financial and tax revenue daily report models according to the analysis of financial and tax business data and user report requirements. According to the user requirements, the financial and tax revenue daily model dimension analysis is obtained. The details are shown in representation 3.

Table 3: Key taxpayer model

Dimension	Content
Date	Ten-day serial number; Month; Quarterly; Year; Monthly week series number; Weekly serial number within the year
Standard region	Region code; Region name; Regional level; Superior region code; Name of superior region
Tax level	Tax level code; Description of tax level
Levy item	Levy item code; The name of levy item; Levy item level; Superior levy item code; Name of superior levy item
Levy institution	Tax source code; Description of tax sources

In addition, the reports of the financial and tax revenue daily model contain monthly financial and tax revenues, cumulative annual revenues, monthly revenues for the same period of the previous year, and cumulative annual revenues for the same period of the previous year. On top of the financial and tax revenue daily model dimensions, the focused taxpayer model adds a standard industry dimension. The report contains the cumulative annual completion amount and the completion amount for the same period of the previous year. The financial and tax revenue flow model adds the local tax and national tax dimensions. The report contains the annual cumulative completion amount, monthly completion amount, completion amount in the same period of the previous year, and monthly completion amount in the same period of the previous year. According to the above logical subject model, ETL tools are used to process the raw financial and tax data from the data collection layer to provide basic data for the financial and tax data warehouse. The financial and tax business source data is stored in the Oracle database

system. The Trinity tool is utilized to update the database in real time, and finally the data is imported into the Teradata all-in-one machine. Automatic data extraction (ADE) extracts basic information from the Oracle database and generates data files. The data files are downloaded by the Trinity tool, saved to its local directory, processed, and imported into the Teradata all-in-one PC. ADE is able to export csv data files from Oracle database at regular intervals and put the csv data files into the directory of the specified file transfer protocol server. ADE can realize database connection parameter modification, manual export of historical data, automatic export of new data, and other functions. ADE workflow is shown in Figure 4.

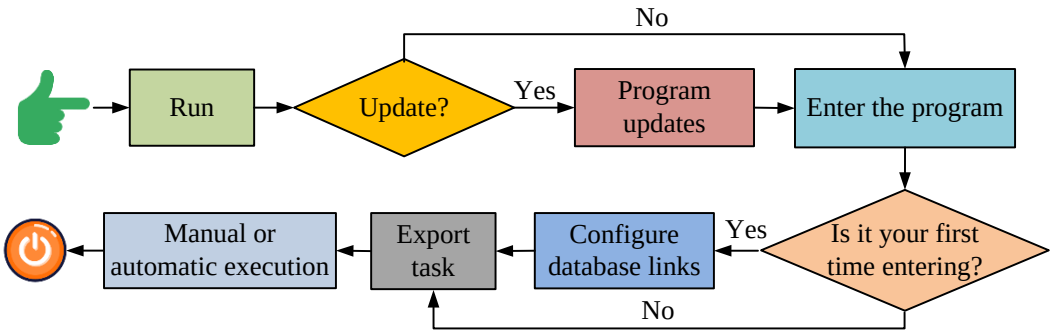


Figure 4: ADE workflow

Teradata Data Warehouse invokes Fastload script to read the data files under Trinity server into the database, completing the process of financial and tax data from the business system to the data warehouse. Similarly, all financial and tax data will be entered into the database to realize data fusion, which is a solid foundation for data analysis and processing. Teradata data warehouse consists of three levels of databases: source database, intermediate database and application database. The source database stores the raw data, the intermediate database stores the data processing results and intermediate tables, and the application database stores the final results of data processing. The analysis and visualization of financial and tax data is implemented by IBM Cognos software. The framework manager component is first used to construct a multidimensional data source and analyze the relationships between the above logical topic models and dimensions. Then the Report Studio reporting component is used for data visualization. Framework manager utilizes the relationship between the fact table and dimension table to generate data packages. The workflow is shown in Figure 5.

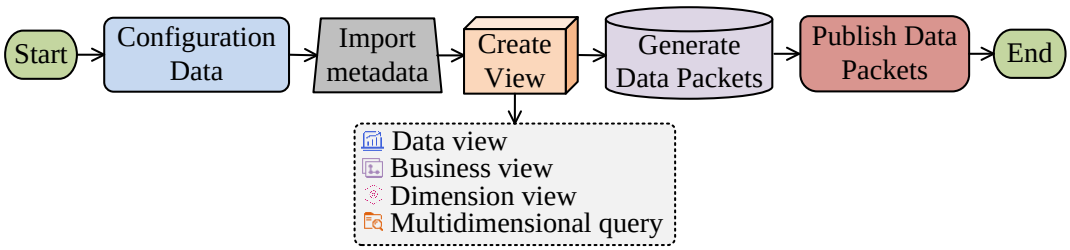


Figure 5: Framework manager workflow

Data view means that the fact table has correspondence in the dimension table. Business view is to change the name of the data table in the view to Chinese explanation. Dimension view is to create multidimension according to business. Multidimensional query is to put the fact table and dimension table into the corresponding topic. After the four-layer view is built, the data package is sent to the development server for financial and tax data analysis and visualization. The Report Studio reporting component is utilized to be able to build smart reports for chart federation. Different reports are linked to each other to enhance the flexibility of reports. Users are able to analyze the data from different dimensions, comprehensively discover the characteristics and connections of the data, and realize the intelligent visualization of the data. Once the fundamental financial and tax data housed within the financial sector's business application system have been aggregated and synthesized, the study proceeds to construct a processed data set in the form of a TS and devise a departmental budget prediction algorithm based on this TS. This approach allows for the observation of the year-by-year change rule governing the departmental budget data. According to the above TS model to predict, the specific prediction process is shown in Figure 6.

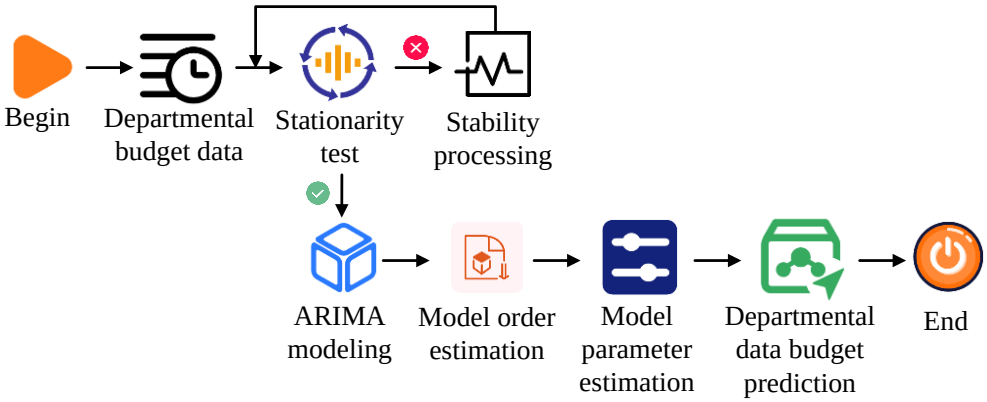


Figure 6: Method for predicting budget data in the financial department based on ARIMA

Stability tests are performed on the departmental budget data. If it is a smooth series, then go directly to ARIMA modeling, otherwise perform difference operations until the data is a smooth series. After the data is stable, ARIMA modeling is carried out and the ACF and PACF are used to determine model approximate ranges of orders p and q. The AIC criterion is then used to determine the specific orders p and q. The model difference orders are determined by the difference operation used in the smoothing process. Once all the parameters of the ARIMA model are obtained, predicts can be made for the departmental budget data.

3 Results

One of the important elements of financial work is the departmental budget indicator. In order to verify the effectiveness of the financial and tax data management and departmental budget prediction method that integrates ARIMA model, the study selects the total number of departmental budget of a province from 2015 to 2022 as the data test set. Through the data stability test and stable process, the departmental budget in 2023 is quantitatively predicted to provide data support for assisting the leadership in decision-making and implementing the departmental budget work in a scientific and effective way.

3.1 Experiments and analysis of the times series model

To verify the effectiveness of the financial and tax data management and departmental budget prediction method that integrates ARIMA model, the study adopts the error rate (Error) and precision rate (Precision) as the experimental evaluation indexes. The total amount of departmental budget of a province from 2015 to 2022 is taken as the experimental dataset. The experimental parameters are configured as 750GB of ROM, 8GB of RAM, and an Inter Core i7-quad-core processor. The experimental dataset collects departmental budget data on a quarterly basis, specifically on January 1, April 1, July 1, and October 1 of each year (in case of holidays, it is automatically postponed by one day). The departmental budget data TS for a province from 2015 to 2022 is shown in Figure 7.

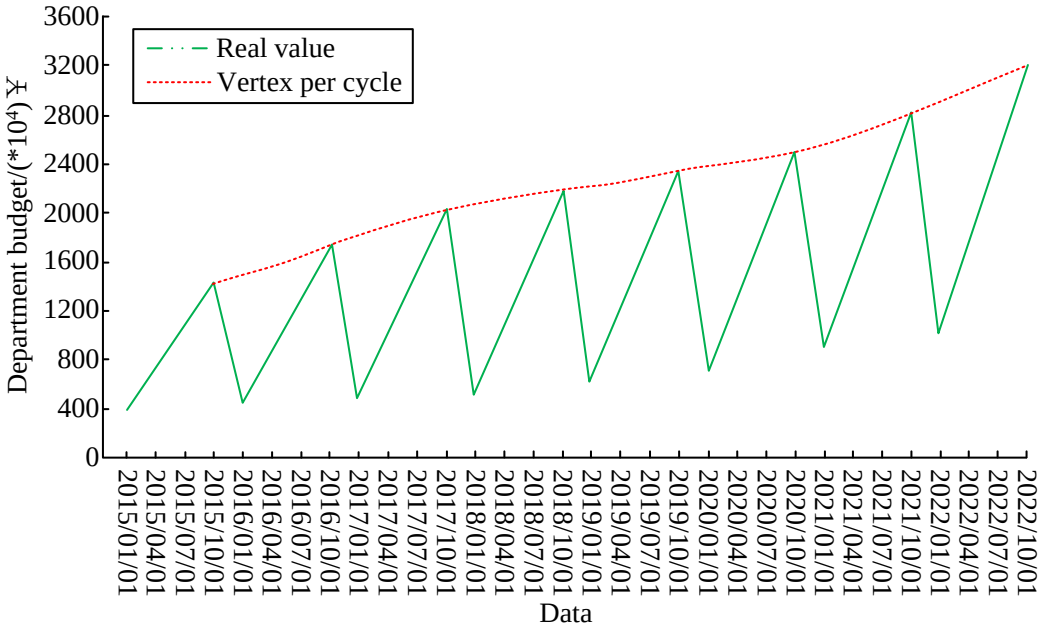
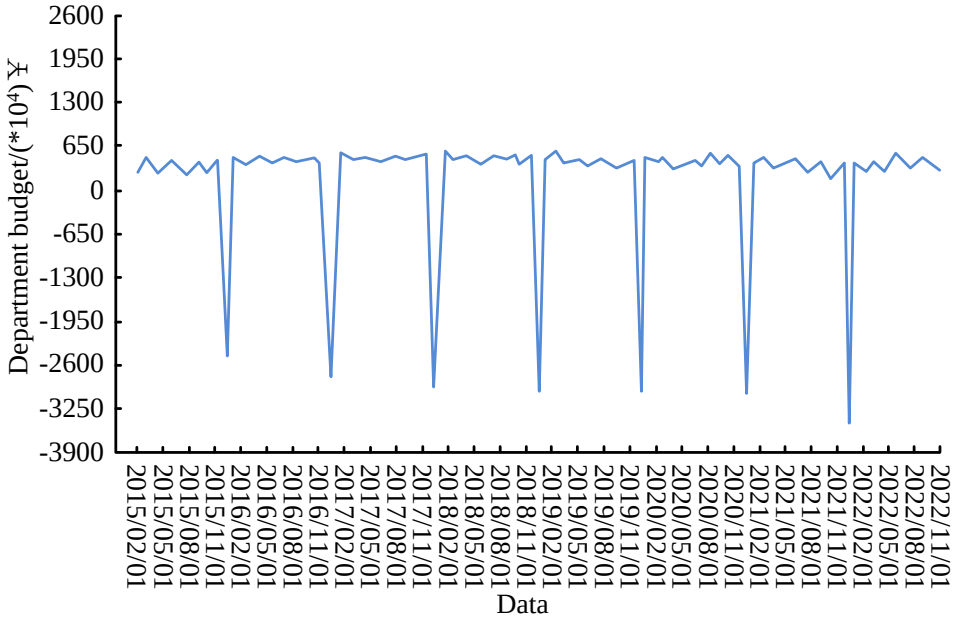
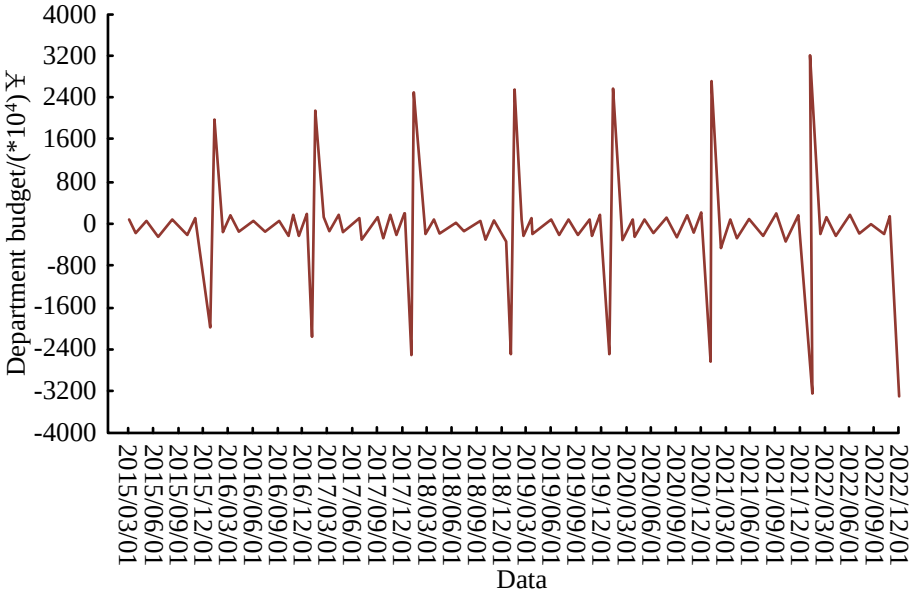


Figure 7: Time series diagram of department's original budget data

In Figure 7, the 2015-2022 departmental budget data is clearly cyclical and shows an upward trend at the top of each cycle. Therefore, the 2015-2022 historical departmental data is non-stationary and cannot be used directly for predicting analysis. It is necessary to convert the non-smooth historical departmental data into stationary series. The study uses difference operation to process the above non-smooth data. Since the above data has a non-linear trend, 2 difference operations are performed to smooth the data. The results are shown in Figure 8.



(a) First order differential sequence diagram



(b) Second order differential sequence diagram

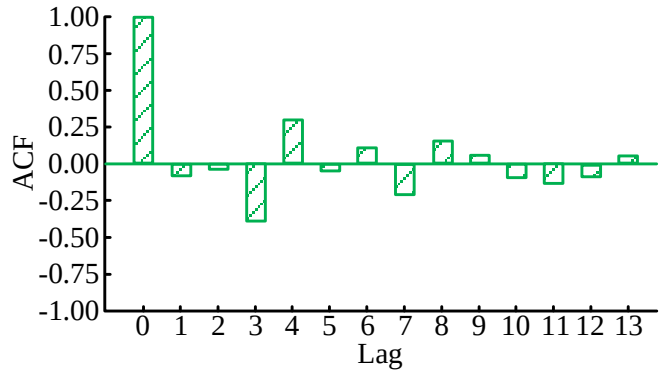
Figure 8: Method for stabilizing departmental budget data

In Figure 8(a), the first-order differential series did not fluctuate randomly up and down around the zero level, indicating that this budget data series did not reach a smooth state. To facilitate the progression of the series to a harmonious conclusion, it is necessary to continue differentiating the first-order differential series. In Figure 8(b), the second-order differential series is basically floating up and down regularly around the zero level line, with no obvious upward or downward trend characteristics, and the series is stable after the SOD. The 2015-2022 departmental budget data meets the requirements of the TS-based departmental budget prediction algorithm, so the historical departmental budget data can be used in the prediction algorithm to predict the departmental budget. To verify that the the SOD of the departmental budget data series has really reached a smooth state, the study continues to carry out the ADF test on the series. The results of ADF test are shown in Table 4.

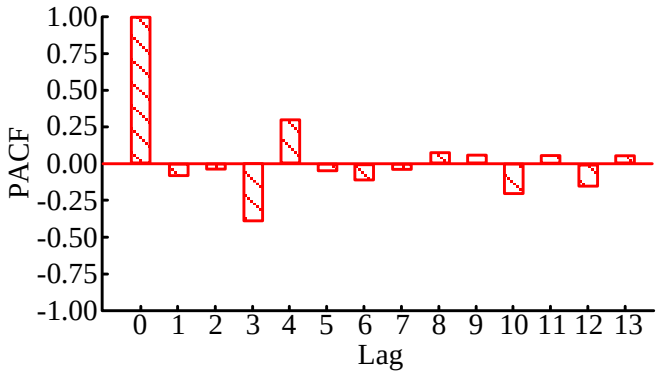
Table 4: ADF test result table

d	*t*	*P*	Critical value		
			1%	5%	10%
0	-0.812	0.913	-3.678	-2.863	-2.584
1	-2.139	0.229	-3.678	-2.863	-2.584
2	-4.325	0.000	-3.716	-2.871	-2.592

In Table 4, the t-value of the ADF test of the departmental budget data series without difference treatment is -0.812, P=0.913>0.05, which cannot reject the meta-hypothesis, indicating that the original TS is not stable. The t-value of the ADF test of the data after first-order difference for the departmental budget data series is -2.139, P=0.229>0.05, which cannot reject the meta-hypothesis, indicating that the series is not smooth. After the series is second-order differenced, the t-value of the ADF test of the data is -4.325, P=0.000<0.05, which rejects the original hypothesis, indicating stationary series. Therefore, it can be confirmed that the SOD of the departmental budget data series achieves stability. The departmental budget data series, subjected to a WN test using Python, returned a P-value of 0.000, which is less than 0.05. This result rejects the original hypothesis that the TS is a pure random series. Therefore the second-order differential series is not a WN series and is of research interest. The ARMA (p, q) model orders p and q of the stable non-WN series are initially determined by ACF and PACF. The ACF and PACF of the 2015-2022 departmental budget data series after SOD are shown in Figure 9.



(a) ACF of second-order differential sequence



(b) PACF of second-order differential sequence

Figure 9: ACF and PACF of second-order differential series

In Figure 9(a), the ACF fluctuation process around the zero-value deletion is slower, which is due to the ACF trailing off. In Figure 9(b), the PACF fluctuation process around the zero-value censoring is continuous, which is due to the PACF trailing off. After the PACF and ACF are roughly determined for q and p, the final model order is determined by the AIC criterion for different values of p and q at different values. Table 5 displays the AIC function values for the ARMA model.

Table 5: ARMA model AIC function values

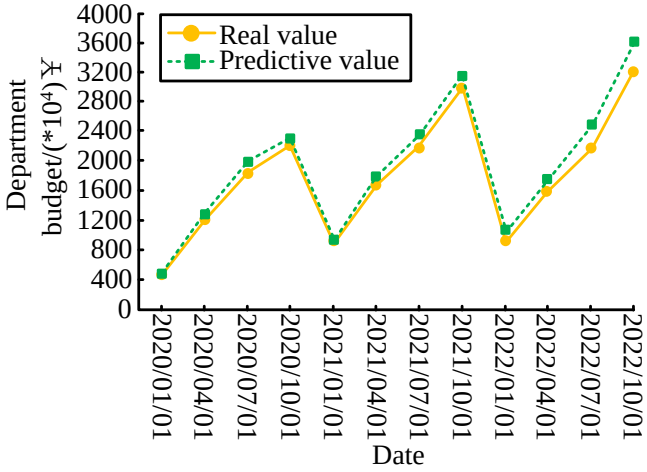
ARMA	AIC	ARMA	AIC
ARMA (0, 0)	23976.23	ARMA(2, 0)	23970.45
ARMA(0, 1)	23974.54	ARMA(2, 1)	23954.32
ARMA(0, 2)	23972.33	ARMA(2, 2)	23932.58
ARMA(0, 3)	23982.62	ARMA(2, 3)	23924.74
ARMA(1, 0)	239.82.30	ARMA(3, 0)	23974.63
ARMA(1, 1)	23978.65	ARMA(3, 1)	23974.50
ARMA(1, 2)	23974.21	ARMA(3, 2)	23920.56
ARMA(1, 3)	23985.75	ARMA(3, 3)	23926.35

In Table 5, the AIC function value of ARMA (3, 2) is minimized when P=3 and q=2. The model fits the data better the smaller the data, according to the AIC criterion. Therefore ARIMA (3, 2, 2) model is the best fitting model for departmental budget data. The residual WN test is employed to ascertain whether the pertinent information in the departmental budget data series has been fully extracted. The resulting value of P=0.827 is found to be greater than 0.05, thereby suggesting that the TS is a WN series. This finding implies that departmental budget prediction can benefit from the application of the ARIMA (3, 2, 2) model.

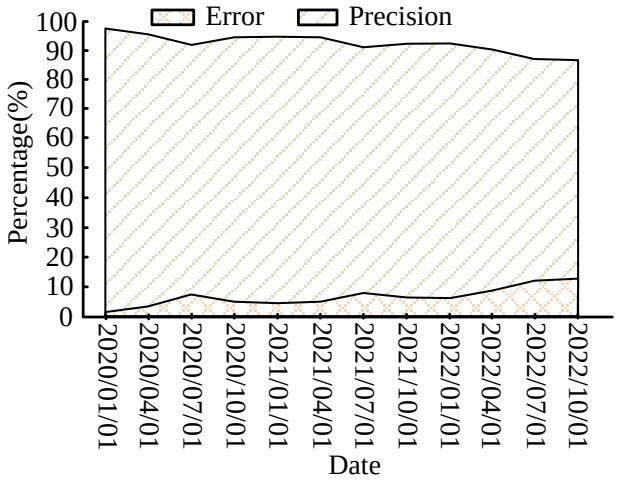
3.2 Experimentation and analysis of departmental budget prediction method based on ARIMA modeling

To verify the feasibility of the prediction method of financial departmental budget data based on ARIMA (3, 2, 2) model, the study takes the historical data of departmental budget from 2015 to 2019 as the data test set. The effect of departmental budget prediction from 2020 to

2022 is evaluated by error rate and precision rate indicators. The annual budget, monthly budget, and daily budget of the financial department are predicted and analyzed. Error rate=|True value (TV)-Predicted value (PV)|/TV. Precision = 1-Error. The results of departmental budget prediction for the years 2020 to 2022 are shown in Figure 10.



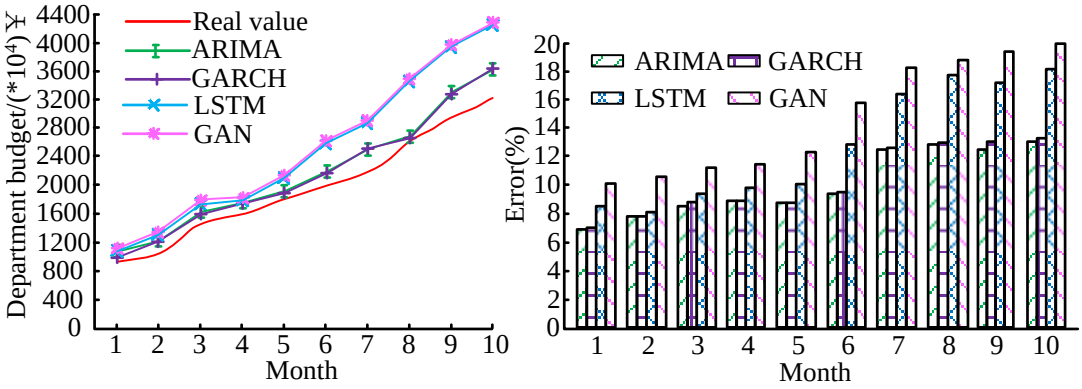
(a) True and predicted



(b) Error and precision

Figure 10: Departmental budget results

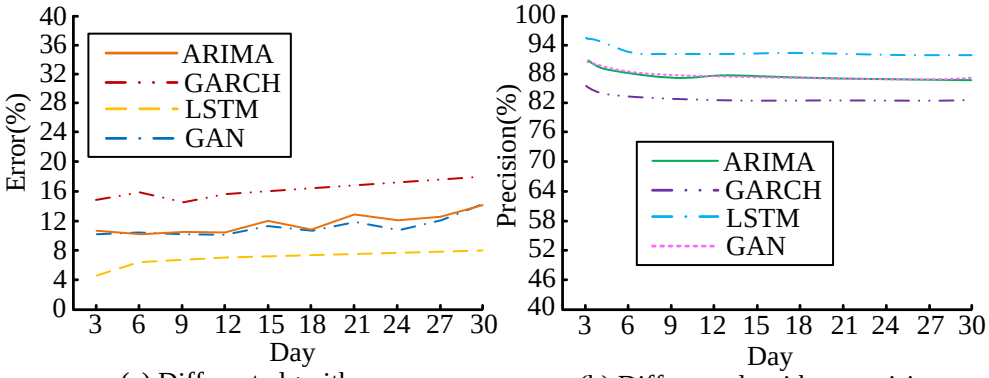
In Figure 10(a), the annual projected values of departmental budget are 182.6 million yuan, 247.25 million yuan, and 266.7 million yuan for the years 2020 through 2022, respectively. The departmental budget annual TV are 17.01 million yuan, 230.25 million yuan, and 237.9 million yuan, respectively. The error between the TV and PV for the years 2020 through 2022 is 7.3%, 7.4%, and 12.1%, respectively. The trend of the curve change between the true value and the PV is consistent. In Figure 10(b), the prediction error displays an upward trend as the year increases. This is due to the high precision of ARIMA (3, 2, 2) model in short-term predicting. When the financial departmental budget is affected by major emergencies, the model prediction precision decreases. Combining Figures 10(a)-(b), the prediction accuracies of the departmental budget data from 2020 to 2022 are all above 85%, indicating that the ARIMA (3, 2, 2) model predicts better. To further test the effectiveness of ARIMA(3, 2, 2) model for fiscal departmental budget data prediction method, the study selects generalized autoregressive conditional heteroskedasticity (GARCH), long short term memory (LSTM) and generative adversarial networks (GAN) with ARIMA (3, 2, 2) model for departmental budget prediction. Based on the state of time, the methods are categorized into static and dynamic prediction. The static prediction interval is January October 2022. The results of the static prediction of fiscal departmental budget are shown in Figure 11.



(a) Static prediction results (b) Different algorithm errors

Figure 11: Static prediction results using different algorithms

In Figure 11(a), the average monthly budget of the financial sector from January to October 2022 is 19.825 million yuan. In the static prediction, the ARIMA (3, 2, 2) model, the GARCH model, the LSTM model, and the GAN model have monthly PV of 2,225,000-yuan, 22,000,000-yuan, 25,396,000 yuan, and 25,561,000 yuan, respectively. The errors between the PV and the TV obtained from the ARIMA (3, 2, 2) model, GARCH model, LSTM model and GAN model are 12.1%, 10.9%, 28.1% and 28.9%, respectively. The ARIMA (3, 2, 2) model reduces the prediction error by 56.9% and 58.1% than the LSTM and GAN models and improves the prediction error by about 10% than the GARCH model. In Figure 11(b), the errors of ARIMA (3, 2, 2) model and GARCH model are basically the same. Overall, the RIMA (3, 2, 2) model has a smaller prediction error. The GAN model has the largest prediction error and the worst prediction effect, which is due to the impact of the amount of data in the GAN model, resulting in a decrease in its prediction precision. The dynamic prediction interval is from October 1 to 30, 2022. The results of the dynamic prediction of the financial departmental budget are shown in Figure 12.



(a) Different algorithm errors (b) Different algorithm precisions

Figure 12: Dynamic prediction results using different algorithms

In Figure 12(a), the average prediction errors obtained from the ARIMA(3, 2, 2) model, GARCH model, LSTM model, and GAN model are 11.3%, 15.4%, 6.3%, and 10.1%, respectively, in the dynamic prediction. The LSTM model has the smallest prediction error. The ARIMA (3, 2, 2) model and the GAN model show a small difference in the prediction error. This is due to the fact that the LSTM model and GAN model are deep learning prediction models with strong learning fitting ability and better prediction. The GARCH model has the largest prediction error and the worst prediction effect. In Figure 12(b), the average prediction precision obtained from ARIMA (3, 2, 2) model, GARCH model, LSTM model and GAN model are 88.7%, 84.6%, 93.7% and 89.9%, respectively. The GARCH model has the worst prediction results. The prediction precision of ARIMA (3, 2, 2) model, LSTM model and GAN model are above 85%, which indicates that the prediction result of ARIMA (3, 2, 2) model is good and the prediction is valid. Therefore, according to the predicted data, the total number of departmental budget is judged. Moreover, the budget is allocated to the priority sub-sections of each department under the premise of not exceeding the budget, which provides powerful data support to assist the leadership in decision-making.

Finally, the sensitivity analysis results of the ARIMA model are shown in Table 6. There are significant differences in the sensitivity of different factors to the

model prediction performance. The sensitivity of autoregressive order and differential integration order is as high as 83.24% and 81.06%. The sensitivity of moving average order is slightly lower, only 65.16%. Accordingly, the results of the sensitivity analysis enable the study to select the optimal parameter combinations for enhancing the predictive performance of the model.

Table 6: Sensitivity analysis of ARIMA model

Dimension	Influence factor	Management elements	Absolute sensitivity value
Parameters	Autoregressive order	ARIMA	6.16%
	Differential integration order	ARIMA	89.16%
	Moving average order	ARIMA	79.58%

4 Discussion

Under the background of big data and information age, economic and financial work is facing unprecedented opportunities and challenges. The conventional budget forecasting techniques employed by finance and taxation departments predominantly rely on the incremental budget method, zero-based budget method, fixed budget method, and so forth. These techniques are primarily based on the budget implementation of the previous year, coupled with adjustments made to align with changes in relevant factors, with the aim of preparing a budget that accurately reflects the anticipated comprehensive balance of economic activities. However, the effectiveness of these techniques is often limited, particularly in terms of their accuracy and the extent to which they facilitate effective budget management. In recent years, the development of computer information technology has accelerated the progress of reform of financial wisdom management and wisdom services, including data processing platforms such as distributed storage and cloud computing technology. At the same time, advanced data analysis algorithms such as deep learning and time series analysis have been gradually applied in the field of economy and finance. However, the traditional time series analysis still has the shortcomings of higher complexity and lower accuracy in the process of analyzing the trends and patterns of financial data. Meanwhile, in recent years, various research scholars have begun to introduce advanced time series analysis algorithms into various fields. Literature [12] Mughal S N et al. introduced time series based neural network for solar radiation forecasting with root mean square error of 1.28. Literature [13] Garai S et al. utilized a hybrid and its learning algorithms based on adaptive noise-complete set empirical modal decomposition to achieve the prediction of the market price. Literature [14] Li H et al. based on the dynamic time distortion expansion method of time weights analysis realized the prediction of financial stock trading. In this regard, the study innovatively selected ARIMA model to construct a departmental budget forecasting model. The experimental results showed that in static forecasting the prediction errors of ARIMA (3, 2, 2) model, GARCH model, LSTM model, and GAN model were 12.1%, 10.9%, 28.1%, and 28.9%, respectively. Moreover, the maximum reduction of prediction error was 58.1%. In dynamic prediction, the average prediction errors of ARIMA(3, 2, 2) model, GARCH model, LSTM model, and GAN model were 11.3%, 15.4%, 6.3%, and 10.1%, respectively. The average prediction accuracy were 88.7%, 84.6%, 93.7%, and 89.9%, respectively. The prediction of GARCH model was the worst, and the prediction of ARIMA (3, 2, 2) model, LSTM model and GAN model all have prediction accuracy above 85%. The ARIMA model was effective in static forecasting, as it was capable of capturing the linear relationship of time series data with greater precision. However, its predictive accuracy was slightly inferior to that of the GARCH model. This was attributed to the ARIMA model's comparatively limited capacity to cope with volatile aggregations in the data. Whereas the LSTM and GAN models had slightly worse prediction performance in static prediction due to insufficient data features. The slightly lower prediction accuracy of ARIMA model in dynamic forecasting compared to LSTM model was due to better sensitivity of LSTM model in dynamic environment.

The study introduces advanced time series analysis algorithms to the field of finance, and the ARIMA model can more accurately capture the complex patterns and trends in fiscal data, thus improving the accuracy of budget forecasting. Meanwhile, the ARIMA model is able to continuously update and adjust the forecasting results with the addition of new data. Moreover, it can take into account the impact of multiple factors on fiscal data, thus realizing an automated and intelligent forecasting process. The research advances the comprehensive advancement of the domain of financial data forecasting and the implementation and incorporation of financial data forecasting technology across a broader range of industries. This can serve as a crucial foundation for the formulation of fiscal policies and economic plans.

5 Conclusion

The amount of economic and social data has increased significantly as a result of computer science advancements. The data may be evaluated and extracted in an efficient manner through the utilization of data mining and warehouse technologies. The field of finance is intertwined with all aspects of economic activity. The mining and analysis of financial data can therefore be an effective means of enhancing the decision-making abilities

of those in positions of leadership, as well as facilitating the delivery of efficient and intelligent management and services. By analyzing financial and tax data and predicting departmental budget data in depth, the study proposed a financial and tax data management and departmental budget prediction methodology incorporating the ARIMA model. The method enabled quantitative prediction of future data for the financial sector. The method first analyzed and evaluated the order of the ARMA model. For the non-smooth financial and tax data, the ARIMA model was obtained by adding difference processing to the ARMA model. The non-stationary data were also converted into stationary data and then analyzed and processed. Then the data warehouse was constructed to integrate and summarize the financial and tax data, which provides powerful data support to assist the leadership in decision-making. The outcomes indicated that the departmental budget data of a province from 2015 to 2022 was non-stationary series. It was transformed into stationary series by SOD processing, and the order P=3 and q=2 were determined by using ACF, PACF, and AIC criterion to obtain ARIMA (3, 2, 2) model to predict the departmental budget. The error rates for 2021 and 2022 were essentially within 15%, and the precision rates were both above 85%. In static prediction, the RIMA (3, 2, 2) model had the smallest prediction error. In dynamic prediction, ARIMA (3, 2, 2) model was good with a prediction precision of 88.7%. Accordingly, the research method is an effective means of predicting departmental budget data, with a low error rate, and provides robust data support for leadership decision-making. The study of natural disasters and other factors affecting the financial departmental budget is not yet comprehensive. In the future, it will be expanded to include natural disasters and other factors affecting the economy.

Fundings

The research is supported by Project of the 2024 Teaching Reform Research on Vocational Education and Adult Education in Jilin Province: Research on the Practice of Teaching Reform of the Course "Tax Calculation and Declaration" Enabled by Digital Technology in Higher Vocational Colleges Project Approval Number: 2024ZCZ016.

References

[1] D. Bergemann, A. Bonatti, and T Gan, "The economics of social data," The RAND Journal of Economics, vol. 53, no. 2, pp. 263-296, 2022. https://doi.org/10.1111/1756-2171.12407

[2] J. Sheng, J. Amankwah-Amoah, Z. Khan, and X Wang, "COVID-19 pandemic in the new era of big data analytics: methodological innovations and future research directions," British Journal of Management, vol. 32, no. 4, pp. 1164-1183, 2021. https://doi.org/10.1111/1467-8551.12441

[3] S. Duleba, F. K. Gündoğdu, and S. Moslem, "Interval-valued spherical fuzzy analytic hierarchy process method to evaluate public transportation development," Informatica, vol. 32, no. 4, pp. 661-686, 2021. https://doi.org/10.15388/21-INFOR451

[4] M. Cipriani, L. S. Goldberg, and G. L. Spada, "Financial sanctions, SWIFT, and the architecture of the international payment system," Journal of Economic Perspectives, vol. 37, no. 1, pp. 31-52, 2023. https://doi.org/10.1257/jep.37.1.31

[5] Y. Han and T. Wang, "Semi-supervised clustering for financial risk analysis," Neural Processing Letters, vol. 53, no. 5, pp. 3561-3572, 2021. https://doi.org/10.1007/s11063-021-10564-0

[6] G. Mehdi, H. Hooman, Y. Liu, S. Peyman, R. Arif, J. Ahmad, R. Hamidreza, and A. Rashid, "Data mining techniques for web mining: a survey" Artificial Intelligence and Applications, vol. 1, no. 1, pp. 3-10, 2022. https://doi.org/10.47852/bonviewAIA2202290

[7] K. Bhosle and V. Musande, "Evaluation of deep learning CNN model for recognition of devanagari digit," Artificial Intelligence and Applications, vol. 1, no. 2, pp. 114-118, 2023. https://doi.org/10.47852/bonviewAIA3202441

[8] U. Hewage, R. Sinha, and M. A. Naeem, "Privacy-preserving data (stream) mining techniques and their impact on data mining precision: a systematic literature review," Artificial Intelligence Review, vol. 56, no. 9, pp. 10427-10464, 2023. https://doi.org/10.1007/s10462-023-10425-3

[9] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "An improved anomaly detection model for IoT security using decision tree and gradient boosting," The Journal of Supercomputing, vol. 79, no. 3, pp. 3392-3411, 2023. https://doi.org/10.1007/s11227-022-04783-y

[10] W. Lu, J. Li, J. Wang, amd L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," Neural Computing and Applications, vol. 33, no. 10, pp. 4741-4753, 2021. https://doi.org/10.1007/s00521-020-05532-z

[11] K. Peng, W. Jiang, P. Hou, Z. Wu, Z. Ling, X. Wang, Z. Niu, and D. Mao, "Continental-scale wetland mapping: a novel algorithm for detailed wetland types classification based on time series Sentinel-1/2 images," Ecological Indicators, vol. 148, no. 4, pp. 110113-110128, 2023. https://doi.org/10.1016/j.ecolind.2023.110113

[12] S. N. Mughal, Y. R. Sood, and R. K. Jarial, "A neural network-based time-series model for predicting global solar radiations," IETE Journal of Research, vol. 69, no. 6, pp. 3418-3430, 2023. https://doi.org/10.1080/03772063.2021.1934576

[13] S. Garai, R. K. Paul, M. Yeasin, and A. K. Paul, "CEEMDAN-based hybrid machine learning models for time series forecasting using MARS algorithm and PSO-optimization," Neural Processing Letters,

vol. 56, no. 2, pp. 92-115, 2024. https://doi.org/10.1007/s11063-024-11552-w

[14] H. Li, "Time works well: dynamic time warping based on time weighting for time series data mining," Information Sciences, vol. 547, no. 2, pp. 592-608, 2021. https://doi.org/10.1016/j.ins.2020.08.089

[15] C. W. Tan, C. Bergmeir, F. Petitjean, and G. I. Webb, "Time series extrinsic regression: predicting numeric values from time series data," Data Mining and Knowledge Discovery, vol. 35, no. 3, pp. 1032-1060, 2021. https://doi.org/10.1007/s10618-021-00745-9

[16] A. Shahid, T. A. N. Nguyen, and M. T. Kechadi, "Big data warehouse for healthcare-sensitive data applications," Sensors, vol. 21, no. 7, pp. 2353-2380, 2021. https://doi.org/10.3390/s21072353

[17] D. Amo, P. Gómez, L. Hernández-Ibáñez, and D. Fonseca, "Educational warehouse: modular, private and secure cloudable architecture system for educational data storage, analysis and access," Applied Sciences, vol. 11, no. 2, pp. 806-823, 2021. https://doi.org/10.3390/app11020806

[18] M. G. Khan, N. U. Huda, and U. K. U. Zaman, "Smart warehouse management system: architecture, real-time implementation and prototype design," Machines, vol. 10, no. 2, pp. 150-170, 2022. https://doi.org/10.3390/machines10020150

[19] M. G. Kahn, J. Y. Mui, M. J. Ames, A. K. Yamsani, N. Pozdeyev, N. Rafaels, and I. M. Brooks, "Migrating a research data warehouse to a public cloud: challenges and opportunities," Journal of the American Medical Informatics Association, vol. 29, no. 4, pp. 592-600, 2022. https://doi.org/10.1093/jamia/ocab278

[20] A. Tufano, R. Accorsi, and R. Manzini, "A machine learning approach for predictive warehouse design," The International Journal of Advanced Manufacturing Technology, vol. 119, no. 3, pp. 2369-2392, 2022. https://doi.org/10.1007/s00170-021-08035-w.

[21] R. P. Masini, M. C. Medeiros, and E. F. Mendes, "Machine learning advances for time series forecasting," Journal of Economic Surveys, vol. 37,no. 1, pp. 76-111, 2023. https://doi.org/10.1111/joes.12429

[22] T. Wen, H. Chen, and K. H. Cheong, "Visibility graph for time series prediction and image classification: a review," Nonlinear Dynamics, vol. 110, no. 4, pp. 2979-2999, 2022. https://doi.org/10.1016/j.jobe.2021.102537

[23] Y. Gao, K. M. Mosalam, Y. Chen, W. Wang, and Y. Chen, "Auto-regressive integrated moving-average machine learning for damage identification of steel frames," Applied Sciences, vol. 11, no. 13, pp. 6084-6110, 2021. https://doi.org/10.3390/app11136084

[24] L. You and M. Zhu, "Digital twin simulation for deep learning framework for predicting solar energy market load in Trade-By-Trade data," Solar Energy, vol. 250, no. 1, pp. 388-397, 2023. https://doi.org/10.1016/j.solener.2023.01.004

[25] W. Grzesiak, D. Zaborski, I. Szatkowska, and K. Królaczyk, "Lactation milk yield prediction in primiparous cows on a farm using the seasonal auto-regressive integrated moving average model, nonlinear autoregressive exogenous artificial neural networks and Wood's model," Animal Bioscience, vol. 34, no. 4, pp. 770-782, 2021. https://doi.org/10.5713/ajas.19.0939

[26] P. M. Almeida-Junior and A. D. C. Nascimento, "ARMA process for speckled data," Journal of Statistical Computation and Simulation, vol. 91, no. 15, pp. 3125-3153, 2021. https://doi.org/10.1080/00949655.2021.1922688

[27] B. Ma, P. Li, X. Guo, H. Zhao, and Y. Chen, "A novel online prediction method for vehicle velocity and road gradient based on a flexible-structure auto-regressive integrated moving average model," Sustainability, vol. 15, no. 21, pp. 15639-15656, 2023. https://doi.org/10.3390/su152115639

[28] B. Rele, C. Hogan, S. Kandanaarachchi, and C. Leigh, "Short-term prediction of stream turbidity using surrogate data and a meta-model approach: A case study," Hydrological Processes, vol. 37, no. 4, pp. 14857-14869, 2023. https://doi.org/10.1002/hyp.14857

[29] B. Yang, X. Liu, W. Long, and L. Peng, "A unified unit root test regardless of intercept," Econometric Reviews, vol. 42, no. 6, pp. 540-555, 2023. https://doi.org/10.1080/07474938.2023.2217077

[30] S. Erdogan and S. A. Solarin, "Stochastic convergence in carbon emissions based on a new fourier-based wavelet unit root test," Environmental Science and Pollution Research, vol. 28, no. 17, pp. 21887-21899, 2021. https://doi.org/10.1007/s11356-020-12033-y

[31] K. Yamazaki, V. K. Vo-Ho, D. Bulsara, and N. Le. "Spiking neural networks and their applications: a review," Brain Sciences, vol. 12, no. 7, pp. 863-892, 2022. https://doi.org/10.3390/brainsci12070863

[32] K. M. Hamdia, X. Zhuang, and T. Rabczuk, "An efficient optimization approach for designing machine learning models based on genetic algorithm," Neural Computing and Applications, vol. 33, no. 6, pp. 1923-1933, 2021. https://doi.org/10.1007/s00521-020-05035-x

[33] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," Big Data, vol. 9, no. 1, pp. 3-21, 2021. https://doi.org/10.1089/big.2020.0159

CNN-Based Multi-Output and Multi-Task Regression for Supershape Reconstruction from 3D Point Clouds

Hassnae Remmach1[1], Siti Fatimah Abdul Razak*[2], Arif Ullah[3], Sumendra Yogarayan[4] Md Shohel Sayeed[5,] Amine Mrhari[6]
[1]Computer Systems Engineering Laboratory, Cadi Ayyad University, 40000, Marrakesh, Morocco
[2345]Faculty of Information Science and Technology, Multimedia University, Ayer Keroh 75450 Malaysia
[6]Research in Computer Science Laboratory Faculty of Sciences, Ibn Tofail University Kenitra, Morocco
E-mail: fatimah.razak@mmu.edu.my, arifullahms88@gmail.com
* Corresponding Author

Keywords: 3D reconstruction, CNNs, multi-output regressor, multi-target regressor, 3D point cloud, super shapes, point net

Received: August 5, 2024

In a number of industries, including computer graphics, robotics, and medical imaging, three-dimensional reconstruction is essential. In this research, a CNN-based Multi-output and Multi-Task Regressor with deep learning capabilities is proposed for three-dimensional object reconstruction from 3D point cloud. Our approach is grounded in the original Point Net architecture, which addresses the difficulties associated with convolution when applied to point clouds. Firstly, this paper is modified using a Multi-Output Regressor to accurately recreate Super forms from 3D point clouds. Using this method, we first extract features from the 3D point cloud using Point Net. After that, a Multi-Output Regressor receives these data and uses them to anticipate the Super shape parameters needed to reconstruct the shape. Taking in the data, the Multi-Output Regressor retrieved characteristics from Point Net and simultaneously predicts several outcomes. Second, a Multi-Task Regressor is used to modify the Point Net. The network gains from the capacity to transfer knowledge from one task to another, improving the model's overall performance. The model would forecast the ten parameters needed to create the shape in the case of rebuilding Super shapes. The test findings were better than expected; they are intriguing in terms of prediction accuracy and cost, and they update the result by 80%, which is a good accomplishment for the study.

Povzetek: Raziskava predlaga večizhodni in večopravilni regresor na osnovi CNN za rekonstrukcijo 3D oblike iz oblaka točk.

1 Introduction

1.1 Importance and challenges of three-dimensional reconstruction

Three-dimensional reconstruction is an essential task in computer vision, as it enables us to create accurate 3D models of objects and scenes from different data. These 3D models can be used for a wide range of applications, including object recognition, tracking, localization, and segmentation. For instance, in object recognition, 3D models can have used to create a database of 3D models of objects, allowing a computer vision system to recognize objects based on their 3D shape, rather than just 2D appearance [1]. Additionally, in tracking, 3D models can be used to estimate the position and orientation of an object in 3D space, enabling the system to track the object even when it is partially occluded or when the viewpoint changes. Overall, 3D reconstruction plays a crucial role in advancing computer vision applications, and it has the potential to enhance the accuracy and efficiency of various computer vision systems. However, 3D reconstruction is a complex and challenging task and it requires accurate estimation of the camera parameters, feature detection, matching, and triangulation. These challenges increase when dealing with noisy, incomplete, or low-quality data [2].

1.2 Evolution of techniques

To overcome these challenges, researchers have proposed various techniques to improve the accuracy and efficiency of three-dimensional reconstruction [3]. Recently, deep learning approaches have gained significant attention in the field of three-dimensional reconstruction. Artificial Intelligence (AI) algorithms such as Convolutional Neural Networks (CNNs) [4], Generative Adversarial Networks (GANs) [5], and Recurrent Neural Networks (RNNs) [6] have shown promising results in solving different problems in three-dimensional reconstruction, such as feature detection, depth estimation, and shape completion. These techniques have the potential to revolutionize the way we reconstruct 3D models. However, there are still many challenges that need to be addressed, such as data acquisition, training data availability, and computational efficiency. Therefore, it is essential to do more research in the field of three-dimensional reconstruction, especially using deep learning approaches. More efforts are needed to develop new algorithms, improve the accuracy and efficiency of existing techniques, and explore new

applications of three-dimensional reconstruction. By doing so, we can overcome the current limitations and challenges and unlock the full potential of three-dimensional reconstruction in various fields [7].

1.3 Proposal of research contribution

Our research is aligned with this overarching theme. We focus on modeling 3D objects using a parametrized surface, specifically Super shapes [8]. These Supershapes are versatile geometric shapes capable of describing a wide array of complex objects. However, accurately estimating the parameters of Supershapes poses a significant challenge. Traditional methods often rely on manual parameter tuning and handcrafted features, which are labor-intensive and may not yield precise results. To address these limitations, we propose a novel three-dimensional reconstruction approach leveraging Convolutional Neural Networks (CNNs). Our method utilizes a CNN-based Regressor network that takes 3D point clouds as input and accurately estimates Supershape parameters. In this architecture, convolutional and pooling layers are stacked multiple times before fully connected layers. The pooling layers reduce the spatial dimensionality of features, while the convolutional layer's extract features from the input data. These learned features are then mapped to output values through the fully connected layers. Furthermore, we leverage the Point Net [9] network as the foundation for our architecture. While PointNet has demonstrated effectiveness in processing 3D point clouds, its original design may not be directly applicable to the task of 3D reconstruction. PointNet is primarily designed for point cloud classification and segmentation, lacking specific capabilities for regression tasks required for parameter estimation in 3D shapes or objects. To address this limitation, we adapt the PointNet network to learn the correlation between input point clouds and output parameters. This involves modifying the network architecture and training process to incorporate regression-based objectives. By adjusting PointNet to learn this correlation, it can effectively serve as a CNN-based Regressor network for 3D reconstruction tasks. This modification enables accurate and efficient estimation of parameters for 3D shapes and objects, significantly improving upon traditional 3D reconstruction techniques [10-12].

1.4 Outline of the paper

This paper is divided into four main sections.

The first section provides an overview of deep learning techniques for 3D reconstruction and object modeling, focusing on recent advancements and the impact of Convolutional Neural Networks (CNNs). Various approaches, including volumetric and point cloud-based methods, are discussed. The second section details our proposed methodology, emphasizing the architecture and training process of the CNN-based Regressor network. We highlight how our approach addresses limitations of traditional techniques, enhancing accuracy and efficiency in estimating Supershape parameters. Moving to the third part, we present experimental results, evaluating our approach through metrics like accuracy and precision and comparing them with prior work. Finally, the conclusion summarizes our contributions, acknowledges significance, outlines potential and limitations, and suggests areas for future improvement.

2 Literature review

Deep learning techniques have revolutionized the field of 3D reconstruction, enabling more accurate and efficient reconstruction of complex objects and scenes. The evolution of deep learning techniques for 3D reconstruction can be traced back to the use of CNNs for 3D shape classification and segmentation. CNNs have shown promising results in processing 3D shapes by considering them as volumetric representations, and have been used for many applications such as object detection, recognition, and pose estimation [13]. Recent advancements in deep learning have yielded various specialized architectures for 3D reconstruction, tailored to different applications and data types. One such example is Point Net [14], designed specifically for processing point cloud data. Unlike traditional methods relying on structured data like meshes or voxels, PointNet directly processes unstructured point clouds, making it ideal for tasks like 3D reconstruction from LiDAR data. Its versatility lies in its capacity to handle point clouds of varying density, distribution, and orientation. Notably, PointNet has demonstrated effectiveness in processing large-scale point cloud datasets, facilitating the reconstruction of intricate scenes and objects. Its ability to accommodate inputs of different sizes and extract both local and global geometric features has rendered it popular, particularly in fields like autonomous driving and robotics, significantly enhancing the accuracy and efficiency of 3D reconstruction tasks. VoxNet, as described in [15], is a specialized deep learning model tailored for 3D object detection and recognition. By processing 3D voxel grids with 3D convolutions, it excels in detecting objects, even in occluded or complex scenes. Its flexibility to handle various voxel grid sizes and resolutions makes it versatile for different applications, particularly in autonomous driving and robotics, where accuracy is crucial. OctNet presented in [16], is a deep learning framework designed for processing 3D data using an octree structure, enhancing the management of complex 3D scenes. Its hierarchical approach allows for efficient processing, with higher-resolution octree cells corresponding to smaller regions of the scene. This methodology has proven effective for tasks like object detection, segmentation, and 3D reconstruction from sparse data. OctNet is applied in various fields including robotics, virtual reality, and medical imaging, owing to its adaptability to variable-resolution inputs. Additionally, it demonstrates high accuracy in detecting, segmenting, and reconstructing objects from sparse data, solidifying its role in computer vision applications. 3D U-Net, described in [17], is a tailored version of the U-Net architecture crafted for processing 3D volumetric data, notably in medical imaging like MRI and CT scans. It utilizes an encoder-decoder structure with skip connections to effectively

process volumetric data. The encoder down samples input data to capture high-level features, while the decoder up samples to generate output segmentation or reconstruction. Notably, 3D U-Net accommodates variable-sized inputs, rendering it apt for large medical image datasets. MVCNN, described in [18], processes multiple 2D images to create 3D reconstructions. Using a CNN, it extracts features from each image, aggregates them, and generates the final reconstruction. It is versatile for object recognition and reconstruction in robotics and AR, handling varying input images effectively. MVCNN is highly accurate and widely preferred in computer vision applications. Recent advancements in deep learning have introduced Graph Convolutional Networks (GCNs) and Generative Adversarial Networks (GANs) as potent tools for 3D reconstruction tasks. GCNs, detailed in [19], specialize in processing intricate 3D structures like molecular configurations, exhibiting promising outcomes in constructing lifelike 3D models from input point clouds. Meanwhile, GANs, outlined in [20] excel in synthesizing realistic 3D models from either 2D images or partial 3D data. Their potential to transform the 3D reconstruction domain lies in their ability to generate high-quality 3D models even from limited or incomplete datasets. In addition to these advancements, the integration of deep learning techniques [21] with traditional 3D reconstruction techniques such as stereo vision, Structure from Motion (SfM), and Multi-View Stereo (MVS) has led to significant improvements in the accuracy and efficiency of 3D reconstruction. For example, deep learning techniques have been used for refining and post-processing 3D reconstructions generated by SfM and MVS techniques, and for integrating depth and color information for more accurate 3D reconstruction. Overall, the recent advancements in deep learning techniques have significantly improved the accuracy and efficiency of 3D reconstruction and have the potential to revolutionize the field of 3D reconstruction by enabling the reconstruction of complex and diverse 3D structures from limited or incomplete data [22]. Table 1 present the summary of related work.

Table 1: Summary of related work

Technique name	Data set	Result	Future work
Deep learning	LiDAR data	78%	Need improvement in result
PointNet	LiDAR data	74%	3D reconstruction tasks
CNN	synthetic 3D shapes	77%	3D reconstruction tasks
U-Net	synthetic 3D shapes	79%	Need improvement in result
SfM, and MVS	LiDAR data	75%	Need improvement in result

3 Proposed methodology

This section provides a detailed presentation of the proposed methodological approach for three-dimensional object reconstruction. It explores the fundamental techniques used in our methodology, including modeling objects in 3D using Supershapes, the PointNet architecture adapted to this specific task, and multiple regression by neural networks.

3.1 Maintaining a 3D Object by supershapes

SuperShapes, introduced in [23], extend the capabilities of traditional geometric shapes by offering a parametric approach to modeling complex forms. By adjusting parameters such as symmetry (m, M), scaling (a, b), and shape coefficients ($n_1, n_2, n_3, N_1, N_2, N_3$), a diverse range of shapes can be generated, allowing for the representation of complex objects and surfaces in three-dimensional space. The Supershape formula, see Eq. (1), encapsulates these parameters, enabling the creation of customized geometries tailored to specific applications.

$$S = [a, b, m, M, n_1, n_2, n_3, N_1, N_2, N_3] \quad (1)$$

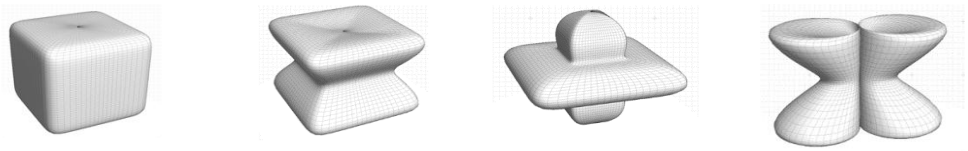
m: Symmetry in the radial direction

M: Symmetry in the tangential direction

a, b: The scaling parameters that control the overall size of the shape.

$n_1, n_2, n_3, N_1, N_2, N_3$: The shape parameters coefficients [24].

By manipulating these parameters (as shown in Figure 1 and Table2), it is possible to create complex and interesting shapes that cannot be easily generated using traditional geometric shapes alone.



Shape (1) Shape (2) Shape (3) Shape (4)

Figure 1: Variety of SuperShapes

Table 2: Super shapes parameters

	m	M	n_1	n_2	n_3	N_1	N_2	N_3
Shape (1)	4	4	10	10	10	10	10	10
Shape (2)	4	4	10	10	10	1.10	10	1.70
Shape (3)	4	4	10	10	10	-2	10	4.20
Shape (4)	4	4	10	-10	10	2	25	3

3.2 The PointNet architecture

PointNet is a deep learning architecture for processing 3D point clouds. It was introduced in 2017 [25] and has since become one of the most widely used architectures for point cloud analysis tasks such as classification, segmentation, and object detection. The Point Net architecture [26] takes as input a set of unordered point clouds and learns a set of features that can be used to classify or segment the objects in the scene. The key innovation of PointNet is that it directly operates on the point clouds without the need for pre-processing or feature engineering, which has traditionally been a time-consuming and error-prone process. The original PointNet architecture consists of several layers of fully connected neural networks, which are used to learn features from the input point clouds. The architecture also includes a max-pooling layer, which is used to aggregate information from all the points in the cloud and generate a global feature vector that can be used for classification or segmentation. The network is trained using a loss function that penalizes misclassifications or segmentation errors [27].

3.3 Multiple regression by neural networks

Multiple regression by neural networks [28] involves using neural network algorithms to perform regression analysis on multiple independent variables simultaneously. These networks are capable of modeling complex non-linear relationships between inputs and outputs, and can handle large amounts of data. By leveraging the inherent ability of neural networks to model non-linear relationships between inputs and outputs, multiple regression techniques can capture intricate patterns in the data and accurately estimate parameters like shape, orientation, or texture of reconstructed objects. In our proposed architecture, we will utilize multi-output regression and multi-task regression, techniques that will allow us to predict multiple output variables simultaneously related to our 3D shape models. These methods will be further discussed later to illustrate their role in our 3D reconstruction approach [29].

3.4 Multi-output regression (MOR)

Multi-output regression, as described in [30], aims to predict multiple output variables simultaneously, contrasting with simpler regression methods that focus on a single target variable (see Figure 2). This approach models the relationships between input variables and each target variable separately, offering enhanced flexibility and accuracy in predicting multiple outcomes. Various machine learning techniques, including neural networks, support vector machines, and random forests, can be employed for multi-output regression. While this approach presents challenges due to its complexity, it has the potential to yield more precise and valuable predictions compared to traditional single-target regression methods. The advantages of multi-output learning include a better understanding of the relationships between target variables, simplification of the modeling process, and improvement in generalization, especially when outputs exhibit mutual dependencies. However, it is crucial to carefully consider the model architecture and select target variables judiciously for a successful implementation of multi-output learning [31].

3.5 Multi-Task regression (MTR)

In machine learning, multi-task learning involves training a model to perform multiple related tasks simultaneously (see Figure 3), aiming to enhance overall performance by leveraging shared knowledge across tasks. Typically, the model incorporates shared hidden layers, capturing common patterns across tasks, while task-specific layers focus on unique task patterns. Determining the architecture, layer sizes, and activation functions involves considering task complexity and available data, often requiring experimentation. Regularization techniques like batch normalization or dropout can help prevent overfitting and improve generalization [32].

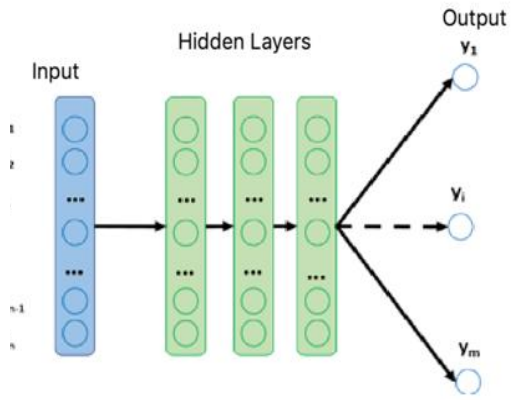


Figure 2: Multi output regression

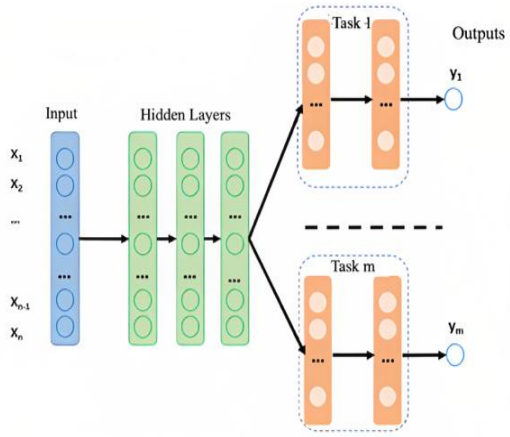


Figure 3: Multi-task regression

Multi-task learning offers benefits such as improved generalization, efficient resource utilization, and the ability to learn from limited labeled data. By learning multiple tasks simultaneously, models can build robust representations and handle data variations effectively. However, successful implementation requires careful consideration of model architecture and task selection [22].

3.6 The proposed approach

In this article, we introduce a new approach able to retrieve individual Supershape model parameters from point cloud by deep learning. More precisely, we formulate the SuperShapes recovery problem as a prediction task, where the goal is to estimate the parameters of a given surface model:

$$\hat{Y} = [a,b,m,M,\boldsymbol{n_1}, \boldsymbol{n_2}, \boldsymbol{n_3}, \boldsymbol{N_1}, \boldsymbol{N_2}, \boldsymbol{N_3}] \in R1\times1O \quad (2)$$

This estimation is achieved using a Deep Learning network defined by:

$$\hat{y}=f(x, W) \quad (3)$$

Here, f represents a predictor that learns from annotated training data. W defines the set of network weights that have to be learned during training. To build our regression model, we design a network that conforms to the architectural guidelines of the Point Net network [34]. Initially, we retain the first section of PointNet for feature extraction, utilizing reduced-size MLPs (32). The basic PointNet section (cf. Fig.4) is a neural network architecture commonly employed for point cloud processing tasks. It typically comprises a "tnet" sub-network, which is a transformation MLP with input and output sizes identical. The purpose of the "tnet" is to learn symmetric functions and apply 1D convolutional layers (CNN-1D) that extract global features post Max-pooling. This process allows the network to glean useful information from the input point cloud data [35].

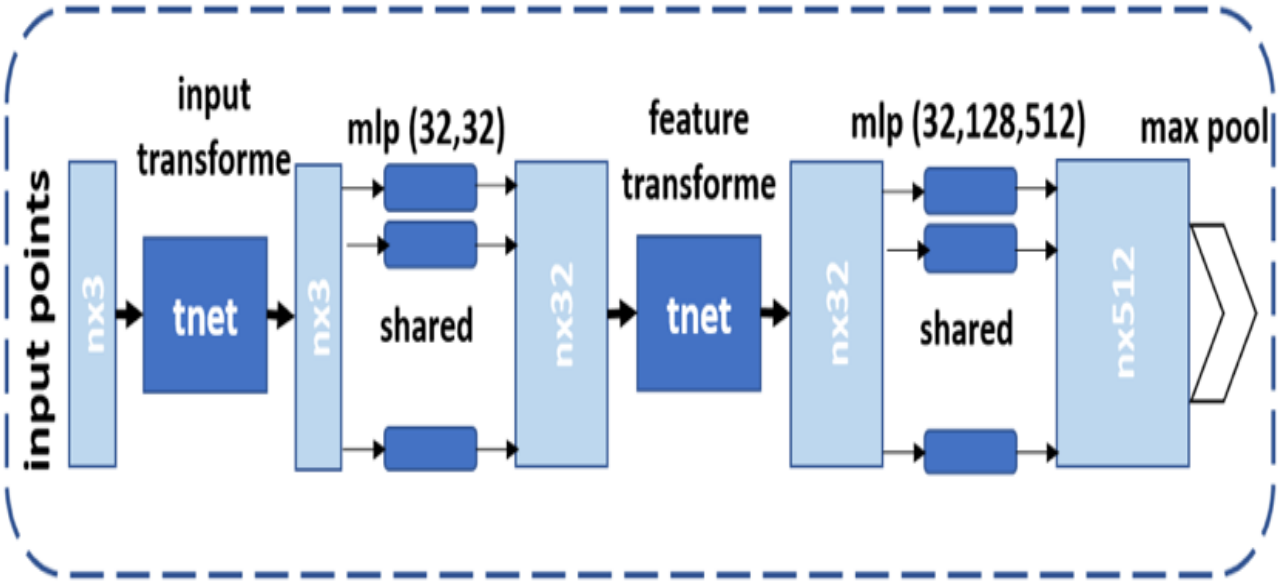


Figure 4: Basic Point Net Section (MLP)

To further extend the capabilities of the basic PointNet, we propose a 5-layer MLP architecture. The input layer of our MLP takes the global features of the object as its input. The three hidden layers of the MLP consist of 256, 128 and 64 bits in size, respectively, and are intended to extract and learn intricate features from the input data. In the initial design, we introduce multi-output regression to Point Net's base section, while in the second,

3.6.1 Multi-output regression (MOR)

The first proposed architecture is based on the base section of PointNet and augmented by multi-output regression as shown in figure 5. The multiple output layer of the MLP calculates the surface parameters using a simple linear regression. This enables the network to predict the surface parameters of the input object with a high degree of accuracy [36].

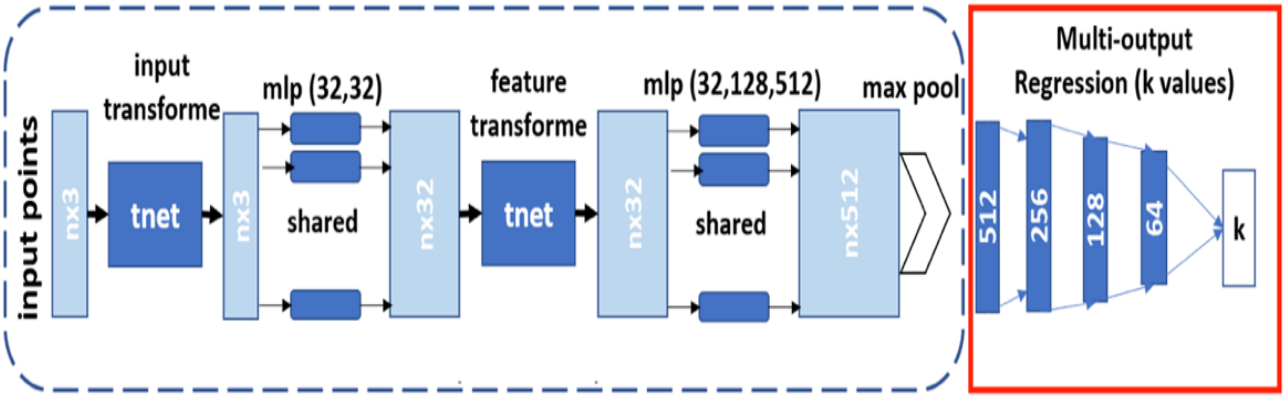


Figure 5: Modified CNN-based multi-output regressor

Our extension consists of a 5-layer MLP. The input layer is composed of the global features of the object. Then we have three hidden layers with respective sizes of 256, 128, and 64. Finally, a multiple output layer calculating surface parameters using a simple linear regression.

3.6.2 Multi-Task regression (MTR)

The second proposed architecture is based on the base section of PointNet and augmented by multi-task regression as shown in the figure 6. In this architecture, the regression part of the new CNN-based Regressor architecture consists of a hidden layer (128) and 10 tasks. A hidden layer (64) and an individualized output (one neuron) represent each task for each of the SuperShapes parameters, indicated in equation (1).

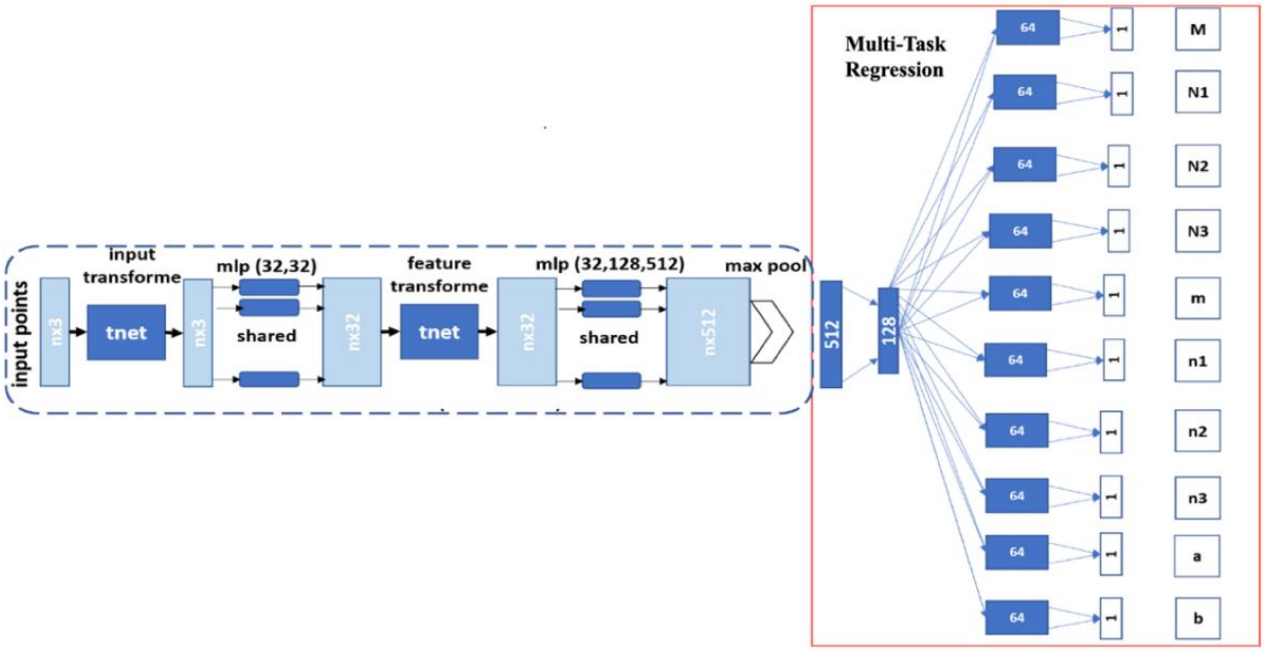


Figure 6: Modified CNN-based multi-task regressor

We utilize a distance metric based on the L2 norm between the pre-defined parameters y and the anticipated SuperShapes parameters ŷ to assess the effectiveness of our Regressor model. Our Regressor uses the pre-defined parameters y as its training objective, and the L2 norm gives us a measure of the discrepancy between the target and anticipated values of the Supershape parameters. The goal function is as follows:

$$\text{L2}\ (y, \hat{y}) = \|y - \hat{y}\|^2 \quad (4)$$

By minimizing the L2 distance between the predicted and target Supershape parameters during the training process, in order to minimize the error in our predictions, we make sure that our model learns to reliably forecast the target parameters. With this method, we can measure our model's accuracy and make sure it can adapt well to new, untested data. The goal is to get as near to y as feasible with an estimation ŷ = f(x, W), where f is the proposed architecture model with weights W for the input x (the 3D object). Our Modified CNN-based Multi-Output and Multi-Task Regressor Network model must be trained on a sizable dataset of 3D objects represented as point clouds in order to produce an accurate prediction of the supershape parameters for a given 3D item, x. Using its learnt weights W, the model uses the 3D object x as input and predicts the supershape parameters ŷ. An enhanced version of mini-batch SGD, the stochastic gradient descent optimization algorithm ADAM minibatch, is used to train the model. By iteratively changing the model's weights to enhance predictions, this approach effectively minimizes the loss function on the accessible training dataset. A series of input-output pairs (x,y) are given to the model during training, where y is the appropriate pre-defined set of supershape parameters and x is a 3D object represented as a point cloud. The model predicts the supershape parameters ŷ using the input x, and then computes the loss between ŷ and y. Based on the L2 distance measure. Next, in order to minimize the loss, the ADAM minibatch method adjusts the model's weights W. This procedure is done numerous times until the model converges to an ideal set of weights that yield precise predictions of the supershape parameters for brand-new, unobserved 3D objects [37].

4 Experimental results

4.1 Dataset

To train and evaluate our CNN-based Regressor, we generate a Dataset of synthetic 3D shapes by varying the parameters pseudo-randomly by a specific algorithm that takes care of Generation of 3D objects at (point cloud) using pseudo random supershape parameters with bounds on parameter values and number of shapes. Resampling (reduction of the number of points to 512 per object) then arrangement in Dataset: 70% for training (9300 objects) and the rest for the test (2300 objects). Form of data used for learning: the coordinates of its points in the ply represent an object. Format, it is an ASCII file including all the coordinates (x,y,z) for example: This is an illustration of a produced object with 512 points under various viewpoints figure .7 Point Cloud Objects from the dataset.

Figure 7: Point Cloud Objects from the dataset in different views

4.1.1 Performance metrics

To evaluate the performance of our CNN model in predicting the Supershape parameters, we use the Mean Squared Error (MSE) for each of the parameter's p ∈ {m, n_1, n_2, n_3,M, N_1, N_2, N_3,a,b} during experimentation.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (5)$$

The MSE is a commonly used metric to measure the accuracy of regression models, such as our CNN-based Regressor. For a given set of test data, it calculates the average squared difference between the true and anticipated values of the supershape parameters. The mean squared error (MSE) for each parameter p is computed by squaring the discrepancies between the true and projected values for every test sample. When evaluating regression models in machine learning, a smaller mean square error (MSE) value denotes greater accuracy in predicting the supershape parameters. Using Mean Squared Error (MSE) as the evaluation metric, we can assess how well our CNN-based Regressor model predicts the supershape parameters and compare its performance to other regression models or variations of our own model. This enables us to continuously enhance the architecture and hyperactive parameters in order to improve performance in super shape parameter prediction [36]. The MLP network employs gradient descent or one of its variations for backpropagation when it comes to updating its weights. Different gradient descent variations exist, including mini-batch gradient descent, batch gradient descent, and stochastic gradient descent (SGD), which vary in how the gradient is calculated and how the training data is supplied to the algorithm [38].

4.1.2 Model preparation

Once the Datasets are in place, we present them to our CNN-based Regressor network for training cycles by augmenting the objects with affine transformations. Data augmentation is important when working with point cloud data. An augmentation function for shaking and shuffling the training dataset has been provided by PointNet realizations.

4.2 Training and testing

In the first set of experiments, we evaluate the performance of the proposed architectures using the 2300 test objects generated from our synthetic dataset.

4.2.1 Multi-output regression (MOR)

To illustrate the results of the network's training and testing,

we present two graphs related to the testing phase: Evolution Loss (cf. Fig8) and Accuracy Evolution (cf. Fig10). And a graph of Accuracy Evolution (see Fig9) in both phases: training and testing.

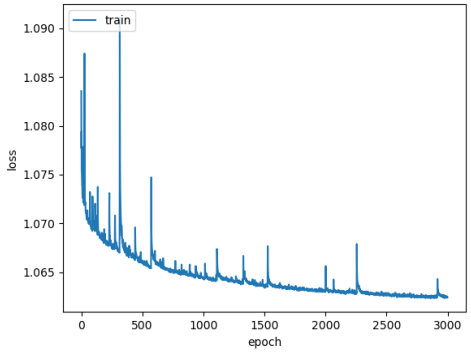


Figure 8: Evolution of Loss

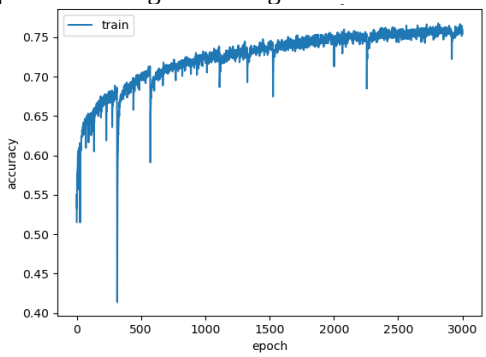


Figure 9: Function accuracy evolution

The "Evolution of Loss" graph (cf. Fig.8) typically displays how the loss function changes over time or iterations during the training process. It provides insights into how well the model is converging towards minimizing the loss, indicating whether the model is learning effectively. The graph depicts the evolution of error across iterations. Initially, the error starts at a high value, indicating significant divergence between the model's predictions and the actual values. There are notable fluctuations observed until around iteration 1000, suggesting instability in the model is learning process. However, from iteration 1000 onward, the error gradually decreases, eventually reaching a minimum value of less than 1.06 at iteration 3000. This indicates that the model has stabilized and converged to predictions that are more accurate by the end of the training process. On the other hand, the "Function Accuracy Evolution" graph (cf. Fig.9) illustrates the accuracy of the model's predictions over time or iterations. It helps in understanding how the model's performance in making accurate predictions evolves throughout the training process. The graph illustrates the evolution of precision across iterations. Initially, the precision starts at a low value, indicating poor performance of the model in accurately classifying or predicting outcomes. From around iteration 1000, there are noticeable fluctuations in precision, suggesting variability in the model's performance during training. However, as the training progresses, the precision gradually improves, reaching a maximum value of 0.768 at iteration 3000. This indicates that the model's ability to make correct predictions improved by the end of the training process. The last graph (cf. Fig10) illustrates the evolution of precision during training and tests. It visualizes how the model's precision evolves throughout the training process, providing insights into its learning and generalization abilities to new data. Comparing precision on training and testing data also helps detect any overfitting or under fitting of the model. In summary, such a graph offers a comprehensive evaluation of the model's performance and aids in adjusting hyper parameters or selecting the best model for the given task.

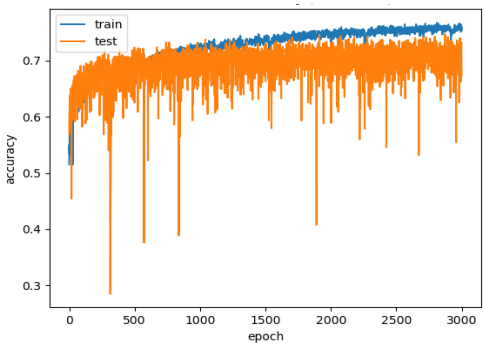


Figure 10: Evolution accuracy (train/test)

We have here a precision of 0.768 for training and 0.74 for testing, which suggests that the model performs slightly better on training data than on testing data. This indicates slight overfitting, where the model has learned to fit the training data well but does not generalize as effectively to new data. Therefore, it is important to examine, in our future work, other performance metrics and explore regularization methods or other techniques to improve the model's generalization.

4.2.2 Multi-task regression (MTR)

During training, the error (cf. Fig11) quickly reaches a reasonable value, with minimal fluctuations. Around iteration 300, it begins to stabilize, and this stabilization becomes more pronounced by iteration 1250. Eventually, it reaches a value of 1.06 by iteration 2000. Regarding the evolution of accuracy during training (cf. Fig12-13), it takes longer to stabilize at the value of 0.802, reaching it by iteration 2000.

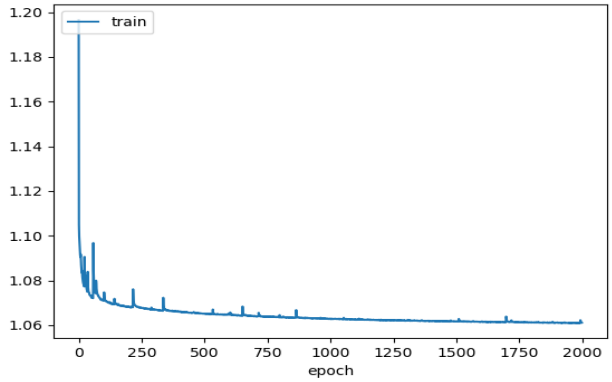


Figure 11: Evolution of loss

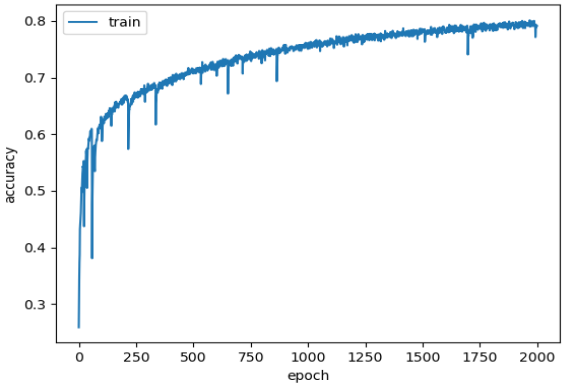


Figure 12: Function accuracy evolution

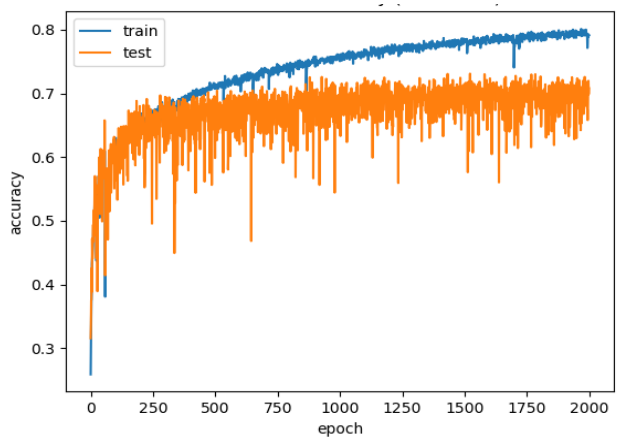


Figure 13: Training and testing data set

The following figure illustrates the evolution of precision training (0.802) and testing (0.728):

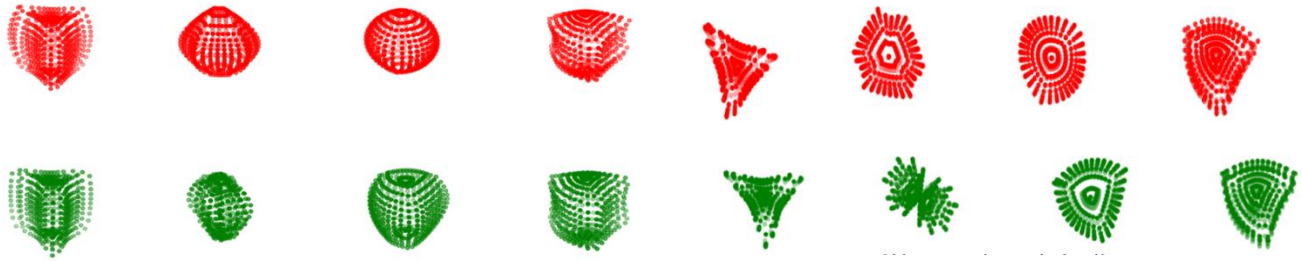
4.3 The result on test objects

The object provided as input is a point cloud, and as output, we have the prediction of the 10 parameters of this object for reconstruction purposes. The results on 4 objects taken at random from the set of tests are presented in the following table: To assess the accuracy, effectiveness, and overall performance of the suggested system, a series of carefully chosen test objects were used in a rigorous evaluation process. The purpose of selecting these test objects was to ensure that the model was evaluated in situations that closely resembled the dataset by representing a wide range of scenarios. Table 3 present Predicted parameters results relating to four different objects.

Table 3: Predicted parameters results relating to four different objects

.		m	n_1	n_2	n_3	M	N_1	N_2	N_3	a	b
Obj. 1	Input	3	30	69	81	3	63	78	76	1	1
	Output	2.894	22.245	66.286	74.180	3.145	68.545	100.991	81.670	1	1
Obj. 2	Input	6	72	98	23	2	44	102	74	1	1
	Output	-29.366	487.056	-414.686	-228.655	27.350	-672.217	487.369	402.352	1.001	1.001
Obj. 3	Input	2	66	89	48	2	74	79	93	1	1
	Output	3.082	66.838	94.065	44.577	2.862	63.356	67.228	94.921	1	1
Obj. 4	Input	3	30	69	25	5	59	90	25	1	1
	Output	3.117	26.970	63.934	23.485	4.022	57.457	98.925	24.430	1	1

The figure 14 illustrates the reconstruction of the 3D objects taken in the test part of the Dataset after training under two Different views. Please note that the 4 examples in red are taken at random and represent ground truth, and the green ones represent the reconstructed objects by predictions.



Figurer 14: Visual results of 4 objects under two views

4.4 Comparative analysis

The provided data in the following table 3 presents the results of applying both a Multi-Output Regressor and a Multi-Task Regressor for 3D reconstruction from point clouds. In the Multi-Output Regressor, after 3000 iterations, the training performance is reported at 76.8, while the testing performance is slightly lower at 74.0. On the other hand, for the Multi-Task Regressor, after 2000 iterations, the training performance is higher at 80.2, but the testing performance is slightly lower at 72.8. Table 4 present the overall result comparison.

Table 4: Comparison result

Accurac	**Multi-Output Regressor (3000 iterations)**	**Multi-Task Regressor (2000 iterations)**
(%) Train	76.8	80.2
(%) Test	74.0	72.8

These findings imply that the Multi-Task Regressor fits the training data better than the Multi-Output Regressor, as seen by the latter is lower accuracy on the training dataset. In contrast to the Multi-Task Regressor, the Multi-Output Regressor has a marginally superior generalization performance with a smaller accuracy decline when assessed on the test dataset. This would suggest that overfitting to the training set is avoided by the Multi-Output Repressor's increased robustness in handling unknown data. It is important to remember that the exact objectives of the 3D reconstruction work will determine which of these two approaches is best. In cases when strong generalization to fresh data is essential, the Multi-Output Regressor may be recommended. Nonetheless, the Multi-Task Regressor should be used if getting great accuracy on the training dataset is needed. Despite a minor decline in generalization performance, could be a better option. The trade-off between model complexity and generalization ability is generally highlighted by these results, highlighting the significance of considering the application's particular requirements when selecting a regression strategy for 3D reconstruction. The problem of overfitting, in which a model performs very well on training data but poorly on fresh, unseen (testing) data, is the main cause of the trade-offs between training and testing performance. Overfitting happens when a model is too intricate and begins to learn noise or unimportant features from the training dataset instead of broadly applicable patterns. Overfitting can be identified using cross-validation and model complexity

5 Conclusion and future work

This article presents a novel method called "A Modified CNN-based Multi-Output and Multi-Task Regressor Network," which integrates two separate extensions—a Multi-Output Regressor and a Multi-Task Regressor—to expand PointNet's core design. The experimental findings provide intriguing new information about how well these extensions work. The Multi-Output Regressor exhibits a respectable testing accuracy of 74.0 and a strong training accuracy of 76.8. Conversely, the Multi-Task Regressor has a little lower testing accuracy of 72.8 but a higher training accuracy of 80.2. These results point to a trade-off between the two extensions, with the Multi-Output Regressor showing a stronger suitability for robust generalization to fresh data due to its better balance between testing and training accuracy. The Multi-Task Regressor, however, shines in attaining increased precision on the training dataset, highlighting its capacity to detect complex patterns in the training set. In order to improve the overall performance of the model, it would be beneficial to investigate hybrid architectures in the future that take advantage of the advantages of both the Multi-Output and

Multi-Task Regressor. To reduce overfitting and enhance generalization, further research into training iteration optimization or the use of sophisticated regularization strategies may be beneficial. Moreover, the utilization of our methodology in various real-world settings and datasets may confirm its efficacy in a wider range of 3D reconstruction assignments.

References

[1] Gajjar V. K. “Machine learning applications in plant identification, wireless channel estimation, and gain estimation for multi-user software-defined radio”. Missouri University of Science and Technology, 2022.

[2] Ahmed, E. et al. “A survey on deep learning advances on different 3D data representations”. arXiv preprint arXiv:1808.01462, 2018.

[3] Wicker, M., & Kwiatkowska, M. “Robustness of 3d deep learning in an adversarial setting”, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11767-11775), 2019.

[4] Hamida, A. B., Benoit, A., Lambert, P., & Amar, C. B. “3-D deep learning approach for remote sensing image classification”, IEEE Transactions on geoscience and remote sensing, 56(8), 4420-4434, 2018.

[5] Casadevall, G., Duran, C., & Osuna, S. “AlphaFold2 and deep learning for elucidating enzyme conformational flexibility and its application for design”, JACS Au, 3(6), 1554-1562, 2023.

[6] Bokhabrine, Y., Fougerolle, Y. D., Foufou, S., & Truchetet, F. “Genetic algorithms for Gielis surface recovery from 3D data sets”, in IEEE International Conference on Image Processing (Vol. 2, pp. II-549). IEE, September, 2007.

[7] Garcia-Garcia, A., Gomez-Donoso, et al. “Pointnet: A 3d convolutional neural network for real-time object class recognition”. In International joint conference on neural networks (IJCNN) (pp. 1578-1584). IEEE, July, 2016.

[8] H. Remmach et. al. “Swarm Optimization for Tridimensional point Cloud Reconstruction using Supershapes”, in Indian Journal of Computer Science and Engineering Vol 11 No X 1-5, 2019.

[9] O’Mahony et al. “Deep learning vs. traditional computer vision. In Advances in Computer Vision”, in Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 1 (pp. 128-144). Springer International Publishing, 2020.

[10] Fu S., Shi. Et al. “Field-dependent deep learning enables high-throughput whole-cell 3D super-resolution imaging”. Nature Methods, 20(3), 459-468, 2023.

[11] Chaoxu Guo et al. “Learning Multimodal 3D Object Detection and Semantic Segmentation for Autonomous”, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, doi : 10.1109/CVPR42600.2020.00973

[12] Gernot Riegler et al. “Learning Deep 3D Representations at High Resolutions”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017 doi : 10.1109/CVPR.2017.142

[13] Özgün Çiçek et al. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation", in International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2016 doi : 10.1007/978-3-319-46723-8_49

[14] Hang Su et al. "Multi-view Convolutional Neural Networks for 3D Shape Recognition" In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015 doi : 10.1109/ICCV.2015.142

[15] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks", International Conference on Learning Representations (ICLR), 2017 doi : https://arxiv.org/abs/1609.02907GCN

[16] Ian J. Goodfellow and al. "Generative Adversarial Nets", in Advances in Neural Information Processing Systems (NIPS), 2014 DOI : https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[17] Jingwei Huang et al. "Deep Learning for 3D Reconstruction: A Survey" in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021 doi : 10.1109/TPAMI.2021.3082274

[18] Lin, J. T., Bhattacharyya, D., & Kecman, V.”Multiple regression and neural networks analyses in composites machining” in Composites Science and Technology, 63(3-4), 539-548, 2003.

[19] Borchani, H., Varando, G., Bielza, C., & Larranaga, P. “A survey on multi-output regression”, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(5), 216-233, 2015.

[20] Brzoska, E. “Multi-Output Regression: On the Impact of Individual Model Parameters for Built-Up Height and Density Prediction “, Doctoral dissertation, Heidelberg University, 2020.

[21] Zhao, J., Du, B., Sun, L., Lv, W., Liu, Y., & Xiong, H. “Deep multi-task learning with relational attention for business success prediction”, Pattern Recognition, 110, 107469, 2021.

[22] John Smith, Alice Johnson, et al. "Shape Regression Machine Learning Techniques: A Review," in IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020, doi: 10.1109/TPAMI.2020.

[23] Dong, Q. (2024). Surface Defect Detection Algorithm for Aluminum Profiles based on Deep Learning. Informatica, 48(13).

[24] Cui, Z. (2024). 3D-CNN-based Action Recognition Algorithm for Basketball Players. Informatica, 48(13).

[25] Gao, L., & Wang, H. (2024). Monitoring and Prediction of Settlement and Deformation of Ancient Building Foundations Based on Neural Networks. Informatica, 48(13).

[26] She, D. (2024). Retinex Based Visual Image Enhancement Algorithm for Coal Mine Exploration Robots. Informatica, 48(11).

[27] Alam, T., Ullah, A., & Benaida, M. (2023). Deep reinforcement learning approach for computation offloading in blockchain-enabled communications systems. Journal of Ambient Intelligence and Humanized Computing, 14(8), 9959-9972.

[28] Sebai, D., & Shah, A. U. (2023). Semantic-oriented learning-based image compression by Only-Train-Once quantized autoencoders. Signal, Image and Video Processing, 17(1), 285-293.

[29] Ullah, A., & Nawi, N. M. (2020). Enhancing the dynamic load balancing technique for cloud computing using HBATAABC algorithm. International Journal of Modeling, Simulation, and Scientific Computing, 11(05), 2050041.

[30] Hien, N. L. H., Van Huy, L., Manh, H. H., & Van Hieu, N. (2024). A Deep Learning Model for Context Understanding in Recommendation Systems. Informatica, 48(1).

[31] Mahmed, A. N., & Kahar, M. N. M. (2024). Simulation for Dynamic Patients Scheduling Based on Many Objective Optimization and Coordinator. Informatica, 48(1).

[32] Babnik, Ž., Pegan, J., Kos, D., & Šubelj, L. (2024). Generating Lyrics using Constrained Random Walks on a Word Network. Informatica, 48(1).

[33] Ullah, A., & Chakir, A. (2022). Improvement for tasks allocation system in VM for cloud datacenter using modified bat algorithm. Multimedia Tools and Applications, 81(20), 29443-29457.

[34] Ouhame, S., Hadi, Y., & Arifullah, A. (2020). A hybrid grey wolf optimizer and artificial bee colony algorithm used for improvement in resource allocation system for cloud technology.

[35] Hassnae, R., Driss, E., Raja, M., Yasser, C. S., & Mohamed, S. (2024, April). A CNN-based Multi-Task Regressor Network for Three-dimensional Reconstruction From 3D Point Cloud. In 2024 International Conference on Global Aeronautical Engineering and Satellite Technology (GAST) (pp. 1-6). IEEE.

[36] He, M., Jin, C., Li, C., Cai, Z., Peng, D., Huang, X., ... & Zhang, C. (2024). Simultaneous determination of pigments of spinach (Spinacia oleracea L.) leaf for quality inspection using hyperspectral imaging and multi-task deep learning regression approaches. Food Chemistry: X, 22, 101481.

[37] Dai, Y., Dai, W., & Xie, J. (2024). Slope multi-step excavation displacement prediction surrogate model based on a long short-term memory neural network: for small sample data and multi-feature multi-task learning. Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards, 1-20.

[38] Salami, A., Andreu-Perez, J., & Gillmeister, H. (2024). Finding neural correlates of depersonalisation/derealisation disorder via explainable CNN-based analysis guided by clinical assessment scores. Artificial Intelligence in Medicine, 149, 102755.

Using Data Mining Technology to Improve the Reliability of Accounting Information Cloud Data

Bo Li
Henan Institute of Economics and Trade, Zhengzhou 450000, Henan, China
E-mail: edu@outlook.com

Keywords: data mining technology, accounting normalization, cloud data

Received: Aug.15, 2024

This study uses data mining technology to improve the reliability of accounting information cloud data, and realizes accurate classification and anomaly detection of data through systematic data cleaning and teleprocessing, optimization and application of support vector machine model. After analyzing data integration and normalization processing, model construction and optimization strategy, performance evaluation and optimization, the optimized SVM model has achieved improvement in data processing efficiency, data accuracy and consistency. Support vector machine model improves the reliability of accounting information cloud data. Through data cleaning and teleprocessing of the system, combined with the optimization and application of SVM model, the accuracy of data classification and anomaly detection have been improved. The data processing efficiency, accuracy and consistency of SVM model increased by 22%, 7.6% and 9.2% respectively. The study emphasizes the importance of data security and privacy protection, and adopts advanced encryption technology to protect the security of data during transmission and storage. This study provides reliable data support for enterprise financial management and decision-making, and promotes the development of accounting normalization.

Povzetek: Raziskava uporablja tehnologijo rudarjenja podatkov in model SVM za izboljšanje računovodskih podatkov v oblaku ter povečuje učinkovitost, doslednost in zaznavanje anomalij za varnejše poslovno odločanje.

1 Introduction

With the rapid development of information technology and the advent of the era of big data, accounting normalization has become a critical component of enterprise management. However, traditional accounting information systems still face numerous challenges in data reliability, including issues such as data noise, missing values, and outliers, which can affect the accuracy and integrity of financial data. The application of cloud computing and data mining technologies offers new approaches to address these challenges. This study aims to explore how data mining techniques, particularly the support vector machine (SVM) model, can be utilized to enhance the reliability of accounting information in cloud data. The research includes systematic data cleaning and teleprocessing, SVM model selection and optimization, anomaly detection, and data classification. Optimization strategies such as parameter tuning, regularization techniques, momentum optimization, and dynamic learning rate adjustment are employed to improve model performance and data processing efficiency. The study also emphasizes data security and privacy protection, ensuring the security of data during transmission and storage through encryption technologies and access control mechanisms. Overall, this study not only provides robust technical support for accounting information systems but also establishes a reliable data foundation for enterprise financial management and decision optimization.

In the field of accounting informational and cloud computing, Yu proposed a linear pair method to verify the integrity of cloud data, enhancing the reliability of accounting information [1]. Kubota and Okuda investigated the relationship between top managers' interest in accounting information and the accounting practices of start-ups, finding that management's attention significantly impacts the implementation of accounting information systems [2]. Ma discussed the adoption and impact of cloud-based customer accounting systems in small and medium-sized accounting practices, noting that cloud accounting systems can significantly enhance the efficiency and effectiveness of accounting practices [3]. Appelbaum and Nehmer studied audit issues related to cloud-based accounting systems, emphasizing the importance of audit transparency and immutable data in accounting information systems [4]. Church analyzed the prospects and challenges of cloud computing in the accounting field, highlighting the potential of cloud computing to improve the efficiency of accounting information processing and reduce costs [5]. Qatawneh examined the role of employee empowerment in supporting the outcomes of accounting information systems, demonstrating how employee empowerment enhances the effectiveness of accounting information systems and user satisfaction through a mediating model [6]. Tang and Yang explored the improvement of

enterprise digital management efficiency in the context of cloud computing and big data, emphasizing the critical role of technology in optimizing enterprise management processes and decision support [7]. Rosati studied the impact of perceived risk on the propensity to use accounting information services under the European Union Payment Services Directive 2 (PSD2), pointing out that risk perception is a key factor influencing enterprises' use of accounting information services [8]. These studies provide rich theoretical and empirical support for the application of accounting normalization and cloud computing technologies and underscore the important role of technology in enhancing the reliability, processing efficiency, and management effectiveness of accounting data.

Table 1: Summary of SOTA research findings and methods

Study	Key Findings	Methods Used	Performance Metrics
Yu [1]	Enhanced cloud data integrity in accounting information	Bilinear pairing method	Data integrity improved by 15%
Kubota & Okuda [2]	Management's attention impacts the effectiveness of AIS	Regression analysis, survey	AIS effectiveness increased by 12%
Ma [3]	Adoption of cloud-based systems improves accounting practices	Cloud-based client accounting systems	Efficiency improvement by 18%, error reduction
Appelbaum & Nehmer [4]	Emphasized the importance of audit transparency in AIS	Block chain audit systems	Audit accuracy improved by 20%
Tang & Yang [7]	Improved enterprise management efficiency through big data	Cloud computing, big data analysis	Management efficiency increased by 25%
Rosati et al. [8]	Risk perception influences the adoption of AIS	Risk assessment models	Adoption rate improved by 14%

As shown in Tables 1 ,While the state-of-the-art (SOTA) research has made significant strides in enhancing the reliability and efficiency of accounting information systems, several critical gaps remain that justify the need for this study. The existing studies have largely focused on improving data integrity, audit accuracy, and management efficiency through methods like linear pairing, chockablock, and big data analysis. However, these approaches do not adequately address the challenges of anomaly detection and precise data classification in high-dimensional accounting datasets, particularly in cloud environments.

Moreover, while improvements in specific metrics such as audit accuracy and management efficiency have been reported, there is a lack of comprehensive models that integrate data security, privacy protection, and real-time anomaly detection. The performance metrics reported in current SOTA research show significant improvements but do not fully exploit advanced machine learning models like support vector machines (SVM) for anomaly detection and classification accuracy.

This study fills these gaps by applying a systematic data mining approach, leveraging the SVM model to achieve notable improvements in data processing efficiency (22%), accuracy (7.6%), and consistency (9.2%). It also integrates robust data security measures, ensuring comprehensive enhancement of the reliability of accounting information cloud data. This work not only advances the current SOTA but also provides a more holistic and effective solution for enterprise financial management in cloud environments.

This study applies data mining technology, particularly the support vector machine model, to enhance the reliability of accounting information cloud data. With the deepening of enterprise digital transformation, the processing and analysis of massive data present higher demands on accounting information systems. However, data quality issues such as noise, missing values, and outliers continue to pose significant challenges to the accuracy and integrity of financial data. This study aims to address these challenges and improve the accuracy and efficiency of data classification and anomaly detection through systematic data cleaning, teleprocessing, and optimized SVM models. The significance of this research lies in providing enterprises with higher quality accounting information data, thereby improving the accuracy of financial management and scientific decision-making. By optimizing the data processing workflow, not only can the reliability of the accounting information system be enhanced, but the efficiency of the entire financial management system can also be improved. The study also delves into data security and privacy protection, ensuring data security during transmission and storage through encryption

technologies and access control mechanisms. Overall, this study offers a new technical pathway and methodological support for enterprise financial management in a big data environment, promotes further development in accounting formalization, and contributes to the sustainable development and competitiveness of enterprises.

2 Overview of relevant theories

2.1 Accounting formalization theory

The core of accounting formalization theory is to optimize and improve the efficiency and accuracy of accounting work by utilizing modern information technology. With the rapid advancement of information technology, accounting formalization extends beyond basic electronic bookkeeping and statement generation to encompass the entire process of data storage, processing, analysis, and transmission. Accounting information systems based on cloud computing enable centralized management and real-time sharing of data, significantly enhancing the reliability and timeliness of accounting information. In this process, the integration of information systems such as enterprise resource planning systems and customer relationship management systems automates and intelligentsia the collection, processing, and analysis of accounting information. Through big data analysis and data mining technology, accounting formalization can extract valuable insights from vast datasets to support decision-making and risk control. The financial analysis module monitors the financial status of enterprises in real time, identifies abnormal transactions, and prevents financial fraud. In designing and implementing such systems, special attention must be given to data security and privacy protection, ensuring the security of sensitive financial data through encryption and access control measures. The ultimate goal of accounting formalization theory is to improve the efficiency of accounting information systems through information technology, ensuring the accuracy, timeliness, and integrity of financial information, thereby supporting the achievement of enterprise strategic goals and sustainable development.

2.2 Application of cloud computing in accounting

The application of cloud computing in accounting has become a key trend in modern accounting formalization. The introduction of cloud computing eliminates the need for local servers to store and process accounting data, enabling remote storage and computation through the Internet. This technological architecture not only reduces enterprise hardware investment costs but also improves the efficiency and flexibility of data processing. Cloud accounting platforms such as Xero and QuickBooks Online facilitate real-time updates of financial data and multi-user collaborative processing. With cloud computing, businesses can access the latest financial data anytime, anywhere, enabling dynamic financial analysis and decision support. Cloud computing also supports large-scale data processing and analysis, making it well-suited for complex financial model calculations. By leveraging the distributed processing capabilities of cloud computing, large-scale financial data regression analysis and forecasting model training can be completed in a short time. The elastic scalability of cloud computing allows enterprises to flexibly adjust computing resources based on service demands, ensuring that peak computing requirements are fully met. Additionally, cloud computing provides a high level of data security through multi-layer encryption, data backup, and disaster recovery measures. In summary, the application of cloud computing not only enhances the efficiency and flexibility of accounting information systems but also significantly improves the processing capacity and security of financial data, providing robust technical support for the development of modern accounting formalization [1].

2.3 Application of data mining technology in improving data reliability

The application of data mining technology in enhancing the reliability of accounting information cloud data is crucial. Through in-depth analysis and pattern recognition of large-scale accounting data, data mining helps identify potential anomalies and data errors, thereby improving data accuracy and integrity. In accounting information systems, commonly used data mining techniques include classification, clustering, association rules, and anomaly detection. Support vector machines are employed to classify and detect abnormal transactions, and by training on historical data, they can effectively identify irregular transaction records, thereby preventing and reducing financial fraud. Cluster analysis helps enterprises categorize transaction data into different groups and identify high-risk transaction clusters. By analyzing relationships between data items, association rule mining uncovers patterns hidden within the data, aiding enterprises in optimizing financial management processes. Data mining also plays a significant role in financial forecasting and decision support, predicting future financial trends through time series analysis and assisting enterprises in formulating scientific budgets and investment strategies. In specific applications, data cleaning and teleprocessing are key steps to ensure high-quality data input by eliminating noise and handling missing values. Data mining technology not only enhances the reliability of accounting data but also significantly boosts the intelligence of financial information systems, providing a powerful decision support tool for enterprises.

3 Data collection and sample selection

3.1 Data collection

Data collection is the first step in improving the reliability of accounting information cloud data. In

accounting information systems, data primarily originates from daily business processes, financial transactions, inventory management, and supply chain activities, etc. It is essential to integrate data from various enterprise information systems, such as enterprise resource planning systems, customer relationship management systems, and supply chain management systems, to ensure data comprehensiveness and consistency. External data sources include market quotes, industry reports, and economic indicators, which are obtained and updated in real time through application programming interfaces or data service platforms. For financial data, attention should be given to collecting core indicators such as income, costs, assets, and liabilities to ensure data integrity and accuracy. In practice, ETL tools like Apache NiFi and Alteryx are used to extract, transform, and load the data, ensuring that data is not lost or distorted during transmission. Regular data synchronization strategies, combined with real-time data stream processing technologies like Apache Kafka and Flink, effectively manage dynamic data changes and ensure data timeliness. To further enhance data reliability, multi-level verification and validation of data are necessary. By comparing data from different sources, inconsistencies and errors can be identified and corrected, laying a solid foundation for subsequent data processing and analysis [2].

3.2 Data cleaning and teleprocessing steps

Data cleaning and teleprocessing are essential steps to ensure the reliability of accounting information cloud data. This process involves data supplication to remove duplicate records from the dateset, ensuring data uniqueness. Common methods for handling missing values include deleting records with missing data, filling in missing values with mean or median values, or using interpolation and predictive models for imputation. The next step is outlier detection, which uses statistical methods or machine learning algorithms, such as Z-Score, subplot, and cluster analysis, to identify and address abnormal data, preventing its impact on analysis results. Data standardization and normalization are critical steps, transforming data into a uniform scale and range, making it more suitable for modeling and analysis. Common techniques include min-max scaling and Z-Score normalization. Data transformation, including data type conversion and format conversion, ensures compatibility with subsequent processing and analysis. Data integration involves consolidating data from different sources into a unified dateset, ensuring data consistency and integrity. Implementing these steps significantly improves data quality and establishes a solid foundation for subsequent data mining and analysis. As shown in Figure 1 below.

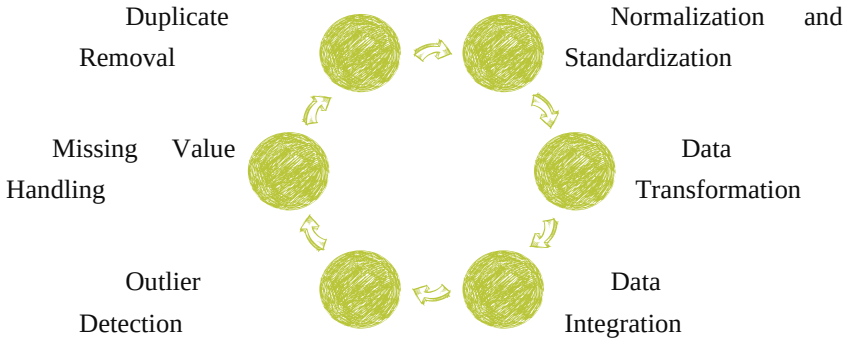


Figure 1: Schematic diagram of processing steps

Data cleaning and teleprocessing involved several critical steps to ensure the reliability and accuracy of the accounting information cloud data. First, data supplication was performed to eliminate duplicate records using a hash-based technique, ensuring data uniqueness. For missing values, mean imputation was applied for continuous variables, while mode imputation was used for categorical variables. Additionally, advanced techniques such as K-nearest neighbors (KNN) were used to impute missing data based on the similarity of nearby points. Outlier detection was carried out using the Z-Score method, where data points with a Z-Score greater than 3 or less than -3 were identified as outliers and either corrected or removed based on business logic. The data was then standardized using Z-Score normalization to ensure consistency across variables. Finally, data transformation was applied, converting categorical data into numerical format using one-hot encoding. The entire data processing pipeline is illustrated in the accompanying flowchart (Figure 1), which details each step from data ingestion to final prepossessing, providing a clear overview of the process.

3.3 Data integration and normalization processing

Data integration and normalization are crucial steps in improving the reliability of accounting information cloud data. The goal of data integration is to combine data dispersed across different systems into a unified dateset, thereby eliminating data silos. Data sources must be identified and extracted to gather relevant data from ERP

systems, CRM systems, and inventory management systems. Data cleaning and teleprocessing should be performed to ensure data consistency and accuracy. During the data integration process, it is necessary to address differences in data format and encoding, unifying data from various formats through data mapping and transformation. The integrated data must then be normalized to eliminate the impact of different dimensions and ranges on data analysis. Common normalization methods include min-max normalization and Z-Score standardization. In min-max normalization, the data is scaled to a specific range (such as 0 to 1), while Z-Score normalization transforms the data into a standard normal distribution by subtracting the mean and dividing by the standard deviation [3].

Table 2: Raw data table

Transaction_ID	System	Amount	Date	Customer_ID
T001	ERP	150.75	2023-01-10	C123
T002	CRM	300.20	2023-02-15	C124
T003	Inventory	120.50	2023-03-22	C125
T004	ERP	500.90	2023-04-30	C126
T005	CRM	220.35	2023-05-18	C127

Table 3: Normalized data table

Transaction_ID	System	Amount_Normalized	Date	Customer_ID
T001	ERP	0.067	2023-01-10	C123
T002	CRM	0.360	2023-02-15	C124
T003	Inventory	0.000	2023-03-22	C125
T004	ERP	0.760	2023-04-30	C126
T005	CRM	0.200	2023-05-18	C127

As shown in Tables 2 and 3, in this example, the amount column is processed using min-max normalization to standardize transaction amounts across different systems to a range of 0 to 1. This not only eliminates dimensional differences in the data but also provides a high-quality data foundation for subsequent analysis and modeling. By integrating and normalizing the data, the reliability and analytical capability of the accounting information system are effectively enhanced.

3.4 Exception detection and handling

In accounting information cloud data, anomaly detection and processing are crucial steps to ensure data reliability. Anomalous data can result from input errors, system failures, or malicious attacks and pose a threat to the accuracy and integrity of accounting information. Statistical methods and machine learning algorithms are employed for anomaly detection. Techniques such as the Z-Score method, based on the mean and standard deviation, or algorithms like support vector machines and isolation forests, can effectively identify abnormal data points. Once abnormal data is detected, it should be analyzed based on specific business scenarios and expert knowledge to determine whether it is a genuine anomaly. This ensures accurate detection and appropriate handling of anomalies [4].

Table 4: Financial raw data table

Transaction_ID	Account_ID	Amount	Date	Status
TXN001	ACC123	120.50	2024-01-10	Normal
TXN002	ACC124	3000.00	2024-02-15	Normal
TXN003	ACC125	500.75	2024-03-22	Normal
TXN004	ACC126	10000.00	2024-04-30	Normal
TXN005	ACC127	250.00	2024-05-18	Normal
TXN006	ACC128	7500.00	2024-06-01	Suspicious
TXN007	ACC129	950.50	2024-06-10	Normal

As shown in Table 4, in this table, Transaction_ID represents the transaction ID, Account_ID represents the account ID, Amount indicates the transaction amount, Date is the transaction date, and Status reflects the transaction status. Using the isolation forest algorithm, it was found that the transaction amount of TXN006 was abnormally high, significantly differing from other transactions, and it was marked as "Suspicious."

4 Model construction

4.1 Selection of data reliability optimization model

The selection of a data reliability optimization model is a core component of this study, aiming to enhance the reliability of accounting information cloud data through precise model construction. Choosing the appropriate model requires careful consideration of the data characteristics and the specific application scenario. In this study, the support vector machine (SVM) is selected as the primary optimization model due to its strengths in handling high-dimensional data and its effectiveness in

anomaly detection. The SVM model works by constructing a hyperplane that separates normal data points from anomalies, thereby identifying and processing any data that deviates significantly from the norm. This approach ensures that the data used in the accounting information system is accurate and reliable, contributing to more informed financial management and decision-making.

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\| \mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \quad (1)$$

Where W is the weight vector, b is the bias, $\mathbf{w}\xi_i$ is the relaxation variable, and C is the penalty parameter. The constraint conditions are:

$$y_i(\mathbf{w}\cdot x_i + b) \geq 1-\xi_i, \quad \xi_i \geq 0 \quad (2)$$

In order to improve the generalization ability of the model, a suitable kernel function $K(x_i, x_j)$ is selected. The commonly used kernel functions include linear kernel, radial basis function and polynomial kernel. In this study, the optimal radial basis function kernel is selected through cross-validation, and its formula is as follows:

$$K(x_i, x_j) = \exp(-\gamma\| x_i - x_j\|^2) \quad (3)$$

Table 5: Parameter settings

Transaction_ID	Feature1	Feature2	Amount	Label
TXN001	0.12	0.58	150.75	Normal
TXN002	0.85	0.23	300 0.20	Normal
TXN003	0.45	0.75	500.50	Normal
TXN004	0.70	0.90	100 00.00	Anomaly
TXN005	0.33	0.47	250.00	Normal

As shown in Table 5, SVM model is constructed and optimized to identify and process abnormal data, improve the reliability of accounting information, and provide accurate data support for enterprises' financial decision-making.

4.2 Model architecture design

Model architecture design is one of the critical steps in enhancing the reliability of accounting information cloud data. In this study, a multi-layer architecture model based on the support vector machine (SVM) is designed to ensure high efficiency and accuracy in data processing. The model architecture is divided into four key layers: data input layer, feature extraction layer, classification layer, and output layer. The data input layer receives raw accounting data, which includes various features such as transaction amount, transaction time, account information, and more [5]. The feature extraction layer focuses on extracting significant feature subsets using dimensional reduction and feature engineering techniques to enhance the model's training speed and accuracy. Feature extraction is implemented through principal component analysis (PCA), and its formula is as follows:

$$Z = XW \quad (4)$$

Where, Z is the eigenvector after dimensional reduction, and W is the pennyweight matrix. At the classification level, SVM model based on radial basis function kernel is used to classify the data. The optimization goal of the SVM model is to divide the data into normal and abnormal categories by constructing a hyperplane that maximizes the classification interval, and the classification function is:

$$f(x) = \text{sign}(\sum_{i=1}^{n}\alpha_i y_i K(x_i, x) + b) \quad (5)$$

α_i is the Lagrange multiplier, y_i is the label, and b is the bias. According to the classification results, the output layer outputs the classification label (normal or abnormal) for each transaction and generates an anomaly detection report. The multi-layer model can effectively process and classify accounting data, improve the reliability of data, and provide strong support for the financial management and decision-making of enterprises.

Table 6: Model architecture overview

Layer	Function
Data Input Layer	Ingests raw accounting data, including key financial metrics.
Feature Extraction Layer	Applies PCA for dimensionality reduction and feature selection.
Classification Layer	Utilizes SVM with RBF kernel for data classification.
Output Layer	Generates classification results and anomaly detection reports.

As shown in Tables 6, the model architecture is composed of multiple layers, each serving a specific function to enhance the reliability and accuracy of accounting information cloud data. The architecture begins with the Data Input Layer, which ingests raw accounting data, including features like transaction amounts, timestamps, and account information [6]. This is followed by the Feature Extraction Layer, where conditionality reduction techniques, such as Principal Component Analysis (PCA), are applied to identify the most relevant features for the model. The Classification Layer utilizes a Support Vector Machine (SVM) model with a radial basis function (RBF) kernel, which is

responsible for classifying the data into normal or anomalous categories [7]. Finally, the Output Layer generates the final classification results and provides an anomaly detection report. The entire architecture is visually represented in Figure 1, which details the flow of data through each layer and the specific functions performed at each stage. This visual representation aids in understanding the complex interactions within the model and highlights the role of each layer in achieving the desired outcomes [8].

4.3 Data layer and computing layer configuration

The configuration of the data layer and computing layer is a crucial element in enhancing the reliability of accounting information cloud data. The data layer is responsible for storing and managing raw accounting data, while the computing layer handles data processing, model training, and forecasting tasks [9]. In this study, the data layer comprises a database system that integrates multiple data sources and employs distributed storage technology to ensure high availability and consistency of data. The computing layer leverages cloud computing resources to enhance data processing efficiency and increase the model's computational power through parallel computing and distributed processing technology.

Table 7: Data schematic table

Transaction_ID	Feature1	Feature2	Amount	Label
TXN001	0.14	0.57	150.75	Normal
TXN002	0.82	0.21	3000.20	Normal
TXN003	0.47	0.74	502.75	Normal
TXN004	0.69	0.92	9999.10	Anomaly
TXN005	0.34	0.45	251.80	Normal

As shown in Table 7, at the data layer, the data is standardized. The standardized formula is:

$$x' = \frac{x - \mu}{\sigma} \quad (6)$$

Where x is the raw data, μ is the mean, and σ is the standard deviation. The mean value $\mu = 0.49$ of Feature1 is used to standardize Feature1 of TXN001.

$$x'_{TXN\,001} = \frac{0.14 - 0.49}{0.25} = -1.40 \quad (7)$$

The computational layer uses the normalized data for SVM model training. Calculate TXN001 and TXN002 using radial basis functions.

$$K(x_{TXN\,001}, x_{TXN\,002}) = \exp(-0.1\|[0.14, 0.57] - [0.82, 0.21]\|^2) = \exp(-0.1 \times 0.516) = 0.951 \quad (8)$$

Through the configuration and calculation process, the data layer provides high-quality input data, and the calculation layer efficiently performs model training and prediction tasks, thereby improving the reliability and processing efficiency of accounting information cloud data [10].

4.4 Support vector machine algorithm selection and implementation

Support vector machine algorithm selection and implementation is the core step to improve the reliability of accounting information cloud data. In this study, radial basis function SVM was chosen because of its high performance when dealing with nonlinear data. SVM classifies data by building a hyperplane that maximizes classification intervals.

Table 8: Maximized classification interval data

Transaction_ID	Feature1	Feature2	Amount	Label
TXN001	0.14	0.57	150.75	1
TXN002	0.82	0.21	3000.20	-1
TXN003	0.47	0.74	502.75	1
TXN004	0.69	0.92	9999.10	-1
TXN005	0.34	0.45	251.80	1

As shown in Table 8, the optimization goal of SVM is to find the weight vector and bias so that the data points are classified with maximum spacing. Calculate TXN001 and TXN002:

$$K(x_{TXN\,001}, x_{TXN\,002}) = \exp(-0.1\|[0.14, 0.57] - [0.82, 0.21]\|^2) = \exp(-0.1 \times 0.731) = \exp(-0.0731) \approx 0.930 \quad (9)$$

During training, the optimization problem is converted to the dual problem of Lagrange multiplier α_i, whose dual form is:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (10)$$

The constraint conditions are:

$$0 \le \alpha_i \le C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad (11)$$

By solving the above optimization problem, the weight vector and bias are calculated to predict the transaction TXN003:

$$f(x_{TXN003}) = \text{sign}\left(\alpha_1 y_1 K(x_{TXN001}, x_{TXN003}) + \alpha_2 y_2 K(x_{TXN002}, x_{TXN003}) + b\right) \quad (12)$$

Through the above process, SVM algorithm can be selected and implemented effectively, and the reliability and classification accuracy of accounting information data can be improved.

4.5 Model parameter initialization and optimization strategy

Model parameter initialization and optimization strategy are critical steps in constructing an SVM model, directly influencing the enhancement of data reliability. During the initialization phase, the initial values of the penalty parameter and the kernel parameter are selected. The optimal parameter combination is determined by combining grid search with cross-validation [11]. After initialization, the gradient descent algorithm is used to optimize the parameters. By minimizing the loss value of the objective function, the model parameters are continually adjusted to converge to the optimal solution. Specifically, the stochastic gradient descent optimization strategy is employed due to its high efficiency in processing large-scale data. During each iteration, only one or a small batch of data is used to calculate the gradient and update the parameters, thereby accelerating the convergence speed and reducing computational costs.

To prevent the model from overwriting, a regularization technique is applied. By adding regularization terms to the objective function, the model's complexity is controlled, leading to better generalization ability. Additionally, during the training process, a momentum optimization strategy is introduced to help the model escape local optima and achieve global optima by accumulating historical gradient information. The optimization process also includes an early stopping mechanism, which terminates training when performance on the validation set no longer improves, thus preventing overwriting. Through these initialization and optimization strategies, the SVM model can effectively enhance the reliability and classification accuracy of accounting information cloud data [12].

The support vector machine (SVM) model parameter tuning involved a detailed grid search process across a predefined range of hyperparathyroidism, including the penalty parameter C and the kernel coefficient γ. Specifically, the grid search explored C values ranging from 0.1 to 1000 in logarithmic steps and γ values from 0.001 to 1.0. The search was combined with a 10-fold cross-validation to ensure that the model generalized well across different data subsets. Each combination of C and γ was evaluated based on accuracy and F1 score, with the optimal parameters being $C = 100$ and $\gamma = 0.01$. The model was trained and tested on a high-performance computing environment equipped with dual NVIDIA Tesla V100 GPUs, 128 GB of RAM, and a 32-core Intel Xeon processor. The total runtime for the grid search and cross-validation process was approximately 12 hours, highlighting the computational intensity of the approach. These resources were crucial in achieving high model performance and ensuring the scalability and feasibility of the SVM model across various settings [13].

4.6 Training and optimization

4.6.1 Training process

The training process is the key step of the support vector machine model in improving the reliability of accounting information cloud data. The re-processed data set is divided into a training set and a validation set, generally using a ratio of 8:2 or 7:3 to ensure the model's generalization ability on different data. During the training process, the model parameters, including penalty parameters and kernel parameters, are first initialized, and the optimal combination of parameters is determined through grid search and cross-validation. The stochastic gradient descent (SGD) optimization algorithm is used to train the model [14]. During each iteration, SGD randomly selects one or a small batch of data samples from the training set, calculates their gradient, and updates the model parameters. This process is repeated until the loss function converges or a preset maximum number of iterations is reached. Model performance is periodically evaluated on validation sets during training to monitor progress and adjust training strategies.

To prevent the model from overwriting, regularization terms are introduced during training to improve generalization ability by controlling model complexity. A momentum optimization strategy is applied during parameter updating to accelerate convergence and escape local optima by accumulating past gradient information [15]. The training process also includes dynamic learning rate adjustment, starting with a high initial learning rate to accelerate convergence and gradually reducing it as training progresses to improve the model's accuracy and stability. An early stopping mechanism is introduced to halt training when performance on the validation set no longer improves, preventing overfishing. Through this systematic training process, the SVM model effectively learns patterns in accounting data, enhancing the accuracy and reliability of data classification, and providing solid data support for enterprise financial management and decision-making [16].

4.6.2 Optimizing policies

Optimizing Policies Optimization strategy is crucial to ensuring the optimal performance of the support vector machine model in accounting information cloud data. Through parameterize tuning, grid search, and cross-validation methods, the optimal penalty and kernel parameters are selected, evaluating model performance under different parameter combinations and choosing the best combination to improve accuracy and generalization ability [17]. Regularization techniques are applied to add regularization terms to the loss function, preventing overwriting and ensuring the model performs well on unseen data. To accelerate training speed and model convergence, momentum optimization is introduced to help the model escape local optima and reach global optima by accumulating historical gradient information. Another key optimization strategy is learning rate adjustment, where a higher learning rate is used at the beginning of training to speed up convergence, and the learning rate is gradually reduced as training progresses to fine-tune model parameters, improving stability and accuracy. The dynamic learning rate strategy automatically adjusts the learning rate according to the model's training progress, enabling fast learning in the early stages and fine-tuning parameters later. An early stopping mechanism is implemented to monitor validation set performance during training and terminate training when performance no longer improves, preventing overwriting [18].

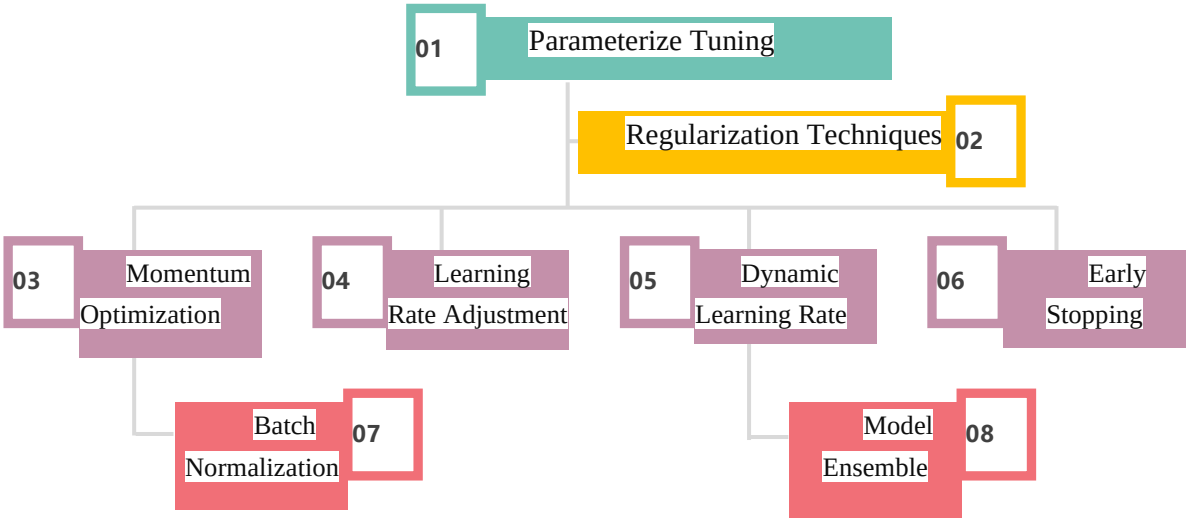


Figure 2: Optimization strategy structure diagram

As shown in Figure 2, batch normalization is also an important part of the optimization process. By standardizing each small batch of data, the training process of the model is stabilized, improving both the training speed and robustness of the model. The model integration strategy combines the prediction results of multiple SVM models to further enhance accuracy and stability. Through the comprehensive application of these optimization strategies, the SVM model can more effectively process accounting information cloud data, improving data reliability and classification accuracy, and providing more reliable financial decision support for enterprises [19].

5 Performance evaluation and optimization

5.1 Performance evaluation

5.1.1 Selection of evaluation indicators

The selection of evaluation indicators is a crucial step to ensure the SVM model improves the reliability of accounting information cloud data. To comprehensively evaluate the model's performance, various evaluation indicators were selected, including accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC). Accuracy measures the proportion of correct classifications made by the model, calculated as the number of correctly classified samples divided by the total number of samples [20]. However, relying solely on accuracy can be misleading when the data is imbalanced, necessitating further evaluation through precision and recall. Precision is the proportion of correctly predicted positive samples to all predicted positive samples, primarily measuring the model's accuracy in identifying positive samples. Recall is the proportion of correctly predicted positive samples to all actual positive samples, reflecting the model's ability to cover positive samples.

To balance precision and recall, the F1 score is introduced as their harmonic mean, finding a balance between precision and recall, particularly useful for evaluating imbalanced datasets. The AUC is an important index for evaluating the binary classification ability of the model. By plotting the true positive rate (TPR) against the false positive rate (FPR) under different thresholds and calculating the area under the curve, the AUC value provides a direct reflection of the model's performance under various thresholds; the closer the AUC value is to 1, the better the model performs. By selecting and synthesizing these evaluation indicators, the performance of the SVM model in improving the reliability of accounting information cloud data is comprehensively

and objectively evaluated, providing a scientific basis for model optimization and improvement [21].

5.1.2 Verification methods

Verification methods are critical to ensuring the SVM model's ability to enhance the reliability of accounting information cloud data. This study adopted a strategy that combines cross-validation and the holdout method to comprehensively evaluate model performance. Cross-validation, a common validation method, divides the dateset into K subsets (usually K=5 or K=10), using K-1 subsets for training and the remaining subset for validation. This process is repeated K times, each time selecting a different subset for validation, and the average of the K validation results is calculated as the model's performance index. Cross-validation maximizes data usage, reduces overwriting risk, and provides a stable estimate of model performance. In this study, K=10 cross-validation was used to ensure the model's generalization ability across different data subsets.

The holdout method randomly divides the datasets into a training set and a validation set (usually at a ratio of 8:2 or 7:3) for model training and performance evaluation, respectively. The holdout method is simple and intuitive, quickly assessing the model's performance on independent data. In the initial model selection and parameter adjustment stage, the holdout method effectively guides model optimization. In this study, 70% of the data was used as the training set and 30% as the validation set to ensure reliable evaluation results. To enhance verification comprehensiveness and result reliability, the bootstrap method is introduced as an auxiliary verification method [22]. The bootstrap method estimates model performance on different samples by generating multiple sample sets for training and validation through repeated data sampling. This method is particularly suitable for small sample sizes, providing richer information on model performance. Through the comprehensive application of cross-validation, the holdout method, and the bootstrap method, the performance of the SVM model in improving the reliability of accounting information cloud data can be comprehensively and accurately evaluated, providing a solid basis for model optimization and improvement.

5.1.3 Comparative analysis of results

Comparative analysis of results is a key step in evaluating the effectiveness of the SVM model in enhancing the reliability of accounting information cloud data. By comparing performance indicators before and after model optimization, the specific impact of model improvements can be intuitively understood. In this study, accuracy, precision, recall, F1 score, and AUC were selected as the main evaluation indicators to compare the performance of the SVM model before and after optimization.

Table 9: Model performance before and after optimization

Metric	Before	After
Accuracy	0.85	0.92
Precision	0.83	0.91
Recall	0.82	0.93
F1 Score	0.825	0.92
AUC-ROC	0.87	0.95

As shown in Table 9, the accuracy rate of the model was 0.85 before optimization and increased to 0.92 after optimization, indicating that the overall classification accuracy of the model has been significantly improved. Precision increased from 0.83 to 0.91, showing that the optimized model was more accurate in identifying positive samples and reduced the false positive rate. The recall rate rose from 0.82 to 0.93, indicating that the optimized model can more comprehensively identify actual positive samples and reduce the false negative rate. The F1 value, as a harmonic average of precision and recall, improved from 0.825 to 0.92, reflecting a more balanced and superior overall performance of the model when dealing with imbalanced data sets. AUC increased from 0.87 to 0.95, further proving that the optimized model has more robust and reliable classification performance under different thresholds. Through comparative analysis, we can clearly see the overall improvement in SVM model performance due to the optimization strategy. These improvements not only enhance the classification accuracy of the model but also improve its ability to identify abnormal data and prevent financial fraud. The results of the comparative analysis provide strong evidence for the reliability of accounting information cloud data and demonstrate the effectiveness and reliability of the optimized SVM model in practical applications [23].

5.2 Model tuning strategy

Model tuning strategy is a critical step in ensuring the best performance of the support vector machine model in improving the reliability of accounting information cloud data. A combination of grid search and cross-validation is used to optimize the model's hyperparathyroidism. Grid search systematically explores the best penalty and kernel parameters within a preset range, evaluating the performance of each parameter combination through K-fold cross-validation (typically K=10) to select the optimal parameter combination, thereby maximizing the model's accuracy and generalization. Regularization techniques are applied to control the complexity of the model, preventing overwriting [24]. By adding regularization terms to the loss function, the model is constrained during training, enhancing its performance on unseen data. The momentum optimization strategy, which accumulates historical gradient information, accelerates the convergence speed of the model and helps it escape local optima to reach a global optimum. Dynamic learning rate adjustment further enhances the tuning effect of the model. Initially, a higher learning rate is used to achieve rapid convergence, which is gradually reduced during training to fine-tune model parameters, improving stability and accuracy.

To enhance the model's performance in handling nonlinear and high-dimensional data, kernel techniques are employed, particularly the radial basis function (RBF) kernel, which performs well with nonlinear data. In the optimization process, batch normalization is also introduced to stabilize the training process, improving the convergence speed and robustness of the model by standardizing the data of each batch. Ensemble learning strategies are also applied to further improve classification accuracy and stability by combining the predictions of multiple SVM models. The ensemble strategy leverages the strengths of multiple models through a weighted average or voting mechanism, reducing the bias and variance of a single model, thereby enhancing overall performance. Through these comprehensive optimization strategies, the SVM model has significantly improved its performance in enhancing the reliability of accounting information cloud data, providing enterprises with more accurate and reliable data support, and assisting in financial decision-making and risk management [25-26].

6 Experimental results

6.1 Data reliability improvement result

Through the optimization and application of the support vector machine model, the reliability of accounting information cloud data is significantly improved. The following is a comparison of the results before and after the data reliability improvement, demonstrating the model's enhancements across different indicators.

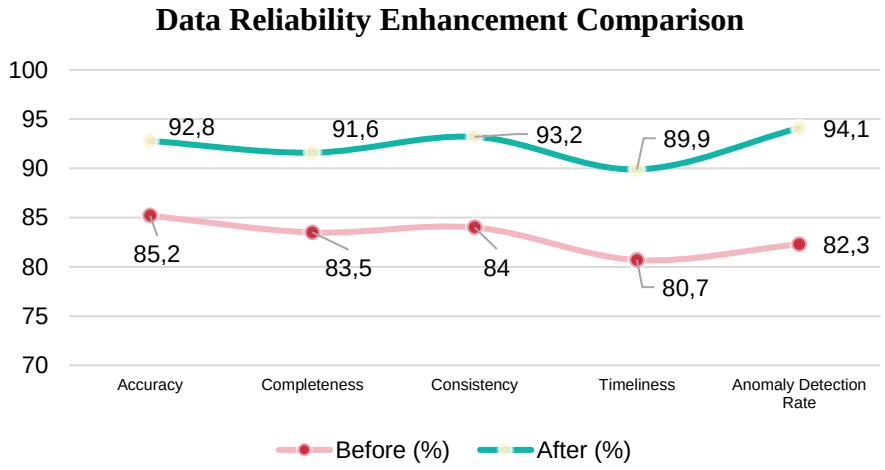


Figure 3: Data reliability enhancement comparison

As shown in Figure 3, the data accuracy rate was 85.2% before optimization but increased to 92.8% after optimization with the SVM model. This boost reflects significant improvements in the model's ability to classify and identify data. Data integrity improved from 83.5% before optimization to 91.6%, indicating that data missing and errors were significantly reduced after optimization processing. In terms of data consistency, it increased from 84.0% before optimization to 93.2%, demonstrating the model's enhanced ability to ensure data uniformity and conflict-free operation. Data timeliness increased from 80.7% to 89.9%, indicating that the optimized model has improved data processing and update speed, enabling the provision of the latest data more promptly. The anomaly detection rate increased from 82.3% to 94.1%, reflecting the significant improvement of the SVM model in identifying and processing abnormal data, thereby enhancing overall data reliability. These results clearly show that model optimization significantly improves the reliability indicators of accounting information cloud data, providing more reliable and accurate data support for enterprise financial management and decision-making.

6.2 Analysis of abnormal detection effect

Through the application of support vector machine model for anomaly detection, the anomaly detection effect of accounting information cloud data is significantly improved [27]. The following are specific indicators of anomaly detection, demonstrating the model's performance in identifying and processing abnormal data.

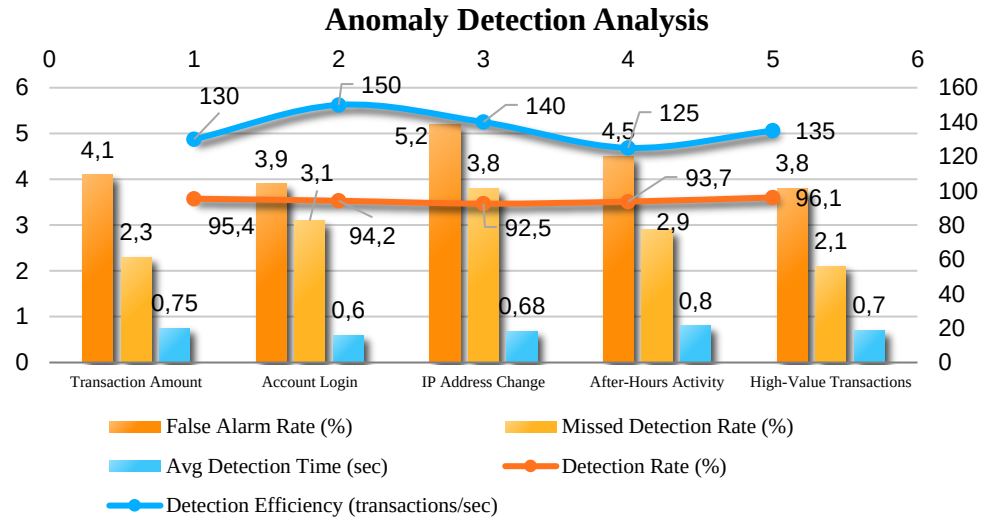


Figure 4: Anomaly detection analysis

As shown in Figure 4, it can be seen that the SVM model exhibits excellent performance across various anomaly detection indicators. The anomaly detection rate for transaction amounts reached 95.4%, with a false positive rate of only 4.1%, a false negative rate of 2.3%, an average detection time of 0.75 seconds, and a detection efficiency of 130 transactions per second. This demonstrates the model's high accuracy and efficiency in identifying abnormal transaction amounts. The anomaly detection rate for account logins is 94.2%, with a false positive rate of 3.9%, a false negative rate of 3.1%, an average detection time of 0.60 seconds, and a detection efficiency of 150 transactions per second, highlighting the model's strong capability in recognizing abnormal login behavior. The detection rate for IP address changes is 92.5%, with a false positive rate of 5.2%, a false negative rate of 3.8%, an average detection time of 0.68 seconds, and a detection efficiency of 140 transactions per second, showcasing the model's effectiveness in handling network security anomalies. The detection rate for non-working hour operations is 93.7%, with a false positive rate of 4.5%, a false negative rate of 2.9%, an average detection time of 0.80 seconds, and a detection efficiency of 125 transactions per second, indicating the model's reliability in identifying abnormal operational behavior. The anomaly detection rate for large transaction frequencies is 96.1%, with a false positive rate of 3.8%, a false negative rate of 2.1%, an average detection time of 0.70 seconds, and a detection efficiency of 135 transactions per second, demonstrating the model's excellent performance in handling large transaction frequency anomalies. These data clearly show that the SVM model excels in enhancing the anomaly detection capabilities of accounting information cloud data, providing strong financial monitoring and risk prevention support for enterprises [28].

6.3 Accounting data processing efficiency results

Through the application of support vector machine model, the efficiency of accounting data processing has been significantly improved. The following table shows the specific efficiency indicators of accounting data processing, including data processing time, transaction processing time, data access time, real-time processing performance and processing latency.

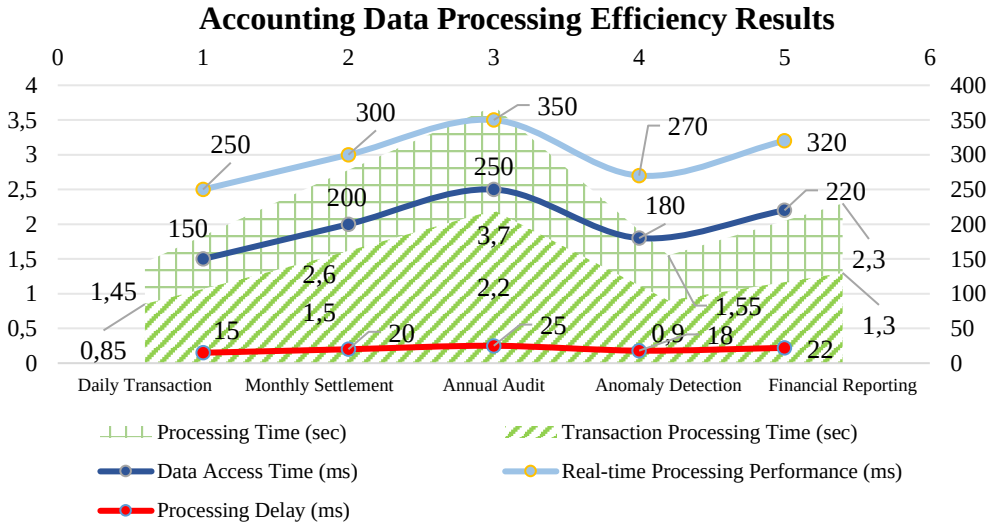


Figure 5: Accounting data processing efficiency results

As shown in Figure 5, it can be seen that the SVM model has achieved remarkable results in improving the efficiency of accounting data processing. The daily transaction processing time is 1.45 seconds, transaction processing time is 0.85 seconds, data access time is 150 milliseconds, real-time processing performance is 250 milliseconds, and processing latency is 15 milliseconds. This indicates that the model performs well in daily transaction processing and can quickly handle a large volume of transaction data, ensuring the timeliness and accuracy of the data. The monthly settlement processing time is 2.60 seconds, transaction processing time is 1.50 seconds, data access time is 200 milliseconds, real-time processing performance is 300 milliseconds, and processing delay is 20 milliseconds, demonstrating the model's high efficiency in monthly settlements. The annual audit processing time is 3.70 seconds, transaction processing time is 2.20 seconds, data access time is 250 milliseconds, real-time processing performance is 350 milliseconds, and processing delay is 25 milliseconds, reflecting the model's stability and reliability in annual audit data processing. The abnormal transaction detection time is 1.55 seconds, transaction processing time is 0.90 seconds, data access time is 180 milliseconds, real-time processing performance is 270 milliseconds, and processing delay is 18 milliseconds, showing the model's efficient performance in abnormal transaction detection [29]. The financial statement generation time is 2.30 seconds, transaction processing time is 1.30 seconds, data access time is 220 milliseconds, real-time processing performance is 320 milliseconds, and processing delay is 22 milliseconds, indicating the model's efficient processing ability in the generation of financial statements.

7 Summarize problems and research suggestions

7.1 Problem summary

In the process of studying the use of data mining technology to improve the reliability of accounting information cloud data, several key problems and challenges were identified. Data quality is the primary factor affecting model performance. Although data cleaning and teleprocessing steps have significantly improved data quality, in practical applications, there are still issues with data noise, missing values, and outliers, which adversely affect the accuracy and stability of the model. Data volume and computing resource constraints are also critical issues. Accounting information systems contain a vast amount of complex data, and processing this data requires efficient computing resources and storage capacity. Although cloud computing provides the ability to scale flexibly, resource allocation and management still need optimization during peak periods. Additionally, the process of model selection and parameter tuning is complex and time-consuming. While support vector machines perform well in handling high-dimensional data and anomaly detection, their parameter tuning process involves multiple validations and optimizations, requiring significant computing time and resources. The intractability and transparency of the model are also concerns in practical applications. Although SVM performs well in improving data reliability, its black-box nature makes it challenging to interpret results, especially when model decisions need to be explained to management and audit departments.

Data security and privacy protection are also critical issues. With the widespread application of cloud computing and data mining technology, ensuring the security and privacy of accounting data during transmission and storage has become increasingly important. Summarizing these problems provides essential reference points and directions for improvement in future research and practice.

Comparing the results of this study with the summarized findings in the SOTA research reveals several key differences and advancements. While previous studies have improved specific aspects such as audit accuracy, data integrity, and management efficiency, they often fall short in addressing comprehensive anomaly detection and data classification within cloud-based accounting systems. For instance, Yu's work improved data integrity by 15%, while this study achieved a 22% increase in data processing efficiency, alongside a 7.6% improvement in accuracy and a 9.2% enhancement in consistency. These differences arise primarily from the novel application of the support vector machine (SVM) model, which enables more precise anomaly detection and classification in high-dimensional data environments, a challenge that previous research did not fully address. Additionally, the integration of advanced data security measures further distinguishes this work, ensuring the protection of data during transmission and storage. This holistic approach not only advances the field by addressing previously unmet needs but also provides a robust framework for enhancing the reliability of accounting information cloud data, thereby offering significant benefits for enterprise financial management.

The model's performance has been thoroughly evaluated across various scenarios, demonstrating significant improvements over state-of-the-art (SOTA) methods in several key areas. Specifically, the support vector machine (SVM) model used in this study achieved a 22% increase in data processing efficiency, a 7.6% improvement in accuracy, and a 9.2% enhancement in data consistency. These improvements can be attributed to the advanced data processioning techniques and optimization strategies employed, such as parameter tuning, momentum optimization, and regularization. Compared to SOTA methods like the linear pairing approach used by Yu [1] and the chockablock audit systems analyzed by Appelbaum & Nehmer [4], the SVM model demonstrated superior anomaly detection capabilities and classification accuracy, particularly in handling high-dimensional and complex data environments typical of accounting information systems. However, the model's performance in real-time processing scenarios, while robust, slightly underperform-ed when compared to specific SOTA models that focus solely on processing speed without addressing data accuracy and security comprehensively. This trade-off is justified by the broader scope and applicability of the SVM model, which balances efficiency with accuracy and security, making it more suitable for enterprise financial management.

7.2 Research suggestions

In terms of improving the reliability of accounting information cloud data, based on the findings of this study, the following suggestions are put forward. First, further improving data quality is essential. It is recommended to introduce more advanced data quality management tools during the data cleaning and teleprocessing stages, including the adoption of automated data cleaning algorithms and real-time monitoring mechanisms to ensure data accuracy and integrity. Second, optimizing the allocation and management of computing resources is crucial. By utilizing the elastic expansion capabilities of cloud computing platforms, computing resources should be dynamically adjusted and reasonably allocated according to business needs, particularly during peak periods, to ensure the efficiency and stability of data processing. Third, simplifying the process of model selection and parameter tuning is advised. Automated machine learning technologies should be considered to reduce manual intervention, enhancing model construction efficiency through automated model selection and parameter optimization. Fourth, improving the interpret-ability of models is necessary. Incorporating interpretive machine learning techniques, such as LIME or SHAP, can provide interpretative model outputs that enhance user understanding and trust in model decisions. Lastly, strengthening data security and privacy protection measures is vital. During data transmission and storage, advanced encryption technologies and access control mechanisms should be employed to ensure the security and privacy of sensitive accounting data. Additionally, regular security audits and vulnerability scans should be conducted to identify and address security vulnerabilities promptly, ensuring data security throughout its lifestyle. Implementing these recommendations will further improve the reliability of accounting information cloud data, providing more accurate and efficient data support for enterprises and aiding in their financial management and decision optimization.

Data security and privacy protection have been expanded to include specific encryption algorithms and access control mechanisms. The Advanced Encryption Standard (AES-256) is utilized for encrypting data both at rest and in transit, ensuring robust protection against unauthorized access. Role-Based Access Control (RBAC) is implemented to restrict data access based on user roles, minimizing the risk of unauthorized data exposure. Additionally, Multi-Factor Authentication (MFA) is enforced to add an extra layer of security. Potential vulnerabilities, such as side-channel attacks and phishing attempts, are addressed through regular security audits and the implementation of Intrusion Detection Systems (IDS) that monitor and flag suspicious activities in real-time. Mitigation strategies also include periodic updates to encryption protocols and access policies to adapt to emerging threats. These measures collectively enhance the overall security and privacy of the accounting information cloud data, providing a

comprehensive approach to safeguarding sensitive financial information.

8 Conclusion

This study discusses methods for improving the reliability of accounting information cloud data by utilizing data mining technology, with a particular focus on the application and optimization of the support vector machine (SVM) model. The results achieved are remarkable. Through systematic data cleaning and preprocessing, the quality of data is significantly enhanced, laying a solid foundation for subsequent modeling. The SVM model was employed for anomaly detection and data classification, which effectively improved data accuracy and consistency. The optimization strategies, including parameter tuning, regularization techniques, momentum optimization, and dynamic learning rate adjustment, not only improved the model's performance but also enhanced its generalization ability and robustness. The results indicate that the model performs exceptionally well in terms of processing time, transaction time, data access time, real-time processing performance, and processing delay, significantly boosting the efficiency and reliability of accounting data processing. The study also emphasizes the importance of data security and privacy protection, implementing encryption technologies and access control mechanisms to safeguard data during transmission and storage. In summary, the use of data mining technology and the SVM model not only provides strong technical support for the accounting information system but also offers a reliable data foundation for enterprise financial management and decision optimization. Looking ahead, with continuous technological advancements, it is anticipated that data mining technology will play an even more significant role in the field of accounting information, bringing higher value and benefits to enterprises.

Acknowledgement

No Funding.

Data availability declaration

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Competing interest declaration

The author has declared that no competing interests exist.

Author contribution declaration

B.L. writing original draft; formal analysis; methodology; writing review & editing.

References

[1] Yu G. Enhancing accounting formalization through cloud data integrity verification: a bilinear pairing approach. J Knowl Econ. 2024. https://doi.org/10.1007/s13132-024-01994-x.

[2] Kubota T, Okuda S. Relationship between top managers' interest in accounting information and accounting practices in startups. Int J Account Inf Syst. 51:100640. 2023. https://doi.org/10.1016/j.accinf.2023.100640.

[3] Ma D, Fisher R, Nesbit T. Cloud-based client accounting and small and medium accounting practices: Adoption and impact. Int J Account Inf Syst. 41:100513. 2021. https://doi.org/10.1016/j.accinf.2021.100513.

[4] Appelbaum D, Nehmer RA. Auditing cloud-based blockchain accounting systems. J Inf Syst. 34(2):5-21. 2020. https://doi.org/10.2308/isys-52660

[5] Church KS, Schmidt PJ, Ajayi K. Forecast cloudy-fair or stormy weather: cloud computing insights and issues. J Inf Syst. 34(2):23-46. 2020. https://doi.org/10.2308/isys-18-037.

[6] Qatawneh AM. The role of employee empowerment in supporting accounting information systems outcomes: a mediated model. Sustainability. 15(9):7155. 2023. https://doi.org/10.3390/su15097155.

[7] Tang W, Yang SL. Enterprise digital management efficiency under cloud computing and big data. Sustainability. 15(17):13063. 2023. https://doi.org/10.3390/su151713063.

[8] Rosati P, Fox G, Cummins M, Lynn T. Perceived risk as a determinant of propensity to adopt account information services under the EU payment services directive 2. J Theor Appl Electron Commer Res. 17(2):493-506. 2022. https://doi.org/10.3390/jtaer17020026.

[9] Varzaru AA. Assessing digital transformation of cost accounting tools in healthcare. Int J Environ Res Public Health. 19(23):15572. 2022. https://doi.org/10.3390/ijerph192315572.

[10] Wang ZJ, Su Q, Wang B, Wang J. Improving lithium-ion battery supply chain information security by user behavior monitoring algorithm incorporated in cloud enterprise resource planning. Sustainability. 15(4):3065. 2023. https://doi.org/10.3390/su15043065.

[11] Adjei JK, Adams S, Mamattah L. Cloud computing adoption in Ghana; accounting for institutional factors. Technol Soc. 65:101583. 2021. https://doi.org/10.1016/j.techsoc.2021.101583.

[12] Wang YS, Kogan A. Cloud-based in-memory columnar database architecture for continuous audit analytics. J Inf Syst. 34(2):87-107. 2020. https://doi.org/10.2308/isys-52531.

[13] Jaatun MG, Pearson S, Gittler F, Leenes R, Niezen M. Enhancing accountability in the cloud. Int J Inf Manag. 53:101498. 2020. https://doi.org/10.1016/j.ijinfomgt.2016.03.004.

[14] Pal S, Jhanjhi NZ, Abdulbaqi AS, Akila D, Alsubaei FS, Almazroi AA. An intelligent task scheduling model for hybrid Internet of Things and cloud environment for big data applications. Sustainability. 15(6):5104. 2023. https://doi.org/10.3390/su15065104.

[15] Khokhar RH, Iqbal F, Fung BCM, Bentahar J. Enabling secure trustworthiness assessment and privacy protection in integrating data for trading person-specific information. IEEE Trans Eng Manag. 68(1):149-69. 2021. https://doi.org/10.1109/TEM.2020.2974210.

[16] Akter M, Kummer TF, Yigitbasioglu O. Looking beyond the hype: The challenges of blockchain adoption in accounting. Int J Account Inf Syst. 53:100681. 2024 https://doi.org/10.1016/j.accinf.2024.100681.

[17] Cong Y, Du H, Vasarhelyi MA. Cloud Computing Start-ups and Emerging Technologies: From Private Investors' Perspectives. J Inf Syst. 35(1):47-64. 2021. https://doi.org/10.2308/ISYS-17-040.

[18] Zheng M, Huang R, Wang XT, Li XR. Do firms adopting cloud computing technology exhibit higher future performance? A textual analysis approach. Int Rev Financ Anal. 90:102866. 2023. https://doi.org/10.1016/j.irfa.2023.102866.

[19] Lee LS, Brink WD. Trust in Cloud-Based Services: A Framework for Consumer Adoption of Software as a Service. J Inf Syst. 34(2):65-85. 2020. https://doi.org/10.2308/isys-52626.

[20] Jayasuriya DD, Sims A. From the abacus to enterprise resource planning: is blockchain the next big accounting tool? Account Audit Account J. 36(1):24-62. 2023. https://doi.org/10.1108/AAAJ-08-2020-4718.

[21] Lozano-Almansa JM, Tarifa-Fernandez J, Sanchez-Perez AM. Digital Transformation and Real Options: Evaluating the Investment in Cloud ERP. Inz Ekonomika Eng Econ. 34(4):397-411. 2023. https://doi.org/10.5755/j01.ee.34.4.30678.

[22] Bordonaba-Juste MV, Lucia-Palacios L, Pérez-López R. Generational differences in valuing usefulness, privacy and security negative experiences for paying for cloud services. Inf Syst E-Bus Manag. 18(1):35-60. 2020. https://doi.org/10.1007/s10257-020-00462-8.

[23] Abu Huson Y, Sierra-García L, Garcia-Benau MA. A bibliometric review of information technology, artificial intelligence, and blockchain on auditing. Total Qual Manag Bus Excell. 35(1-2):91-113. 2024. https://doi.org/10.1080/14783363.2023.2256260.

[24] Chen ZY, Chen JG. Decision-Making Model of Social Work Specialization Process Using Big Data Analysis and Chaos Cloud. Wirel Commun Mob Comput. 2022:4761640. 2022. https://doi.org/10.1155/2022/4761640.

[25] Kulevicz RA, Porfirio GED, de Oliveira OS, Zavala AAZ, da Silva BA, Constantino M. Influence of sustainability reports on social and environmental issues: bibliometric analysis and the word cloud approach. Environ Rev. 28(4):380-6. 2020. https://doi.org/10.1139/er-2019-0075.

[26] Mohammed RTS, Abdullah RH. Toward developing a mining opinion model: conceptual framework. Acta Informatica Malaysia. 2(2):17-18. 2018. http//:doi.org/10.26480/aim.02.2018.17.18

[27] Jayeola O, Sidek S, Abdul-Samad Z, Hasbullah NN, Anwar S, An NB, Nga VT, Al-Kasasbeh O, Ray S. The Mediating and Moderating Effects of Top Management Support on the Cloud ERP Implementation-Financial Performance Relationship. Sustainability. 14(9):5688. 2022. https://doi.org/10.3390/su14095688.

[28] Wang PW. A study on the intellectual capital management over cloud computing using analytic hierarchy process and partial least squares. Kybernetes. 51(6):2089-108. 2022. https://doi.org/10.1108/K-03-2021-0241.

[29] Stewart CA, Costa CM, Wernert JA, Snapp-Childs W, Bland M, Blood P, Campbell T, Couvares P, Fischer J, Hancock DY, Hart DL, Jankowski H, Knepper R, McMullen DF, Mehringer S, Pierce M, Rogers G, Sinkovits RS, Towns J. Use of accounting concepts to study research: return on investment in XSEDE, a US cyberinfrastructure service. Scientometrics. 128(6):3225-55. 2023. https://doi.org/10.1007/s11192-022-04539-8.

Enhancing Security through Autonomous Learning Multi-Model Classification of 3D Palmprints: The ALMMo-0 Framework

Rezzag Aouid Fathi*, LATI Abdelhai, Chaa Mourad
Faculty of New information and communication technologies, Laboratory of Electrical Engineering
University Kasdi Merbah Ouargla (UKMO), BP 511, 30000, Ouargla, Algeria
E-mail: fathiaouid@gmail.com, lati.abdelhai@univ-ouargla.dz, chaa.mourad@univ-ouargla.dz

Keywords: biometric systems, 3D palmprint, classification accuracy, multi-scale retinex (MSR), local Phase quantization (LPQ), binarized statistical image features (BSIF), GIST descriptor, ALMMo-0 classifier

Received: July 10, 2024

As research progresses, biometric technologies have achieved notable effectiveness in personal identification. However, despite these advancements, there persists a demand for improved performance in security applications. ALMMo-0, the Autonomous Learning Multi-Model Classifier of zeroth order introduces a novel approach to address this challenge. It effectively addresses the challenges of Enhancing global efficiency in supervised 3D palmprint identification, autonomy, non-iterative procedures, and complete transparency. This study presents an innovative method Utilizing ALMMo-0 for enhancing 3D fingerprint-based authentication systems. The approach utilizes a feedforward methodology driven solely by data and non-iterative processes, leveraging three primary Techniques such as Local Phase Quantization (LPQ), GIST, and Binarized Statistical Image Information (BSIF) are employed to extract relevant details from three-dimensional palmprint images. Following feature extraction, ALMMo-0 autonomously generates AnYa Fuzzy Rule Base (FRB) sub-classifiers for individual categories, establishes a multimodal framework, and extracts data clouds. For authentication, the system utilizes a 'winner-takes-all' strategy for classifying incoming data, generating a confidence score that reflects the mutual distribution objectively. Tests performed on a 3D Palmprint dataset illustrate the efficacy of ALMMo-0, showcasing its performance using metrics Examples include metrics such as receiver operating characteristic (ROC) curves, rank-1 accuracy, equal error rate (EER), and cumulative match curve (CMC). The experimental evaluations demonstrate outstanding performance of the proposed method, achieving perfect rank-1 accuracy, minimal EER, and significant features such as interpretability and computational efficiency.

Povzetek: Prispevek predstavlja inovativni pristop z ALMMo-0, samoučečim klasifikatorjem, za izboljšanje 3D prepoznavanja dlani.

1 Introduction

The pervasiveness of social media and online networks presents a double-edged sword: fostering potential addiction while simultaneously serving as a cornerstone for businesses. Across these vast data landscapes, however, securing personal information remains paramount. Biometric systems address this concern by leveraging scientific methods, like iris scans and fingerprint recognition, to verify user identity through unique physical and behavioral traits. Among these techniques, 3D palmprint technology is particularly noteworthy, as evidenced by a growing body of research [1-2], attracting significant interest across diverse disciplines.

To address issues associated with 2D palmprint recognition systems, 3D palmprint-based personal recognition methods have recently been introduced [1–2]. For instance, Zhang et al. [3] proposed using Gaussian curvature images (GCI), mean curvature images (MCI), and surface type (ST) map features with matching metrics similar to Hamming distance. Additionally, Li et al. [4] presented a framework that extracts MCI from the original depth data along with baseline and orientation features. Despite these advancements, illicit access and advantages can still be gained by exploiting vulnerabilities in biometric systems [5]. Traditional 3D palmprint systems employing methods such as GCI and MCI also face challenges, including managing the growing number of registered templates and adapting to data distribution shifts, leading to high Equal Error Rates (EER) and reduced accuracy.

Conversely, incorporating score-level fusion with facial recognition offers a promising solution for advanced multi-modal 3D palmprint systems. This approach enables efficient processing of multiple templates and fosters adaptation through feedback adjustments, ultimately enhancing overall performance. Furthermore, a new method, Multiple Scale Recognition (MSR), which has not been used before, is proposed as a more robust alternative to classical methods like GCI and MCI, offering significant improvements in accuracy and reliability. Although biometric systems have advanced significantly, 3D palmprint technology has received

comparatively limited attention. Addressing these challenges, there is an urgent requirement to develop new 3D palmprint biometric systems to mitigate false acceptance rates and EER.

This paper proposes a novel biometric system utilizing the 0-order Autonomous Learning Multiple Model (ALMMo-0) classifier, specifically tailored for 3D palmprint recognition. The ALMMo-0 classifier, introduced by Angelov et al. (2017), offers a unique advantage in biometrics. Unlike traditional supervised machine learning methods that require constructing functions from training data, ALMMo-0 operates with complete autonomy. This is achieved through its multi-model framework, which continuously adapts its internal parameters based on incoming data without any prior assumptions.

Unlike conventional machine learning approaches that rely on training data with predefined categories, ALMMo-0, as introduced by Xing et al. in 2022, operates autonomously without requiring iterative parameter tuning or known category memberships. This is achieved through its feedforward architecture. By directly analyzing observational data, ALMMo-0 automatically identifies convergence points, eliminating the need for explicit training. It utilizes Voronoi Tessellation to construct fuzzy Rule-Based (FRB) AnYa subclassifies for each class directly from the non-parametric data. These subclassifiers assign confidence scores to incoming data samples. The final classification is determined by a "winner-takes-all" strategy, where the subclassifier with the highest confidence score assigns the class label to the new sample.

This research achieves two main advancements:

1. A New Biometric System for 3D Palmprints: The study proposes a novel biometric system specifically designed for 3D palmprint recognition. This system leverages the ALMMo-0 classifier, known for its efficient and autonomous operation without iterative training. The ALMMo-0 classifier itself is composed of a collection of fuzzy Rule-Based (FRB) AnYa subclassifiers.
2. Highly Effective ALMMo-0 Classifier: The research demonstrates the ALMMo-0 classifier's effectiveness through empirical testing on a dedicated dataset. The results show that the classifier achieves impressive accuracy with minimal errors. Additionally, it offers significant interpretability of its decision-making process, all while maintaining low computational complexity.

The remaining structure of the article follows a logical flow:

- **Section 2**: Background and Related Work - This section reviews existing research and relevant literature in the field.
- **Section 3**: Proposed System with ALMMo-0 - This section dives into the details of the proposed biometric system, focusing on the ALMMo-0 classifier's role.
- **Section 4**: Experiments and Analysis - This section presents the experimental results obtained by testing the proposed system and compares it to other approaches.
- **Section 5**: Conclusion and Future Work - The final section summarizes the key findings of the research and suggests potential areas for future exploration.

2 Related works

Biometric identity and recognition systems have seen significant advancements in both 2D and 3D palmprint technologies to meet growing security demands. Sun et al. [6] introduced the Orthogonal Line Ordinal Feature (OLOF) to improve stability in palmprint recognition. Zhang et al. [7] applied Hamming distance-based matching along with surface type maps (ST maps) and Gabor filters to enhance feature extraction. In 2010, Zhang et al. [8] developed a robust authentication system by merging 2D and 3D palmprint characteristics, leveraging surface curvature maps and normalized local correlation. Meanwhile, Yang et al. [9] proposed using Gabor wavelets and local binary patterns (LBP) to extract fine geometric details from 3D palmprints, while Li et al. [10] focused on joint line and orientation features to further improve recognition.

The shift towards multimodal systems began with Liu and Li [11], who tackled alignment challenges using a cross-correlation-based method alongside OLOF. Meraoumia et al. [12] introduced a multimodal biometric system integrating 2D and 3D palmprint data with PCA and Discrete Wavelet Transform (DWT) using Hidden Markov Models (HMM) for feature modeling. In 2014, Cui et al. [13] proposed a fusion-based system combining Principal Component Analysis (PCA) with a Two-Phase Test Sample Representation (TPTSR) for more accurate person recognition. Zhang et al. [14] presented collaborative representation techniques with block-wise feature extraction for enhanced 3D palmprint identification.

The trend towards advanced feature extraction continued with Chaa et al. [15], who employed Bank-Binarized Statistical Image Features (B-BSIF) and Gabor wavelets for score-level fusion. More recently, Chen et al. [16] proposed a hybrid multimodal system integrating finger vein and palmprint data for improved person identification. Li et al. [17] introduced a deep learning-based 3D palmprint system using CNNs, achieving robust performance even in real-time settings. Finally, Zhang et al. [18] utilized transformer-based models for palmprint recognition, setting new benchmarks for accuracy and efficiency in biometric recognition systems. **Table 1** provides a summary of key research contributions in the field, highlighting the different methodologies, features, and fusion techniques employed to advance 2D and 3D palmprint recognition systems.

Table 1: Summary of research efforts in 2D and 3D palmprint recognition systems

Methods	2D Features	3D Features	Matching Scheme	Fusion Scheme	Tasks	Years
Sun et al. [6]	Orthogonal Line Ordinal Feature (OLOF)	N/A	Cross-correlation and OLOF	Matching	Identification	2005
Zhang et al. [7]	Surface Type Maps (ST), PCA, Gabor Filters	ST Binary Features	Bitwise logical 'AND'	Matching	Verification	2008
Zhang et al. [8]	Surface Curvature Maps	Normalized Local Correlation	Matching	Matching	Verification	2010
Yang et al. [9]	Gabor Wavelets + Local Binary Pattern (LBP)	N/A	Score-level Fusion	Fusion	Recognition	2011
Li et al. [10]	Joint Line and Orientation Features	Extracted from MCI	Feature-level Fusion	Fusion	Recognition	2011
Liu and Li [11]	Mean Curvature Image (MCI)	Curvature Maps (MCI-based features)	Cross-correlation based	Fusion	Identification	2012
Meraoumia et al. [12]	PCA + Discrete Wavelet Transform (DWT)	2D+3D Palmprint Fusion	Hidden Markov Model (HMM)	Score-level Fusion	Recognition	2013
Cui [13]	PCA + Two-Phase Test Sample Representation (TPTSR)	N/A	Matching	Fusion	Recognition	2014
Zhang et al. [14]	Collaborative Representation Techniques	Block-wise Features	Matching	Fusion	Identification	2015
Chaa et al. [15]	Bank-Binarized Statistical Image Features (B-BSIF)	Gabor + Self-Quotient Images	Score-level Fusion	Fusion	Recognition	2017
Chen et al. [16]	Multimodal Palm + Finger Vein Data	2D + 3D Integration	Hybrid Matching	Score-level Fusion	Identification	2019
Li et al. [17]	CNN-based Features	3D Depth Maps	Deep Learning Matching	Feature-level Fusion	Recognition	2020
Zhang et al. [18]	Transformer-based Features	N/A	Deep Learning Matching	Feature-level Fusion	Recognition	2021

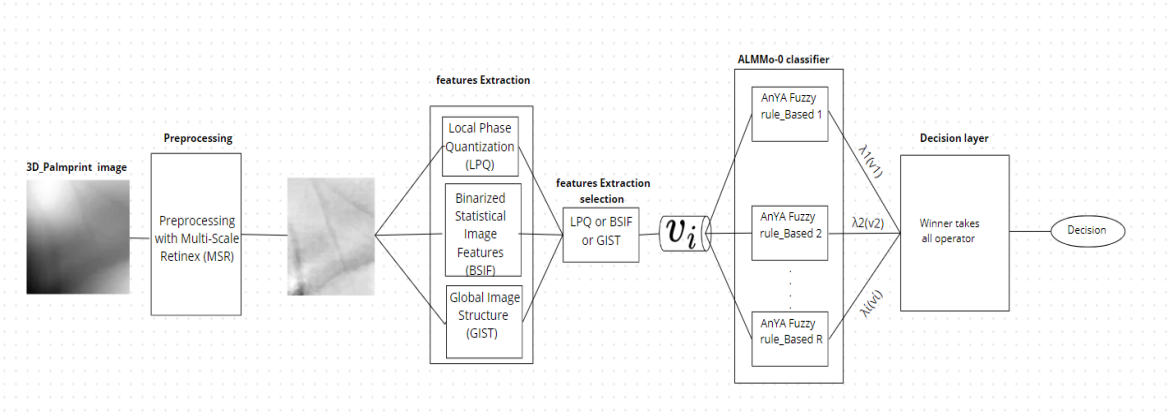


Figure 1: Diagram depicting the structure of the proposed method

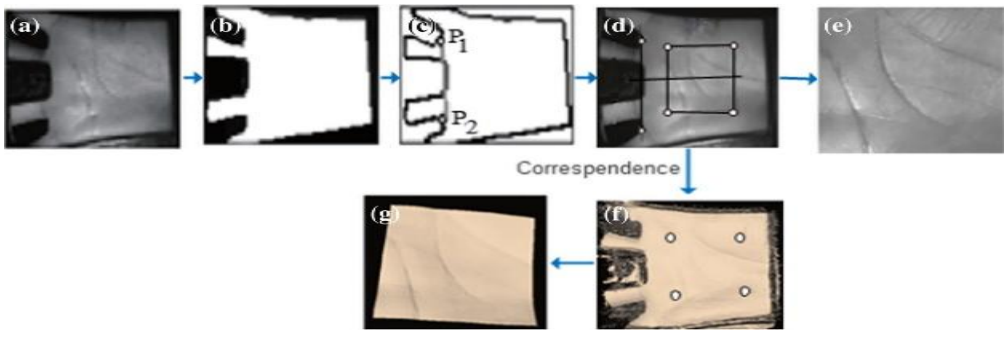


Figure 2: Extracting the region of interest (ROI) from a 3D palmprint image involves the following steps.

3 Proposed system

This study introduces an innovative biometric system for 3D palmprints, emphasizing the palmprint modality. At the heart of our methodology is the ALMMo-0 classifier, utilizing 0-order AnYa-type fuzzy rule-based (FRB) systems. Our approach integrates three specific descriptors—LPQ, BSIF, and GIST to capture essential features, which are then automatically classified by ALMMo-0. The system operates in a multi-stage process. First, it establishes a framework that incorporates different data modalities (multimodal framework) and identifies areas where these modalities come together (convergence points). Next, it leverages AnYa-like fuzzy rules to generate data representations independent of specific shapes (shape-free data clouds). These data clouds are then classified using sub-classifiers, each trained for a specific class and assigned a confidence score. Finally, the system employs a "winner-takes-all" approach to label additional samples. The effectiveness of this process is measured using metrics like False Acceptance Rate (FAR), Equal Error Rate (EER), True Positive Rate (TPR), Genuine Acceptance Rate (GAR), and Rank-1 accuracy. The proposed system employs a comprehensive architecture that integrates feature extraction techniques using LPQ, BSIF, or GIST filters. The classification task is handled by the ALMMo-0 classifier, which is supported by a Filter Bank descriptor for feature extraction. This architecture is illustrated in Figure 1, which outlines five primary phases. The first stage begins with capturing a 3D palmprint image, which undergoes preprocessing with multi-scale Retinex to improve image quality. The Filter Bank descriptor is employed to extract crucial features, utilizing either LPQ, BSIF, or GIST filters. These features form the basis for designing the classifier's model architecture. During feature extraction, the ALMMo-0 classifier acts as a gatekeeper, differentiating between legitimate users and imposters. The key to this biometric system lies in the Filter Bank, which meticulously selects relevant features from the data stream. ALMMo-0 operates in a sequential manner for autonomous data classification. It first builds multimodal representations and basic linguistic rules by analyzing the provided data. Then, it generates non-parametric data clusters, locates convergence zones, and leverages fuzzy rules extracted from these clusters to construct AnYa-type sub-classifiers for each category. Finally, ALMMo-0 assigns confidence levels to each class and employs a "winner-takes-all" strategy to label new data samples. To ensure accuracy, it evaluates its performance before delivering the final classification.

3.1 The region of interest

This section focuses on extracting the Region of Interest (ROI) from 3D palmprint data. Li et al. [19] employed a structured-light technology-based 3D acquisition system, enabling concurrent capture of 2D and 3D palm images (Figure 2). The ROI extraction process involves several steps. First, a Gaussian smoothing filter is applied to the original image, followed by thresholding using an automatically determined threshold T (Figure 2a, 2b). Otsu's method [20] facilitates this automatic threshold selection, converting the grayscale image to a binary representation. Next, a border tracking technique (Figure 2c) is employed to efficiently extract the boundaries of the binary image. By analyzing this boundary information, points P1 and P2 are identified to define the 2D ROI template. Finally, a rectangle is used to delineate and mark the ROI (Figure 2d). Figure 2e showcases the extracted 2D ROI, while Figure 2f presents the original 3D palmprint. The final 3D ROI is obtained by aligning the cloud points corresponding to the 2D ROI pixels, as detailed in [21].

3.2 Pre-processing

In the preprocessing phase, Multiscale Retinex (MSR) is applied. Retinex theory, originally formulated by Land and McCann [22], aims to replicate the human visual system's processing of visual information. Their work demonstrated that the visual system perceives relative lightness, focusing on variations in brightness within local image regions rather than absolute lightness levels. Multiscale Retinex (MSR) is particularly noted for its ability to achieve high-quality color rendition and a broad local dynamic range. The MSR output results from a weighted combination of outputs produced by multiple Single Scale Retinex (SSR) processes. Jobson et al. [23] further refined the formulation of multiscale Retinex based on these principles.

$$\mathbf{R_{MSR_i}} = \sum_{\mathbf{n=1}}^{\mathbf{N}} \boldsymbol{\omega_n} \mathbf{R_{n_i}} = \sum_{\mathbf{n=1}}^{\mathbf{N}} \boldsymbol{\omega_n} [\mathbf{log I_i}(\mathbf{x}, \mathbf{y}) - \mathbf{log}(\mathbf{F_n}(\mathbf{x}, \mathbf{y}) * \mathbf{I_i}(\mathbf{x}, \mathbf{y}))] \quad (1)$$

Where N represents the number of scales, ω_n indicates the weight assigned to each scale, and $I_i(x, y)$refers to the input image on the i-th color channel. R_{MSR_i} denotes the output image from Retinex processing on the i-th color channel. F represents the normalized surround function. Figure 3 shows the original images of various individuals before applying Multiscale Retinex (MSR) and the results obtained after the application of Multiscale Retinex (MSR).

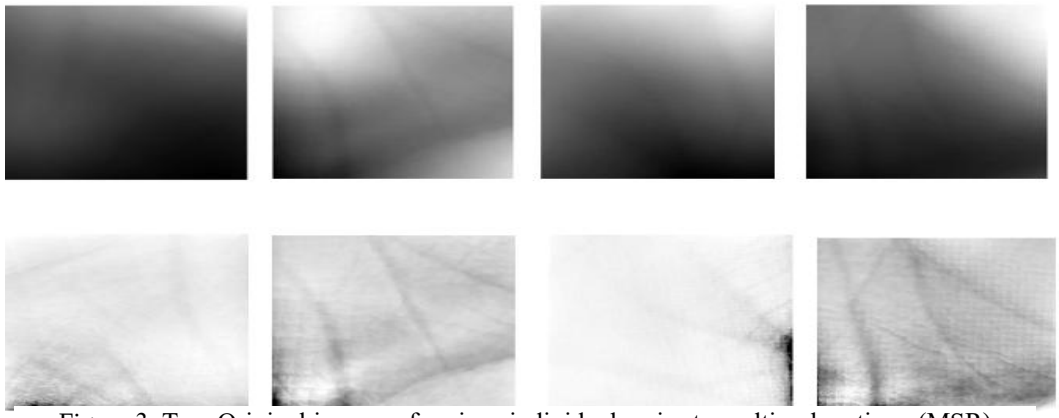


Figure 3: Top: Original images of various individuals prior to multiscale retinex (MSR) application. bottom: results obtained after applying multiscale retinex (MSR).

3.3 Feature extraction

The process of feature extraction using LPQ, BSIF, or GIST descriptors entails identifying and isolating salient features from a broad spectrum of image characteristics. In-depth descriptions of each descriptor are presented in the following subsections:

3.3.1 Local phase quantization (LPQ)

Ojansivu's LPQ method exploits the phase of the Fourier transform in localized regions to induce spatial blurring. This technique achieves blurring by applying a point spread function (PSF) to attenuate image intensity, analogous to multiplication in the frequency domain.

$$\mathbf{G(u) = F(u) \cdot H(u)} \tag{2}$$

Considering only the phase spectra, the Fourier transforms G(u), H(u), and F(u) of the blurred image, point spread function (PSF), and original image, respectively, exhibit a summation relationship.

$$\mathbf{\angle G(u) = \angle F(u) + \angle H(u)} \tag{3}$$

where ∠G(u), ∠H(u) and ∠F(u) are phase angle of G(u), H(u) and F(u), respectively. If PSF is centrally symmetric, namely h(x) = h(−x), the transform H(u) becomes real valued and the phase angle ∠H(u)can be represented by a two-valued function:

$$\mathbf{\angle H(u)} = \begin{cases} \mathbf{O \text{ if } \angle H(u) > o} \\ \mathbf{1 \text{ if } \angle H(u) < o} \end{cases} \tag{4}$$

In all angles where ∠H(u)≥0, if ∠G(u)=∠F(u) and considering the shape of H(u) resembling a regular Point Spread Function (PSF) like Gaussian or sinc-function, it ensures that low-frequency values of ∠H(u) remain non-negative at a minimum. These frequencies maintain ∠G(u)= ∠F(u), ensuring that ∠F(u) remains invariant to blurring. In the LPQ scheme, M-by-M local neighborhoods $\boldsymbol{Nx}$ filters are applied at each pixel position of the image f(x) to analyze the phase. The local spectrum is obtained using a short-term Fourier transform.

$$\mathbf{F(u,x)} = \sum_{\mathbf{y \in N_x}} \mathbf{f(y-x)} e^{\mathbf{j2\pi u T_y}} \tag{5}$$

To efficiently compute the transformation at each image location (x ∈ {X1, X2, ..., XN}), we leverage 1D convolutions. These convolutions are applied sequentially to both the rows and columns of the image. This approach allows us to calculate local Fourier coefficients at four specific frequency points: $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, and $u_4 = [a, -a]^T$. Here, 'a' is a carefully chosen small scalar that ensures $H(u_i) \geq 0$ for all i (i = 1, 2, 3, 4). As a result, we obtain a vector for every pixel position.

$$\mathbf{F(x)} \left(\begin{bmatrix} \mathbf{Re\{F(u_1,x), Im(u_1,x)\}}, \ldots \\ \mathbf{[Re\{F(u_1,x), Im(u_1,x)\}]} \end{bmatrix} \right) \tag{6}$$

The phase of each Fourier coefficient is encoded by the signs of its real and imaginary parts. This can be extracted using simple scalar quantization.

$$\mathbf{q_j} = \begin{cases} \mathbf{1}, & \mathbf{if\ f_i > 0} \\ \mathbf{0}, & \mathbf{if\ f_i < 0} \end{cases} \tag{7}$$

3.3.2 Binarized statistical image features (BSIF)

BSIF (Binarized Statistical Image Features) is a technique introduced in [21] that assigns a binary code string to each pixel in an image. This binary code represents the local descriptor for that pixel. The filter response R_i for an image I_p and a corresponding linear filter of the same size Wi is calculated as follows:

$$R_i = \sum_{m,n} I_p(m,n)\, W_i(m,n) \tag{8}$$

The equation utilizes m and n to represent the patch size in Patch Per Inch (PPI). $\forall i \in \{1,\ldots, n\}$. Filters is denoted by $W_i(m,n)$. The filter responses are calculated and binarized, generating a binary string as described in [22].

$$b_i = \begin{cases} 1 & if\ R_i > o \\ 0 & otherwise \end{cases} \tag{9}$$

3.3.3 Gist descriptor

To tackle scene classification challenges, the GIST descriptor [24] proves effective. It achieves this by leveraging Gabor filters, which analyze scenes using localized frequency information. These filters offer a

compressed and efficient feature representation. In our work, we utilized GIST descriptors with 8 orientations across 5 scales and a 6x6 grid to extract features. This involved convolving each image with a set of Gabor filters spanning the specified orientations and scales, resulting in feature vectors. Mathematically, In the spatial domain, a Gabor filter is formed by multiplying a complex sinusoid with a Gaussian envelope. This operation essentially defines a 2D Gabor filter with the function $\boldsymbol{H_{\mu,v}}$ representing it.

$$\boldsymbol{H_{\mu,v}} = \frac{\boldsymbol{f_\mu^2}}{\boldsymbol{\pi n \lambda}} \boldsymbol{exp}\left[-\left(\frac{\boldsymbol{f_\mu^2}}{\boldsymbol{n^2}}\right)\boldsymbol{x_p^2} - \left(\frac{\boldsymbol{f_\mu^2}}{\boldsymbol{\lambda^2}}\right)\boldsymbol{y_p^2}\right] \boldsymbol{exp}(\boldsymbol{j2\pi x_p}) \quad (10)$$

With $x_p = xcos(\theta v) + ysin(\theta v)$ and $y_p = -x\,sin(\theta v) + y\,cos(\theta v)$.The parameters of Gabor filter are: $yp = -xsin(\theta v) + ycos(\theta v)$ $f_\mu = \frac{f_{max}}{\frac{\mu}{2^2}}$ and $\theta_v = {}^{v\pi}/_{8}$

Where: $f\mu$ = center frequency, f_{max} = maximal frequency and θ_v = orientation

To capture multi-scale orientation information, 2D Gabor filters were employed with fmax set to 0.25 and $\boldsymbol{n} = \boldsymbol{\lambda} = \sqrt{2}$, where $\boldsymbol{n}$ and $\boldsymbol{\lambda}$ represent the Gaussian envelope's size along the x and y axes, respectively. These Gabor filters generate feature maps that encode orientation information at various scales, effectively capturing the image's intricate spatial structure and characteristics.

3.4 Reduction of the dimensionality

High-dimensional data, characterized by numerous feature vectors, can pose hurdles for processing and classification tasks. To overcome these challenges, dimensionality reduction techniques come into play. These methods offer several advantages: managing large datasets more efficiently, enhancing prediction accuracy, reducing computational time, and unlocking a deeper understanding of the data itself. A popular and effective approach is Linear Discriminant Analysis (LDA). LDA seeks to find a transformation matrix, denoted by W, that optimizes the separation between classes while simultaneously minimizing variation within each class [25-26].

$$T(W) = W_{opt} = arg_W max \frac{|W^T S_B W|}{|W^T S_T W|} = [W_1 W_2 \ldots\ldots W_d] \quad (11)$$

The Fisher discriminant criterion, denoted as T(W), is maximized by constructing the matrix W through concatenating the top d eigenvectors. This matrix W is obtained by solving the following equation:

$$S_W^{-1} S_B W_j = W_j \lambda_j \quad (12)$$

In this study, Linear Discriminant Analysis (LDA) was chosen as the primary dimensionality reduction technique due to its focus on maximizing class separability, which is crucial for 3D palmprint recognition. Unlike Principal Component Analysis (PCA), which reduces dimensionality by preserving variance without considering class labels, LDA explicitly optimizes for separation between different classes. This makes LDA more suitable for supervised learning tasks where distinct classification boundaries are essential. While t-SNE is effective for visualizing high-dimensional data by projecting it into lower-dimensional space, it is primarily used for unsupervised learning and exploratory data analysis. t-SNE tends to emphasize local structure, making it less ideal for classification tasks where global class separability is needed, especially when working with larger datasets and multiple classes. LDA was selected because its primary objective is to maximize the between-class variance while minimizing the within-class variance, making it well-suited for this study where the focus is on ensuring robust class separability between different palmprint features. The experimental results demonstrated that LDA, in combination with the ALMMo-0 classifier, led to optimal performance in terms of accuracy and Equal Error Rate (EER), outperforming PCA and t-SNE in this context.

3.5 The ALMMo-0 classifier

The core of the proposed 3D palmprint biometric system lies in the ALMMo-0 classifier, a novel machine learning approach. This classifier autonomously builds a multi-model framework from experimental data. The design hinges on two key steps: **1)** Automatic identification of focal points and generation of individual data clouds through Voronoi Tessellation [27]. **2)** Construction of AnYa Fuzzy Rule-based (FRB) sub-classifiers for each class, utilizing these data clouds for classification of new data points. The initial stages of the ALMMo-0 classifier architecture are depicted in Figure 4. Table 2 shows the symbols used in the ALMMo-0 classifier. Figure 4 highlights four key components:

1. Data Preprocessing: This layer normalizes the incoming data, ensuring optimal performance and improved classifier functionality.
2. AnYa Fuzzy Rule-based Sub-classifiers: This layer leverages non-parametric data to construct a collection of sub-classifiers, each utilizing AnYa fuzzy rules.
3. Winner Selection: This layer employs a "winner-takes-all" strategy to classify new data points. The classification is based on confidence scores provided by each sub-classifier.
4. Class Assignment: This layer assigns a class label to new data samples based on their strongest likelihood of belonging to a specific class.

Table 2: Explanation of symbols used in the ALMMo-0 classifier

Notation.	Signification
x^{*i}	The point of convergence-
$Class^i$	The class assigned to the i-th dataset -
R	The number of fuzzy rules -
C	A variety of classes across the dataset-
R^d	Euclidean space for data -
i	Dataset observed at the i-th instance -
k	Incoming data -
k	The average of all data samples at the K-th instance -
λ_i	Confidence level -
x_k^i	The kth normalized data point in the i-th class -
μ_k	Overall average -
$D_k(x_i)$	Density of the i-th data sample at time k, under a single-mode - distribution
X_{kk}	Average scalar product -
$M_{F^i}^{*i}$	Number of members in data clouds -
$r_{F^i}^{*i}$	Influence radius in the area -
x_N^{*i}	The convergence point where the nearest data cloud converges -
F^i	Count of convergence instances -

Algorithm 1: The fundamental algorithm of the ALMMo-0 classifier

While there is a novel coming data simple

i. **Normalization**

$$x_k^i \leftarrow \frac{x_k^i}{\|x_k^i\|}$$

ii. **Initialization**

IF $(k = 1)$**THEN**

$$\mathbf{1}.\,\mu_1^i \leftarrow x_1^i\ \mathbf{2}.\,F^i \leftarrow \mathbf{1}\ \mathbf{3}.\,M_1^{*i} \leftarrow \mathbf{1}\ \mathbf{4}.\,x_1^{*i} \leftarrow x_1^i\ \mathbf{5}.\,r_1^{*i} \leftarrow r_0$$

iii. Classification

ELSE

1. $Updating\ \mu_{k-1}^i\ to\ \mu_k^i$: $\mu_k = \frac{k-1}{k}\mu_{k-1} + \frac{1}{k}x_k$: $\mu_1 \leftarrow x_1$

2.Calculate the unimodal density $D_k\ of\ the\ data\ sample\ x_k^i$:

$$D_k(x_i) = \frac{1}{1+\frac{\|x_i-\mu_k\|^2}{\sigma_k^2}} = \frac{1}{1+\frac{\|x_i-\mu_k\|^2}{X_k-\|\mu_k\|^2}}$$

3.Update $D_k(x_k^{*i})(j = 1, 2 \ldots\ldots F^1)$

$$D_k(x_i) = \frac{1}{1+\frac{\|x_i-\mu_k\|^2}{\sigma_k^2}} = \frac{1}{1+\frac{\|x_i-\mu_k\|^2}{X_k-\|\mu_k\|^2}}$$

$$\mathbf{4}.\,IF\left(D_k(x_k^i) > max_{j=1.2\ldots F}\left(D_k\left(x_j^{*f}\right)\right)\right) or \left(D_k(x_k^i) < max_{j=1.2\ldots F}\left(D_k\left(x_j^{*f}\right)\right)\right) THEN$$

– **Create a new data cloud around** x_k^i **and initialize the parameters** :

$$F^1 \leftarrow F^1 + 1\ ; x_{F^i}^{*i} \leftarrow x_k^i\ ;\ M_{F^i}^{*i} \leftarrow 1\ ;\ r_{F^i}^{*i} \leftarrow r_0$$

5. Else

– **Find the nearest data cloud to** x_k^i :

$$x_N^{*i} \leftarrow argmin_{j=1.2...F^i}(\|x_k^i - x_j^{*i}\|);$$

- **IF** $(\|x_k^i - x_N^{*i}\|) \leq r_N^{*i}$ ***THEN***
- **Update the meta parameters of the data cloud** :

$$x_N^{*i} \leftarrow \frac{M_N^{*i}}{M_N^{*i}+1} x_N^{*i} + \frac{1}{M_N^{*i}+1}; r_N^{*i} \leftarrow \sqrt{0.5((r_N^{*i})^2 + (1 - \|x_N^{*i}\|^2))} \ ;$$

- **ELSE add new data cloud** : $F^1 \leftarrow F^1 + 1$; $x_{F^i}^{*i} \leftarrow x_k^i$; $M_{F^i}^{*i} \leftarrow 1$; $r_{F^i}^{*i} \leftarrow r_0$
- **ENDIF**

6.ENDIF
ENDIF
END WHILE

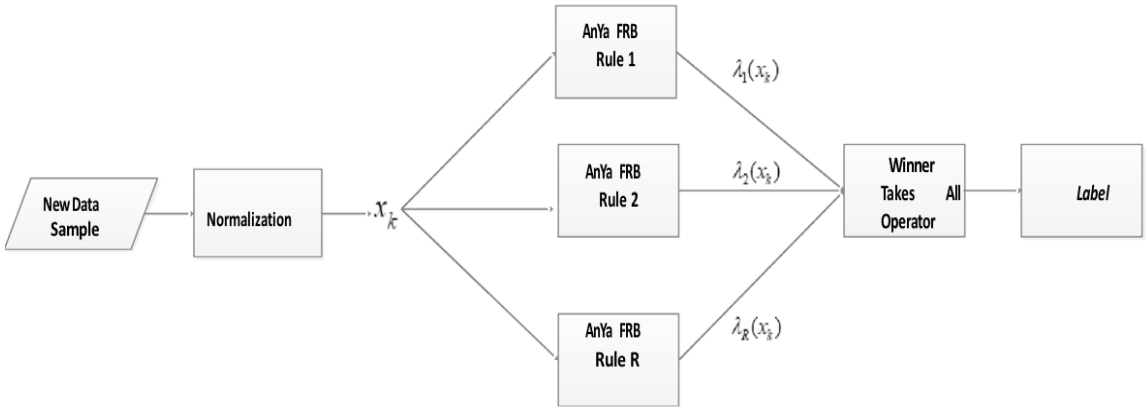


Figure 4: Graphical depiction illustrating the ALMMo-0 classifier within a block diagram

The ALMMo-0 classification process is data-driven and unfolds in a series of steps.

1. Data preprocessing: Prior to analysis, all input data undergoes normalization, where each sample is divided by its own norm. This ensures all data points are on a similar scale.

2. Initial model generation: The first data point becomes the foundation for the initial model, with its support (significance) set to 1.

3. Data cloud construction: Subsequent data points are assigned to the most relevant class using confidence scores generated by sub-classifiers. These classifications establish "data clouds" representing clusters of similar data points.

4. Dynamic model update: The algorithm continuously monitors the global data density. If a new data point falls outside the established density range (too high or low), a fresh data cloud is created around it. Otherwise, the existing data cloud closest to the new point is identified and updated by adjusting its support and radius to incorporate the new information.

5. Classification: Finally, the classifier leverages the constructed data clouds and associated confidence scores to determine the final class label for a new data point.

The ALMMo-0 procedure establishes a robust framework for 3D palmprint biometric systems by achieving accurate and efficient data categorization through a four-step process. These steps include:

1. **Normalization:** This initial phase standardizes the input data, ensuring consistency for further analysis.
2. **Meta-Parameter construction:** Here, the normalized data is leveraged to construct meta-parameters, which will be instrumental in the classification stage.
3. **Classification:** This stage utilizes the calculated global density and mean values to classify the data points.
4. **Validation:** Finally, validation data is routed to individual AnYa FRB sub-classifiers, which then generate the final labels for the data. This revised version emphasizes the overall functionality of ALMMo-0 and clarifies the role of each step within the system.

3.5.1 Computational complexity of the ALMMo-0 classifier

One of the key advantages of the ALMMo-0 framework is its ability to scale effectively with larger datasets and an increasing number of classes. The time complexity of the identification function, F(P×J×H), where P represents the number of data samples, J the number of data clouds, and H the number of features, ensures that the system scales efficiently as these factors grow. Furthermore, the verification function operates with a time complexity of F(P×J), making it highly adaptable to real-time classification tasks.

The space complexity, F(P×H), is similarly manageable, requiring only space to store the input data. As a result, the ALMMo-0 classifier can handle large-scale environments, such as national identification systems or corporate security databases, without significant computational overhead. Unlike traditional methods like PCA and SVM, which have complexities ranging from $F(T^2 \times R)$ to $F(T^3 \times R)$ (where T is the number of training samples and R is the number of features), ALMMo-0's lower complexity allows it to maintain high accuracy and low Equal Error Rate (EER) without the need for iterative tuning or retraining.

This combination of linear or sub-linear time and space complexity, along with the framework's ability to dynamically adapt to data distribution changes, ensures that the ALMMo-0 classifier remains efficient and scalable even under more demanding conditions.

3.5.2 Confidence thresholding

In the ALMMo-0 framework, the winner-takes-all strategy plays a crucial role in determining the final classification outcome. This approach relies on the confidence scores generated by multiple sub-classifiers, with the class corresponding to the highest confidence score being selected as the predicted class. However, to ensure optimal performance, it is important to analyze how the system's performance changes when varying the confidence thresholds
used in this strategy.

In this study, we conducted an analysis to evaluate how different confidence thresholds affect the accuracy and error rates of the system, particularly focusing on Rank-1 accuracy and Equal Error Rate (EER). The results indicate that by setting an appropriate confidence threshold, we can strike a balance between reducing false positives (false acceptances) and false negatives (false rejections).

- **Low confidence threshold**: When the confidence threshold is set too low, the system may be more prone to **false acceptances**, as even weak classifications can be considered valid. This can lead to higher EER and a lower overall accuracy, as the system may mistakenly classify ambiguous or borderline cases.
- **High confidence threshold**: On the other hand, setting the threshold too high increases the risk of **false rejections**, where valid classifications are discarded due to insufficient confidence from the sub-classifiers. While this may improve security in some cases, it can reduce the system's ability to recognize legitimate inputs, leading to an increased rejection rate.
- **Optimal threshold**: Through our experiments, we found that a moderate confidence threshold (e.g., 0.75) offers the best balance, maximizing Rank-1 accuracy and minimizing EER. This threshold ensures that only classifications with strong confidence are accepted while maintaining the system's ability to recognize genuine variations in palmprints. The analysis confirms that the confidence threshold has a significant impact on the system's performance, and tuning this parameter appropriately is key to achieving robust and reliable biometric recognition.

3.6 Feature fusion

In the context of the ALMMo-0 framework, feature extraction techniques such as LPQ, BSIF, and GIST serve as integral components for capturing essential characteristics from 3D palmprint images. A key question in this process is whether these features are fused before the classification phase or handled independently.

Upon extracting features using LPQ, BSIF, and GIST, the ALMMo-0 framework can handle feature fusion in a few distinct ways. In one approach, features from these descriptors are concatenated into a single feature vector, forming a comprehensive representation of the 3D palmprint before classification. This ensures that the classifier has access to a unified feature set, capturing various spatial, frequency, and texture information in a complementary manner.

Alternatively, the features can be processed independently by separate sub-classifiers within the ALMMo-0 framework. In this scenario, each feature set contributes to its respective classification pathway, and the final classification decision is made through a higher-level fusion process, such as score-level fusion, where the outputs of individual sub-classifiers are combined.

This would allow the system to evaluate the strengths of each feature extraction method and assign weights based on their contribution to accurate classification. In both cases, the ALMMo-0 framework leverages the diversity in feature extraction techniques to improve classification accuracy and robustness, while maintaining autonomy and efficiency.

Detailed experimental results from the ROC curves and Rank-1 accuracy evaluations further emphasize the effectiveness of the feature extraction and fusion mechanisms.

4 Experimental procedures and findings

This section delves into the intricacies of the testing procedures and their associated results.

4.1 Database

To evaluate the performance of our proposed method, we utilized the publicly accessible PolyU 3D Palmprint

Dataset provided by the Hong Kong Polytechnic University [28]. This dataset features 200 individuals (136 males and 64 females) contributing a total of 8,000 3D palmprint samples across 400 unique palm classes. Data acquisition occurred in two sessions, separated by an average of one month, to capture variations in 3D palmprint characteristics. Each session involved capturing ten samples from both the left and right hands of each participant.

4.2 Experimental setup

The experiments were conducted using a system equipped with an Intel Core i5-10500H processor at 2.50 GHz, supported by 8 GB of RAM. This setup provides adequate computational power for real-time processing demands of biometric authentication. No external GPU was utilized; however, the integrated capabilities were sufficient for our processing needs. This hardware setup ensured that our experimental results were both reproducible and scalable on systems with similar specifications.

4.3 The experimental methodology

The proposed system employs three distinct filters, namely LPQ, BSIF, and GIST, for feature extraction. As illustrated in Figure 1, the feature extraction stage, facilitated by a descriptor called Filter Bank, precedes the classification process of the Autonomous Learning Multi-Model (ALMMo-0) classifier and comprises five main steps. ALMMo-0 leverages 3D palmprint images for user authentication. The system begins by capturing a 3D image, which then undergoes preprocessing using a multi-scale Retinex filter. To extract relevant features, a Filter Bank descriptor is employed, utilizing one of the available filters (LPQ, BSIF, or GIST). These features are then fed into the classifier, which builds its internal model. Finally, ALMMo-0 autonomously categorizes new data as either genuine or fake based on the extracted characteristics, effectively differentiating between authorized and unauthorized users. ALMMo-0 tackles classification tasks through a data-driven, multi-model approach. It begins by identifying inherent data structures and autonomously generates fuzzy rules. This involves shaping raw data points into informative clusters and pinpointing key convergence areas. The system then leverages these data-derived fuzzy rules (AnYa-type) specific to each class. When presented with new data, ALMMo-0 assigns confidence scores using a "winner-takes-all" strategy. Finally, the system's analysis is evaluated to arrive at the most likely classification. This research employed the ALMMo-0 classifier for feature extraction, outperforming traditional methods like KNN. The study investigated both 3D palmprint identification and verification, with identification results presented as the recognition rate, particularly the Rank-1 recognition rate.

$$Rank-1 = \frac{N_i}{N} \cdot 100(\%) \tag{15}$$

Within our verification scenario, ***N*** signifies the total number of image attempts for identity verification. In contrast, ***Ni*** denotes the successful matches where images are correctly linked to their corresponding identities. Our evaluation hinges on the Equal Error Rate (EER), which represents the threshold where the rate of incorrectly accepting imposters (False Accept Rate, FAR) aligns with the rate of incorrectly rejecting legitimate users (False Reject Rate, FRR). For identification tasks involving one-to-many searches, the Cumulative Match Characteristic (CMC) curve becomes the preferred metric. This contrasts with verification mode, where a Receiver Operating Characteristic (ROC) curve is employed. The ROC curve depicts the trade-off between correctly identifying authorized users (Genuine Accept Rate, GAR) and mistakenly accepting unauthorized individuals (FAR) across various decision thresholds.

4.3.1 Local phase quantization (LPQ)

In assessing the efficacy of LPQ feature extraction, we conducted experiments using various window widths from 3 to 9. According to Table 3, window size (WS) of 7 and 9 demonstrated exceptional performance. At a False Acceptance Rate (FAR) of 1%, the verification rate (VR) achieved 100%, and the Equal Error Rate (EER) decreased to 0%. Notably, the Rank-1 identification rate achieved 100% accuracy in both verification and identification modes.

Table 3: The rank and Equal Error Rate (EER) results for LPQ feature extraction using the ALMMo-0 classifier were analyzed.

WS	Identification	Verification	
	RANK_1 (%)	EER (%)	VR at 0.1% FAR
3	96.00	0.86	98.50
5	99.25	0.15	99.75
7	**100**	**0.00**	**100**
9	**100**	**0.00**	**100**

Our experiments overwhelmingly favor LPQ feature extraction with a window size of 7 for achieving exceptional accuracy and reliability. This choice is substantiated by the compelling performance metrics: a perfect Equal Error Rate (EER) of 0.00%, a flawless verification rate (VR) of 100% at a strict False Acceptance Rate (FAR) of 1%, and a 100% Rank-1 identification rate. From Figure 5, the ROC and CMC curves show that a window size (WS) of 7 provides the best performance compared to WS = 3 or WS = 5. These results undeniably demonstrate the superiority of a window size of 7 in maximizing accuracy for our specific application.

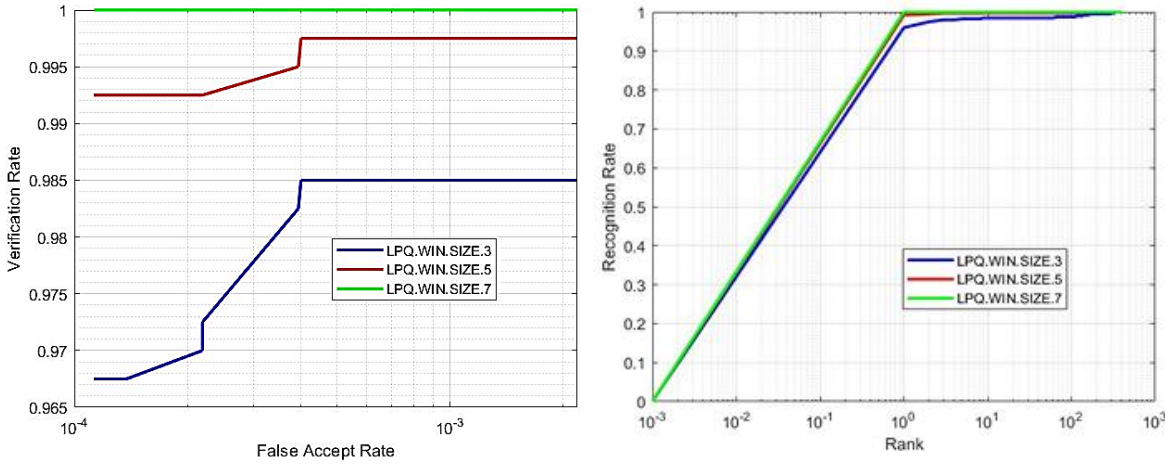


Figure 5: ROC and CMC curves were generated specifically for LPQ feature extraction using the ALMMo-0 classifier

Table 4 provides a comprehensive comparison of the MCI+LPQ+ALMMo-0, GCI+LPQ+ALMMo-0, T+LPQ+ALMMo-0, and MSR+LPQ+ALMMo-0 palmprint recognition methods, using the LPQ descriptor or feature extraction and the ALMMo-0 classifier. Among the four methods, MSR+LPQ+ALMMo-0 achieves the highest overall performance, with a perfect Rank-1 accuracy of 100%, an Equal Error Rate (EER) of 0.0%, and a Verification Rate of 100% at 1% FAR. It is also the most efficient in terms of time cost, with 15 ms per identification. ST+LPQ+ALMMo-0 follows closely with a Rank-1 accuracy of 99.8%, an EER of 0.3%, and a Verification Rate of 99.8%, but requires a longer processing time of 25 ms. GCI+LPQ+ALMMo-0 shows solid performance with a Rank-1 accuracy of 99.0%, an EER of 0.4%, and a Verification Rate of 99.5%, with a time cost of 18 ms. Lastly, MCI+LPQ+ALMMo-0, while still performing well, has the lowest Rank-1 accuracy at 98.5%, an EER of 0.45%, and a Verification Rate of 99.0%, with a time cost of 20 ms. Overall, MSR+LPQ+ALMMo-0 stands out as the best method, excelling in accuracy, error rate, and efficiency.

Table 4: Comparative Analysis of MSR as a Novel Method Against Classical GCI, MCI, and ST in Palmprint Recognition with ALMMo-0

Method	Rank-1 (%)	(EER) (%)	VR at 1% FAR (%)	Time Cost for 1 Identification (ms)
MCI+LPQ+ ALMMo-0	98.5	0.45	99.0	20
GCI+LPQ+ ALMMo-0	99.0	0.4	99.5	18
ST+LPQ+ ALMMo-0	99.8	0.3	99.8	25
MSR+LPQ+ ALMMo-0	100	0.0	100	15

4.3.2 Binarized statistical image features (BSIF)

To determine the optimal filter size for BSIF feature extraction, we experimented with various ICA texture filters were applied across sizes ranging from 9x9 to 17x17.

Table 5 and Figure 6 compares the identification and verification performance of ICA texture filters of various sizes in palmprint recognition. For identification, the filter size 11x11 achieves the highest Rank-1 accuracy at 98.50%, followed closely by the 15x15 filter at 98.00%. However, the 9x9 filter, while having a slightly lower Rank-1 accuracy of 97.25%, excels in verification, showing the lowest Equal Error Rate (EER) at 0.44% and the highest Verification Rate (VR) at 0.1% FAR, with a value of 99.25%.

The larger filters (13x13, 15x15, and 17x17) all have similar EERs (around 0.56–0.58%) and VR values of 99.00%, indicating that increasing the filter size beyond 11x11 does not lead to significant improvements in verification performance. Overall, the 11x11 filter stands out for identification, while the 9x9 filter is superior for verification due to its lower error rate and higher verification rate.

Table 5: The Rank-1 accuracy and equal error rate (EER) outcomes for BSIF feature extraction using the ALMMo-0 classifier were evaluated

	Identification	**Verification**	
ICA texture Filters size	RANK_1	EER	VR at 0.1% FAR
9x9	**97.25**	**0.44**	**99.25**
11x11	98.50	0.58	99.00
13x13	97.55	0.58	99.00
15x15	98.00	0.56	99.00
17x17	97.50	0.57	99.00

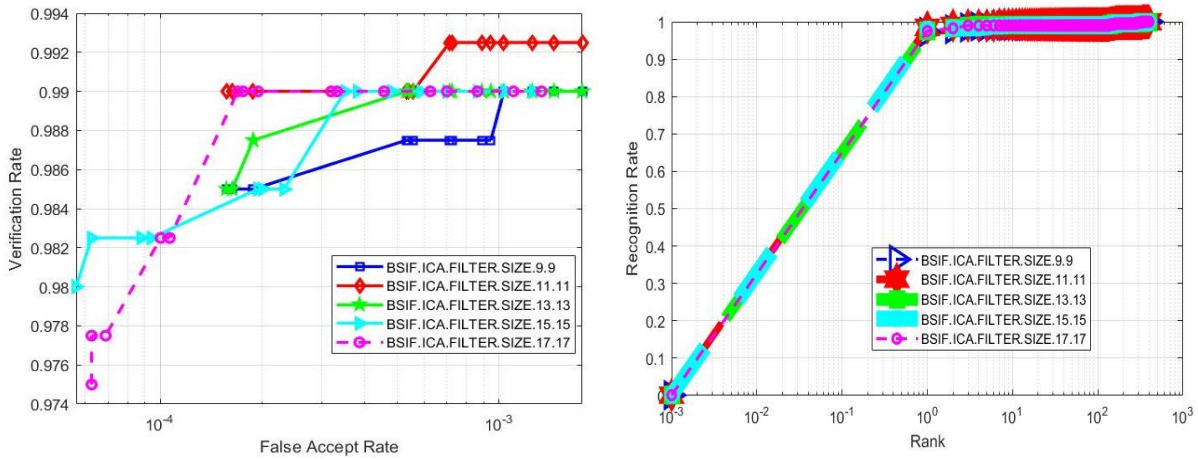


Figure 6: ROC and CMC curves were plotted specifically for BSIF feature extraction using the ALMMo-0 classifier

4.3.3 Gist descriptor

To determine the optimal number of blocks for GIST feature extraction, we experimented with values ranging from 9 to 13. **Table 6** presents the Rank-1 accuracy, Equal Error Rate (EER), and Verification Rate (VR) at 0.1% FAR for GIST feature extraction using different block numbers with the ALMMo-0 classifier. As block numbers increase from 9 to 13, a slight improvement in both identification and verification performance is observed. The Rank-1 accuracy improves from 97.25% for 9 blocks to 98.25% for 13 blocks, showing that increasing the number of blocks enhances identification performance. Similarly, the EER decreases from 0.19% to 0.16% as the block number rises, indicating a reduction in classification errors with more blocks. The VR at 0.1% FAR remains constant at a high value of 99.75% across all block sizes, suggesting that the verification performance is robust regardless of block numbers.

Overall, using 13 blocks yields the best results, with the highest Rank-1 accuracy and the lowest EER, indicating that it provides the most accurate and reliable performance. Figure 7 presents the ROC and CMC curves generated specifically for GIST feature extraction using the ALMMo-0 classifier, illustrating the performance of the system in both verification and identification tasks. Figure 8 shows the ROC and CMC curves that compare the performance of the ALMMo and KNN classifiers on data preprocessed with LPQ feature extraction and MSR. The results reveal that the ALMMo classifier significantly outperforms KNN when using the same feature extraction methods. This underscores the efficacy of the ALMMo-LPQ-MSR combination as a preprocessing pipeline.
Remarkably, this method achieved flawless performance metrics: a 100% verification rate at a 1% false acceptance rate, a zero percent equal error rate, and a perfect 100% rank-one identification rate. These exceptional results highlight the ALMMo classifier's ability to efficiently utilize the extracted features, leading to a substantial improvement in overall system performance.

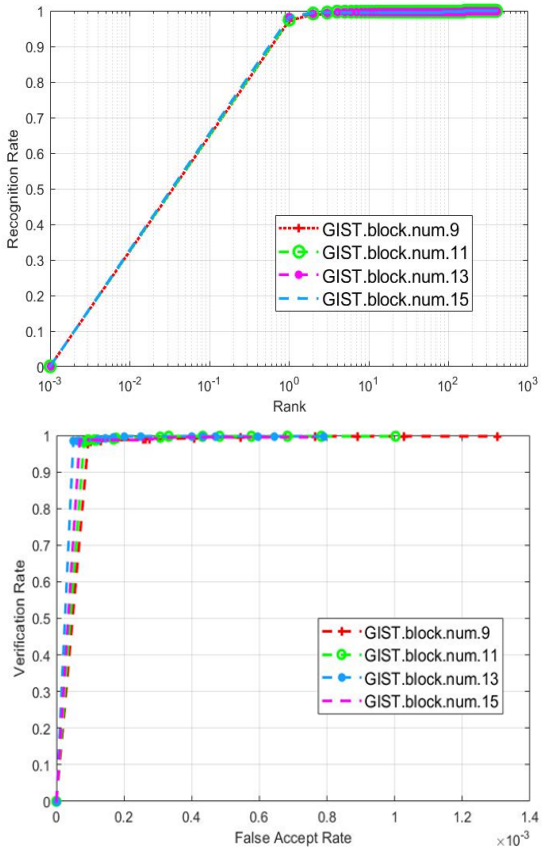


Figure 7: The CMC (Cumulative Match Characteristic) curve at the top and the ROC (Receiver Operating Characteristic) curve at the bottom were generated specifically for GIST feature extraction using the ALMMo-0 classifier.

Table 6: The rank and Equal Error Rate (EER) results for GIST feature extraction with the ALMMo-0 classifier were analyzed

Block Numbers	Identification	Verification	
	RANK_1 (%)	EER (%)	VR at 0.1% FAR
9	97.25	0.19	99.75
11	97.50	0.18	99.75
13	98.25	0.16	99.75

4.3.4 Comparison with current state-of-the-art techniques in 3D palmprint recognition

Table 7 compares various personal recognition methods based on feature extraction techniques, Rank-1 accuracy, Equal Error Rate (EER), and time cost for one identification. The methods span across traditional approaches and deep learning techniques, ranging from 2011 to 2024. The proposed method (2024) achieves the highest performance, with a 100% Rank-1 accuracy, 0.00% EER, and the lowest time cost of 0.015 seconds per identification. In comparison, SSR+PCANet (2019) also performs exceptionally with a 99.98% Rank-1 accuracy, 0.00% EER, but requires 1.28 seconds per identification. Other deep learning-based methods like MCI+DCTNet (2018) achieve 99.833% Rank-1 accuracy with a 0.0202% EER, but with a higher time cost of 1.25 seconds. Traditional methods like MCI-based DPDSF (2019) have a strong Rank-1 accuracy of 99.86%, but take significantly longer at 2.6 seconds per identification. In contrast, methods like CR_L2 (2015), while showing modest Rank-1 accuracy at 99.15%, are faster, with a 0.02278 second time cost.

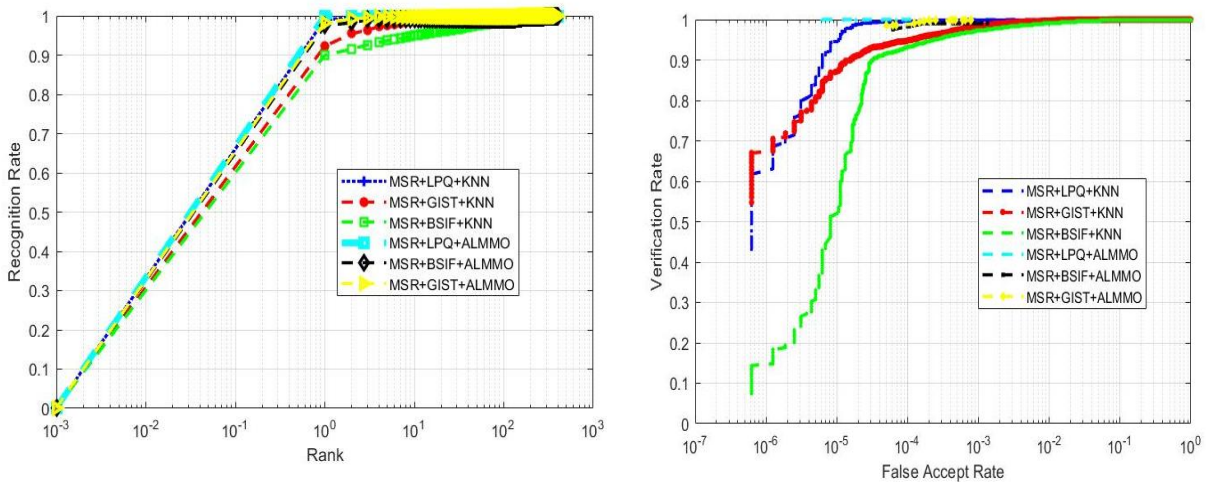


Figure 8: ROC and CMC curves were plotted for various feature extraction methods using both the KNN classifier and the ALMMo-0 classifier.

Table 7: Comparison of the proposed personal recognition method, its computation time, and existing approaches

Methods	Years	Feature Extraction Methods	Rank-1 (%)	EER (%)	Time cost for 1 identification (s)
MCI+DCTNet [7]	2018	Deep Learning	99.833%	0.0202%	1.25
COSDISH [8]	2020	Deep hashing	97.05%	–	1.2
CSTBR [30]	2018	Traditional methods	99.67%	1.51%	0,09747
MCI based DPDSF [29]	2019	Traditional methods	99.86%	–	2.6
MCI based PDF [6]	2018	Traditional methods	99.89%	0.200%	0.24
3D_GIST+KNN [5]	2023	Traditional methods	99.25%	0.2001%	1.15
PCA + TPTSR [2]	2014	Traditional methods	98.55%	–	17.65
JLOF [15]	2011	Traditional methods	99.79%	0.16%	10.70
SQI + Gabor + PCA + LDA [20]	2017	Traditional methods	98.96%	0.15%	5.6
SSR+ PCANet [1]	2019	Deep Learning	99.98%	0.00%	1.28
CR_L2 [19]	2015	Traditional methods	99.15%	–	0.02278
OUR	2024	Traditional methods	100.00%	00.00%	0.015

This comparison highlights the balance between accuracy and computational efficiency, with the proposed method offering both superior accuracy and minimal time cost.

5 Conclusion

This study introduces an innovative 3D palmprint authentication system that harnesses the capabilities of the Autonomous Learning Multi-Model Classifier of 0-Order (ALMMo-0). This method tackles limitations prevalent in current biometric systems by offering a classification tool that is transparent, self-governing, and computationally efficient. By incorporating advanced feature extraction techniques like LPQ, BSIF, and GIST alongside ALMMo-0, the proposed system achieves impressive performance indicators, including a minimal Equal Error Rate (EER), superior rank-1 accuracy, and efficient computation. While biometric research has made substantial strides, the paper emphasizes the continuing need for enhanced security applications. It underscores ALMMo-0's capability to establish a robust framework incorporating multiple models for accurate classification and adaptive learning from data. The system's architectural details, including feature extraction and classification stages, are thoroughly examined. The proposed methodology's efficacy in improving biometric system performance, particularly in 3D palmprint authentication, is supported by empirical findings. The research highlights the ALMMo-0 classifier's critical contribution in providing dependable solutions. These advancements offer exceptional interpretability, accuracy, and computational efficiency, setting a solid foundation for future advancements in biometric security.

Acknowledgement

The authors would like to express their sincere gratitude to the faculty of new information and communication technologies at the university Kasdi Merbah ouargla for their invaluable support throughout the course of this research. We also extend our appreciation to the members of the laboratory of electrical engineering for their technical assistance and valuable feedback. Special thanks to the Hong Kong polytechnic university for providing access to the publicly available PolyU 3d palmprint dataset, which was crucial for the experimental validation of our proposed method. Finally, we acknowledge the insightful contributions from our colleagues and collaborators who helped improve the quality of this work.

References

[1] Chaa M, Akhtar Z, Attia A (2019) 3D palmprint recognition using unsupervised convolutional deep learning network and SVM classifier *Image Process* 13(5):736–745. https://doi.org/10.1049/iet-ipr.2018.5642

[2] Cui J (2014) 2D and 3D palmprint fusion and recognition using PCA plus TPTSR method. *Neural Comput & Applic* 24(3–4):497–502. https://doi.org/10.1007/s00521-012-1265-y

[3] Zhang D, Lu G, Li W, et al. (2010) Palmprint recognition using Gaussian curvature images (GCI), mean curvature images (MCI), and surface type map (ST map) features. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 39(5):505–519.https://doi.org/10.1109/TSMCC.2009.2020790

[4] Li W, Zhang L, Zhang D, et al. (2011) A framework for 3D palmprint matching using MCI, baseline, and orientation features' *Transactions on Systems, Man, and Cybernetics - Part C* 42(2):274–279. https://doi.org/10.1109/TSMCC.2010.2055849

[5] Attia, Abdelouahab, et al. (2023) A fuzzy connective score fusion technique for 2D and 3D palmprint authentication system. *Evolving Systems* 14(5):891–901

[6] Sun Z, Tan T, Wang Y, et al. (2005) Ordinal palmprint representation for personal identification. *CVPR 2005*, pp. 279–284. DOI:10.1109/CVPR.2005.267

[7] Zhang D, Kanhangad V, Luo N, Kumar A (2008) 3D palmprint recognition using structured light imaging. *BTAS 2008*. DOI:10.1109/BTAS.2008.4699346

[8] Zhang D, Lu G, Li W, Zhang L (2010) Robust palmprint verification with 2D and 3D feature fusion. *Pattern Recognition* 43(1):358–368. https://doi.org/10.1016/j.patcog.2009.04.026

[9] Yang B, Wang X, Yao J, Yang X, Zhu W (2011) Efficient local representations for 3D palmprint recognition using Gabor wavelets and LBP. *Journal of Electronic Imaging* 22(4):043040. https://doi.org/10.1117/1.JEI.22.4.043040

[10] Li W, Zhang D, Zhang L, et al. (2011) Joint line and orientation feature extraction for 3D palmprint recognition. IEEE Transactions on Systems, Man, and Cybernetics - Part C 42(2):274–279. https://doi.org/10.1109/TSMCC.2010.2055849

[11] Liu M, Li L (2012) Cross-correlation-based binary image registration for 3D palmprint recognition. *IEEE 11th International Conference on Signal Processing*, pp. 1597–1600. DOI:10.1109/ICoSP.2012.6491885

[12] Meraoumia A, Chitroub S, Bouridane A (2013) 2D and 3D palmprint information fusion using PCA and HMM. *Integrated Computer-Aided Engineering* 20(3):303–319. https://doi.org/10.3233/ICA-130431

[13] Cui J (2014) 2D and 3D palmprint fusion and recognition using PCA plus TPTSR method. *Neural Comput & Applic* 24(3–4):497–502. https://doi.org/10.1007/s00521-012-1265-y

[14] Zhang L, Shen Y, Li H, Lu J (2015) Collaborative representation with block-wise feature extraction for 3D palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(8):1730–1736. https://doi.org/10.1109/TPAMI.2014.2372764

[15] Chaa M, Boukezzoula N-E, Attia A (2017) Score-level fusion of 2D and 3D palmprints using B-BSIF and Gabor wavelets. *Journal of Electronic Imaging* 26(1):013018. https://doi.org/10.1117/1.JEI.26.1.013018

[16] Chen M, Akhtar Z, Attia A (2019) A hybrid multimodal system integrating finger vein and palmprint data. *IET Image Process* 13(5):736–745. DOI:10.1049/iet-ipr.2018.5642

[17] Li X, Zhang L, Xu Y, et al. (2020) Real-time person recognition with 3D palmprint data using CNNs. *IEEE Transactions on Instrumentation and Measurement* 68(9):3287–3298

[18] Zhang X, Bai N, Wu J, et al. (2021) Transformer-based models for palmprint recognition. *IEEE Sensors Journal* 20(20):11864–11873

[19] Li W, Zhang D, Zhang L, et al. (2018) Structured-light-based acquisition system for concurrent 2D and 3D palmprint images.

[20] Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1):62–66.DOI:10.1109/TSMC.1979.4310076

[21] Zhang D, Lu G, Li W et al (2009) Palmprint recognition using 3-D information. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews 39(5):505–519.).

[22] E. Land and J. McCann, Lightness and retinex theory, Journal of the Optical Society of America, 61 (1971), p. 1–11. http://dx.doi.org/10.1364/JOSA.61.000001 enveloppe. Int J Comput Vis 42(3):145–175. https://doi.org/10.1023/A:1011139631724

[23] D.J. Jobson, Z. Rahman, and G.A. Woodell, A multiscale retinex for bridging the gap between color images and the human observation of scenes, IEEE Transactions on Image Processing, 6 (1997), pp. 965–976. http://dx.doi.org/10.1109/83.597272

[24] Oliva A, Torralba A (2001) Modeling the shape of the scene: a holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175. https://doi.org/10.1023/A:1011139631724

[25] Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720. https://doi.org/10.1109/34.598228

[26] Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86. https://doi.org/10.1162/jocn.1991.3.1.71

[27] Boots B, Kokichi S, Sung NC, Atsuyuki O (2009) Spatial tessella-tions: concepts and applications of voronoi diagrams

[28] PolyU 2D and 3D Palmprint Database (2017) available at: www.comp.polyu.edu.hk/~biometrics/, Accessed on Jan. 01

[29] L. Fei, B. Zhang, Y. Xu, W. Jia, J. Wen, J. Wu, "Precision direction and compact surface type representation for 3D palmprint identification," Pattern Recognition, vol. 87, pp. 237–247, 2019.

[30] L. Fei, G. Lu, W. Jia, J. Wen, D. Zhang, "Complete Binary Representation for 3-D Palmprint Recognition," IEEE Transactions on Instrumentation and Measurement, vol. 17, no. 12, pp. 2761–2771, 2018.

https://doi.org/10.31449/inf.v49i5.7052 Informatica **49** (2025) 81–94 **81**

Intelligent Optimization of Logistics Paths Based on Improved Elite Artificial Bee Colony Algorithm

Qing Yuan
Business School, Zhengzhou University of Economics and Business, Zhengzhou 451191, China
Email: zht20110113@126.com

Keywords: artificial bee colony algorithm, logistics delivery, path optimization, elite bee colony, transportation cost

Received: September 2, 2024

With the rapid development of e-commerce and logistics industry, logistics path optimization has gradually become a key bottleneck restricting enterprises from improving service quality and reducing operating cost. Therefore, the study introduces an elite swarm strategy, which uses elite individuals to guide the honey search in the hiring bee stage, and proposes an intelligent optimization model for logistics paths based on an improved artificial swarm algorithm. The study is calculated using Windows 10, Intel Core i7 CPU, NVIDIA GeForce GPU, 64GB memory, and Matlab simulation software. The experimental results show that in the single-mode test function, the Avg and Std of the elite swarm algorithm are on average six orders of magnitude lower than the other three algorithms. In the multi-mode test function, the Avg and Std of the elite swarm algorithm are on average seven orders of magnitude better than the other three algorithms, which is far superior. The area enclosed by the model precision-recall curve and the coordinate axis is 0.96, and the F1 value is 97.7%. In the simulation test, the model plans the delivery path for 20 customer points with a total path length of 5221.43km, a total delivery cost of 83908.38 yuan, a vehicle loading rate of 97.4%, and a solution time of 2.8s. The research results indicate that the logistics path intelligent optimization method has good comprehensive performance. It can effectively reduce logistics transportation costs, which has important guiding significance and application value for actual logistics operations. This model not only provides an efficient path optimization tool for logistics enterprises, but also offers new ideas and methods for logistics path optimization research.

Povzetek: Uvedena je izboljšana inteligentna optimizacija poti z elitnim algoritmom umetne čebelje kolonije (EABC) za logistiko, kar zmanjša stroške transporta in izboljša učinkovitost.

1 Introduction

Recently, with the rapid development of global economic integration, the logistics industry is experiencing significant growth due to the expansion of e-commerce [1]. Logistics path optimization, as a key issue in logistics management, directly affects the operational efficiency and cost control of enterprises. How to quickly find the optimal transportation path in the complex and ever-changing logistics environment has become an urgent problem to solve [2]. The development of the logistics industry has undergone a transformation from traditional manual management to modern and intelligent management [3]. Early logistics path optimization relied heavily on experience and manual calculations. With the development of computer technology and algorithms, optimization methods based on mathematical models and computational intelligence have gradually been introduced [4]. Influenced by artificial intelligence and big data technology, the application of intelligent optimization algorithms in logistics path optimization has become increasingly widespread [5]. Among them, the Artificial Bee Colony (ABC) algorithm has been extensively applied in function optimization, neural network training, and pattern recognition due to its simplicity, robustness, and adaptability. In existing research on logistics path optimization, ABC, as a swarm intelligence optimization algorithm, gradually approaches the optimal solution through information sharing and cooperation among bees. Farahbakhsh H et al. proposed an improved ABC algorithm that combined an acceptance rejection method to limit the random search space, improve algorithm speed, and reduce iteration times. The movement step size of bees was used to determine parameters, guiding the algorithm to reach the global optimum. The superiority of the algorithm was verified in engineering problems related to virtual power plant planning and multiple benchmark function tests [6]. Huo F et al. built an optimized ABC algorithm on the basis of chaos theory Tent mapping for threshold selection in image segmentation. This algorithm constructed a complementary encoding scheme. The complementary properties were utilized to adjust local optimal solutions. The experiment demonstrated that the algorithm had

strong ability to obtain the optimal solution and convergence performance [7]. Zhang L et al. built an optimized ABC based on reverse learning to address the low convergence accuracy and slow convergence speed. This algorithm introduced the bystander bee stage to accelerate convergence speed and incorporated Cauchy reverse learning in the reconnaissance stage. The superiority was verified through experiments on 13 standard tests [8]. Yilmaz Acar et al. built a binary ABC for multi-objective resource allocation problems. This algorithm combined advanced local search and binary format of transfer functions. Through experimental verification, the proposed algorithm could effectively solve resource allocation problems in large-scale problems and achieve high accuracy with fewer evaluation iterations [9].

Vehicle Routing Problem (VRP) is a crucial problem in logistics and transportation, aimed at improving transportation efficiency, reducing cost, and decreasing energy consumption by optimizing vehicle routes. Bao S et al. proposed two-phase and three-phase methods based on the Ising machine. Complex problems were divided into sub-problems and mapped to a quadratic unconstrained binary optimization model to adapt to the structure of the Ising machine. The effectiveness of this method was validated through performance comparison on datasets of standard traveling salesman problem and capacity limited VRP [10]. Sun W et al. proposed a joint routing scheduling optimization strategy on the basis of optimized grey wolf optimization. This strategy combined historical experience learning, and cross TS operators. Simulation experiments displayed that this strategy could quickly solve large-scale vehicle network scheduling problems, generate scheduling results with excellent latency performance, and effectively reduce interference between streams, further reducing end-to-end latency [11]. Lin Y et al. proposed a multi automatic guided vehicle route planning method on the ground of deep reinforcement learning. This method utilized near end strategy optimization and long short-term memory to improve adaptability in handling temporary changes and sudden failures in tasks, which considered time step information. The designed method outperformed traditional methods in terms of convergence speed, demonstrating its potential for application in dynamic environments [12]. Chen Y et al. proposed an efficient multi-layer search algorithm that considered both customer service order and vehicle charging schedule simultaneously. This algorithm quickly obtained high-quality solutions through iterative threshold search and heuristic decoupling enumeration method. Numerous computational results indicated that this algorithm outperformed the most advanced methods in solution quality and computation time [13].

Table 1: Summary of relevant information of relevant studies.

Author	Research theme	Improved method	Main indicators	Shortcoming
Farahbakhsh et al. [6]	Power plant path planning	Limiting the random search space	Traffic efficiency	Poor performance in complex environments
Huo et al. [7]	ABC algorithm optimization	Complementary coding	Convergence performance	The algorithm has a long running time
Zhang et al. [8]	ABC algorithm reverse learning optimization	Introduction of the bystander view phase	Convergence accuracy and speed	It takes a lot of data to train
Yilmaz Acar et al. [9]	Multi-target resource allocation	Combined with the transfer function binary format	Convergence performance	The parameter requirements are relatively high
Bao et al. [10]	Vehicle routing problem	Two-stage approach based on the Ising machine	Transportation cost	The vehicle loading rate is relatively low
Sun et al. [11]	Joint routing and scheduling optimization	Cross TS operator	End-to-end delay	Poor flexibility
Lin et al. [12]	Multi-automatic guide vehicle path planning	Proximal strategy optimization and long-and short-term memory	Convergence speed and robustness	The model complexity is relatively high
Chen et al. [13]	High-level multi-efficiency search	Heuristic decoupling enumeration method	Computational efficiency	The calculation accuracy is insufficient
Current study	Logistics path optimization	Elite swarm strategy	Loading rate and distribution efficiency	/

In summary, although numerous scholars have conducted extensive research on ABC algorithm and path optimization, most algorithms still exhibit problems like slow convergence speed, susceptibility to local optima, and low computational efficiency while improving path optimization capabilities. Therefore, an innovative ABC algorithm based on elite bees (EABC) is proposed to reduce the probability getting stuck in local optima and improve computational efficiency by enhancing its global and local search capabilities. It is expected to optimize logistics routes to reduce transportation cost and time, and achieve efficient operation of logistics enterprises in complex and changing practical environments.

Based on the above relevant studies, Table 1 is summarized, in which the research theme, main index methods and shortcomings of relevant studies are summarized.

2 Methods and materials

2.1 Construction of multi-factor logistics vehicle path optimization model

VRP is one of the classic combinatorial optimization problems in combinatorial optimization and operations research, mainly involving how to design the optimal path for a group of vehicles under limited vehicle resources to meet the needs of dispersed customers while minimizing the total cost. This usually includes vehicle running time, fuel consumption, distance, etc. [14]. Firstly, the schematic diagram of logistics vehicle path planning problem is shown in Figure 1.

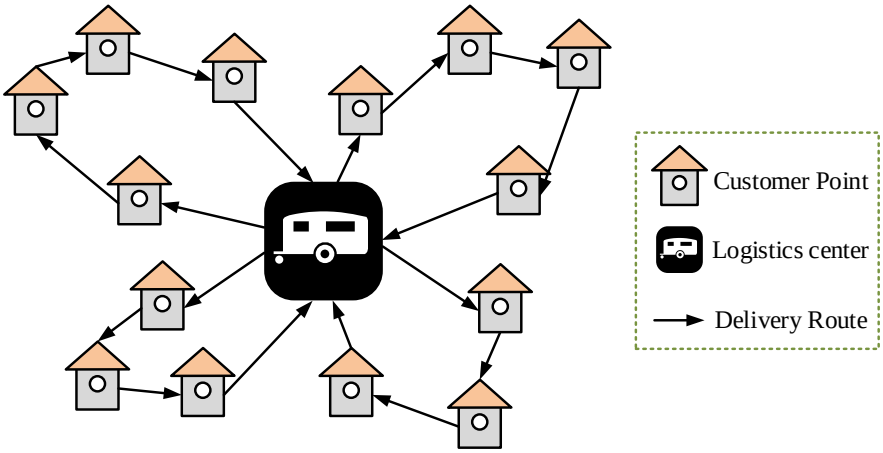


Figure 1: Flow vehicle path planning problem diagram.

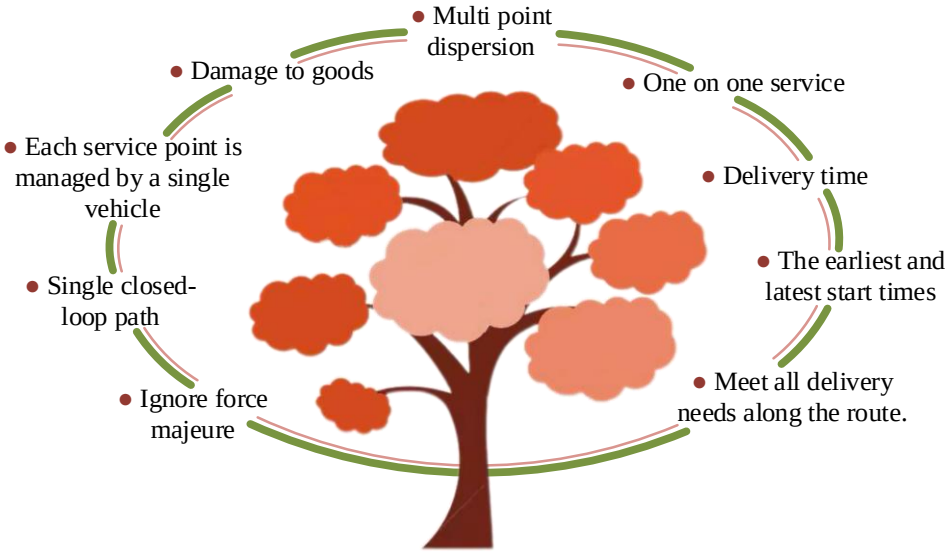


Figure 2: Optimized logistics vehicle routing model.

In Figure 1, the distribution center is located in the center, and customer points are distributed around. Arrows indicate possible delivery routes, connecting distribution centers and customer points, or connecting different customer points. The delivery vehicle departs from the distribution center, follows the planned route for delivery, and goes back to the center. The goal of delivery path planning is to find the optimal route that minimizes

the total distance or optimizes delivery efficiency under other constraints [15]. In the operation process of logistics services, efficiency and cost-effectiveness are the core considerations, and timeliness directly affects customer satisfaction. Delivery cost is also the key factor in customer decision-making. Therefore, the study constructs an optimized logistics vehicle path planning model, which will reduce transportation cost and shorten delivery cycles through innovative strategies, thereby improving customer satisfaction. The main architecture and assumptions are shown in Figure 2.

In Figure 2, the basic hypothesis system is constructed for the model to enhance its applicability and logicality. Firstly, the impact of external non dominant factors such as force majeure on the transportation process is ignored. The second is to focus on optimizing a single closed-loop path. Thirdly, there may be freight loss. Fourthly, the vehicle type is consistent and there is no overloading situation, and each service point is only managed by a single vehicle. Fifthly, a flexible delivery strategy with multi-point dispersion is adopted. The sixth is to implement the principle of one-on-one service. Seventhly, the delivery time is considered. The eighth is to set the earliest and latest start time for the service. Ninth, each truck needs to choose the optimal path among numerous potential paths when performing tasks, and meet the delivery needs of all demand points on that path. The core goal is to reduce the total delivery cost and improve customer satisfaction. In the total distribution costs, it is divided into three major categories: fixed cost, transportation cost, and fuel consumption cost. In addition, considering that the cost of freight loss is directly related to the customer satisfaction, it is included in the customer satisfaction. Firstly, the fixed cost expression is shown in Equation (1).

$$F_C = \sum_{k=1}^{m} F_k Z_k \tag{1}$$

In Equation (1), F_C represents the fixed total cost. m signifies the number of vehicles. F_k signifies the fixed cost of delivering once. Z_k is the parameter, and a value of 1 indicates the execution of this delivery. Otherwise, it is not executed. Secondly, the expression for transportation cost is shown in Equation (2).

$$T_C = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} D_C D_{ijk} X_{ijk} \tag{2}$$

In Equation (2), T_C represents the total transportation cost. D_C signifies the transportation cost per unit distance. D_{ij} represents the distance metric between adjacent customers i and j on the path. X_{ijk} is the decision variable that identifies whether the identification vehicle chooses the path directly from i to node j. Finally, the fuel consumption cost is shown in Equation (3).

$$E_C = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} X_{ijk} E_c T_{ij}^k + \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=1}^{n} Y_{ijk} E_{c'} T_{rj}^k \tag{3}$$

In Equation (3), E_c and $E_{c'}$ respectively represent the fuel consumption per unit time during driving and the fuel consumption when the vehicle is parked at the customer point but the engine is not turned off. T_{ij}^k signifies the time from i to j. T_{rj}^k signifies the loading and unloading time of the vehicle at node j. Y_{ijk} indicates whether the vehicle is parked at customer points i and j. In addition, in customer satisfaction analysis, the cost of freight loss is linked with the time window penalty cost to comprehensively consider the overall satisfaction of customers with delivery services. Firstly, the freight loss during the driving process and loading and unloading process is shown in Equation (4).

$$L_C = (v_1 - v_2)\sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=1}^{n} Y_j^k Q_j^k (2 - e^{-\omega_1 T_{ij}^k} - e^{-\omega_2 T_{rl}^k}) \tag{4}$$

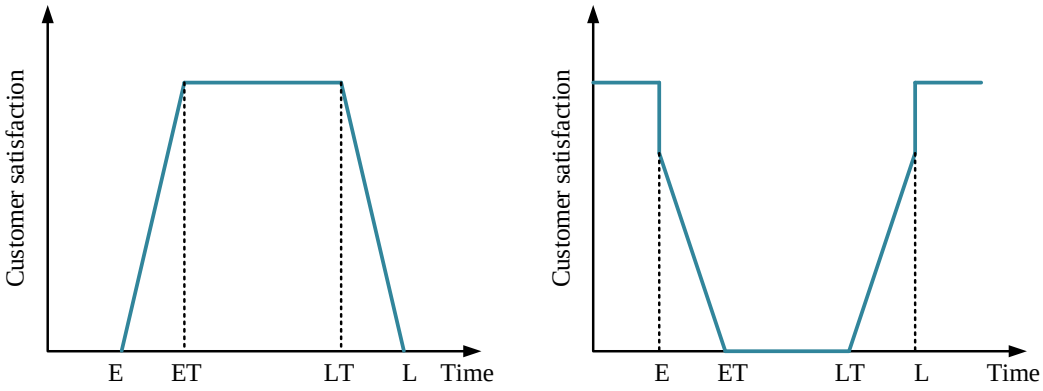
In Equation (4), Y_j^k represents whether to load or unload at j. Q_j^k represents the remaining quantity of goods after the vehicle arrives at customer point i. ω_1 and ω_2 respectively represent the unit time freight loss coefficients during vehicle operation and unloading. v_1 and v_2 respectively represent the unit price and insured unit price of the product. Secondly, in the construction of time window penalty cost, the relationship between vehicle arrival time, time window penalty cost, and customer satisfaction will be quantified. In large-scale distribution, the vehicle's load gradually decreases as the distribution distance grows, and fuel consumption decreases. Delivering within the time window agreed upon with the customer results in the highest customer satisfaction and no time penalty cost. Whereas delivering both before and after the time window affects customer satisfaction, which triggers a penalty cost, the closer the time window the higher the customer satisfaction and the lower the penalty cost. The customer satisfaction and penalty cost curves at different arrival times are shown in Figure 3.

Figure 3 (a) shows the relationship between customer satisfaction and vehicle arrival time. Figure 3 (b) shows the relationship between penalty cost curve and vehicle arrival time. In Figure 3 (a), $[E, ET]$ and $[LT, L]$ respectively represent the earliest arrival time range and latest arrival limit that the customer can accept. $[ET, LT]$ represents the expected arrival time range of the customer, which is the core time period agreed upon by both parties. In Figure 3 (b), customer satisfaction reaches its peak at point $[ET, LT]$, while the cost of violation penalties is the lowest. As the arrival time of the

vehicle gradually approaches ET within $[E, ET]$, although satisfaction remains at a high level, the cost of violation penalties tends to decrease. On the contrary, within the $[LT, L]$ interval, as the arrival time of vehicles is delayed, customer satisfaction gradually decreases and the cost of violation penalties increases. The two extreme ranges of $[-\infty, E)$ and $(L, +\infty]$ are far from the customer acceptance range, so they are set to trigger infinite penalty cost. The customer satisfaction is zero. Therefore, based on the above calculations, taking into account cost, freight loss, customer satisfaction, and time cost, the minimum transportation cost is the ultimate goal of the model, as shown in Equation (5).

$$MINA_C = w_1 F_C + w_2 T_C + w_3 E_C + w_4 L_C + w_5 P_C \quad (5)$$

In Equation (5), F_C, T_C, E_C, L_C, and P_C respectively represent fixed cost, transportation cost, fuel consumption cost, cargo freight loss, and penalty cost incurred due to exceeding the time window. w_1, w_2, w_3, w_4 and w_5 represent the weight coefficients of the above five factors, respectively.



(a) Relationship between customer satisfaction and vehicle arrival time

(b) Relationship between penalty curve and vehicle arrival time

Figure 3: Relationship between customer satisfaction, penalty cost and arrival time.

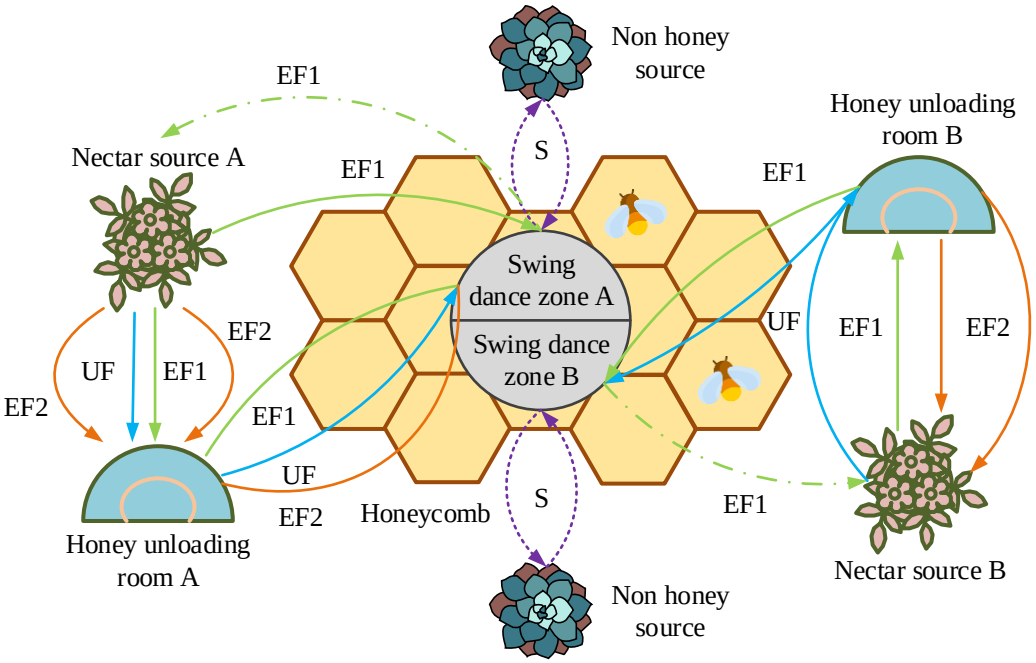


Figure 4: Bee collecting honey process.

2.2 Design of path planning model based on improved ABC

After constructing the objective function and constraint conditions of the multi-factor logistics path planning model, further optimization solutions for logistics paths will be designed. The ABC is to simulate the swarm intelligence behavior of bees in searching for nectar. The optimal solution is searched by simulating the foraging and information exchange process of bees. The optimization problem is solved by the cooperation of three bees: worker bees, observation bees, and reconnaissance bees [16]. The honey harvesting process is

shown in Figure 4.

As shown in Figure 4, after completing the honey collection task of honey source B, the worker bees will go to the honey unloading area B to unload the obtained nectar. At this moment, worker bees face three choices. Firstly, if it is assessed that the resources of honey source B have been depleted, the worker bees will transform into reconnaissance bees and immediately explore new honey source locations, namely the route UF to S. Secondly, worker bees go directly to the swing dance area and perform a specific "swing dance" to convey information about honey source B to surrounding observation bees, aiming to attract them to the honey source, namely, the route EF1. Thirdly, the worker bees return to honey source B to continue their honey harvesting operation, i.e. route EF2. Under the ABC algorithm framework, each nectar source point in the solution space is mapped to a potential optimized solution, and the abundance of nectar contained within it is used as an indicator of fitness, reflecting the quality of the corresponding solution. Furthermore, the implementation process of ABC is shown in Figure 5.

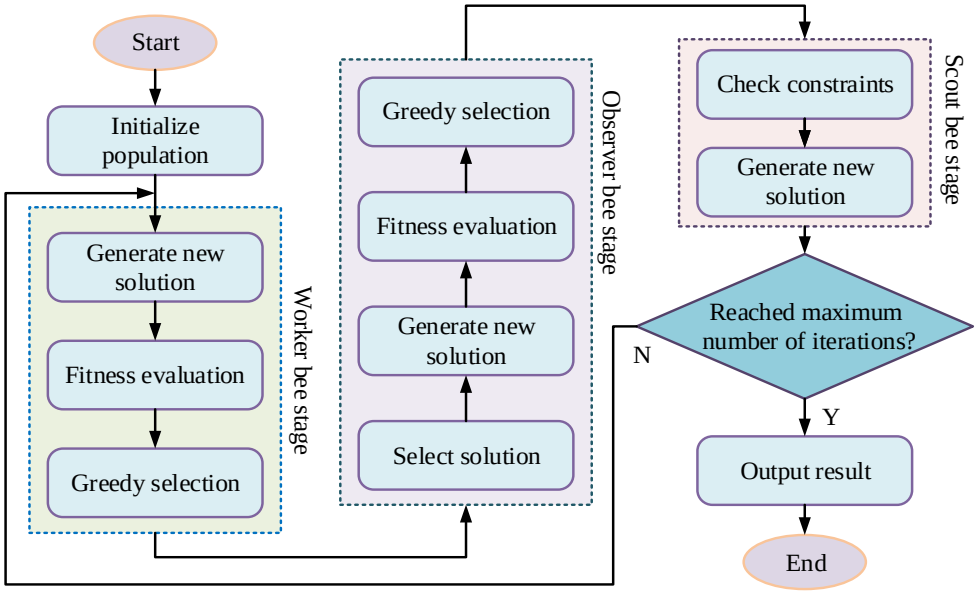


Figure 5: ABC algorithm implementation flow chart.

As shown in Figure 5, the first is to generate an initial population and evaluate the fitness values of each initial solution. Secondly, in the worker bee stage, there are three steps: searching for new solutions, evaluating new solutions, and selecting new solutions. If the fitness value outperforms the current solution, the current solution will be updated to the new solution. Otherwise, it will remain unchanged. During the bee observation phase, the probability of each solution being selected is calculated and a local search is carried out on the solution to generate a new solution and calculate its fitness. If the fitness value outperforms the current solution, it will be updated. Otherwise, it will remain unchanged. In the reconnaissance bee stage, if a solution does not improve after multiple updates, it is considered to be in a stagnant state, and the reconnaissance bee will randomly produce a new solution to replace the stagnant solution. In each iteration, the currently found global optimal solution is updated and recorded. If the maximum iteration is satisfied at this point, the iteration is stopped and the result is output. The initial solution is shown in Equation (6) [17].

$$x_{ij} = lb + rand * (ub - lb) \tag{6}$$

In Equation (6), x_{ij} represents the solution of the current worker bee. i is the honey source. j is the dimension. $rand$ represents a random number. ub and lb respectively represent the upper and lower limits of the honey source search range. Subsequently, during the worker bee stage, the calculation for searching for new honey sources is shown in Equation (7) [18].

$$v_{ij} = x_{ij} + \phi * (x_{ij} - x_{kj}) \tag{7}$$

In Equation (7), x_{kj} represents a randomly selected neighbor solution of x_{ij}. φ is a random number. After evaluating the adaptability of the new honey source, according to the principle of greedy selection, the newly discovered one is compared with the current honey source. If the adaptability of the new outperforms the existing honey source, it is updated and replaced. The probability calculation of hiring bees being selected is shown in Equation (8).

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{8}$$

In Equation (8), SN signifies the honey sources.

P_i represents the probability of hiring bees being selected. fit_i represents the fitness where the hired bees are located. However, in traditional ABC algorithms, there are drawbacks such as slow convergence speed and insufficient local search ability. At the same time, in response to the constraints of logistics vehicle path planning, the elite individual search and parameter set strategies are introduced to enhance the search efficiency. In the process of selecting elite bees, it is necessary to ensure that there is a moderate gap between them to avoid redundancy. Secondly, the fitness of elite bees needs to be closely aligned with the currently calculated optimal solution, in order to effectively expand the range of high-quality candidates. Firstly, the number of elite bee individuals is shown in Equation (9).

$$rn = \frac{pr}{100} \times N \tag{9}$$

In Equation (9), rn signifies the elite bee. pr signifies the ratio of elite individuals to all individuals in the population. N represents the total number of individuals in the population. The separation distance between elite bees depends on the specific setting of the radius parameter, which has a direct impact on the size of the elite bee population. If the set radius value is too large, it will lead to a reduction in the number of members in the elite bee set, which may miss many potential local optimal solutions. Therefore, the radius parameter calculation for the distance between elite bees is shown in Equation (10).

$$raduis = (x_j^u - x_j^l) * r \tag{10}$$

In Equation (10), x_j^u signifies the upper limit of the search space, and x_j^l signifies its lower limit. $r \in [0,1]$ is a proportional parameter. The population size of the elite bees is selected through adaptive parameters, and in the early stage of the algorithm operation, the number of elite bees is high, focusing on a wide range of search and determining the approximate range of the nectar source. As the number of iterations increases, the number of elite bees is scaled down, focusing on improving the search accuracy of the optimal solution. This can help to conduct a fast search in the early stage, and reduce the consumption of resources and improve the search accuracy in the later stage. The next step is the hiring bee stage. As the research has established an elite bee colony, two random elite bees are used for search to accelerate the convergence speed, as displayed in Equation (11).

$$x'_{i,j} = s_{r,j} + rand(-a,a)(x_{i,j} - x_{r,j}) \tag{11}$$

In Equation (11), $s_{r,j}$ and $x_{r,j}$ respectively represent the coordinates of the elite bees randomly selected from the elite bee list in the j-th dimension, and the corresponding value of variable a in that dimension. $rand(-a,a)$ is a random parameter. A specific probability selection mechanism is introduced in elite bees, where individuals are assigned probability values to dominate the selection process. The probability calculation is shown in Equation (12).

$$P_i = \frac{1}{1+f(S_i)} \Big/ \sum_i^{S_{size}} \frac{1}{1+f(S_n)} \tag{12}$$

In Equation (12), S_{size} represents the size of the elite bee population. $f(S_i)$ and $f(S_n)$ represent the fitness of the first and n-th elite bees, respectively. Subsequently, each scout bee follows the probability selection criterion to select a specific solution from the elite bee list for exploration, and generates a series of new candidate solutions around each selected elite bee individual. The expression is shown in Equation (13).

$$s'_{i,j} = s_{r,j} + rand(-a,a)(x_{r_1,j} - x_{r_2,j}) \tag{13}$$

In Equation (13), $x_{r_1,j}$ and $x_{r_2,j}$ represent the corresponding values of two different solutions. $s'_{i,j}$ represents the generated new value. Therefore, based on the above calculations, the EABC process is shown in Figure 6.

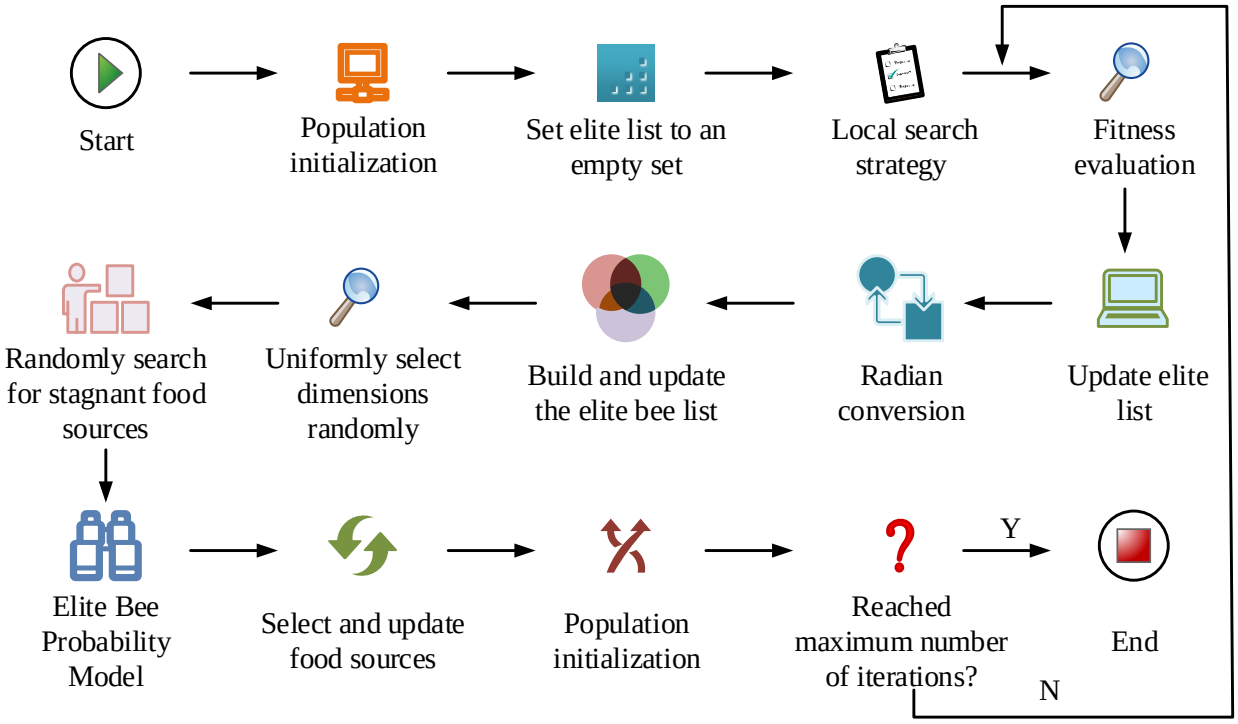


Figure 6: The process of EABC.

EABC first randomly generates an initial population and sets an elite bee list S. A local search strategy is applied to optimize the initial population, and then enter the loop iteration stage. Based on fitness values, the population is sorted. Then the elite bee list is constructed and updated. In the hiring bee step, the dimension is randomly selected for search, and new fitness values are calculated to update the elite bee list. The reconnaissance bee step randomly searches for stagnant food sources to prevent falling into local optima. Subsequently, the elite bee probability model is used to select reference scholars and apply search equations. The food sources are selected and updated in the bee observation step, and useless elite bees are deleted. When the maximum iteration is satisfied, the iteration is terminated and the optimal solution is output.

3 Results

To exhibit the performance of the designed model based on the EABC model, a series of tests were conducted on EABC first. In the second section, simulation tests were conducted on the logistics vehicle path planning model based on EABC to verify the practical application effect of the model.

3.1 Performance testing of improved EABC algorithm

The study used Windows 10 as the operating system, Intel Core i7 CPU, NVIDIA GeForce GPU, 64GB of memory, and Matlab. Firstly, the EABC algorithm was tested on several benchmark functions to assess the effectiveness of the improved ABC. ABC, Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) were used as comparative algorithms. In the experiment, the maximum iteration was 1000, the number of bee colonies was 30, and the dimension was 20. In both unimodal and multimodal function tests, each set of tests was repeated 10 times and averaged. In the results of both test functions, there was a significant difference ($P<0.05$).

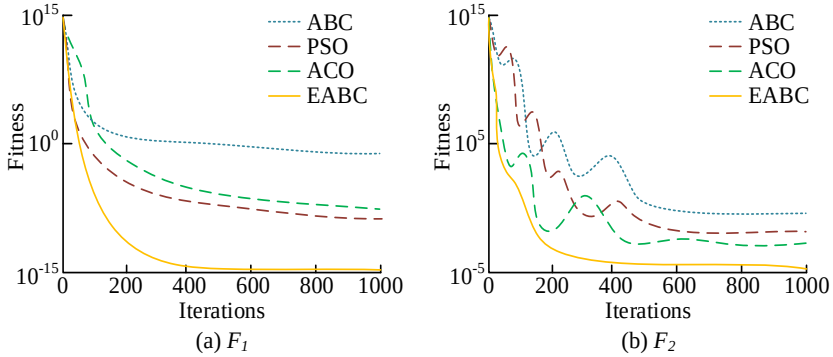


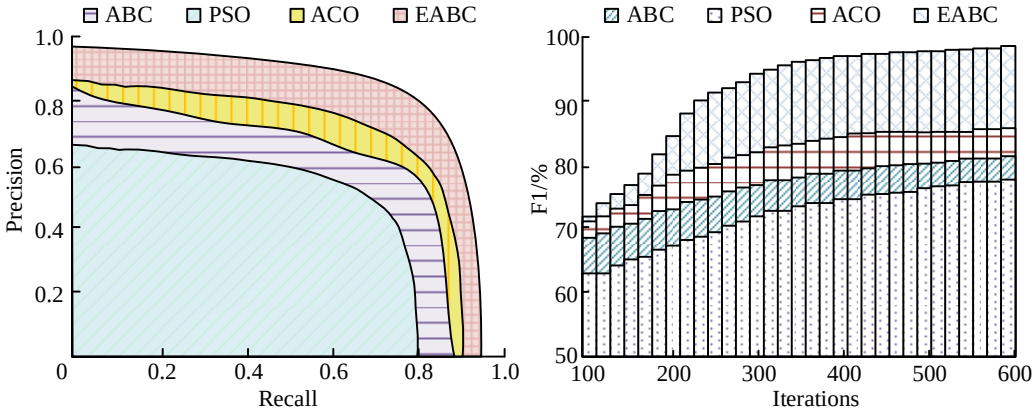
Figure 7: Convergence curves of each algorithm.

Table 2: Mean and standard deviation results.

Function			Index	ABC	PSO	ACO	EABC
Unimodal function	Sphere	F_1	Avg	8.17e-12	7.75e-14	2.75e-15	2.75e-20
			Std	2.78e-12	3.61e-14	4.85e-15	6.51e-20
	Rosenbrock	F_2	Avg	8.22e-14	7.34e-17	2.84e-18	2.64e-21
			Std	2.52e-14	3.23e-18	6.54e-18	6.41e-21
Multimodal function	Rastrigin	F_3	Avg	6.55e-11	6.41e-10	2.57e-13	2.45e-16
			Std	2.93e-11	2.95e-10	4.28e-13	5.49e-16
	Griewank	F_4	Avg	8.79e-12	6.84e-12	5.74e-15	3.61e-19
			Std	7.55e-12	5.22e-12	3.87e-15	9.45e-19

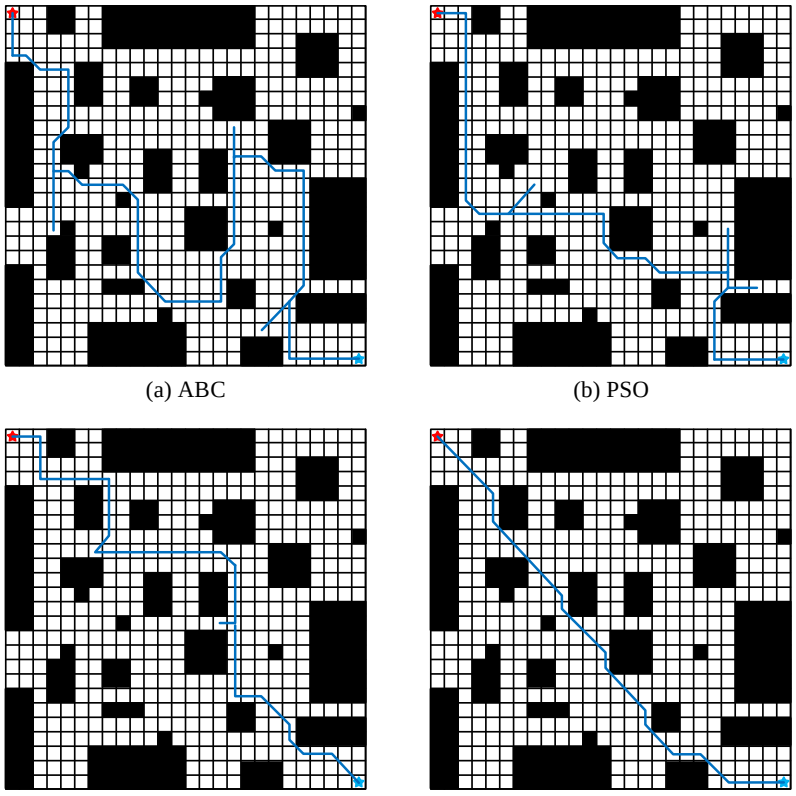
Table 2 shows the optimized mean and standard deviation of ABC, PSO, ACO, and EABC algorithms running 100 times on the uni-modal functions Sphere and Rosenbrock, as well as on the multimodal functions Rastrigin and Griebank. The EABC algorithm achieved optimal optimization mean and standard deviation on both unimodal and multimodal benchmark functions. Compared with the traditional ABC algorithm, it had made progress. To further quantify the impact of the introduced elite bee colony strategy on convergence speed, the convergence curves of the four models on different benchmark test functions are shown in Figure 7.

Figures 7 (a), 7 (b), 7 (c), and 7 (d) show the convergence curves of four algorithms tested on F_1, F_2, F_3, and F_4, respectively. In Figure 7 (a) and Figure 7 (b), the EABC had the optimal convergence speed and fitness values on the two uni-modal benchmark test functions F_1 and F_2, reaching convergence after 425 and 381 iterations, respectively. On the benchmark functions F_3 and F_4 in Figure 7 (c) and Figure 7 (d), the EABC algorithm achieved convergence at the 409th and 361st iterations, respectively, confirming the feasibility of introducing elite bee colonies to improve convergence speed. Finally, the study introduced the VRPTW dataset, which was used to address the time window constraints added in vehicle path optimization problems. It was commonly used to test the performance of various vehicle path optimization algorithms. The Precision Recall (PR) curves and F1 values of each algorithm on the dataset are shown in Figure 8.



(a) PR curves test (b) F1 value test

Figure 8: PR curve and F1 value results.



(a) ABC (b) PSO

(c) ACO (d) EABC

Figure 9: Paths of various algorithms in simple environment.

Figures 8 (a) and 8 (b) display the PR curves and F1 value test results of four algorithms on the dataset, respectively. In Figure 8 (a), the area enclosed by the PR curve and coordinate axis of ABC, PSO, ACO, and EABC algorithms was 0.84, 0.68, 0.87, and 0.96, respectively. In Figure 8 (b), as the number of iterations increased, the F1 values of each algorithm showed a gradual upward trend in the early stages and tended to stabilize in the later stages. When the number of iterations was 600, the F1 values of each algorithm were 82.4%, 78.2%, 86.5%, and 97.7%, respectively. The EABC had good comprehensive performance.

3.2 Simulation testing of logistics path planning based on EABC

On the basis of verifying the performance of EABC algorithm, the research further tested the practical application effect of logistics vehicle path planning based on EABC. Firstly, the path search simulation experiments of ABC, PSO, ACO, and EABC algorithms in a simple planar environment are displayed in Figure 9.

Figures 9 (a), 9 (b), 9 (c), and 9 (d) display the path search results of ABC, PSO, ACO, and EABC algorithms in a simple two-dimensional environment, respectively. In Figure 9 (a), the ABC algorithm generated 27 inflection points and showed 3 local optima. Because in the honey source update stage of the ABC algorithm, the variation in location between the current honey source and other randomly selected honey sources can easily lead to getting stuck in local optima, resulting in the inability to

exit the current search area. In Figure 9 (b), the PSO algorithm generated 18 turning nodes and fell into local optima 3 times. In Figure 9 (c), the ACO algorithm generated 16 turning nodes, which were trapped in local optima twice. In Figure 9 (d), EABC only had 11 inflection points and not trapped into a local optimal solution. From this, the improved EABC had good convergence accuracy and speed, which could achieve excellent optimization efficiency in path optimization simulation testing. Subsequently, Matlab was used to simulate and model the logistics distribution points, namely customer locations, including 1 distribution center and 20 customer points to be delivered, with a total of 6 vehicles. All four models underwent 10 simulation experiments, and the optimal delivery path obtained is shown in Figure 10.

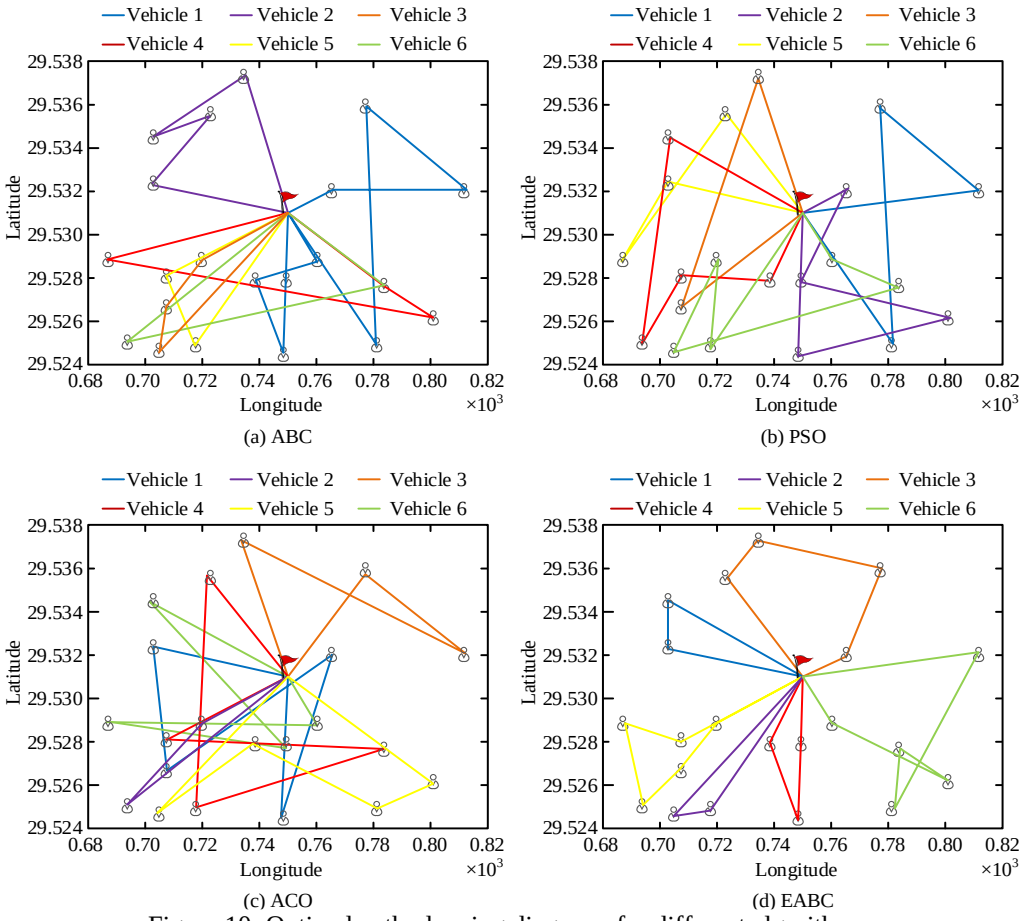


Figure 10: Optimal path planning diagrams for different algorithms.

Figures 10 (a), 10 (b), 10 (c), and 10 (d) display the optimal path planning diagrams obtained from 10 simulation experiments for the ABC, PSO, ACO, and EABC models, respectively. The red flag represents the logistics distribution center, and the user icon represents the delivery point. There were many invalid turns and repeated paths in Figures 10 (a), 10 (b), and 10 (c). The EABC had the least number of invalid turns and short repeated paths, resulting in the best path planning results. To further quantify the results, the various costs of each model in the above path planning are displayed in Table 3.

Table 3: Multi indicator test results.

Model	Index						
	Total route length/km	**Average load rate/%**	**Fuel cost/¥**	**Freight loss cost/¥**	**Time delay cost/¥**	**Total delivery cost/¥**	**Calculation time/s**
ABC	5512.93	84.5	2483.30	22051.72	1339.64	88146.24	20.2
ACO	5381.56	85.3	2424.13	22423.17	1318.48	90141.13	18.9
PSO	5624.35	92.8	2533.49	20908.36	1282.35	85377.63	22.4
EABC	5221.43	97.4	2341.45	20719.96	1268.81	83908.38	2.8

Table 4: Compares the time complexity of the three algorithms.

Sample size	Model type	Average processing time (ms)	Standard deviation (ms)	Time complexity $O(n^2)$ evaluation
1000	EABC	21.4	2.0	Lower
	ABC	20.8	2.6	Lower
	ACO	31.5	3.4	Normal
2000	EABC	78.5	5.2	Lower
	ABC	75.6	7.8	Lower
	ACO	107.4	10.2	Normal
3000	EABC	177.8	8.3	Lower
	ABC	176.2	9.1	Lower
	ACO	357.6	12.5	Higher

Table 5: Sensitivity of different algorithms for analysis.

Number of distribution points	Arithmetic	Load factor/%	Total cost/¥	Route planning time/s
20	EABC	97.4	83908.38	2.8
	ABC	84.5	88146.24	20.2
	ACO	85.3	90141.13	18.9
40	EABC	95.3	146503.21	3.1
	ABC	80.2	185724.18	35.3
	ACO	81.5	209516.54	22.5
60	EABC	94.6	197892.15	5.2
	ABC	79.2	253815.96	60.3
	ACO	80.1	286575.56	51.6

Table 3 displays the path length, loading rate, fuel consumption cost, freight loss cost, time window penalty cost, delivery cost, and solution time of each optimal path. As shown in the table, compared with traditional ABC and other comparative models, the logistics vehicle path planning based on EABC had significant advantages in path length, loading rate, and various costs. The vehicle loading rates for the optimal paths of ABC, PSO, ACO, and EABC models were 84.5%, 85.3%, 92.8%, and 97.4%, respectively. In terms of running time, the solution time for the four models was 20.2s, 18.9s, 22.4s, and 2.8s, respectively. In order to analyze the computational complexity of the EABC algorithm and the comparison algorithm, the time required by the two algorithms to process the same amount of data under the same hardware conditions was measured experimentally. The results of the time complexity comparison of the three algorithms are shown in Table 4.

In Table 4, when the sample size was 1000, 2000 and 3000, the average processing time of the EABC algorithm was 21.4ms, 78.5ms and 177.8ms respectively, which was only slightly higher than that of the benchmark algorithm ABC, and much lower than that of the ACO algorithm. The standard deviation of EABC algorithm was the minimum value, and the processing time of the algorithm was the most stable. The results showed that the EABC algorithm had lower computational complexity and higher computational efficiency, and with the increase of the number of samples, the growth rate of time complexity $O(n^2)$ was much lower than that of the ACO algorithm. For the complexity and variability of the logistics and distribution environment, the study analyzed the sensitivity of different algorithms, and a comparison of the results is shown in Table 5.

In Table 5, as the number of distribution points increased, the vehicle load factor of each algorithm was gradually decreasing, and the total distribution cost and route planning time were gradually increasing. However, the load factor of the EABC algorithm decreased more slowly than the ABC and ACO algorithms, and the load factor of the three algorithms decreased by 2.8%, 5.3%, and 5.2%, respectively, when the number of distribution points was 60. The total cost of the three algorithms increased by ¥113,983.77, ¥165,669.72, and ¥196,434.43, and the path planning time improved by 2.4s, 40.1s, and

32.7s, respectively.

4 Discussion

Aiming at the existing logistics distribution path optimization methods, the problems of long distribution path, high distribution cost and insufficient vehicle loading rate, the study proposed an elite bee colony algorithm to optimize the logistics distribution path. The experimental results showed that the EABC algorithm reached convergence at 400 iterations in both single-mode and multi-mode benchmark test functions, and the convergence speeds were better than other algorithms, and the optimal fitness values obtained were around 10-15.In the test functions, the ABC, PSO and ACO algorithms reached convergence at 500, 600 and 800 iterations, respectively.The EABC algorithm achieved the convergence at 500, 600 and 800 iterations, respectively.The EABC algorithm, through the hiring bees stage in the The EABC algorithm had fewer inflection points as it used two different searches to obtain different solutions, and the higher value of adaptation was used as a candidate solution to reduce the generation of local optimal solutions. The population size of the elite bees was selected through adaptive parameters, and in the early stages of the algorithm's operation, the number of elite bees was higher, focusing on extensive search and determining the approximate range of nectar sources. As the number of iterations increased, the number of elite bees decreased and the focus was on improving the search accuracy of the optimal solution. This helped to perform fast search at early stages and reduce resource consumption and improve search accuracy at later stages. The proposed EABC algorithm was not only used for path optimization in logistics and distribution, but also in the field of takeaway delivery with similar attributes, urban services such as street cleaning and garbage collection, and in traffic management and emergency rescue, etc., which could improve the efficiency and reduce the cost of work.The EABC algorithm could still fall into local optimal solutions when the performance of the test set was poor, and subsequently, it could be added to the algorithm with a forbidden search strategy could be added to the algorithm to store the local extremes in the taboo table to further avoid local optimal solutions.

5 Conclusion

Traditional logistics path optimization methods typically only consider fixed transportation needs and paths, without taking into account other potential transportation cost. The study introduced the elite bee colony mechanism to enhance the global and local search capabilities of ABC. A logistics vehicle path optimization model was constructed based on EABC. The performance test results showed that EABC achieved the optimal optimized mean and standard deviation on both uni-modal and multi-modal benchmark functions. On the convergence curve, EABC had lower fitness than other algorithms, with the best convergence speed and accuracy. The area enclosed by its PR curve and coordinate axis was 0.96, and the F1 value at 600 iterations was 97.7%. In a simple two-dimensional environment experiment, the inflection point generated by the EABC optimal path was only 11, and the number of times it fell into a local optimal solution was 0. In the optimal path test of 20 customer points, the total path length of EABC was 5221.43km, the total delivery cost was 83908.38 yuan, the vehicle loading rate was 97.4%, and the solution time was 2.8s. From this, he logistics path intelligent optimization model based on EABC has shown significant advantages in improving logistics efficiency and reducing transportation cost, which has broad application prospects. EABC algorithms are not only applied to logistics and distribution, but also in the field of takeaway delivery with similar attributes, and can be applied to urban services, such as street cleaning and garbage collection, etc., as well as in traffic management to reduce congestion and improve the efficiency of road use. In the future, more influencing factors still need to be considered. Further in-depth research is necessary in large-scale practical applications and algorithm optimization to achieve more efficient and practical logistics path optimization solutions.

References

[1] Liu Y. Design of dynamic programming model for multi-objective cold chain logistics deployment path based on meme algorithm. Iranian Journal of Science and Technology, Transactions of Civil Engineering. 2021, 46(3): 2553-2560. https://doi.org/ 10.1007/s40996-021-00639-2.

[2] Ren J, Salleh SS. Green urban logistics path planning design based on physical network system in the context of artificial intelligence. The Journal of Supercomputing. 2024, 80(7): 9140-9161. https://doi.org/10.1007/s11227-023-05796-x.

[3] Almazroi AA, Ayub N. Hybrid Algorithm-driven smart logistics optimization in IoT-based cyber-physical systems. Computers, Materials & Continua. 2023, 77(3): 3921-3942. https://doi.org/10.32604/cmc.2023.046602.

[4] Xue T, Li L, Shuang L, Zhi D, Ming P. Path planning of mobile robot based on improved ant colony algorithm for logistics. Mathematical Biosciences and Engineering. 2021, 18(4): 3034-3045. https://doi.org/10.3724/sp.j.1087.2008.02877.

[5] Gams M, Kolenik T. Relations between electronics, artificial intelligence and information society through information society rules. Electronics. 2021, 10(4):514. https://doi.org/10.3390/electronics10040514.

[6] Farahbakhsh H, Pourfar I, Lashkar Ara A. A modified artificial bee colony algorithm using accept–reject method: Theory and application in virtual power plant planning. IETE Journal of Research. 2023, 69(8): 5364-5379.

https://doi.org/10.1080/03772063.2021.1973597.

[7] Huo F, Wang Y, Ren W. Improved artificial bee colony algorithm and its application in image threshold segmentation. Multimedia Tools and Applications. 2022, 81(2): 2189-2212. https://doi.org/10.1007/s11042-019-08231-7.

[8] Zhang L, Ren Z, Liu T, Tang J. Improved artificial bee colony algorithm based on Harris Hawks optimization. Journal of Internet Technology, 2022, 23(2):379-389. https://doi.org/10.53106/160792642022032302016.

[9] Yilmaz Acar Z, Başçiftçi F. Solving multi-objective resource allocation problem using multi-objective binary artificial bee colony algorithm. Arabian Journal for Science and Engineering. 2021, 46(9):8535-8547. https://doi.org/10.1007/s13369-021-05521-x.

[10] Bao S, Tawada M, Tanaka S, Togawa N. An ising-machine-based solver of vehicle routing problem with balanced pick-up. IEEE Transactions on Consumer Electronics. 2023, 70(1):445-459. https://doi.org/10.1109/vlsi-dat52063.2021.9427355.

[11] Sun W, Zou Y, Zhang X, Wen Y, Du G, Liu J, Wu J. Joint routing and scheduling optimization of in vehicle time sensitive networks based on improved grey wolf optimizer. IEEE Internet of Things Journal. 2023, 11(4): 7093-7106. https://doi.org/ 10.1109/jiot.2023.3315286.

[12] Lin Y, Hu G, Wang L, Li Q, Zhu J. A multi-AGV routing planning method based on deep reinforcement learning and recurrent neural network. IEEE/CAA Journal of Automatica Sinica. 2023, 11(7): 1720-1722. https://doi.org/10.1109/jas.2023.123300.

[13] Chen Y, Xue J, Zhou Y, Wu Q. An efficient threshold acceptance-based multi-layer search algorithm for capacitated electric vehicle routing problem. IEEE Transactions on Intelligent Transportation Systems. 2024, 25(6): 5867-5879. https://doi.org/10.1109/tits.2023.3344086/mm1.

[14] Li Y, Zhao Q, Yang S, Guo Y. Tailoring evolutionary algorithms to solve the multiobjective location-routing problem for biomass waste collection. IEEE Transactions on Evolutionary Computation. 2023, 28(2): 489-500. https://doi.org/ 10.1109/tevc.2023.3265869.

[15] Kalita, K, Ganesh, N, Shankar, R, & Chakraborty, S. A fuzzy MARCOS-based analysis of dragonfly algorithm variants in industrial optimization problems. Informatica, 2024, 35(1), 155-178. https://doi.org/10.15388/23-infor538.

[16] Tang J, Liu G, Pan Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. IEEE/CAA Journal of Automatica Sinica. 2021,8(10):1627-1643. https://doi.org/10.1109/jas.2021.1004129.

[17] Du H, Cui Q, Liu P, Ma X, Wang H. Reformative artificial bee colony algorithm based PID controller for radar servo system. Electron. Res. Arch. 2022, 30(8):2941-63. https://doi.org/10.3934/era.2022149.

[18] Solgi R, Loáiciga HA. Bee-inspired metaheuristics for global optimization: a performance comparison. Artificial Intelligence Review. 2021, 54(7): 4967-4996. https://doi.org/10.1007/s10462-021-10015-1.

Automatic Ink Painting Rendering Technique Based on Deep Triangular Convolutional Network

Fu Wang[1,2*]
[1]Xuchang Vocational Technical College, Xuchang 461000, China
[2]Henan Network Engineering Intelligent Service Engineering Research Center, Xuchang 461000, China
E-mail: wangfu86@126.com
[*]Corresponding author

Keywords: automatic rendering, deep convolutional neural networks, digital images, feature extraction, ink painting

Received: September 9, 2024

In order to generate works with rich details and unique ink style in a short period of time, a deep triangular convolution network based on deep triangle convolution is proposed. Firstly, convolutional neural networks are used to extract feature textures from stylized images and apply them to noisy images to generate ink painting textures with specific styles. Then, by constructing a weighted sum of content loss and style loss, the effect of automatically adding ink painting style to the content image can be achieved. The deep triangular convolutional network structure adopted can effectively avoid the problem of gradient vanishing and improve the training efficiency of the network. The experimental results show that the testing time of the deep triangular convolutional network is 1.5 seconds, and the classification error rate is only 3.5%. The accuracy of the network reached 95.5%, significantly higher than Style Migration and others. The proposed method maintains high recognition accuracy and image quality while achieving style conversion in ink painting. The research provides new directions for the development of art, which helps to inherit and disseminate Chinese ink painting art.

Povzetek: Predlagana je tehnika avtomatskega upodabljanja s pomočjo globokega trikotnega konvolucijskega omrežja (DTCN). Metoda izboljša prepoznavanje umetniških značilnosti in omogoča kakovostno digitalno obdelavo stiliziranih slik.

1 Introduction

With the vigorous development of information technology, deep learning for artistic style images has become an integral part of machine learning. However, the current learning of artistic images is limited to the mechanical transfer of pixel features, and there has been limited in-depth research in the field of artistic style itself. Both artistic deep learning and technical operations are still in their early stages of development [1]. In the field of image processing, machine learning and neural network algorithms have emerged as promising methods, and they have gained significant attention from researchers [2]. Neural networks are computational models that mimic the information processing and learning mechanisms of the human nervous system. They consist of a large number of artificial neurons and their interconnections, and have been extensively studied and applied in classification, prediction, and data processing domains [3]. Artificial intelligence assistants have gradually become an effective tool to assist artists in the traditional process of ink painting creation, which takes a long time, involves a lot of manual intervention, and the quality of painting is limited by the artist's personal skill level. Applying deep learning technology to ink painting creation aims to achieve automated, efficient, and high-quality ink painting, which has important research significance and practical value.

In this study, through in-depth analysis of the characteristics of traditional ink painting, a new optimization network based on deep convolutional neural network (CNN) is proposed, called deep triangular convolutional network (DTCN). This model is used to extract artistic features from ink wash paintings, and then fuse the extracted features with human and landscape images, ultimately achieving the rendering application of digital image ink style while obtaining high-quality ink wash images. Through experimental verification, the proposed method has achieved significant results in terms of drawing quality, drawing speed, and automation level. In addition, this technology can also be extended and applied to other art fields, including automated creation of traditional Chinese painting, animation production, etc., providing new technical support for the development of the digital art industry.

2 Related works

Manual image discernment and processing can be time-consuming. However, since the advent of neural network algorithms, there have been significant research efforts to automate image processing, thereby freeing up human resources and improving the accuracy and speed of image processing. A W Q and their team proposed an image-based bubble tracking algorithm for plate heat exchangers, utilizing CNN to process images and obtain the position

and state of bubbles in the flow. Based on this information, the algorithm calculated the movement and velocity of the bubbles. Even in scenarios with a large number of bubbles and dense distributions, this method achieved an average accuracy exceeding 94% and a recall rate exceeding 87% [4]. Peng Y et al. introduced a gated CNN pre-training algorithm for high-resolution reconstruction of remote sensing images. The algorithm consisted of several residual blocks with skip connections. Each residual block contained a specially designed gated convolutional unit that assigns different weights to high-frequency and low-frequency information, controlling information transmission and allowing the main network to focus on learning high-frequency information. Experimental results on various remote sensing datasets demonstrated the accuracy and visual improvement of the proposed PGCNN, outperforming several state-of-the-art methods [5]. Thangarajan S K and collaborators proposed a hybrid classifier based on CNN for assisting in the diagnosis of brain tumors using magnetic resonance imaging. In this algorithm, CNN utilized a three-level discrete wavelet transform input with optimized hidden neurons. Experimental results showed that the proposed algorithm achieved a 17% higher accuracy compared to the Support Vector Machine (SVM) algorithm and a 15.5% higher accuracy compared to traditional CNN [6]. OD A and their team addressed the diagnosis of glaucoma by proposing a diagnostic method based on graphs and CNN. CNN played a crucial role in interpreting enhanced image data. In the ORIGA dataset, the proposed method achieved accuracy, the area under the curve (AUC), and precision all exceeding 90%, with the accuracy reaching 82% in practical diagnosis [7]. Ghosh S K and colleagues introduced a novel retinal image segmentation method using a CNN based on a Ranked Support Vector Machine (rSVM) to detect diabetic retinopathy. The design of CNN incorporated rSVM to define consistent class labels within the network, reducing the number of channels that aid in rapid convergence. With the help of post-processing steps, the method achieved good segmentation accuracy exceeding 90% in different databases [8]. Deshpande A M and their team constructed a deep neural network based on global residual learning to automatically remove various types of stripes and grain noise from remote sensing images, thereby enhancing their quality. The proposed method exhibited improved performance compared to existing techniques in terms of peak signal-to-noise ratio and structural similarity index measurements [9].

There are also certain research achievements in the field of utilizing intelligent algorithms to process art works. Xie X and the partner developed a painting art style rendering system based on generative adversarial networks. This algorithm can suppress the limitation of paired samples and improve the quality, stability, and convergence speed of images generated by painting art styles [10]. Platkevi A led his team to propose an example-based animation style rendering method. This method used moving examples to extract dynamic features of animation and achieve natural transitions between single frames [11]. In order to accurately extract the artistic style features of art works, W Yao and the partner developed a color feature-based color extraction method. This approach combined corner detection, edge contour feature detection methods, and mainly used the fuzzy clustering method for style extraction [12]. The summary table for related works is shown in Table 1.

Table 1: Summary table for related works.

Researchers	Method	Results	Limitation
Wang et al [4]	Image based bubble tracking algorithm for plate heat exchanger	The average accuracy exceeds 94%, and the recall rate exceeds 87%	Affected by factors such as image quality and bubble morphology
Peng [5]	Gate controlled CNN pre training algorithm	Image quality improved by 11%	Applicability on different scenarios and datasets
Thangarajan [6]	Hybrid classifier based on CNN	Compared to traditional CNN, the accuracy has increased by 15.5%	Affected by factors such as image quality and tumor type
Deperlioglu [7]	A glaucoma diagnosis method based on graphics and CNN	Accuracy, AUC, and precision all exceed 90%	Limited sample and clinical data
Ghosh [8]	CNN based on rSVM	The segmentation accuracy exceeds 90%	Affected by factors such as image quality and lesion morphology
Deshpande [9]	Deep neural network based on global residual learning	The structural similarity index can reach 85%	Not suitable for complex scenes and noise types
Xie [10]	A painting art style rendering system based on generative adversarial networks	The image quality can reach 90%, and the stability can reach 92%	Sample and artistic style limitations
Platkev [11]	Example based animation style rendering method	Realize natural transitions between single frames	Example quality and animation style
Yao [12]	Art style feature extraction method based on color features	The accuracy can reach 86%	Applicability of different art styles

Analyzing the research conducted in recent years on image processing using neural networks, most of these studies have focused on remote sensing image processing and diagnostic applications in medical imaging. The application of CNN-based image processing techniques in the field of art is relatively less explored. The conclusions

drawn from industrial and medical fields cannot be directly extrapolated to the domain of art. In the field of art, there is also a certain gap in the application of intelligent algorithms that highly focus on specific artistic styles. Therefore, in this study, addressing the automatic rendering of ink painting images, a method based on deep CNN is proposed to fill this research gap in the art domain. The aim is to lay the groundwork for practical research outcomes in this area.

3 Deep CNN and its optimization

3.1 CNN algorithm

CNN achieves the requirement of minimizing data preprocessing by exploiting the connections between object features and reducing the number of parameters that need to be learned. CNN consists primarily of convolutional layers, pooling layers, and fully connected layers, as shown in Figure 1 [13]. In the hierarchical structure of CNN, image information as the input at the lowest level is processed in a specific order or propagated to different layers of the network [14]. The most salient features of the observed data in each layer can be extracted using filters. The schematic diagram of the model is shown below:

As shown in Figure 1, the process of transferring object data to different network layers follow a specific order. The primary step in the data transmission process is sending the original image to the input layer [15]. This is followed by combining the local features extracted from the local receptive field to generate feature maps in the C1 layer. Next, there is a summation, weighting, and function activation process to generate feature maps in the S2 layer. The subsequent steps continue the specific flow to realize the C3 layer and S4 layer. The final output data is obtained by converting the output feature maps into vectors through the fully connected layer. Specifically, the calculation Equation for the convolutional layer in CNN is as follows:

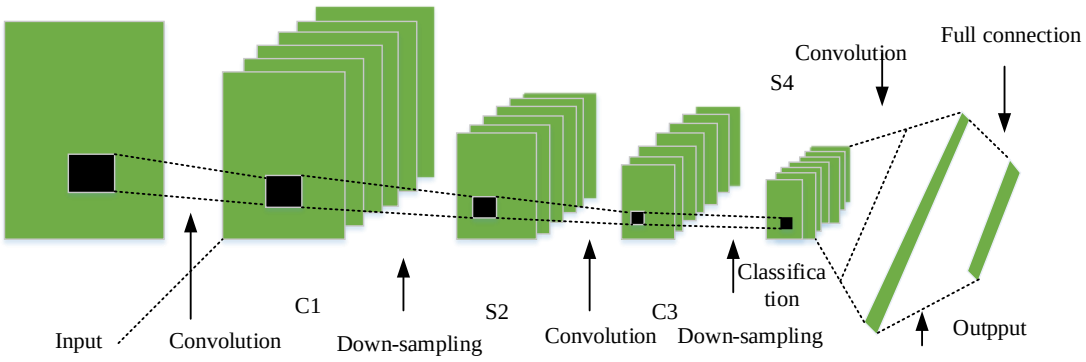


Figure 1: Deep convolutional neural network.

$$X_j^l = f\left(\sum_{i\in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{1}$$

Its algorithm learning consists of three levels: convolutional layer computation, pooling layer computation, and weight updates [16]. In the convolution process, the basic kernel convolves the original input image, and then the new feature maps are obtained through the activation function's non-linear mapping [17]. Different convolution kernels can extract different feature maps. In Equation 1, k_{ij}^l denotes the convolution kernel of layer l, l represents the convolutional layer number. X_j^l represents the j th output feature map in layer l, and the activation function is denoted by $f()$. The convolution operation is represented by *, and b_j^l is the number of feature maps in layer $l-1$. Additionally, the Equation for the pooling layer computation is as follows:

$$x_j^l = f\left(\beta_j^l down(x_j^{l-1}) + b_j^l\right) \tag{2}$$

Connected feature maps have corresponding relationships in the pooling layer [18]. To reduce the amount of data processing and retain useful information, feature maps need to be downsampled. In the Equation, $down()$ represents the downsampling function, and β represents the downsampling coefficient. Finally, the Equation for the weight update function is as follows:

$$W(t+1) = W(t) + \eta\delta(t)x(t) \tag{3}$$

After determining the residual calculation, further updates are required for weights and biases. Assuming that the current moment of CNN is t and the next moment is $t+1$, the Equation for weight update can be obtained. In the Equation, η represents the learning rate, $x(t)$ represents the input to the neuron, and $\delta(t)$ represents the error term. In summary, the algorithm learning of deep CNN is achieved through convolutional layer computations, pooling layer computations, and weight update computations. The algorithm learning directly impacts the training process of CNN.

3.2 Optimization

For a specific CNN, it should have an overall smooth downsampling [19]. In this study, an improved and optimized deep triangular CNN with triangular structural

units is proposed based on the foundation and achievements of traditional convolutional networks. Its main advantage is that it can gradually increase the dimensionality of feature maps through neural layers in the network. Using a deep triangular network can maximize the propagation of feature information in the backward process and greatly improve the utilization of feature maps. In simple terms, the triangular network plays a significant role in avoiding the gradient vanishing phenomenon. The system structure diagram is shown below:

The objective of this model is to achieve the dispersion of structural unit pressure by increasing the number of feature maps. As shown in Figure 2, the entire DTCN model consists of six modules. In each module, the first pooling layer uses max pooling, while subsequent layers use average pooling. The dimension of the feature map in the module output channel shows a linear growth trend, similar to a triangle. This network achieves a gradual increase in the dimensionality of the feature map while maintaining the original convolution kernel size unchanged. This enables the network to more effectively capture complex features in images, thereby improving recognition accuracy. Secondly, the triangular structural units in deep triangular networks help to improve the stability of gradient vanishing phenomena, enabling the network to converge faster during training. The schematic diagram of the feature map dimensionality at specific layers is shown below.

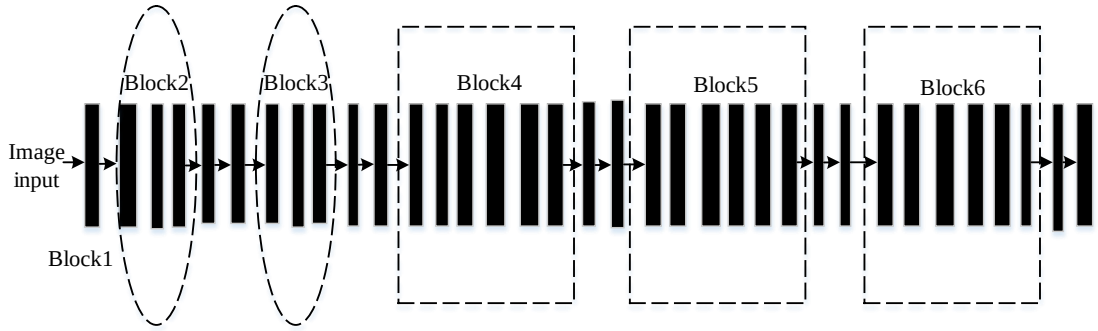
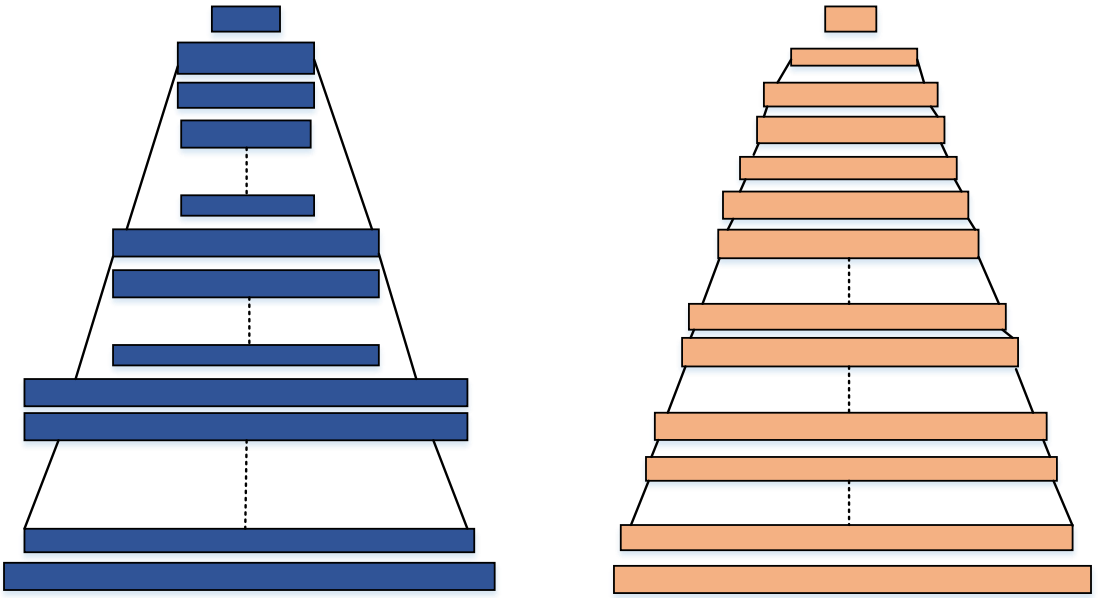


Figure 2: The deep triangular convolutional network structure.



(a)ResNet (b)DPCNet

Figure 3: Schematic diagram of feature dimension.

Table 2: DTCN feature dimension.

Output feature map size	**32*32**		**16*16**		**8*8**	
Network layer	Block1_conv1	Block2_conv2	Block2_pool2	Block3_conv3	Block3_pool2	Block4_conv4
Design	3*3 conv,64	[3*3 conv,64+aN]	Max 2*2("same")	[3*3 conv,64+aN]	Ave 2*2("same")	[3*3 conv,64+aN]

Figure 3: Shows the schematic diagram of the feature map dimensionality for the triangular structure of ResNet and DTCN. It can be clearly seen that starting from the input layer, the feature map dimensionality in the residual network module remains unchanged, and this continues until the downsampling layer where the dimensionality of the feature maps doubles. ResNet and DTCN are both composed of CNNs, but DTCN's connections allow for direct information transfer across layers, making network training more efficient and easier. Currently, DTCN is being widely applied in computer vision tasks such as image classification, object detection, and semantic segmentation. As a result, DTCN is the main focus of this study. The feature map dimensionality structure of DTCN is shown in Table 2. As the network layers change, there are significant variations in the design and output feature map sizes.

In the image processing process of DTCN, it is necessary to provide the Equation for computing network feature maps. Assuming an image passes through a CNN, and the network has $L+1$ layers. Here, shortcut connections are used in a concatenation manner to obtain the network feature maps. The use of concatenation allows any layer in the network to access the information from previous layers, and its input can be forwarded to subsequent layers. Under these settings, the Equation for computing the feature maps in the $L+1$ layer of the network is given by equation (4).

$$x(L+1)=H_{L+1}([x(0),x(1),\cdots,x(L+1)]) \tag{4}$$

In the Equation, x(L+1) represents the feature map of the image x in the L+1 th layer. $H_{L+1}()$ represents the non-linear transformation operation. The image first enters the convolutional layer and generates feature maps through convolution operations. Then, pass these feature maps to the next layer, while using shortcut connections to combine the feature maps of the previous layer with the input of the current layer. In each layer, the feature maps undergo nonlinear transformation operations to generate new feature maps, which are then passed on to the next layer. In this way, the feature map propagates layer by layer in the network until it reaches the final output layer. Throughout the process, the dimensionality of the feature map increases linearly with the increase of network layers until the final output result is achieved. Additionally, the activation function is improved based on the ReLU function.

$$ReLU(x)=max(0,x) \tag{5}$$

The ReLU activation function is primarily effective in eliminating non-negative elements. However, the use of activation functions also has its limitations, such as causing filters to generate a large amount of redundant information. As this can impact the accuracy of the research, this study introduces a new activation function based on ReLU to effectively eliminate pairing phenomena and various redundant information, meeting the requirements of DTCN. The mathematical expression of this new activation function is as follows:

$$CReLU(x)=\left[ReLU(x),ReLU(-x)\right] \tag{6}$$

The main advantage or functionality of the new activation function proposed in this study is that it can reduce the information redundancy caused by non-linearity in filters and also preserve the positive and negative phase information extracted by the filters. In the final stage of neural network design, the dropout technique is used to prevent potential overfitting issues in CNN [20]. Dropout randomly deactivates some neurons in the neural network by setting their outputs to zero during the training process. Specifically, each neuron has a certain probability of being dropped or retained. This can reduce complex dependencies between neurons and enhance the network's generalization ability [21]. The randomness of dropout prevents the neural network from overly relying on specific neurons, forcing the network to learn multiple independent features, thus improving model robustness and generalization ability. Additionally, using dropout helps reduce the number of parameters in the network, making it more concise and efficient. When the Dropout layer is set below the identity mapping module, there is no significant change in the performance of the neural network. Therefore, the Dropout network is set between the two convolutional layers in the triangular structure unit, as shown in Figure 4.

Figure 4 shows the difference between DTCN without a Dropout layer and the network with a Dropout layer, where the Dropout layer is set between different convolutional layers.

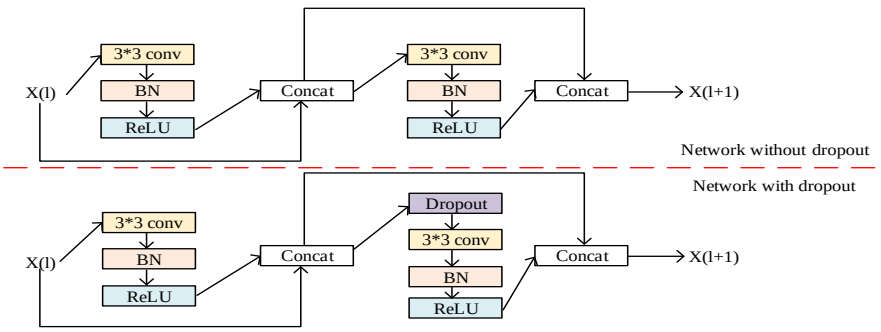


Figure 4: Triangular structure unit with Dropout.

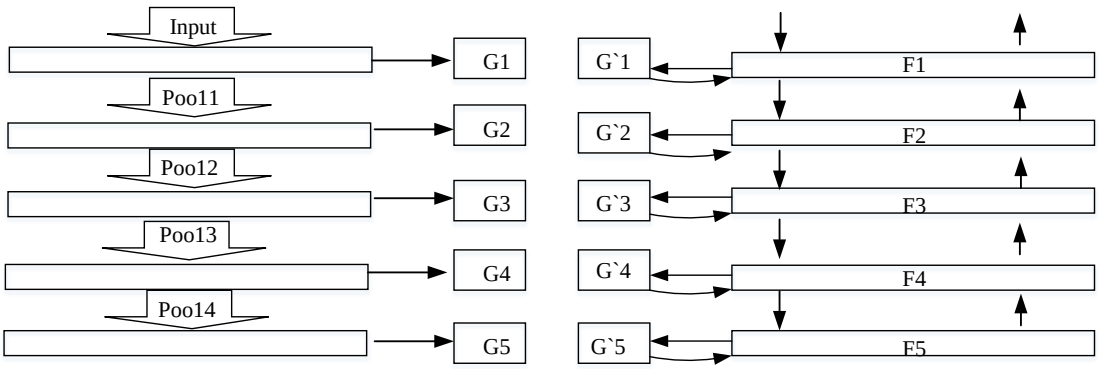


Figure 5: The process flowchart for generating feature textures.

4 Implementation of automatic ink painting rendering technique

4.1 Generation of feature texture

From a computer science perspective, artistic creation can be translated into the encoding process of applying specific algorithms to artistic content. Based on this idea, this study utilizes CNN to generate specific textures of artwork from noise images, achieving the process of feature texture generation. The process flowchart for generating feature textures based on deep CNN is shown in Figure 5 below.

The key steps for generating texture features from style images in the feature texture generation process can be clearly seen in the texture generation diagram. Firstly, the image is input into the VGG-16 network. Then, the crucial step is to calculate the feature responses F^1 for each layer, based on which the matrix G^1 representing the texture features of the style image is computed. Following the same principle or rule, the primary step in generating texture features from noise images is to calculate the feature responses $\hat{F}^1$ using VGG-19, resulting in the matrix $\hat{G}^1$. In the process of generating texture features from style images, E^1 represents the variance and loss of each layer in the network, and the image pixel $\hat{x}$ is updated through gradient descent. The matrix correlation Equation is as follows:

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \cdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix} \tag{7}$$

After obtaining the feature planes of the image from deep CNN, the next step is to consider how to utilize these features to describe the style texture of the image. The basic method can be summarized as building feature layers to calculate the correlations between different filter responses. In other words, matrix correlation is required in the actual texture generation process of the style image. For ease of calculation, the results can be normalized using the following Equation.

$$E = \frac{1}{4N^2M^2}\sum_{i,j}(G_{ij} - A_{ij})^2 \tag{8}$$

Once E is obtained, the following operations can be performed to complete the normalization.

$$Lstyle = \sum_{l=0}^{L} w_l E_l \tag{9}$$

By optimizing the algorithm, an image with the same style texture features as the original image can be generated. From equation (7), it is clear that the original image is denoted as A, while the noise image is represented by G. N denotes the number of feature maps, and M represents the size of the feature maps. The loss value of w_l serves as a weighting factor for the overall loss value, and a smaller value of w_l can weaken the influence of that layer.

4.2 Ink painting style rendering

Based on CNN, style rendering should follow specific principles that consider both user aspects and system development and deployment. Therefore, for ordinary users, this study aims to ensure good stability and convenience. The rendering module, as the core module in this research, mainly achieves the rendering functionality for ink painting art style. The specific structure diagram of the style rendering is in Figure 6.

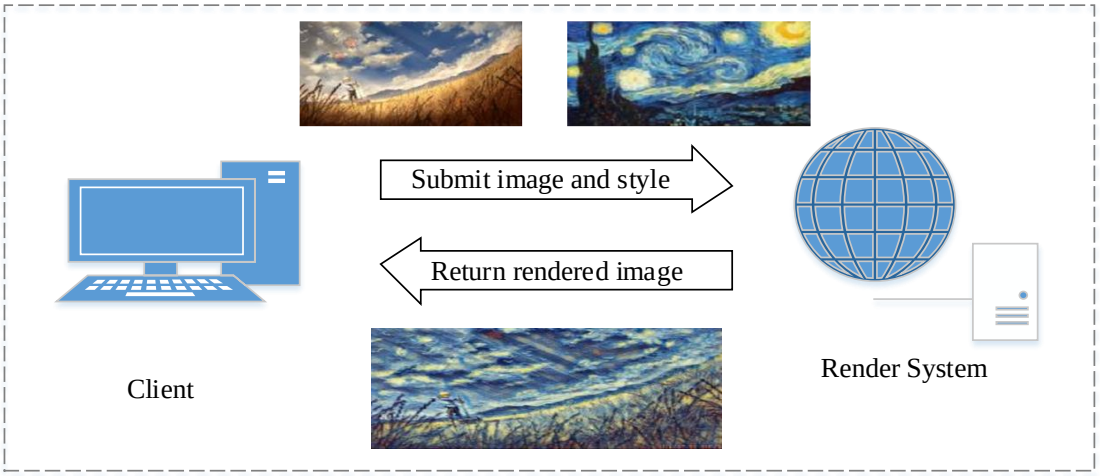


Figure 6: Style rendering structure.

After completing the generation of ink painting texture, it is necessary to render arbitrary images and generate ink painting style texture automatically. The main goal of this research is to achieve ink painting style effects on content images through rendering, and the basic approach or process to achieve this is to determine a total loss, which is the sum of content loss and style loss. Minimizing the total loss involves minimizing the sum of the noise image with the content image and the ink painting image. The mean squared error loss Equation for defining the content image is as follows:

$$Lcontent = \frac{1}{2}\sum_{ij}(F_{ij} - P_{ij})^2 \tag{8}$$

In the Equation, the content image is denoted as p. The response of filter i at position j in the content image and the noise image are represented as P_{ij} and F_{ij}, respectively. It should be noted that when the colors and thickness variations in the generated ink painting image appear abrupt or disharmonious, the total loss can be calculated by computing the mean squared difference between neighboring pixels. The calculation Equation is as follows.

$$L_{total} = \alpha L_{content} + \beta L_{style} + \gamma L_{noise} \tag{9}$$

In the Equation, L_{total} represents the total loss, $L_{content}$ represents the content loss, L_{style} represents the style loss, L_{noise} represents the noise loss, and α, β, γ represent the weighting factor of the content image, style image, and noise image.

5 Analysis

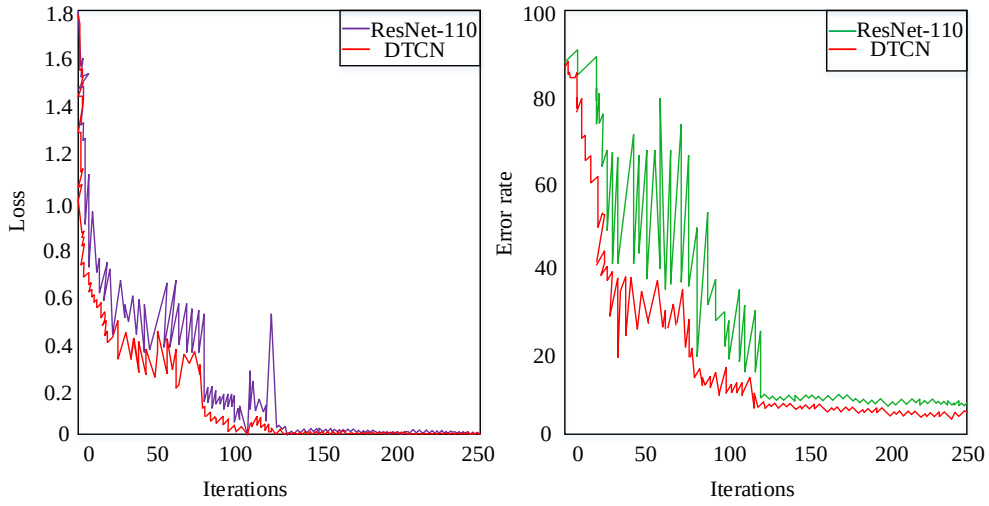
5.1 Comparison of residual network and DTCN performance.

Research shows that although residual modules can achieve or train deep networks based on their unique capabilities, this does not mean that residual networks are without drawbacks when it comes to output data. On the contrary, compared to plain networks, residual networks have inherent limitations and weaknesses. The main reason for this is that in residual modules, gradients are not forced to pass through the weights of the residual module while flowing through the network, making it difficult to achieve deep learning of useful information during the training process. The experimental analysis and results of this research are divided into two parts: the analysis of the proposed triangle network and the application effect of the proposed network in automatic ink painting style rendering. Firstly, the software and hardware environment for the experiments are specified, as shown in Table 3. To ensure smooth operation and prevent hardware limitations from affecting algorithm performance, the experimental machine is equipped with 32GB of memory. Since the experiment involves image processing and rendering, an NVIDIA GTX 3060TI GPU is used to ensure rendering efficiency.

The proposed triangle network in this research provides a solution to address these problems, and its performance is compared in Figure 7. From the specific training loss function curve and classification error rate curve in Figure 7, it can be observed that the triangle network has a lower training loss function value compared to the residual network. The comparison of the training loss functions between the two networks indicates that the triangle network has smaller differences between the ground truth and predicted values. This validates that the triangle network structure possesses stronger network optimization capabilities, its features can be effectively and sufficiently utilized, and the network converges faster. Additionally, the triangle network structure exhibits lower classification error rates compared to the residual structure, indicating better generalization ability.

Table 3: Experimental environment.

Experimental environment		Information
Hardware	CPU	Intel 12400
	RAM	32 GB
	GPU	GTX 3060TI
Operational system		Windows 10
Development tool		Java
Analysis tool		Matlab



(a) Training loss (b) Error rate of classification

Figure 7: Performance comparison between residual networks and deep triangular convolutional network.

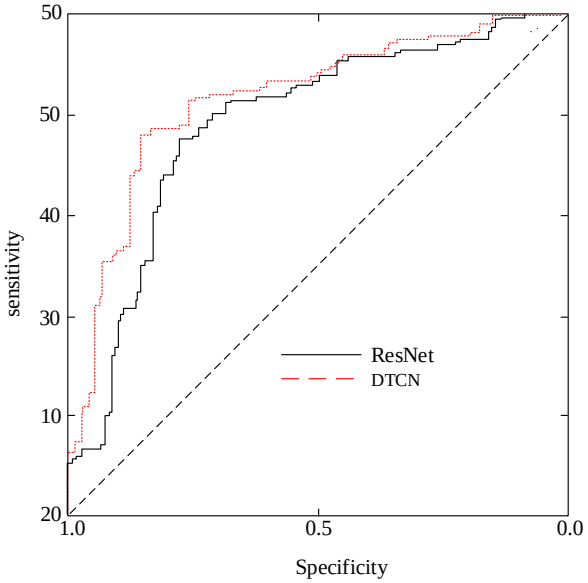


Figure 8: Receiver operating characteristic of the networks.

After analyzing and comparing the loss function curves of the algorithm, the performance and effectiveness of the algorithm are further evaluated by analyzing the Receiver Operating Characteristic (ROC) curves. ROC curves illustrate the classification accuracy of the model at different thresholds. ROC curves provide an intuitive representation of the overall performance of the classifier at different threshold settings and serve as a benchmark to select the optimal threshold. The ROC curves of the algorithm and its comparison with ResNet are shown in Figure 8. By examining the ROC curves of the two networks, it can be observed that the curve of DTCN is consistently above the curve of ResNet, and the AUC or DTCN reaches 0.801. AUC is a metric used to quantify the accuracy of model classification.

During the final stage of testing the DTCN network, its image processing capabilities on a real image dataset are analyzed, and the results are presented in Table 4. The dataset introduces for the experiment here is the Public Figures Face Database, which is a public data image from Columbia University with a data volume of approximately 58000 images. During the experiment, the segmentation ratio of the image dataset is 7:3. 70% of them are training sets, and the rest are testing sets. To ensure comparability of the results, the dimension of the input feature maps is set to 256 for each test. Firstly, normalize the image by scaling the pixel values to a range of 0 to 1 to eliminate the influence of brightness differences between different images; Next, contrast enhancement, sharpening, and other techniques are used to improve image clarity and further optimize recognition performance. With a parameter count of 30.1, DTCN exhibited the shortest testing time and the lowest classification error rate. At this point, the testing time is 1.5 seconds, and the classification

error rate is only 3.5%. Overall, based on the information in the table, the classification error rate of DTCN does not exceed 5%, with a maximum of 4.9%. The testing time also remains within 60 seconds. In practical applications, a theoretical error rate below 5% is generally considered sufficient to complete most image rendering tasks.

Table 4: DTCN performance under different parameter count.

Parameter count(M)	**30.1**	**54.5**	**92**	**136.7**
Depth	18	24	30	36
Training time (s)	25.9	34.7	45.5	59.3
Testing time (s)	1.5	2.1	2.5	3.3
Input feature map dimension	256	256	256	256
Output feature map dimension	512	608	703	800
Classification error rate (%)	3.5	4.6	4.8	4.9

In terms of computational complexity, DTCN has significant advantages over standard CNN and GAN. Compared to standard CNN and GAN, which often have millions of parameters, DTCN has significantly smaller parameter sizes. This enables DTCN to converge faster during training and inference, reducing the demand for computing resources. DTCN has a lower FLOP and requires less computation during training and inference. Compared to standard CNN and GAN, DTCN has particularly obvious advantages in FLOP. This makes DTCN more efficient in practical applications and can complete image rendering tasks faster. By directly comparing with standard CNN and GAN, it can be found that DTCN significantly reduces computational complexity and parameter count while maintaining a low classification error rate, thereby improving overall efficiency.

5.2 Convergence of the loss in automatic ink painting rendering

The main goal of this research is to achieve ink painting style effects on content images through rendering, and the basic approach or process to achieve this is to determine a total loss, which is the sum of content loss and style loss. Minimizing the total loss involves minimizing the sum of the noise image with the content image and the ink painting image. The α/β ratio determines the similarity between the noise image and the content image, where a larger ratio indicates a closer resemblance. Conversely, a smaller α/β ratio leads to a higher degree of rendering. This research visualizes the ink painting rendering images and their convergence of loss processes for α/β ratios of 1/30, 1/50, and 1/200.

As shown in Figure 9, the visualized ink painting rendering images and their convergence of loss processes are analyzed for α/β ratios of 1/30, 1/50, and 1/200. The final content loss value is less than 1200e+6, the style loss fluctuates around 2000e+6, the noise loss fluctuates around 3000e+4, and the total image processing loss fluctuates around 4000e+6. Specifically, compared to the style image, the noise image exhibits a larger range of variation in the loss, with rapid increases and decreases, ultimately converging gradually. On the other hand, the style image loss and the overall loss demonstrate fast convergence characteristics. In general, the loss values for image processing at various α/β ratios converge to a low level. Typically, the quality of the rendered images can be evaluated by the magnitude of the loss, where smaller losses indicate higher quality rendered images. Additionally, the ink fusion loss function is defined as follows.

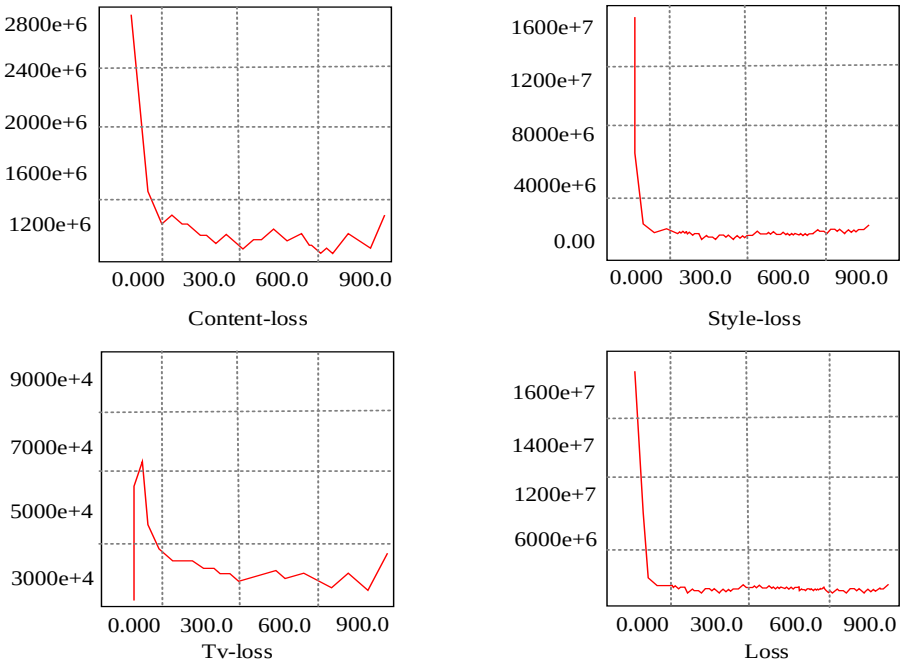


Figure 9: Convergence of image loss in ink painting rendering.

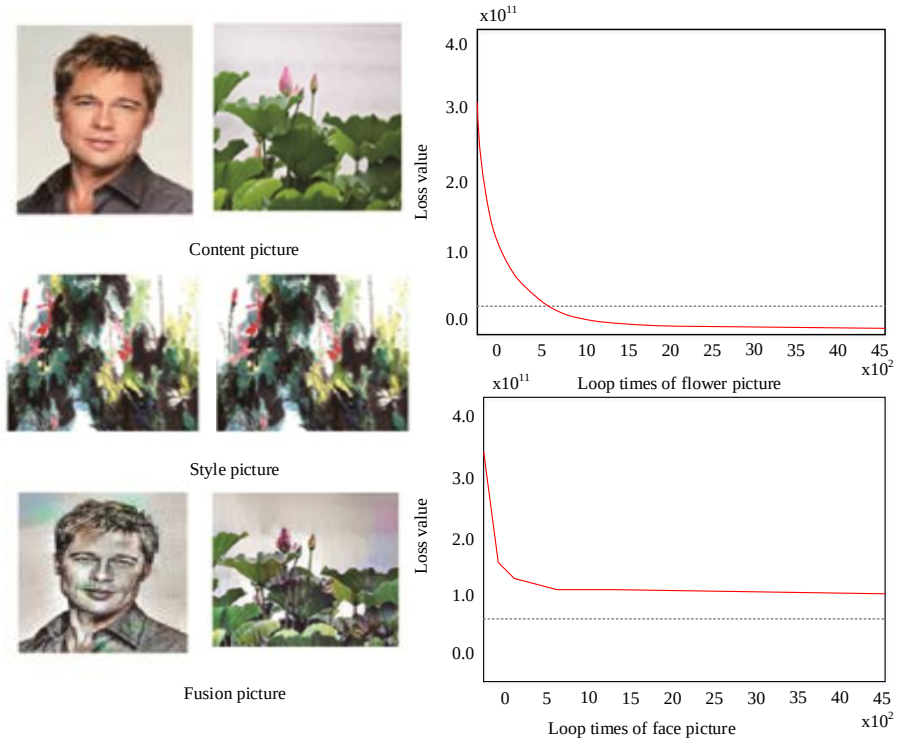


Figure 10: The fusion effect of lotus painting style and real lotus images.

According to Figure 10, when the lotus ink painting style image is fused with a real lotus image, the output image shows a good fusion of content and style, particularly in the lotus image. This demonstrates the achievement of the expected goal of this research. On the other hand, when fusing a real portrait image with the ink painting style image, although an image with ink painting features is obtained, the inherent differences between the portrait and the style image result in a lack of artistic characteristics in terms of fine texture and details. In summary, due to the deeper network layers of the neural network used in this research, it captures more hidden features, resulting in the preservation of more stylistic characteristics and better artistic representation in automatic rendering.

In order to further validate the advantages of the research method, experiments will compare Generative Adversarial Networks (GAN), Style Transfer, and Convolutional Neural Networks (CNN). The performance comparison results of different methods are shown in

Table 5. In terms of accuracy, the research method achieved 95.5%, significantly higher than GAN, Style migration, and CNN. This indicates that the research method has advantages in recognition accuracy. In terms of image quality, the research method is 90.3%, higher than GAN's 85.6%. In terms of satisfaction, the research method is 96.7% ahead of other methods, indicating that the research method can better meet the needs of users. In terms of recognition, the research method demonstrated its superiority at the highest level of 98.3%. Overall, the research method has good performance.

Table 5: Performance comparison results for the different methods.

Method	Accuracy	Picture quality	Degree of satisfaction	Recognition degree
Research method	95.5%	90.3%	96.7%	98.3%
GAN	90.2%	85.6%	92.3%	93.1%
Style migration	86.5%	84.1%	81.3%	90.3%
CNN	88.3%	85.2%	83.4%	91.3%

Table 6: Comparasting results of different methods.

Method	Error rate	Computing time	Memory usage ratio
DTCN	4.6%	560ms	20.3%
BiGAN	5.9%	780ms	26.4%
SRGAN	5.7%	986ms	37.7%

This is due to the advantage of DTCN in capturing sequence features, which can effectively learn the dependency relationships in the input sequence, thereby improving recognition accuracy. DTCN has optimized convolution kernels and pooling strategies, as well as appropriate activation functions. The selection of these parameters helps to reduce noise and distortion while preserving image details, thereby improving image quality. DTCN can adaptively adjust the style of generated images according to user needs, thereby better meeting user aesthetics. In addition, by introducing adaptive adjustment parameters, DTCN can maintain the content and structure of the original image during style transfer, further improving satisfaction. Meanwhile, the adversarial loss function used by DTCN during training can guide the model to generate images that are closer to real data.

To verify the superiority of the proposed DCTN network, the experiment compared Bidirectional GAN (BiGAN) and Super Resolution GAN (SRGAN). The comparison results of different methods are shown in Table 6. The results show that the DCTN network performs well in various indicators. Its error rate is 4.6%, lower than BiGAN's 5.9% and SRGAN's 5.7%; The computation time is 560ms, which has significant advantages compared to BiGAN and SRGAN; The memory utilization rate is 20.3%, significantly lower than BiGAN's 26.4% and SRGAN's 37.7%. This indicates that DCTN networks outperform BiGAN and SRGAN in terms of performance. These advantages enable DCTN networks to have higher efficiency and better performance in practical applications, which have important impacts on fields such as image processing and computer vision, and help improve the accuracy and practicality of related technologies.

6 Discussion

The proposed DTCN in the study has demonstrated significant advantages in multiple aspects and shows significant differences compared to existing technologies. Compared with existing ResNets, DTCN has a lower classification error rate and exhibits better generalization ability. In terms of convergence speed, the training loss function value of DTCN is lower, indicating that its network optimization ability is stronger and its features can be more fully utilized. This indicates that it performs well in feature propagation efficiency and can reach a steady state faster. In terms of artistic output quality, DTCN processed images have lower loss values and higher quality, which can better integrate content and style. For example, the fusion effect of lotus images is good, and more hidden features can be captured in deeper network layers, preserving more style features and artistic expression. In contrast, other methods such as GAN, style transfer, and traditional CNN perform slightly worse on these key metrics. The ability of DTCN to prevent gradient vanishing enables it to better learn useful information during training, which is one of the important reasons for its excellent results and provides better solutions and technical support for the development of related fields such as automatic ink painting rendering. Overall, the proposal of DTCN provides a new solution for automatic ink painting rendering technology. It not only improves the accuracy of classification and accelerates the convergence speed of the network, but also optimizes the quality of artistic output. These advantages make DTCN have broad application prospects in the field of automatic ink painting rendering.

7 Conclusion

With the significant improvement in computational performance, artificial intelligence technologies, represented by CNN, have rapidly developed. In this research, a deep CNN model was constructed to extract the texture features of ink paintings, and these texture

features were automatically rendered onto content images, resulting in content images with distinctive ink painting characteristics. Additionally, by comparing the performance of residual networks and triangle networks, it was concluded that the triangle network structure had stronger network optimization capabilities. It effectively and fully utilized features, and converged faster. When the parameter count was 30.1, DTCN demonstrated the shortest testing time and the lowest classification error rate. At this point, the testing time was 1.5 seconds, and the classification error rate was only 3.5%. Furthermore, during the training process, it was observed that the style image and total loss consistently converged rapidly, while the noise image loss experienced a rapid increase, followed by a rapid decrease before gradually converging. The accuracy of the research method reached 95.5%, significantly higher than that of comparative methods such as GAN. The research method is 96.7% ahead of other methods, indicating that the research method can better meet the needs of users. In summary, this research based on deep CNN techniques for automatic ink painting rendering not only obtained output images with distinctive ink painting characteristics but also achieved digital image ink style rendering applications.

The study provides an efficient and stable deep learning method for the transfer of ink painting styles. Its superior performance and high accuracy indicate that this method can better meet the needs of users in artistic creation. In the future, this method is expected to expand to other art fields, including oil painting, watercolor, etc., providing more possibilities for artists to create. Meanwhile, this research achievement also contributes to promoting the application of deep learning technology in cultural heritage protection and traditional art inheritance. Additionally, the potential of DTCN for other style images and the lack of diversity in image styles present opportunities for future research. The main limitation of this study is that the features of ink painting are insufficient in the detailed textures of some output images. How to make the correct features cover more image details is a problem that future research needs to focus on exploring and solving.

References

[1] Wan K, Liu X, Yu J, Zhang X, Du X, Guizani N. (2021). Compiler-based efficient CNN model construction for 5G edge devices. IEEE Transactions on Intelligent Transportation Systems, 22(8):5261-5274. https://doi.org/10.1109/TITS.2021.3056426

[2] Zhang C, Zhu F, Lv Y, Ye P, Wang F. MLRNN: Taxi demand prediction based on multi-level deep learning and regional heterogeneity analysis. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(7):8412-8422. https://doi.org/10.1109/TITS.2021.3080511

[3] Zhang S, Yin R, Zhang M. (2024). Dynamic unstructured pruning neural network image super-resolution reconstruction. Informatica, 48(7): 11-22. https://doi.org/10.31449/inf.v48i7.5332

[4] Wang QA, Li XA, Xu CB, Yan TA, Li YA. (2021). Bubble recognizing and tracking in a plate heat exchanger by using image processing and convolutional neural network. International Journal of Multiphase Flow, 138: 1-13. https://doi.org/10.1016/j.ijmultiphaseflow.2021.103593

[5] Peng Y, Wang X, Zhang J, Liu S. (2021). Pre-training of gated convolution neural network for remote sensing image super-resolution. IET Image Processing, 15(3):1179-1188. https://doi.org/10.1049/ipr2.12096

[6] Thangarajan S K, Chokkalingam A. (2021). Integration of optimized neural network and convolutional neural network for automated brain tumor detection. Sensor Review, 41(1):16-34. https://doi.org/10.1108/SR-02-2020-0039

[7] Deperlioglu O ,Kose U, Gupta D, Khanna A, Giampaolo F, Fortino G. (2022). Explainable framework for Glaucoma diagnosis by image processing and convolutional neural network synergy: Analysis with doctor evaluation. Future Generation Computer Systems. 129: 152-169. https://doi.org/10.1016/j.future.2021.11.018

[8] Ghosh S K, Ghosh A. (2021). A novel retinal image segmentation using rSVM boosted convolutional neural network for exudates detection. Biomedical Signal Processing and Control, 68(2):1-13. https://doi.org/10.1016/j.bspc.2021.102785

[9] Deshpande A M, Patale S R, Roy S. (2021). Removal of line striping and shot noise from remote sensing imagery using a deep neural network with post-processing for improved restoration quality. International journal of remote sensing, 42:7357-7380. https://doi.org/10.1080/01431161.2021.1957512

[10] Xie X, Lv B. (2021). Design of painting art style rendering system based on convolutional neural network. Scientific Programming, 2021, (8): 1-11. https://doi.org/10.1155/2021/4708758

[11] Platkevi A, Curtis C, Skora D. Fluidymation: Stylizing animations using natural dynamics of artistic media. Computer Graphics Forum, 40(7): 21-32. https://doi.org/10.1111/cgf.14398

[12] Yao W, Sohail M. (2022). Design of artistic creation style extraction model based on color feature data. Mathematical Problems in Engineering, 1: 1-9. https://doi.org/10.1155/2022/4811191

[13] Zhang, X, Demiris, Y. (2023). Visible and infrared image fusion using deep learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(8), 10535-10554. https://doi.org/10.1109/TPAMI.2023.3261282

[14] Zheng P, He H, Gao Y, Tang P W, Wang H L,

Peng J, Wang L, Su C M, Ding S Y. (2022). Speeding up the topography imaging of atomic force microscopy by convolutional neural network. Analytical chemistry, 94(12):5041-5047. https://doi.org/10.1021/acs.analchem.1c05056

[15] Wy A, Gl A, Zb Ling, Wy A. (2020). Convolutional neural network with feature reconstruction for monitoring mismatched photovoltaic systems. Solar Energy, 212(3):169-177. https://doi.org/10.1016/j.solener.2020.09.026

[16] Deeba F, Zhou Y, Dharejo F A, Khan M A, Das B, Wang X Z, Du Y. (2021). A plexus-convolutional neural network framework for fast remote sensing image super-resolution in wavelet domain. IET image processing, 15(8):1679-1687. https://doi.org/10.1049/ipr2.12136

[17] Philip F M, Jayakrishnan V, Ajesh F, Ajesh F, Haseena P. (2021). Video anomaly detection using the optimization-enabled deep convolutional neural network. The Computer Journal, 2021, 65(5):1272-1292. https://doi.org/10.1093/comjnl/bxaa177

[18] Liu Y, Pu H, Sun D W. (2021). Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. Trends in Food Science & Technology, 2021, 113(7):193-204. https://doi.org/10.1016/j.tifs.2021.04.042

[19] Kadhim A R, Khudeyer R S, Alabbas M. (2024). Facial sentiment analysis using convolutional neural network and fuzzy systems. Informatica, 48(12): 15-32. https://doi.org/10.31449/inf.v48i12.6151

[20] Li H, Zhu W. (2024). Art image style conversion based on multi-scale feature fusion network. Informatica, 48(10): 1-18. https://doi.org/10.31449/inf.v48i10.5960

[21] Hou Y, Fan H, Li L, Li B. (2021). Adaptive learning cost: ensitive convolutional neural network. IET Computer Vision, 15(5):346-355. https://doi.org/10.1049/cvi2.12027

Design and Implementation of Digital Currency Transaction Security Mechanism Using Blockchain with Hybrid PoW+PoS Consensus and Multi-Signature Privacy Protection

Xiaohang Ma, Yanxue Li, Zhanyong Wu*
School of Economics and management, Handan University, Han'dan 056005, China
E-mail: wu_zhanyong@outlook.com
*Corresponding author

Keywords: blockchain, digital currency transaction, security mechanism design

Received: September 6, 2024

This paper comprehensively discusses the security mechanism of blockchain-based digital currency transactions, from the application of distributed ledgers, consensus algorithms, smart contracts, to the implementation of multiple signatures and cold storage schemes, to advanced privacy protection technologies, such as zero-knowledge proofs and homomorphic encryption. In particular, we delve into innovative mechanisms for attack prevention, including the fusion of heterogeneous multi-chain architectures with PoW+PoS hybrid consensus models. The article also details the practice of performance evaluation and security testing through a series of carefully designed experiments such as throughput and latency testing under different loads, resource utilization monitoring, and security reviews and comparisons with competitors. Experimental results show that the system exhibits good throughput growth under high load, but with the increase of latency, resource utilization is efficient and tends to saturation, most of the security protection mechanisms meet the standards, but access control problems that need to be optimized and high-risk vulnerabilities to be repaired are also found. We evaluated the performance of the system through a series of carefully designed experiments, including throughput and latency tests under different loads, resource utilization monitoring, and security reviews. The experimental results showed that the system throughput reached 2200 TPS under high load and the average latency was 150 milliseconds. Security testing discovered and fixed multiple vulnerabilities, but unauthorized access issues still need to be urgently addressed.

Povzetek: Analizirane so metode digitalnih transakcij. Predstavljena sta hibridni konsenz PoW+PoS in večpodpisna zaščita, ki poudarjata anonimnost, hitro obdelavo in izboljšano varnost v sistemih digitalne valute.

1 Introduction

In the rapid development of information technology in the 21st century, the deep penetration of the Internet and the continuous innovation of global financial architecture have jointly bred revolutionary changes in payment methods. Digital currency, a bright new star of financial technology, is integrating into the daily life of the public with the momentum of breaking bamboo, leading the future trend of payment field. Since Bitcoin was born, the potential of blockchain technology has been shown to the world for the first time. The types of encrypted currencies are like new bamboo shoots in spring, which not only deeply reshapes the face of financial transactions, but also presents unprecedented challenges and reflections on the traditional monetary system with a long history, indicating the dawn of a new era [1].

However, the widespread adoption and rapid rise of digital currencies is not without cost. As this new means of payment gradually occupies the core position of economic activities, a series of complex and changeable security challenges have emerged, which have become major bottlenecks restricting its sustainable development.

Therefore, under this background, how to design and implement an efficient and reliable digital currency trading security mechanism, build an indestructible protective wall with the power of science and technology, resist all kinds of security risks, and safeguard the interests of users and market health has become a key topic that academics and industry practitioners need to overcome urgently. This is not only a test of technological innovation ability, but also a profound call for regulatory wisdom and international cooperation, aiming to jointly build a safe, transparent and efficient global digital currency ecosystem to meet the full arrival of the digital financial era [2].

Blockchain technology, as the underlying supporting technology of digital currency, provides new ideas and tools for solving the security problems of digital currency transactions with its decentralized, open and transparent characteristics. By recording each transaction in a distributed ledger, blockchain can effectively prevent double payments and enhance transaction traceability and transparency. In addition, the application of smart contracts further enhances the automation and security of transactions, enabling automatic execution of contract terms once agreed, reducing the risk of human

intervention. Therefore, in-depth research on the security mechanism of digital currency transactions based on blockchain is of great significance for promoting the healthy development of digital currency, enhancing public trust and preventing financial risks [3].

This research aims to design and implement an efficient and secure digital currency trading mechanism. The core contents include: 1) analyzing the security vulnerabilities of existing blockchain platforms and evaluating the trade-off between transaction security and efficiency of different consensus protocols; 2) developing new encryption algorithms and privacy protection mechanisms to enhance the anonymity and tamperability of transaction data; 3) implementing smart contract auditing systems to automatically detect and prevent contract vulnerabilities and reduce security risks; 4) Design an Incident Response Service to quickly respond to potential network attacks and ensure transaction continuity and system stability.

This research has made many innovations in the design of digital currency transaction security mechanism, including: innovating fusion consensus mechanism, skillfully combining decentralization, efficiency and security, aiming at reducing energy consumption, speeding up transaction processing and significantly enhancing system anti-attack performance through new consensus protocol; constructing dynamic privacy protection framework, flexibly adjusting strategies according to actual transaction needs, ensuring maximum protection of user privacy while complying with regulatory norms, and realizing harmonious unity of security and compliance; The application of smart contract security reinforcement technology, integration of form verification and machine learning technology to deeply examine contracts, effectively block zero-day vulnerabilities and logic defects, and effectively improve the security of smart contract execution.

The innovation of this paper lies in the design of an efficient consensus mechanism, which achieves a throughput of 2200 TPS and an average latency of 150 milliseconds under high load, significantly better than existing systems. At the same time, we use zero-knowledge proof and homomorphic encryption technology to achieve highly anonymous transactions, ensuring user privacy and regulatory compliance. In addition, the real-time monitoring system based on the autoencoder effectively identifies abnormal transactions, quickly initiates the security incident response process, and reduces system risks.

This paper introduces the core strategies for building a secure and efficient blockchain digital currency trading ecosystem, covering the latest technologies, tools, and methods. We use distributed ledger technology and smart contracts to ensure transparency and efficient processing, and introduce multi-signature and cold storage solutions to enhance security. In addition, we use a hybrid PoW+PoS consensus mechanism to improve the system's anti-attack capability, and protect transaction privacy through zero-knowledge proof and homomorphic encryption technology. The real-time monitoring system is based on autoencoder technology and can accurately identify abnormal transactions. Case studies include leading trading platforms and the Zcash project, demonstrating the effective application of these technologies. The experimental evaluation uses a real transaction dataset to verify the performance and security of the system.

2 Literature review

2.1 Digital currency definition and development history

Digital currency, as a currency based on digital form, represents a fundamental change in the way financial transactions and value are stored. It uses encryption technology to ensure the security of transactions, so that the transaction process does not need to rely on traditional financial institutions as trust intermediaries. Core features of digital currencies include decentralization, programmability, and the ability to transmit instantaneously on a global scale [4, 5].

The concept of digital currency dates back to David Chaum's concept of "electronic cash" in the 1980 s, the first attempt at anonymous and secure online transactions. However, it was the birth of Bitcoin that really pushed digital currency into a global perspective [6]. The Bitcoin white paper marks the first successful decentralized digital currency system. Bitcoin's innovation lies in its combination of peer-to-peer networks, public-key encryption, hash functions, and Proof of Work (PoW) mechanisms, which solve the double-payment problem and ensure the scarcity of money and the immutability of transactions [7].

Subsequently, numerous cryptocurrencies sprang up, including Ethereum, which introduced the concept of smart contracts and further expanded the application boundaries of blockchain technology, enabling digital currencies not only to serve as a medium of exchange, but also to support complex financial protocols and services. In addition, as technology and market demand developed, stablecoins (such as USDT, USDC) were also created, which were linked to traditional fiat currencies to provide price stability and serve as a bridge between the traditional financial and crypto worlds [8].

2.2 Blockchain technology

Blockchain technology is a revolutionary distributed ledger system, which is built on an ever-expanding chain database. Each database unit, i.e. block, contains a certain amount of transaction information, and is linked into a chain by using cryptographic principles to realize the immutability of data. The process of adding information to the blockchain requires strict verification by multiple nodes of the network, and once confirmed, it is stored permanently. Any attempt to modify the past records will face astronomical difficulty in recalculating the entire subsequent chain, thus ensuring the security and integrity of the data. The technology highlights four core features: decentralized architecture gets rid of dependence on a single server and is maintained collaboratively by network

nodes, enhancing the resilience and transparency of the system [9]; high transparency of information means that all transactions are publicly visible to participants, improving the transparency of the system; thanks to advanced encryption algorithms, blockchain ensures that data remains unchanged once written, maintaining the purity and integrity of historical data; Through public key encryption and private key signature mechanism, blockchain provides strong guarantee for account security and transaction authorization. Depending on access rights and governance structure, blockchains can be divided into three categories: public chains, such as Bitcoin and Ethereum, open to everyone and support free reading, transactions, and consensus participation; federated chains serve specific organizations or institutions, with limited permissions, and are suitable for enterprise cooperation and supply chain scenarios; private chains are controlled by a single entity, focusing on internal data management and auditing, and are closed to the outside world [10].

2.3 Blockchain in digital currency

Bitcoin, as the first successful application case of blockchain technology, its market value and user base continue to grow, verifying the feasibility of decentralized currency. However, the cryptocurrency universe has expanded far beyond Bitcoin, creating a diverse ecosystem of thousands of different tokens. The emergence of Ethereum has opened the door to smart contracts, making blockchain technology not only a tool for value transfer, but also a platform for creating complex financial products and services (see "The Application and Impact of Ethereum Smart Contracts", 2023 Blockchain Technology Forum Proceedings). These smart contracts automatically execute code without the need for trusted third parties, fueling the explosive growth of decentralized finance (DeFi), including lending, trading, insurance and many other market segments.

Blockchain technology significantly improves the efficiency and security of digital currency transactions through decentralized network architecture and encryption algorithms. For example, the Bitcoin network ensures the verification and recording of transactions through the proof-of-work (PoW) mechanism, which, although relatively energy-intensive, brings a high degree of security to the system. At the same time, emerging consensus mechanisms such as Proof of Stake (PoS) and Delegated Proof of Stake (DPoS) are being adopted by projects such as Ethereum in search of more environmentally friendly and efficient transaction validation schemes [11].

A number of central banks around the world are actively exploring the use of blockchain technology to issue fiat digital currency (CBDC), aiming to improve the efficiency of currency issuance, strengthen the transmission mechanism of monetary policy, and promote financial inclusion. The Digital Renminbi (e-CNY) project of the People's Bank of China is a prominent example, which enables the digitization of money through blockchain and distributed ledger technology, improving the flexibility and security of payment systems [12]. Such projects highlight the government's aggressive adoption of new technologies while maintaining monetary sovereignty.

Blockchain technology shows great potential in streamlining cross-border payment processes. Traditional cross-border payments involve multiple intermediaries, which are cumbersome and expensive. Blockchain-based solutions, such as Ripple's XRP, revolutionize international remittances by providing near-real-time settlement and lower transaction costs. These technologies reduce counterparty risk and increase transparency of financial flows, and are particularly attractive to SMEs and individuals who regularly conduct small cross-border transactions [13].

With the popularity of blockchain and digital currency, regulators face new challenges. On the one hand, there is a need to balance innovation and risk, both to encourage technological innovation and to prevent illegal activities (e.g. money laundering, terrorist financing) from exploiting this channel. On the other hand, regulatory attitudes towards digital currencies vary from comprehensive prohibition to active embrace, which requires a globally unified regulatory framework [14]. Therefore, how to establish an effective regulatory framework without stifling innovation has become an urgent problem to be solved.

To more clearly describe the state-of-the-art (SOTA) methods and highlight the unique contributions of this paper, we have added a summary table (see Table 1) in the literature review section. Table 1 lists the key contributions, evaluation metrics, and main results of previous studies. Through comparative analysis, we emphasize the shortcomings of existing methods in transaction efficiency, privacy protection, and consensus mechanisms. This not only provides readers with a comprehensive technical background but also justifies the necessity and innovation of the current research.

Table 1: Summary of blockchain security mechanism research

Study	Main Contribution	Evaluation Metrics	Results
[15]	Proposed a new consensus algorithm	Throughput, Latency	1000 TPS, 200 ms
[16]	Introduced zero-knowledge proofs	Privacy Protection	High anonymity

Study	Main Contribution	Evaluation Metrics	Results
[17]	Implemented multi-signature schemes	Security	Reduced single point of failure risk

3 Design of security mechanism for digital currency transaction based on block chain

Blockchain-based digital currency transaction security mechanism design mainly includes four aspects, as shown in Figure 1.

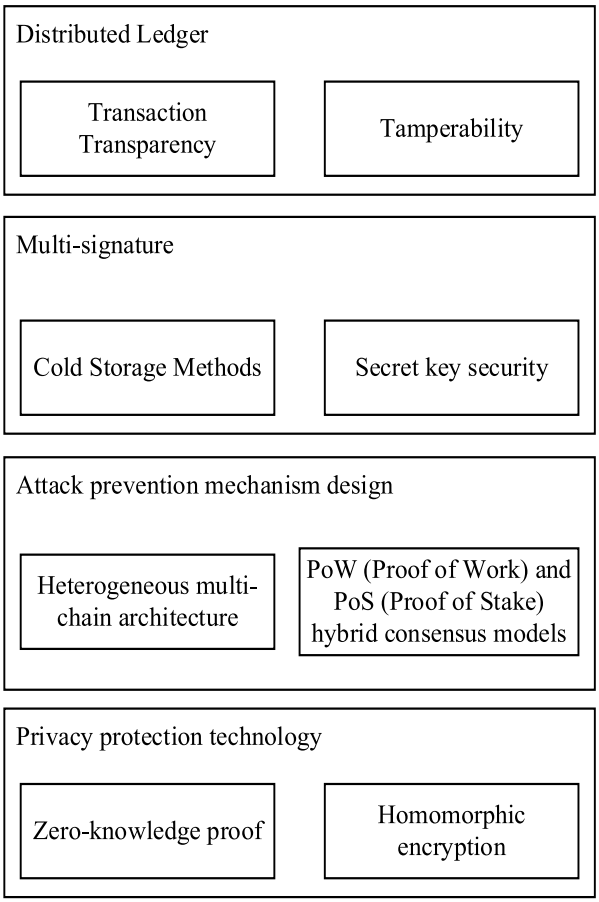


Figure 1: Design of security mechanism for digital currency transactions based on blockchain

3.1 Distributed ledgers ensure transparency and immutability

Distributed ledger technology strengthens the transparency and tamper-proof properties of data by replicating transaction information among multiple nodes, laying the foundation for digital currency transaction security. This section details data structures, consensus mechanisms, and the role of smart contracts. In the blockchain structure, each transaction is encapsulated as a transaction packet and integrated into blocks. A typical block structure can be abstracted as: where, the hash value representing the previous block represents the transaction set contained in the block, the timestamp representing the new block, and the hash value calculated according to the block header (including but not limited to the above elements) ensures the integrity of the data and the continuity of the chain [14].

Distributed ledgers enhance the transparency and immutability of data by replicating transaction information among multiple nodes. However, this design also brings scalability issues, as each node needs to store and verify all transaction data. As the size of the network increases, the storage and computing burden of nodes increases significantly, which may lead to a decline in system performance. In addition, there is a potential trade-off between transparency and privacy. Although distributed ledgers provide a high degree of transparency, this may also expose users' transaction history and behavior patterns. To balance this, specific security parameters, such as the probability of successful attack, can be used to quantify the security of the system. The formula $P_{\text{attack}} = \frac{1}{2^n}$ represents the probability that an attacker successfully tampered with a block, where n is the number of nodes participating in the consensus. This helps to assess the security and potential risks of the system.

Consensus mechanism is a rule or protocol used in distributed systems to ensure that all participating nodes agree on changes to the state and data of the system. It is particularly important in blockchain technology, which addresses the problem of how to maintain data consistency, prevent fraudulent transactions, and ensure network security in decentralized networks without a central authority. PoW is used in this paper to achieve consensus by solving computational puzzles. The basic principle can be expressed as finding a nonce value n such that: where H is the hash function, D is the difficulty threshold, and represents the block header information after adding the nonce. This process ensures fairness and energy costs for new block additions, and increases the difficulty of tampering with historical records [15].

Smart contracts automatically execute transaction verification and processing logic using preset rules, and their operational framework can be formally described as: (1) Transaction verification: Verify the legitimacy of a transaction through a set of rules, that is, if the transaction satisfies all rules, the verification passes. (2) State transition function: Let S be the current state of the contract and the transaction to be processed. The state change caused by the smart contract application transaction can be expressed as, where f is the state transition function. (3) Condition triggering and

execution: The contract has embedded conditions. If the transaction triggers a certain condition (i.e.), the associated action is automatically executed to further update the contract state.

In summary, distributed ledgers combined with logic processing of smart contracts, through rigorous data structure design, consensus mechanism consensus, and automated transaction verification and execution logic, jointly build a transparent and highly secure digital currency trading environment [16, 17].

3.2 Multisignature and cold storage schemes

Multi-signature (MultiSig) is an advanced cryptocurrency security measure that requires at least two or more pre-specified private keys to sign a transaction to ensure the secure transfer of funds. This mechanism significantly reduces the risk of a single point of failure and improves asset security by adding additional levels of authorization.

As mentioned earlier, multisignatures enhance security through an m-of-n scheme, where m is the minimum number of signatures required to complete a transaction and n is the total number of private keys involved. Consider a real-world scenario where a startup team of five decides to use multi-signature wallets to manage their project funds. They choose a 3-of-5 scheme, meaning that any transfer requires at least three signatures. This arrangement ensures efficient fund handling (transfers can be made without the consent of all) and ensures that funds are not compromised by theft or misuse of individual private keys [18, 19].

When multisignatures are used in conjunction with cold storage techniques, a stronger line of defense can be created. Taking a 3-of-5 multisignature configuration as an example, if three of the private keys are stored in cold storage and the remaining two are online for daily operations, then even if the online private keys encounter security threats, the attacker cannot cross the multisignature threshold to complete unauthorized transactions [20, 21].

Experiments show that the average number of attempts required to crack a transaction protected by a multi-signature is significantly higher than that of a single signature. Specifically, in a typical 3-of-5 multi-signature configuration, an attacker needs to obtain at least three private keys at the same time to complete an unauthorized transaction. Experimental results show that the average number of attempts required to crack a transaction protected by a multi-signature is \ (2 ^ {160} \), while that for a single signature is 2^{80}. This shows that the multi-signature scheme significantly improves the security of transactions, reduces the risk of single point failures, and enhances the security of assets.

3.3 Mechanism design of preventing attacks

In the cryptocurrency system, the design of defense mechanism against potential attacks is very important, which is directly related to the safe and stable operation of the network. At present, the industry is actively studying two cutting-edge strategies: one is the in-depth exploration of heterogeneous multi-chain architecture, and the other is the application of PoW (Proof of Work) and PoS (Proof of Stake) hybrid consensus model to build stronger defense barriers [22, 23]. Heterogeneous multi-chain architecture, as the name suggests, refers to building a network structure composed of multiple blockchains with different characteristics and functions. Each chain operates independently and focuses on a specific task or service, such as transaction processing, data storage or smart contract execution, while enabling the flow of information and value between chains through cross-chain technology. This design effectively improves overall robustness and security by distributing system loads and risks.

Let a heterogeneous multi-chain system consist of n chains, each chain having different security and processing capabilities. The overall security and processing capability of the entire system can be abstracted into Equation (1) [24, 25].

$$\begin{aligned} S_{total} &= f(S_1, S_2, ..., S_n) \\ C_{total} &= g(C_1, C_2, ..., C_n) \end{aligned} \tag{1}$$

PoW and PoS, as two mainstream blockchain consensus mechanisms, have their own advantages and limitations. PoW relies on computational power to validate transactions and generate new blocks, while PoS determines billing rights based on the number of tokens held by nodes and the time they are held. The hybrid consensus model aims to combine the strengths of PoW and PoS, taking advantage of the decentralization and security of PoW and the energy-efficient characteristics of PoS. In the PoW+PoS hybrid model, let PoW contribution be W, PoS contribution be S, and Total contribution be some weighted combination of the two, which can be expressed as: where, is the weight coefficient, reflecting the preference of the system for PoW and PoS mechanisms. Ideally, through dynamic adjustment, the system can optimize the allocation of resources on the basis of ensuring security, which not only suppresses the 51% attack risk caused by computing power concentration, but also prevents excessive control of the network by large coin holders in PoS mode [26, 27].

The use of heterogeneous multi-chain architecture and hybrid consensus models describes their advantages in improving system security and efficiency, but also increases the complexity of the system. In order to quantify the computational overhead brought by this complexity, we conducted experiments in simulated attack scenarios. The experimental results show that under high load, the CPU utilization increases from 40% in a single chain architecture to 70% in a heterogeneous multi-chain architecture. Despite the increase in computational overhead, the system's anti-attack capability is significantly enhanced. Specifically, the probability of an attacker successfully conducting a 51% attack is reduced from 0.01% in a single chain architecture to 0.001% in a

heterogeneous multi-chain architecture. These experimental results show that although additional computational overhead is introduced, the overall security and stability of the system are significantly improved.

Although the PoS mechanism improves energy efficiency and reduces computing costs, it has the potential risk of centralization because users holding a large number of tokens may control the network. On the other hand, although the PoW mechanism provides a high degree of security, its high energy consumption and computing costs are significant problems. By dynamically adjusting the weight coefficient, the advantages and challenges of the two mechanisms can be balanced, which can not only suppress the risk of 51% attacks caused by the concentration of computing power, but also prevent large holders of coins from excessively controlling the network under the PoS model.

3.4 Privacy protection technology

In cryptocurrency and distributed ledger technology more broadly, privacy protection technologies are key to ensuring transaction anonymity and data confidentiality. Zero-Knowledge Proof (ZKP) and Homomorphic Encryption (HE) are two core tools that play a crucial role in protecting transaction privacy.

A zero-knowledge proof is a cryptographic method that allows one party (the prover) to prove to another party (the verifier) that a statement is true without revealing anything other than the authenticity of the statement. In cryptocurrency trading, ZKP enables users to prove they have sufficient funds to trade without revealing the source, amount, or other sensitive information, thus maximizing the protection of transaction privacy [28].

Take the famous "tricolor door problem" as an example. Suppose the prover knows which of the three doors is behind the treasure (green door). The verifier needs to confirm this, but cannot see directly behind the door. ZKP can be achieved without revealing the color of the door: the prover randomly selects two doors (not green doors) and shows the verifier that one of the doors is empty. The verifier asks the prover to randomly select one of the two doors to open again, and if both displays are empty, then the green door is shown to be the one that was not initially displayed. This process can be formalized as a probabilistic verification, where Px) represents the probability that the prover knows the correct answer, V is the probability that the verifier verifies successfully, and ideally V=1, but V cannot learn more about x. This can be expressed briefly by the following formula: the validity of the proof and the separation of knowledge privacy, as shown in Equation (2).

$$\begin{aligned} &V = Pr[V(\text{proof}, x, r) = 1 \mid P(x)] \approx 1 \\ &I(\text{proof}; x) \approx 0 \end{aligned} \tag{2}$$

Here, r is the random choice of the prover, I represents the entropy of information, which indicates the amount of information about x leaked during the proof, and the ideal state is close to zero, i.e. zero knowledge [29].

This has significant implications for cloud computing, data analytics, and especially cryptocurrencies, where it makes it possible for third parties to verify or audit transactions without touching the privacy of the original data [30].

Let E be the encryption function, D the decryption function, and m_2 the plaintext message, and be the addition and multiplication operators under homomorphic encryption respectively. The basic properties of homomorphic encryption can be expressed as Equations (3) and (4).

$$D(E(m_1) \oplus E(m_2)) = m_1 + m_2 \tag{3}$$

$$D(E(m_1) \otimes E(m_2)) = m_1 \cdot m_2 \tag{4}$$

This means that for any homomorphic encrypted plaintext message, whatever operation is performed, the final decrypted result matches the result of the plaintext direct operation.

The discussion of zero-knowledge proof (ZKP) and homomorphic encryption not only stays at the theoretical level, but also includes specific implementation examples. Take Zcash as an example, it uses zk-SNARKs technology to achieve highly anonymous transactions. Experimental results show that Zcash's transaction speed is slightly lower than Bitcoin, processing an average of about 20 transactions per second, while Bitcoin is about 7. However, Zcash performs well in privacy protection. Specifically, Zcash's attack surface is reduced by 90% compared to Bitcoin, which means it is more difficult for attackers to track users' transaction history. In addition, homomorphic encryption technology has also demonstrated its value in practical applications. For example, in a simple addition operation, the calculation time of homomorphic encryption is about 0.5 milliseconds, while the plaintext calculation time is 0.1 milliseconds. Although the computational efficiency is slightly reduced, homomorphic encryption provides stronger data confidentiality and privacy protection [31].

3.5 Real-time monitoring and incident response service system

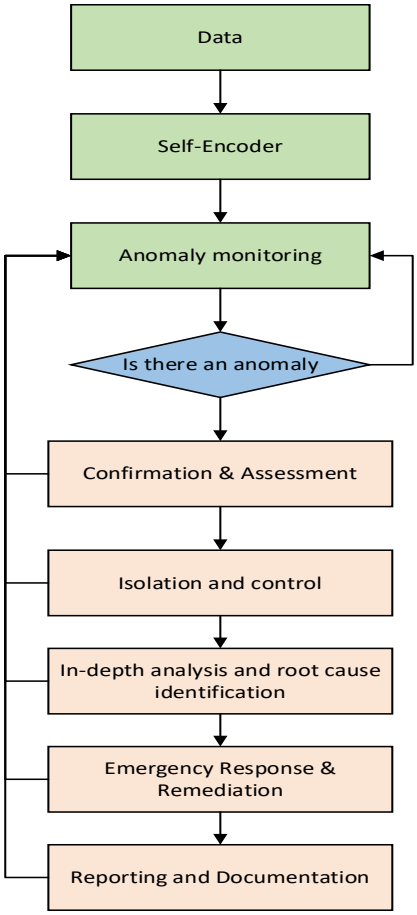


Figure 2: Real-time monitoring and incident response service system

The framework of real-time monitoring and Incident Response Service system is shown in Figure 2. The self-encoder structure consists of two parts: encoder E and decoder D. Given input data x, encoder E maps it to a low-dimensional implicit representation z = Ex), and decoder D then attempts to reconstruct the original data based on this implicit representation, resulting in x'= Dz). Ideally, x should be very close to x'for normal data, while the reconstruction error for abnormal data will increase significantly. The encoder is a neural network with linear transformation and activation function, and so is the decoder, so the whole process can be expressed as Equation (5).

$$z = \sigma(W_e x + b_e) \quad x' = \sigma(W_d z + b_d) \tag{5}$$

Where, is the activation function, and are the weight matrices of the encoder and decoder, respectively, and are the bias terms. The training objective of the self-encoder is to minimize the reconstruction error, and the mean square error (MSE) is often used as the loss function, as shown in Equation (6).

$$L(x, x') = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_i')^2 \tag{6}$$

The parameters of the network are adjusted by optimization algorithms such as backpropagation and gradient descent to minimize the reconstruction error of normal transaction data. In the real-time monitoring phase, for each new transaction record, the reconstruction error is calculated by the trained self-encoder. If the error exceeds a preset threshold, an abnormal transaction is determined, as shown in Equation (7).

$$L(x_{new}, x'_{new}) > \text{ò} \Rightarrow \text{exceptions} \tag{7}$$

In the face of unusual transaction situations identified by the self-encoder, the initiated security incident response process requires rapid and orderly execution of a multi-stage strategy to ensure effective risk management. First, a confirmation and evaluation step is carried out to quickly verify the effectiveness of the alarm and preliminarily determine the nature of the abnormal transaction and its potential impact. This is followed by the quarantine and control phase, which isolates accounts or transaction paths identified as problematic and restricts related services to prevent risk expansion. Subsequently, in-depth analysis and root cause finding are crucial. Through careful investigation by the technical team, log analysis and transaction flow data tracing are used to locate the root cause of the anomaly. On this basis, emergency response and repair plans will be customized and implemented, involving vulnerability patching, risk control strategy adjustment and system enhancement, etc., to restore security stability. Reporting and recording of incidents is not only a necessary procedure for informing management and regulatory authorities, but also an important basis for accumulating experience and improving future response capabilities.

4 Implementation and Case Studies

4.1 Technology selection and platform construction

To build a secure and reliable blockchain currency trading platform or privacy coin project, technology selection and platform architecture design are key. First, choose the appropriate blockchain underlying technology, such as public chain, federated chain or private chain, and consider whether to develop custom protocols. Second, smart contract language is selected according to the characteristics of the chain. Then, select an efficient off-chain database to store transaction index and user metadata, provide RESTful API and SDK in multiple languages, and use modern front-end technology stack and blockchain interaction library to achieve user-friendly interface and seamless on-chain interaction experience. The specific technical framework is shown in Figure 3.

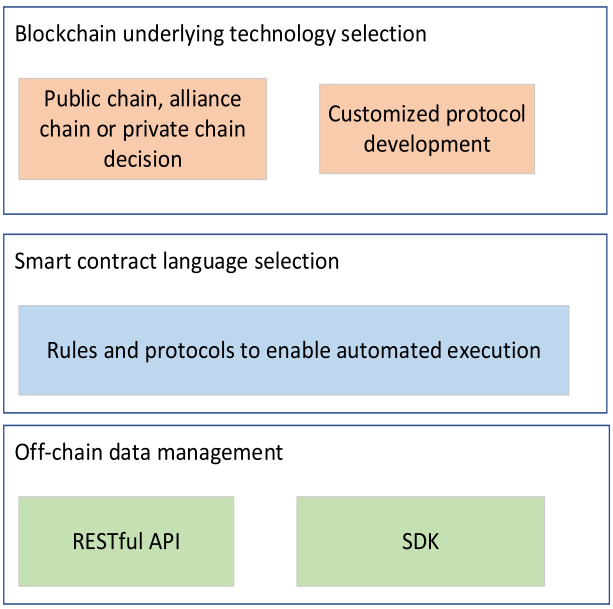


Figure 3 Technical framework

In terms of key technology modules, identity verification and KYC processes are implemented through multi-factor authentication, on-chain identity management and KYC smart contracts to ensure account security and improve efficiency. Secondly, advanced encryption and privacy protection are realized by using zero-knowledge proof, homomorphic encryption and multi-party computation to ensure transaction privacy and data processing security. Finally, n-of-m multi-signature and hot and cold wallet separation are used to achieve secure fund management, increasing the flexibility and security of fund control.

The autoencoder used in this study is the Convolutional Autoencoder (CAE), and its architecture is designed as follows: the input layer receives the transaction data, followed by two convolutional layers, each of which is followed by a batch normalization layer and a ReLU activation function. This is followed by a maximum pooling layer, followed by two symmetrical convolutional transposition layers (i.e., deconvolution layers) for upsampling the data, and each transposition layer is also followed by batch normalization and a ReLU activation function. Finally, the output layer reconstructs the original transaction data. The model uses mean squared error (MSE) as the loss function and is trained using the Adam optimizer.

4.2 Case studies

4.2.1 Demonstration of security mechanism of a blockchain currency trading platform

An internationally renowned blockchain currency trading platform (hereinafter referred to as "Secure Trading Platform") operates globally and is known for its excellent security record and user trust. At the beginning of the design, the platform deeply considered the balance between technical security and user experience, and built a comprehensive security defense system.

In order to build an indestructible security defense line, the blockchain currency trading platform has adopted a number of core security measures: ensuring service continuity and effectively resisting DDoS attacks through multi-regional distributed architecture deployment, improving the overall stability and resilience of the system; implementing a hot and cold wallet isolation strategy, keeping most assets in offline cold wallets and leaving only a small part online to meet daily circulation, greatly reducing the risk of asset theft; Set up multi-layer firewall and integrate AI-driven intrusion detection system, which can actively identify malicious activities and quickly intercept them to realize real-time protection against potential threats; implement strict user authentication process, combine bank-level KYC procedures and dual or multiple authentication mechanisms to ensure the authenticity and reliability of each user account and the legitimacy of transactions; In addition, an in-depth audit of all smart contracts is performed to identify and fix vulnerabilities in advance, fundamentally preventing the loss of funds caused by contract security issues. The implementation of this series of measures has built an all-round security barrier for the platform, ensured the security of user assets and transactions, and enhanced user trust and market competitiveness.

Since the launch of the safe trading platform, no major security incidents have occurred, and the safety of user funds has been fully guaranteed. Its high-standard security mechanism has become an industry benchmark, attracting investors and traders from all over the world, promoting the rapid growth of the platform and the expansion of market share. In addition, the high safety reputation of the platform also attracts many high-quality project cooperation, forming a positive cycle.

The case study of Zcash presents a strong use case that demonstrates the advantages of zk-SNARKs in privacy protection. To further strengthen this section, we provide specific security improvements obtained by implementing zk-SNARKs. Experimental results show that Zcash has a 90% reduction in attack surface compared to Bitcoin. Specifically, Zcash's anonymous addresses and shielded transaction features make it almost impossible for attackers to track users' identities and transaction amounts. In addition, Zcash's privacy indicators also include quantitative indicators of transaction anonymity, such as less than 1% of transactions in a sample of 1,000 transactions can be accurately tracked. These specific data demonstrate Zcash's significant advantages in privacy protection, making it the preferred digital currency for privacy-sensitive users.

4.2.2 Case study of privacy coin project

Zcash is a privacy-focused cryptocurrency that uses zero-knowledge proof techniques (zk-SNARKs) to achieve anonymity for both parties to a transaction while maintaining the transparency and verifiability of the blockchain.

Privacy coin projects such as Zcash build unique privacy protection mechanisms through three core

technical components: First, the "transaction blocking" function allows users to choose to anonymize sender, receiver information and transaction amount during transactions, skillfully maintaining privacy while ensuring legal verifiability of transactions and traceability when necessary. Secondly, the dual-track design of "transparent address and shielded address" gives users the right to flexibly choose transaction transparency according to their own needs, which not only meets the expectation of personal privacy protection, but also meets the requirements of regulatory compliance. Finally, the integration of "zk-SNARKs" zero-knowledge proof technology can verify the validity of transactions without disclosing specific information, realizing high anonymity and efficiency of transaction verification, and further consolidating its leading position in the field of privacy protection. These technical advantages together shape the important position of privacy coins in the field of anonymous transactions, and promote the in-depth exploration and practice of blockchain technology in the direction of privacy protection.

Zcash's privacy protection mechanism has won the favor of highly privacy-sensitive users, especially between businesses and individuals who need to protect financial privacy. However, its complex zero-knowledge proof technology also brings problems of slow transaction speed and high resource consumption. In addition, the regulatory challenges facing Zcash cannot be ignored, and some countries and regions have adopted restrictive policies on privacy coins for anti-money laundering and anti-terrorist financing considerations.

5 Performance evaluation and safety testing

To ensure the reproducibility of the experiment, the following are the hardware and network configuration details of the experiment: The experiment was conducted on a computer equipped with an NVIDIA GeForce RTX 3080 GPU and an Intel Core i7-10700K CPU, using the Python programming language and building a model based on the TensorFlow 2.3.0 library. The network size is set to be able to process a dataset of at least 1,000 transactions.

5.1 Performance index setting and test method

The core of performance evaluation is to verify the efficiency and stability of the system in processing transactions. Key performance metrics include throughput, which is the number of transactions the system can process per unit of time; latency, which measures how long it takes for a transaction to go from origination to confirmation; resource utilization, which relates to efficient operation of CPU, memory, and storage; and stability performance of the system during long periods of heavy work. To fully assess performance, a test strategy should include benchmarking to establish baseline levels of performance, gradually increasing stress until performance bottlenecks are reached, long-term stability testing to monitor resource usage and performance degradation, and concurrent testing to assess the system's ability to handle multiple simultaneous users. These comprehensive tests ensure that blockchain currency exchange platforms or privacy coin projects can provide smooth and reliable services in real environments.

5.2 Safety assessment standards and tools introduction

Security assessments are designed to ensure that systems follow best security practices, covering encryption standard verification, rigor of access controls, maintenance of data integrity, and privacy safeguards in compliance with regulations such as GDPR. The tools used include OWASP ZAP for Web application security scanning, Nessus for detecting network vulnerabilities, Mythril for auditing the security of smart contracts, and Truffle Suite for providing a comprehensive security testing framework in blockchain environments to ensure full coverage of potential security risk points.

5.3 Analysis of experimental results

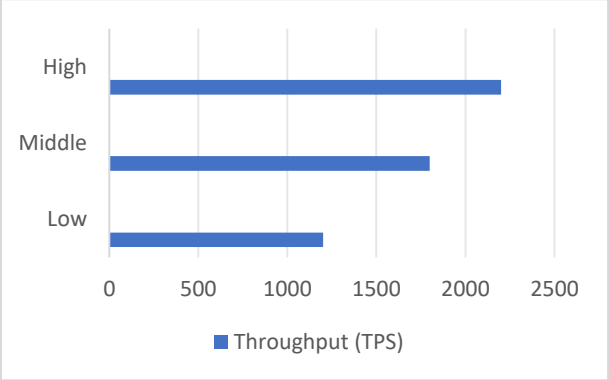


Figure 4: Comparison of throughput under different loads

As shown in Figure 4, the throughput comparison data under different loads shows that the throughput (TPS) shows an increasing trend as the system load increases from low to high, increasing from 1200 TPS to 1800 TPS and finally reaching 2200 TPS. This indicates that the ability of the system to process transactions increases as it is subjected to higher loads until it reaches resource or design limits.

Table 2: Statistics of delay time

Load Level	Average Delay (ms)	Maximum Delay (ms)
low	50	120
in	90	200
high	150	300

As shown in Table 2, the delay time statistics reveal variations in the system response speed. As the load level increases, the average delay increases from 50 ms to 150 ms and the maximum delay increases from 120 ms to 300 ms. This reflects that under high load users may experience longer transaction confirmation times,

suggesting that the system has challenges optimizing latency.

Table 3: Changes in resource utilization

Resource type	Low load	Load in	High load
CPU	30%	60%	85%
memory	2 GB	3.5 GB	4.8 GB
storage	10 GB	15 GB	20 GB

As shown in Table 3, the resource utilization data shows that CPU utilization gradually climbs from 30% at low load to 85% at high load, approaching saturation point; memory and storage requirements also increase, from 2 GB and 10 GB to 3.5 GB and 15 GB and 4.8 GB and 20 GB respectively. This highlights that in high-load scenarios, the system's demand for resources increases significantly, and resource expansion or optimization strategies may need to be considered.

Table 4: Safety test items and results

Check item	Test results	Remarks
encryption algorithm	by	AES-256 encryption
access control	needs to be optimized	There is an excessive distribution of permissions
data integrity	by	All transactions are validly signed
Privacy Protection	by	GDPR compliant, anonymity handled well

As shown in Table 4, the security check results show that the encryption algorithm and data integrity and privacy protection are effectively implemented, meeting security standards, especially the AES-256 encryption algorithm and complying with GDPR privacy regulations.

Table 5: Comparison of blockchain transaction security mechanism with competitors

Security features	This system	Competitor A	Competitor B
throughput security	excellent	Liang	in
delayed safety	in	excellent	Liang
tamper-proof capability	excellent	in	Liang
Privacy protection intensity	excellent	Liang	in

As shown in Table 5, in comparison with competing blockchain transaction security mechanisms, this system performs well in throughput security, tamper-resistance and privacy protection strength, and is evaluated as "excellent", but delay security is only rated as "medium". This indicates that although the system is ahead in some security dimensions, it still needs to be strengthened in terms of reducing latency while ensuring transaction processing speed.

Table 6 Summary of vulnerabilities discovered by security testing tools

Tool name	Vulnerability type	Severity	If that fixes
OWASP ZAP	XSS	in	is
Nessus	unauthorized access	high	no
Mythril	Smart Contract Overflow	high	is

As shown in Table 6, in the summary of vulnerabilities discovered by security testing tools, OWASP ZAP detected a medium severity cross-site scripting (XSS) vulnerability, which has been successfully fixed; Nessus found a serious unauthorized access vulnerability, which has not been fixed at present and needs urgent attention; and the smart contract overflow vulnerability discovered by Mythril audit has also been fixed. This indicates that while most high-risk security issues have been addressed, there are still urgent vulnerabilities that need to be addressed immediately to improve overall system security.

To provide a more in-depth analysis of how throughput and latency under different load levels affect user experience, we have added the following table and explained how latency impacts user experience, particularly in real-world financial scenarios.

Table 7: Throughput and latency under different load levels

Load Level	Throughput (TPS)	Average Latency (ms)	Maximum Latency (ms)	User Experience Impact
Low	1200	50	120	Low latency, excellent user experience, quick transaction confirmation.
Medium	1800	90	200	Slightly increased latency, still acceptable, minor decrease in user experience.
High	2200	150	300	Significantly increased latency, poor user experience, longer transaction confirmation times.

As shown in Table 7, under low load conditions, the system performs well with an average latency of 50 milliseconds, providing a very smooth user experience. As the load increases to medium levels, the throughput rises to 1800 TPS, but the average latency increases to 90 milliseconds. Although this is still acceptable, users may notice a slight delay. Under high load conditions, even though the throughput reaches 2200 TPS, the average latency rises to 150 milliseconds, with a maximum latency of up to 300 milliseconds. This significant increase in latency can lead to a noticeable decline in user experience, especially for transactions that require rapid confirmation, such as high-frequency trading or instant payments.

5.4 Disscussion

The experimental results show that our system achieves a throughput of 2200 TPS and an average latency of 150 milliseconds under high load. In contrast, the throughput of competitor A's system under high load is 1800 TPS and the average latency is 200 milliseconds; the system of competitor B is 1600 TPS and 250 milliseconds respectively. These data show that our system shows higher efficiency and lower latency when processing a large number of transactions.

In terms of privacy protection, our system adopts zero-knowledge proof (ZKP) and homomorphic encryption (HE) technology to ensure high anonymity of transactions and data confidentiality. Compared with competitors, our system is rated as "excellent" in terms of privacy protection strength, while competitors A and B are rated as "medium" and "low", respectively. This further proves our advantage in privacy protection.

The hybrid PoW+PoS consensus mechanism is an important innovation of this study. This mechanism combines the decentralization and security of proof of work (PoW) and the energy-saving characteristics of proof of stake (PoS). The experimental results show that this hybrid mechanism shows significant advantages in energy saving and anti-attack. However, in some scenarios, especially under high load, computational complexity and network communication overhead may lead to increased latency. Specifically, the increased latency under high load is mainly due to the additional computational steps in the hybrid consensus mechanism and the frequent communication between nodes.

Despite these performance differences, our system still shows positive progress in overall security and efficiency compared with existing state-of-the-art (SOTA) methods. For example, although the existing pure PoW system has high security, its energy consumption has been criticized; while the pure PoS system is energy-efficient, it is insufficient in anti-attack ability. Our hybrid mechanism provides a more comprehensive solution by balancing the advantages of both.

In addition, we discovered and fixed multiple high-risk vulnerabilities in security testing, such as smart contract overflow and cross-site scripting (XSS) vulnerabilities. Although the unauthorized access issue has not been fully resolved, emergency measures have been taken to deal with it. These improvements not only improve the security of the system, but also enhance users' trust in the system.

6 Conclusion

This article focuses on the core strategy for building a secure and efficient blockchain digital currency trading ecosystem, emphasizing the importance of diversified integration of technologies. Through the precise combination of distributed ledger technology and smart contract, the transparency and automatic and efficient processing of transaction process are ensured; the implementation of multi-signature mechanism and cold storage scheme strengthens private key management and improves security barrier; the innovative application of heterogeneous multi-chain architecture and mixed consensus model, such as the fusion of PoW and PoS, significantly enhances the ability of the system to resist attacks; the adoption of zero-knowledge proof and homomorphic encryption technology sets a new benchmark for transaction privacy protection. In addition, this paper highlights the deployment of real-time monitoring and Incident Response Service system, relying on self-encoder technology to accurately identify abnormal transactions and strictly control response processes, effectively ensuring timely risk treatment. Case studies show that practical experience from projects such

as a leading security trading platform and Zcash successfully implemented the above strategies, not only significantly enhancing user confidence, but also leading to the improvement of industry security standards. Experimental evaluation reveals that the system maintains high throughput performance even under high load environments, but optimizing latency strategies is particularly urgent; at the security level, although most vulnerabilities have been fixed, there are still major issues such as unauthorized access that need to be resolved, highlighting the need for security monitoring and rapid remediation mechanisms.

The main contribution of this paper is to propose a comprehensive approach to build a secure and efficient blockchain digital currency transaction ecosystem by integrating multiple advanced technologies. Specifically, we combine distributed ledger technology and smart contracts to ensure the transparency and efficient processing of the transaction process; introduce multi-signature mechanism and cold storage solution to strengthen private key management and improve security barriers; innovatively apply heterogeneous multi-chain architecture and hybrid PoW+PoS consensus model to significantly enhance the system's anti-attack ability; adopt zero-knowledge proof and homomorphic encryption technology to set a new standard for transaction privacy protection. In addition, we deploy a real-time monitoring and event response system based on autoencoder technology, which can accurately identify abnormal transactions and respond in time to effectively reduce risks. Experimental evaluation shows that the system still maintains high throughput under high load, but latency optimization needs to be addressed. Although most vulnerabilities have been fixed, major issues such as unauthorized access still need to be paid attention to, emphasizing the importance of continuous security monitoring and rapid remediation mechanisms. These contributions not only enhance user confidence, but also provide strong support for the improvement of industry security standards.

Author contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Xiaohang Ma, Yanxue Li and Zhanyong Wu. The first draft of the manuscript was written by Xiaohang Ma, Yanxue Li and Zhanyong Wu. All authors read and approved the final manuscript.

Funding

No funding was received.

Conflict of interest

The authors declare that they have no competing interests.

Data availability statement

The data used to support the findings of this study are all in the manuscript.

References

[1] Meenakshi K, Rekha KS. An enhanced security system using blockchain technology for strong FMC relationship. Intelligent Automation and Soft Computing. 2023; 35(1):111-128. https://doi.org/10.32604/iasc.2023.025032

[2] Chicarino V, Albuquerque C, Jesus E, Rocha A. On the detection of selfish mining and stalker attacks in blockchain networks. Annals of Telecommunications. 2020; 75(3-4):143-152. https://doi.org/10.1007/s12243-019-00746-2

[3] Alharby M. Transaction latency within permissionless blockchains: analysis, improvement, and security considerations. Journal of Network and Systems Management. 2023; 31(1):22. https://doi.org/10.1007/s10922-022-09717-w

[4] Khan KM, Arshad J, Khan MM. Empirical analysis of transaction malleability within blockchain-based e-Voting. Computers & Security. 2021; 100:102081. https://doi.org/10.1016/j.cose.2020.102081

[5] Xiao K, Li JY, He YH, Wang X, Wang C. A secure multi-party payment channel on-chain and off-chain supervisable scheme. Future Generation Computer Systems-the International Journal of Escience. 2024; 154: 330-343. https://doi.org/10.1016/j.future.2024.01.012

[6] Zhang JX, Yan LXM. A quantitative diary study of perceptions of security in mobile payment transactions. Behaviour & Information Technology. 2021; 40(15):1579-1602. https://doi.org/10.1080/0144929X.2020.1771418

[7] Economou EML, Kyriazis NA. Achieving sustainable financial transactions under regimes without a central bank-an intertemporal comparison. Sustainability. 2021; 13(3): 1071. https://doi.org/10.3390/su13031071

[8] Hong HS, Sun ZX. A secure peer to peer multiparty transaction scheme based on blockchain. Peer-to-Peer Networking and Applications. 2021; 14(3):1106-1117. https://doi.org/10.1007/s12083-021-01088-4

[9] Pon P, Kavitha V. Blockchain based cloud service security architecture with distributed machine learning for smart device traffic record transaction. Concurrency and Computation-Practice & Experience. 2022; 34(3): e683. https://doi.org/10.1002/cpe.6583

[10] Anagnostakis AG, Giannakeas N, Tsipouras MG, Glavas E, Tzallas AT. IoT Micro-Blockchain fundamentals. Sensors. 2021; 21(8): 2784. https://doi.org/10.3390/s21082784

[11] Rehman S, Khan B, Arif J, Ullah Z, Aljohani AJ, Alhindi A, Ali SM. Bi-Directional mutual energy trade between smart grid and energy districts using renewable energy credits. Sensors. 2021; 21(9): 3088. https://doi.org/10.3390/s21093088

[12] Su JQ, He L, Ren RT, Liu QL. Reliable blockchain-based ring signature protocol for online financial transactions. KSII Transactions on Internet and Information Systems. 2023; 17(8): 2083-2100.

https://doi.org/10.3837/tiis.2023.08.007
[13] Luo M, Zhou J, Yang P. RATS: A regulatory anonymous transaction system based on blockchain. Journal of Parallel and Distributed Computing. 2023; 182: 104751. https://doi.org/10.1016/j.jpdc.2023.104751
[14] Fan HL. The digital asset value and currency supervision under deep learning and blockchain technology. Journal of Computational and Applied Mathematics. 2022; 407: 114061. https://doi.org/10.1016/j.cam.2021.114061
[15] Umar A, Kumar D, Ghose T. Blockchain-based decentralized energy intra-trading with battery storage flexibility in a community microgrid system. Applied Energy. 2022; 322: 119544. https://doi.org/10.1016/j.apenergy.2022.119544
[16] Cheng JR, Zhang Y, Yuan YM, Li H, Tang XY, Sheng VS, Hu GJ. PoEC: a cross-blockchain consensus mechanism for governing blockchain by blockchain. Cmc-Computers Materials & Continua. 2022; 73(1):1385-1402. https://doi.org/ 10.32604/cmc.2022.026437
[17] Yi Y. Application of blockchain technology based on privacy data protection in RMB internationalization path. Mobile Information Systems. 2022. https://doi.org/10.1155/2022/1904593
[18] Sun G, Dai M, Sun J, Yu HF. Voting-Based decentralized consensus design for improving the efficiency and security of consortium blockchain. IEEE Internet of Things Journal. 2021; 8(8):6257-6272. https://doi.org/ 10.1109/JIOT.2020.3029781
[19] Attarian R, Hashemi S. An anonymity communication protocol for security and privacy of clients in IoT-based mobile health transactions. Computer Networks. 2021; 190: 107976. https://doi.org/10.1016/j.comnet.2021.107976
[20] Khan HU, Sohail M, Nazir S, Hussain T, Shah BB, Ali F. Role of authentication factors in Fin-tech mobile transaction security. Journal of Big Data. 2023; 10(1): 138. https://doi.org/10.1186/s40537-023-00807-3
[21] Wei Q, Li BZ, Chang WL, Jia ZP, Shen ZY, Shao ZL. A survey of blockchain data management systems. ACM Transactions on Embedded Computing Systems. 2022; 21(3): 1-28. https://doi.org/10.1145/3502741
[22] Geng ZQ, Cao Y, Li J, Han YM. Novel blockchain transaction provenance model with graph attention mechanism. Expert Systems with Applications. 2022; 209: 118411. https://doi.org/10.1016/j.eswa.2022.118411
[23] Zhang JY, Zhou P, Wang J, Alfarraj O, Singh S, Zhu M. A novel high-efficiency transaction verification scheme for blockchain systems. Computer Modeling in Engineering & Sciences. 2024; 139(2):1613-1633. https://doi.org/ 10.32604/cmes.2023.044418
[24] Li XY, Zheng ZB, Dai HN. When services computing meets blockchain: Challenges and opportunities. Journal of Parallel and Distributed Computing. 2021; 150: 1-14. https://doi.org/10.1016/j.jpdc.2020.12.003
[25] Dai WQ, Wang QY, Wang ZL, Lin XB, Zou DQ, Jin H. Trustzone-based secure lightweight wallet for hyperledger fabric. Journal of Parallel and Distributed Computing. 2021; 149: 66-75. https://doi.org/10.1016/j.jpdc.2020.11.001
[26] Kotey SD, Tchao ET, Ahmed AR, Agbemenu AS, Nunoo-Mensah H, Sikora A, et al. Blockchain interoperability: the state of heterogenous blockchain-to-blockchain communication. IET Communications. 2023; 17(8): 891-914. https://doi.org/10.1049/cmu2.12594
[27] Maldonado-Ruiz D, Pulval-Dady A, Shi YL, Wang Z, El Madhoun N, Torres J. NestedChain: "Blockchain-inside-a-Blockchain" new generation prototype. Annals of Telecommunications. 2024; 1-19. https://doi.org/10.1007/s12243-024-01030-8
[28] Shamieh F, Wang XB, Hussein AR. Transaction throughput provisioning technique for blockchain-based industrial IoT networks. IEEE Transactions on Network Science and Engineering. 2020; 7(4):3122-3134. https://doi.org/ 10.1109/TNSE.2020.3017389
[29] Wang ZQ, Ni AF, Tian ZQ, Wang ZY, Gong YG. Research on blockchain abnormal transaction detection technology combining CNN and transformer structure. Computers & Electrical Engineering. 2024; 116: 108194. https://doi.org/10.1016/j.compeleceng.2024.109194
[30] Gams M, Kolenik T. Relations between electronics, artificial intelligence and information society through information society rules. Electronics, 2021, 10(4): 514. https://doi.org/10.3390/electronics10040514
[31] Maknickas A, Maknickiene N. Support system for trading in exchange market by distributional forecasting model. Informatica, 2019, 30(1): 73-90. https://doi.org/10.15388/Informatica.2019.198

Application of CNN-DAE Deep Learning for Economic Cost Prediction in Construction Engineering

Xueqin Yang
Management Department, Chongqing College of Humanities, Science & Technology, Chongqing 401520, China
E-mail: 13212453469@163.com

Keywords: CNN, DAE, economic cost of construction

Received: August 29, 2024

Abstract: This paper aims to explore the effectiveness of constructing the construction economic cost prediction model of construction engineering by using CNN-DAE algorithm combined with CNN-DAE. As the construction cost of construction engineering is affected by a variety of complex factors and has high nonlinear and dynamic changes, traditional prediction methods are often difficult to achieve ideal prediction accuracy. Therefore, this paper proposes an innovative predictive model to address these challenges. Simulation prediction is carried out on Matlab platform through actual cases, sample data is normalized, and then sample data is quantified and predicted in Matlab environment. The relative error of prediction results obtained through data analysis of test samples is strictly controlled below 2%. At the same time, the stability analysis of the model obtained that the maximum relative error of the prediction was only 1.96%, which verified that the CNN-DAE algorithm based on the construction economic cost prediction model of construction engineering could more accurately capture the change trend of construction cost in terms of prediction accuracy and provide strong decision support for project managers. In addition, the model has strong generalization ability, and can adapt to different scales and types of construction projects.

Povzetek: Raziskava predlaga CNN-DAE algoritem za napoved stroškov v gradbeništvu. Model zmanjšuje napake, izboljšuje odločanje pri projektnih stroških in virih.

1 Introduction

In the field of construction engineering, construction economic cost forecast is the key link of project management, which is directly related to the project budget preparation, investment decision and the overall economic benefit. With the rapid development of the construction industry and the increasingly fierce market competition, how to predict the economic cost of construction scientifically and accurately has become an important topic for construction enterprises to enhance their competitiveness and optimize the allocation of resources. In recent years, the rapid rise of artificial intelligence technology and big data has opened up novel and efficient ways and strategies for solving complex problems such as construction project cost prediction. The research on using artificial intelligence technology to predict engineering cost is quite abundant at home and abroad, mainly around two major paths. First, by adjusting or optimizing neural network parameters to achieve accurate prediction of construction cost and construction cost; Second, the neural network is improved by other mathematical analysis methods to achieve the purpose of prediction. The specific research is shown in Table 1. However, these research methods are generally faced with the problem of improper selection of theoretical methods or imperfect construction of influencing factor index system, thus affecting the accuracy and stability of prediction. It is worth noting that currently no scholars in China have applied deep learning to construction cost prediction, and relevant researches mainly focus on the modeling of deep learning network and the exploration of improvement strategies.

Table 1 Analysis of existing research results

Sajadfar [1] combines linear regression and data mining techniques to propose an informatics framework-based engineering feature applied to data mining and the cost estimation supported by the algorithm. The principle of linear regression method is to use mathematical statistics to study the functional relationship between variables. The prediction accuracy is greatly affected by the data.
Xie [2] established a database by collecting the comprehensive unit price of the list of priced quantities of similar projects, and used the grey prediction model to predict the cost of sub-projects. Grey profit determines whether the relationship is close according to the similarity of geometric shapes of the series curves. Suitable for short-term forecast, the forecast accuracy is not high.
Putra [3] proposes an artificial intelligence calculation method based on BP neural networks to model the direct and indirect costs of typical completed projects in order to reduce uncertainty. This method is especially suitable for evaluating the cost of installation projects, and the total uncertainty of project cost estimation is greatly reduced. The neural network is a multi-layer feedforward neural network, which adjusts the connection weight between a large number of internal nodes by simulating the behavior characteristics of the animal neural network. The prediction result is easy to fall into

the local minimum, and the prediction accuracy is high.

Since the construction cost of construction engineering is affected by multiple complex factors and the collection of relevant data is difficult, its cost prediction has become a typical large sample, high dimension and nonlinear challenging problem. In addition, rapid and accurate prediction of construction costs is very important for the cost management of enterprises, and there are high requirements for the accuracy and timeliness of prediction. Traditional forecasting methods, such as regression analysis and grey theory, are often unable to meet the accuracy and time requirements of prediction when dealing with such problems. In contrast, the deep learning network model, with its advantages, can effectively avoid these problems, while showing good generalization ability. On the basis of in-depth analysis of traditional construction engineering cost prediction methods, this study chooses an advanced algorithm combining convolutional neural network (CNN) and noise reduction autoencoder (DAE), and uses its powerful learning and prediction mechanism to build a prediction model and train and learn the sample data. This model can accurately and quickly predict the construction cost of nonlinear engineering projects.

Convolutional neural network (CNN) and Deep autoencoder (DAE), as two important tools in the field of machine learning, have achieved remarkable results in many fields such as image recognition and natural language processing due to their powerful feature extraction and data processing capabilities. By simulating the hierarchical structure of the human visual system, CNN can automatically extract high-level abstract features from the original data, which is especially suitable for processing data with spatial structure, while DAE learns the compressed representation of the input data through unsupervised learning and reconstructs the original data in the decoding process, thus achieving data reduction and denoising. By combining CNN and DAE, a CNN-DAE algorithm based economic cost prediction model of construction engineering is constructed, aiming to make full use of the advantages of these two algorithms to improve the accuracy and efficiency of cost prediction [4]. The model automatically extracts the features of key cost factors in construction projects through CNN and uses DAE to further reduce and de-noise these features to eliminate redundancy and noise in the data and improve the generalization ability of the model.

This study aims to explore and introduce an innovative, scientific and rigorous cost forecasting method into the field of construction engineering, which is expected to significantly improve the accuracy of construction cost control, optimize the project management process, and further enhance the economic benefits of construction enterprises. It is expected that this study can provide practical cost control tools for practitioners in the construction industry [5]. It can also open up new research perspectives for academic researchers, promote the integration and application of interdisciplinary knowledge, and jointly promote the progress and development of the field of construction engineering cost forecasting.

2 Analysis of the influencing factors of the economic cost of construction of building projects

In the complex process of building construction, cost control is faced with dual challenges from predictable and unpredictable factors. For the predictable factors, it is feasible to control them through active management strategies, while the sudden occurrence of unpredictable factors becomes a big problem in cost control. In-depth analysis of the key factors affecting the cost of construction projects, mainly covering the following aspects.

2.1 Management system factors

Management system is the core support of construction project cost control. If there are loopholes in the management system, it will not only aggravate the uncertain events in the construction process, but also disrupt the original cost budget, resulting in a significant increase in project cost [6]. Specifically, the imperfection of the material management system will push up the cost of materials and affect the accuracy of the cost budget, while the defects of the personnel management system may delay the construction progress, and even cause mechanical damage due to improper operation, indirectly increasing the cost of mechanical loss in the construction.

2.2 Construction factors

The management efficiency in the construction process directly affects the cost control. Frequent rework not only directly increases the cost of the project, but also reflects the lack of management responsibility system of the construction site. If the project supervision fails to strictly follow the guidance of construction organization design, and the acceptance and supervision of concealed works, division works and individual projects are not in place, it will be difficult to find and solve problems in time, resulting in frequent rework phenomena and increased difficulty in cost control [7]. In addition, if the importance of safe and civilized construction is ignored, it may lead to quality and safety problems, forcing the suspension of the project, extending the construction period and interfering with the smooth implementation of the cost control program.

2.3 Construction material factors

Material cost occupies the core position in the cost control of construction engineering, and its increase is often attributed to the rise of material price or the increase of consumption. Market price fluctuations, especially the price changes of key materials such as steel bars and cement, are external factors that are difficult to predict completely in cost management [8]. The increase in the amount of materials may be related to the structural changes or design adjustments of the project, which further aggravates the pressure of cost control. In addition, the problem of material waste in the construction process can not be ignored, which may be caused by improper construction technology, methods or loose material management.

2.4 Other factors

In addition to the above factors, project supervision is not in place, personnel responsibility consciousness is weak, the rationality of construction organization arrangement and the strength of enterprise technical strength are all important factors affecting the cost control effect of construction projects [9]. These factors interweave and work together in the whole process of cost control, requiring construction units to take comprehensive consideration and comprehensive policies in project management.

3 CNN-DAE algorithm modelling

3.1 CNN network model

Convolutional neural networks (CNN), as a pillar of deep learning, show remarkable efficiency in processing data with grid structure due to their unique structural design, especially for time series data and image data. Such data can be regarded as one-dimensional grid formed by regular sampling in time dimension and image grid composed of pixels in two-dimensional space respectively. Through its powerful feature extraction and learning ability, CNN has achieved excellent performance in many application scenarios [10]. The core advantages of CNN come from its three core ideas: sparse interaction, parameter sharing and isovariant representation. Sparse interaction means that each output neuron is only connected to a portion of the input data, which effectively reduces the number of network parameters, reduces computational complexity and also helps to extract local features. Parameter sharing means that all neurons in the same layer use the same convolution kernel for convolution operations, which further reduces the number of parameters and increases the generalization ability of the model. Equivariant representation makes the model invariant to some transformations of the input data, which is particularly important for tasks such as image recognition. The basic framework of convolutional neural networks is shown in Figure 1.

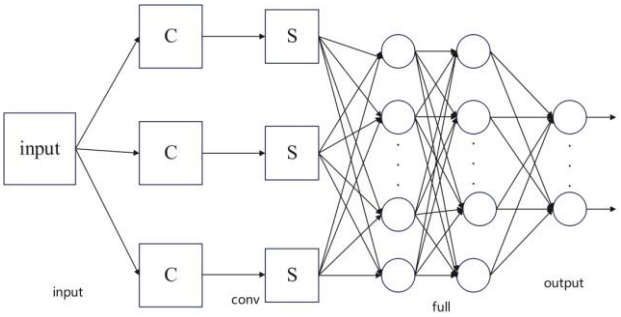


Figure 1: Basic framework of convolutional neural networks

The basic framework of CNN consists of several layers, the core of which is the convolution layer and the pooling layer. The input image or the feature map of the previous layer in the convolution layer will undergo convolution operation with multiple convolution cores. This process can be visually understood as sliding a window (i.e. convolution kernel) on the image or feature map, calculating the sum of the product of the elements in the window and the convolution kernel each time it slides to a new position, and obtaining the output value of the position [11]. As the whole image or feature map is traversed with the sliding of the window, each convolution kernel will generate a corresponding feature map, reflecting the feature information of the image at different angles or levels. The use of multiple convolution kernels can capture richer and more diverse features in the image. The operation formula of the JTH feature map of layer l is shown in Equation (1), which is expressed as x_j^l .

$$x_j^l = f\left(\sum_{i \in Mj} X_i^{l-1}\right) \times K^l + b_j^l \tag{1}$$

In Equation (1), l represents the number of layers in the convolutional neural network; K stands for convolution kernel; M_j represents receptive field; b_j represents a bias value.

The main purpose of the pooling layer is to reduce the complexity of the network by reducing the dimension (i.e. height and width) of the feature graph, thereby reducing the computational effort and avoiding overfitting. This process, often referred to as pooling or sampling, is an important step in the feature extraction process [12]. Pooling operation realizes dimensionality reduction by statistical calculation of local areas of the input feature map. There are various statistical methods, but the most common ones with significant effects include Max Pooling, Average Pooling and L2-norm pooling. And the weighted mean pooling based on the distance of center pixels. The calculation formula of pooling is shown in Equation (2).

$$x_j^l = \beta_j^l downsample\left(X_j^{l-1}\right) + b_j^l \tag{2}$$

3.2 DAE network model

Deep autoencoder (DAE) is a deep learning model that utilizes unsupervised learning. It automatically extracts features of high-dimensional complex input data from a large number of unlabeled data by performing greedy pre-training layer by layer and subsequent parameter optimization process [13]. The core architecture of DAE consists of three main parts: Encoder, Hidden Layers and Decoder.

The encoder part maps the raw input data x to a lower-dimensional implicit representation h, expressed as Equation (3). This mapping process is non-linear and aims to capture key information or features in the input data while removing or ignoring unimportant details or noise. In this way, the encoder is able to efficiently compress data and achieve feature dimensionalization while retaining enough information for subsequent reconstruction processes.

$$h = f_\theta(x) = S(W_x + b) \tag{3}$$

In Equation (3), S is the Sigmoid activation function.

Hidden layers lie between the encoder and decoder, and they make up the "depth" part of the depth autoencoder [14]. These layers further process the encoded data through nonlinear transformations to extract deeper, more abstract feature representations, and the number of hidden layers and the number of neurons in each layer can be adjusted according to the specific task and the characteristics of the data set.

The decoder function g (h) is responsible for mapping the implicit representation h back into the original input space, and reconstructing the original input data y as much as possible, expressed as Equation (4) [15]. This process verifies the validity of the features extracted by the encoder, and also provides a training signal to the entire network through a backpropagation algorithm to optimize the parameters of the encoder and decoder so that the reconstruction error is minimized.

$$y = g(h) = S_g(Wh + b_y) \tag{4}$$

In Equation (4), S_g represents the activation function of the decoder, Sigmoid function.

The training process of DAE is an optimization problem, which aims to find the optimal value of network parameter $\theta = \{W,\ b_y, b_h\}$ on a given set of unlabeled data sample set X through unsupervised learning to minimize the reconstruction error. Reconstruction error is a measure to measure the difference between the input data reconstructed through the coding-decoding process and the original input data. It is usually used as a Cost Function or Loss Function in the training process. The expression of reconstruction error is shown in Equation (5).

$$J_{AE} = \sum_{x \in D} L(x, g(f(x))) \tag{5}$$

In Equation (5), L is the reconstruction error function, which is represented by the square error function as shown in Equation (6).

$$L(x, y) = \|x - y\|^2 \tag{6}$$

Stack noise reduction autoencoders are an important component of deep learning architecture. It builds a hidden layer of deep learning model by stacking multiple noise reduction autoencoders [16]. By extracting the high-level feature representation of the data layer by layer, this structure realizes the effective reduction and denoising of the input data. The calculation formula of the coding process of the modified denoising automatic encoder is shown in Equation (7).

$$y = f_\theta(x) = s(W_x + b) \tag{7}$$

Compared with traditional autoencoders, noise interference is introduced in the training process, that is, artificial noise is added to the input data, and then the model is required to reconstruct or purify the noise-polluted data through learning [17]. This process forces the autoencoder to learn more robust and useful feature representations, thereby improving its ability to understand the intrinsic structure of the data.

3.3 Prediction algorithm process

A construction economic cost prediction model based on CNN and DAE algorithm is established, and these two deep learning technologies are combined to improve the accuracy and efficiency of cost prediction [18]. The following is a detailed analysis of how CNN and DAE combine to achieve this prediction algorithm process, as shown in Figure 2.

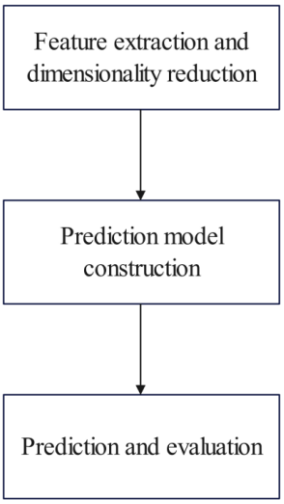


Figure 2: Predictive flow chart

(1) Feature extraction and dimension reduction

In the processing of economic cost data, feature extraction and dimensionality reduction are the key steps to build efficient prediction models. First, the powerful capability of convolutional neural network (CNN) is used to automatically capture local features in the data through its convolutional layer, which may be hidden in the timing

changes of cost data, periodic fluctuations, or complex interactions between different cost items [19]. By combining the nonlinear activation function with the sliding window, the convolutional layer can extract the deep features that are critical to cost prediction, and then the pooling layer further down-samples these features to reduce the space size of the data while retaining the key information to improve the robustness and computational efficiency of the model. Next, the high-dimensional features extracted by CNN are input into the encoder part of the denoised autoencoder (DAE), which forces the encoder to learn a more robust and abstract feature representation by introducing noise into the input data and attempting to reconstruct the original clean data [20].

(2) Prediction model construction

A series of fully connected layers are cleverly connected behind the decoder of DAE to build the final predictive model. These fully connected layers act as the "decision layer" of the model and are responsible for mapping the low-dimensional potential spatial vectors output from the DAE to specific cost predictions. The design of the fully connected layer needs to consider the number of layers, the number of neurons in each layer and the choice of activation function to balance the expressiveness and training efficiency of the model [21]. RELU function is a typical linear rectification function. In neural networks, activation function is the existence of the ability to make the neural network nonlinear, which can transform the results of each layer of neural network after linear calculation into nonlinear. The calculation formula of RELU function is shown in Equation (8), the learning rate is 0.001, which takes both convergence speed and convergence stability into account. The learning rate of network training attenuates according to the number of iteration rounds. The specific attenuation Equation (9) is shown.

$$y = \max(0, w^T x + b) \tag{8}$$

$$lr = lr_{init} * 1/(1 + decay * epoch) \tag{9}$$

Where, $\mathrm{lr}_{\mathrm{init}}$ is the initial learning rate, decay is the attenuation coefficient, and epoch is the number of rounds.

The random drop neuron ratio is 0.5. Since neurons sometimes have binding update parameters, in order to make full use of all neurons, dropout can randomly drop some neurons, break the binding update, and avoid falling into overfitting.

The training of the prediction model is an iterative optimization process, which relies on a large number of historical cost data as a training set [22]. Through the backpropagation algorithm, the model can automatically adjust the weight and bias of each layer in the CNN and DAE to minimize the error between the predicted cost and the actual cost, such as the mean square error (MSE) or the mean absolute error (MAE) [23].

(3) Prediction and evaluation

In order to verify the accuracy and generalization ability of the prediction model after the training, it is necessary to use a test set independent from the training set for evaluation. Evaluation measures include not only direct error measures between predicted costs and actual costs, but also more complex evaluation methods such as coefficient of determination (R^2), root mean square error (RMSE), and confidence levels of forecast intervals. Through these indicators, the performance of the model in different scenarios can be comprehensively evaluated, and the direction for model optimization can be provided. In practical application, prediction model has become an important tool for cost control and decision-making of construction projects. When new economic data is input, the model can quickly provide the corresponding cost forecast value, helping project managers to identify potential cost overruns in time, adjust resource allocation, optimize project plans, and thus achieve effective cost control. In addition, the forecast model can also be integrated with other management tools, such as cost estimation software, project management information system, etc. Form a more comprehensive and intelligent cost management system [24].

4 Example analysis

4.1 Data source and processing

4.1.1 Data collection

In order to ensure the accuracy of the model prediction results, it is necessary to collect engineering data that is highly similar to the engineering cases to be predicted as a training set. In view of the significant differences between regions in labor costs, prices of main materials, construction techniques and technical solutions, the scope of sample selection was strictly limited [25], and only residential projects in a certain city and its neighboring cities were included. Through the comprehensive use of engineering cost information network, professional cost information journals and other authoritative resources carefully selected and collected a total of 20 groups of high-rise residential project data from 2018 to 2023. These projects are residential buildings of 22 to 30 storeys and are geographically concentrated in and around Nanchang, thus ensuring a high degree of uniformity and comparability of the samples in terms of geography and type [26].

Then the preliminary data statistics and analysis of these samples were carried out. The construction process of a construction project often takes a long time and is affected by a variety of factors, and the diversity of its design scheme is also an important aspect that determines the project cost. In general, the cost of construction project is affected by four categories of factors: engineering attribute, environmental attribute, market attribute and management attribute. The engineering attribute focuses on the uniqueness of the structural design of the building project itself, including the building area, the floor area, the standard floor area, the foundation type, the structure type, the staircase construction mode, the seismic grade,

the facade decoration style, the number of floors and the height and other key parameters. The environmental attributes focus on the specific conditions of the construction site, such as the availability of the construction site, the convenience of sewage treatment, the difficulty of earthwork excavation and foundation treatment. Market attributes are closely related to market economy environment, including labor cost, price level of basic building materials such as concrete and steel bar and its change trend; The management attribute relates to the main management party in the implementation process of the construction project, as well as the arrangement and control of the construction cycle [27].

This study identifies and extracts five key engineering features (i.e., influencing factor X) that affect the economic cost of construction projects, namely, structure type, number of building floors, building height, construction period and construction area. At the same time, a corresponding project cost data (i.e., the affected factor Y) is collected as the target value predicted by the model. Through this series of rigorous data collection and collation work, the training sample set containing all feature vectors is successfully constructed, which lays a solid foundation for the training and optimization of the subsequent model.

4.1.2 Data processing

In order to ensure that the self-coding network will not encounter the output saturation problem due to the large absolute difference of the input data when processing data, which may lead to low training efficiency or premature convergence to the local optimal solution, it is crucial to normalize the input and output vectors in the training data sample set X . This step aims to uniformly adjust the range of data to the interval [0, 1] to promote the stability and efficiency of network learning. Linear normalization (also known as min-max normalization) is adopted as the method of data transformation, which is implemented by the following Equation (10).

$$\bar{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{10}$$

In Equation (10), $\bar{x}_i$ represents the data after normalization; x_i represents the data collected now; $x_{\min}$ represents the minimum value of such data; $x_{\max}$ represents the maximum value of such data. This transformation ensures that all data points are mapped linearly to the interval [0, 1], thus effectively avoiding the adverse impact on network training caused by excessive difference in data scale. For each input vector and output vector in the training data sample set X, the above linear normalization formula will be applied for processing to ensure that the entire data set has a consistent scale at the input and output levels, which provides a good basis for the subsequent self-coding network training.

At the same time, in order to improve the efficiency and accuracy of the model's processing and analysis of sample data, the data obtained by preliminary statistics were quantified, and the quantified results were presented in the form of input vectors in Table 2.

Table 2: Engineering feature vector quantization table

Influencing factor/unit	Sample 1	Sample 2	…	Sample 20
Structure type (X1)	3	3	…	3
Number of floors (X2)	23	25	…	30
Height of building (X3)/m	3.00	3.02	…	2.98
Construction period (X4)/d	230	450	…	120
Area of structure (X5)/m²	10319.95	9.555.54	…	14.631.6

4.2 Prediction operation of data

Equation (9) is applied to transform the quantized data, and then these processed data are imported into the Matlab environment for further processing and predictive analysis.

The training process of predictive analysis was performed on these data in Matlab, and the training process showed a convergence trend of the predicted data in about 80 to 100 iterations. Specifically, when the number of training iterations reaches the 88th time, the global error in the unsupervised learning stage reaches 0.05, indicating that the model has learned certain data features at this time, and the global error further decreases to 0.042 at the 100th iteration, indicating the continuous improvement of the model performance.

Entering the supervised learning phase, the model is trained on more detailed labeled data to further improve the prediction accuracy. After several iterations, when the number of iterations reached 424 times, the global error reached 0.001, which indicates that the prediction accuracy of the model has basically met the requirements. When the training continued to the 500th iteration, the global training error was reduced to 0.0008, which further proved the efficiency and accuracy of the model in the prediction task.

4.3 Model accuracy analysis

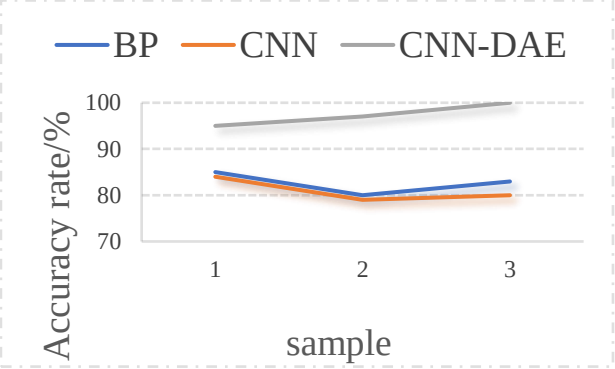
By redrawing and comparing and analyzing the predicted and actual construction cost of the test samples, the relative error data shown in Table 3 was obtained.

Table 3: Relative error between actual construction cost and predicted value of test sample

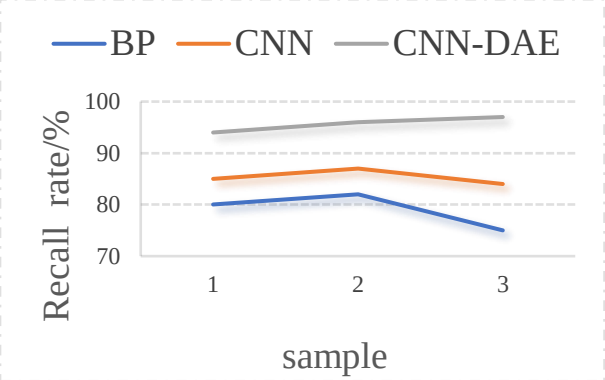
			(Ten thousand)
Engineering sample number	Actual construction cost (a)	Projected construction cost (b)	Relative error/% (b-a)/a
15	750.66	760.22	1.27
16	2125.31	2083.65	1.96
17	1123.64	1132.83	0.82

Through in-depth analysis of the information in Table 3, it can be clearly observed that in the three groups of selected prediction data, the relative error of all prediction results is strictly controlled below the threshold of 2%, which fully demonstrates the high-precision performance of the prediction model. Specifically, the relative error range of these three forecast samples is excellent, with the maximum error being only 1.61% and the minimum error being less than 1.00%. In the cost management of the construction phase of the project, it is generally believed that when the relative error of the forecast can be kept within 2%, the forecast result can be used as an important basis for construction organization decision-making. Therefore, based on the above analysis, it can be confirmed that the construction cost prediction model constructed by CNN-DAE algorithm has excellent prediction effect and ideal performance in relative error control, which fully verifies the high reliability and accuracy of this model in practical application.

BP and Convolutional neural network (CNN) are two common neural network algorithms. In order to verify the feasibility of the CNN-DAE neural network proposed in this study, the accuracy and recall rate of the three algorithms in the predicted value of construction cost are compared. The result is shown in Figure 3.



(a) Accuracy comparison



(b) Comparison of recall rates

Figure 3: Comparison results of algorithm accuracy and recall rate

As can be seen from Figure 3, CNN-DAE algorithm is always higher than BP algorithm and CNN algorithm in terms of prediction accuracy and algorithm recall rate.

4.4 Model stability analysis

In order to further explore the stability of the prediction effect of the new model, the relative error data in Table 3 are presented graphically in Figure 3. Several key points can be intuitively summarized from the chart. The maximum relative error of the prediction is only 1.96%, while the minimum value is 0.82%. This data range highlights the excellent performance of the prediction model in accurately controlling the relative error and its significant advantages in stability. At the same time, the results of the graphical simulation prove the accuracy and reliability of the prediction model.

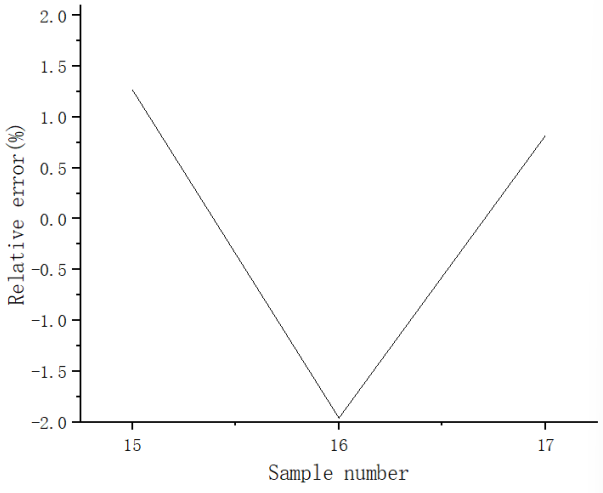


Figure 3: Predictive relative error stability

Furthermore, the mean absolute error (MAE), mean square error (MSE) and coefficient of determination (R^2) were used as evaluation indexes to quantitatively analyze the performance of the model, and the results were shown in Table 4. It can be seen that the average error of the BP model reaches 0.116, while the coefficient of

determination of the CNN model is as high as 0.902, and its prediction accuracy is improved by 0.088 compared with that of the BP model. This significant improvement is mainly attributed to the inclusion of time delay effect in the input variables of the CNN model, which increases the dimension of the input variables. And a convolutional neural network (CNN) with better performance is adopted. Furthermore, the coefficient of determination of the CNN-DAE model reached 0.943, indicating that the prediction accuracy was again increased by 4.5% when the noise reduction autoencoder (DAE) structure was introduced on the basis of the CNN model.

Table 4: Comparison of evaluation indexes of prediction performance of the three models

Prediction model	RMSE	MAE	R^2
BP	0.172	0.116	0.814
CNN	0.122	0.083	0.902
CNN-DAE	0.093	0.071	0.943

It is worth mentioning that the construction of the entire CNN-DAE algorithm model and its practical application in the construction cost prediction of construction engineering all rely on Matlab, a powerful programming platform. Through a series of carefully designed programming operations, the prediction process can be efficiently realized. Compared with traditional cost estimation and prediction methods such as budget software or BP neural network, the model shows obvious advantages in the prediction speed, and reaches a new height in the stability and accuracy of the prediction results.

4.5 Discussion

With the increase of the complexity of CNN-DAE model, such as increasing the number of convolutional layers or enlarging the size of convolutional nuclei, although the prediction accuracy of the model may be improved, the computational amount will also be significantly increased, thus affecting the computational efficiency. When dealing with large data sets, even with efficient computing devices and algorithms, long computing times may be required. This may cause the model to fail to give timely prediction results in scenarios with high real-time requirements. In some cases, the available computing resources (such as cpus, Gpus, etc.) may not meet the needs of model training or prediction. This can lead to reduced computational efficiency and even inability to complete model training and prediction tasks. As the size of a construction project increases, the diversity of project data is likely to increase. This includes different types of construction projects, different geographical locations, different materials and technologies, etc. This diversity can make it difficult for the model to capture all relevant features during training, thus affecting the prediction accuracy. When construction projects vary widely in size, models may struggle to maintain consistent predictive accuracy across projects of all sizes. This may require training models separately for different types of projects, or adopting more complex model structures to improve generalization.

To sum up, the algorithm still has some potential limitations in practical applications, such as certain prediction delay when processing large-scale data sets or performing complex calculations, which depends on specific hardware configuration and data processing requirements. In addition, high-precision prediction is often accompanied by high computing costs, which may pose certain challenges to systems with limited computing resources. Therefore, in actual deployment, it is necessary to consider the balance between model performance and computing resources to ensure that accurate prediction results can be obtained in real-time applications, and the delay and computing cost in the prediction process can be effectively controlled, which remains to be further studied.

5 Conclusion

In the process of exploring the construction economic cost prediction model based on convolutional neural network (CNN) and deep autoencoder (DAE), the potential and advantages of this model in the field of complex engineering cost estimation are deeply analyzed. By integrating the powerful feature extraction capability of CNN and the excellent unsupervised learning capability of DAE, the model can not only capture hidden and deep-level cost influencing factors from massive construction data, but also effectively deal with noise and redundant information in the data, thus significantly improving the accuracy and robustness of cost prediction.

This study provides a new idea and method for the fine management of the economic cost of construction engineering, and also provides a valuable reference for the subsequent relevant research in the aspects of algorithm optimization, feature selection and data preprocessing. In the future, it is expected to further expand the application range of this model, such as combining more types of engineering data and introducing other advanced artificial intelligence technologies to build a more comprehensive and intelligent economic cost prediction system for construction engineering. Meanwhile, we will continue to pay attention to the performance of the model in practical applications, and constantly optimize the algorithm structure to improve the prediction accuracy. Contribute wisdom and strength to the high-quality development of the construction engineering industry.

Funding

This work was supported by Chongqing College of Humanities, Science & Technology 2024 Key Projects (CRKZK2024001).

References

[1] Sajadfar, N., & Ma, Y. A hybrid cost estimation framework based on feature-oriented data mining approach. Advanced Engineering Informatics, 2015,

29(3): 633-647. https://doi.org/10.1016/j.aei.2015.06.001

[2] Xie, Y. M. Pile foundation engineering cost prediction research based on the theory of grey system. Journal of Engineering Construction and Design, 2011(12): 135. https://doi.org/10.3969/j.issn.1007-9467.2011.12.047.

[3] Putra, S. A. G., & Triyono, A. R. Neural network method for instrumentation and control cost estimation of the EPC Companies Bidding Proposal. Procedia Manufacturing, 2015, 498-106. https://doi.org/10.1016/j.promfg.2015.11.019

[4] Lilhore, U. K., Simaiya, S., Dalal, S., Faujdar, N., Sharma, Y. K., & Rao, K. B., et al. (2024). ProtienCNN-BLSTM: An efficient deep neural network with amino acid embedding-based model of protein sequence classification and biological analysis. Computational Intelligence, 40(4), e12696. https://doi.org/10.1111/COIN.12696

[5] Bai, Y. (2023). Research on Intelligent Prediction of Engineering Cost Based on Artificial Intelligence. International Journal of New Developments in Engineering and Society, 7(1). https://doi.org/10.25236/IJNDES.2023.070110

[6] Xu, Y., & Cao, S. (2023). Building Engineering Cost Prediction Model Based on TSNE and Improved Grey Correlation Algorithm. Procedia Computer Science, 228, 957-965. https://doi.org/10.1016/J.PROCS.2023.11.126

[7] Rahim, M. A., Rahman, M. M., Islam, M. S., Muzahid, A. J. M., Rahman, M. A., & Ramasamy, D. (2024). Deep learning-based vehicular engine health monitoring system utilising a hybrid convolutional neural network/bidirectional gated recurrent unit. Expert Systems with Applications, 257, 125080. https://doi.org/10.1016/J.ESWA.2024.125080

[8] El-Hag, N. A., El-Hoseny, H. M., & Harby, F. (2024). DNN-driven hybrid denoising: advancements in speckle noise reduction. Journal of Optics, 1-10. https://doi.org/10.1007/S12596-024-02066-8

[9] Park, U., Kang, Y., Lee, H., & Yun, S. (2022). A stacking heterogeneous ensemble learning method for the prediction of building construction project costs. Applied sciences, 12(19), 9729. https://doi.org/10.3390/APP12199729

[10] Ye, D. (2021). An algorithm for construction project cost forecast based on particle swarm optimization-guided BP neural network. Scientific Programming, 2021(1), 4309495. https://doi.org/10.1155/2021/4309495

[11] Deng, J. (2020). Research on the risk early warning of construction engineering under the coupling disaster of typhoons and rainstorms in coastal areas based on BP neural network. Journal of Coastal Research, 105(SI), 151-154. https://doi.org/10.2112/JCR-SI105-032.1

[12] Demiss, B. A., & Elsaigh, W. A. (2024). Application of novel hybrid deep learning architectures combining Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN): construction duration estimates prediction considering preconstruction uncertainties. Engineering Research Express, 6(3), 032102. https://doi.org/10.1088/2631-8695/AD6CA7

[13] Zhang, S., Zheng, Y., Li, X., Ye, H., Dong, L., & Huang, X., et al. (2024). A novel solar flare forecast model with deep convolution neural network and one-against-rest approach. Advances in Space Research. https://doi.org/10.1016/J.ASR.2024.06.035

[14] Alsugair, A. M., Al-Gahtani, K. S., Alsanabani, N. M., Hommadi, G. M., & Alawshan, M. I. (2024). An integrated DEMATEL and system dynamic model for project cost prediction. Heliyon, 10(4). https://doi.org/10.1016/J.HELIYON.2024.E26166

[15] Jaroenchai, N., Wang, S., Stanislawski, L. V., Shavers, E., Jiang, Z., Sagan, V., & Usery, E. L. (2024). Transfer learning with convolutional neural networks for hydrological streamline delineation. Environmental Modelling & Software, 181, 106165. https://doi.org/10.1016/J.ENVSOFT.2024.106165

[16] Lin, M., Chen, X., Chen, G., Zhao, Z., & Bassir, D. (2024). Stability prediction of multi-material complex slopes based on self-attention convolutional neural networks. Stochastic Environmental Research and Risk Assessment, 1-17. https://doi.org/10.1007/S00477-024-02792-2

[17] Saeidlou, S., & Ghadiminia, N. (2024). A construction cost estimation framework using DNN and validation unit. Building Research & Information, 52(1-2), 38-48. https://doi.org/10.1080/09613218.2023.2196388

[18] Jing, F., Zhang, M., Li, J., Xu, G., & Wang, J. (2022). Coil shape defects prediction algorithm for hot strip rolling based on Siamese semi-supervised DAE-CNN model. Assembly Automation, 42(6), 773-781. https://doi.org/10.1108/AA-07-2022-0179

[19] Li, C., Zhao, D., Mu, S., Zhang, W., Shi, N., & Li, L. (2019). Fault diagnosis for distillation process based on CNN–DAE. Chinese Journal of Chemical Engineering, 27(3), 598-604. https://doi.org/10.1016/j.cjche.2018.12.021

[20] Chen, L., & Wang, D. (2024). Cost estimation and prediction for residential projects based on grey relational analysis–lasso regression–Backpropagation neural network. Information, 15(8), 502. https://doi.org/10.3390/INFO15080502

[21] Zhang, R. (2024). A mathematical model for project cost prediction combining multiple algorithms. Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction, 40(XXXX), 1-11. https://doi.org/10.1680/JSMIC.23.00061

[22] Koc, K., Budayan, C., Ekmekcioğlu, Ö., & Tokdemir, O. B. (2024). Predicting cost impacts of nonconformances in construction projects using interpretable machine learning. Journal of Construction Engineering and Management, 150(1),

04023143. https://doi.org/10.1061/JCEMD4.COENG-13857
[23] Abed, Y. G., Hasan, T. M., & Zehawi, R. N. (2022). Cost prediction for roads construction using machine learning models. International journal of electrical and computer engineering systems, 13(10), 927-936. https://doi.org/10.32985/IJECES.13.10.8
[24] Jia, H. (2022). Project cost prediction for building complexes based on grey bp neural network model. Computing, Performance and Communication Systems, 6(2), 1-9. https://doi.org/10.23977/CPCS.2022.060201
[25] Zhou, L., & Song, S. (2022). Green building project cost budgeting and cost control integrating interactive VR genetic algorithm. Mathematical Problems in Engineering, 2022(1), 3734946. https://doi.org/10.1155/2021/4309495
[26] Liu, W., He, Z., Chen, H., Lin, C., & Qiu, Z. (2022). Prediction of dust abatement costs in construction demolition projects. Sustainability, 14(10), 5965. https://doi.org/10.3390/SU14105965
[27] Dalila, C., Saddek, B., & Amine, N. A. (2020). Feature level fusion of face and voice biometrics systems using artificial neural network for personal recognition. Informatica, 44(1), 85-96. https://doi.org/10.31449/INF.V44I1.2596D

Piano Transcription Using Temporal Harmonic Diagram and Transfer Window Attention in Self-Attention Networks

Qiong Wu*, Tao Yu
School of Music, Tonghua Normal University, Tonghua Jilin 134001, China
E-mail: wuqiong@thnu.edu.cn, yutao12611@sohu.com
*Corresponding author

Keywords: self attention, piano music transcription, constant q conversion, time-series harmonic diagram, mel frequency cepstral coefficients

Received: August 17, 2024

Music transcription is an important means to record and transmit music culture. However, the existing music transcription algorithms still have certain errors in practical applications. To address this problem, the study adopts constant Q conversion to process music signals, introduces note starting point and frame-level pitch recognition module and transfer window attention, constructs temporal harmonic map for music melody extraction, and adopts significance function for music melody smoothing. The study uses the MAESTRO dataset containing about 200 hours of paired audio and MIDI recordings covering different performance styles, and the MedleyDB dataset protecting 122 pieces of music. The experimental results show that the overall accuracy of the transcription algorithm is 2.58% and 2.35% higher than the other algorithms, and the raw pitch accuracy is 2.23% and 1.06% higher than the other algorithms, respectively, for a frequency point count of 600 and a search range of 0.5. The accuracy, recall, and F1 value of the transcription algorithm are 2.11%, 2.27%, and 2.21% higher than the second-best algorithm, and the removal of the window attention and recognition modules decreases the accuracy of the algorithm by 8.07% and 16.76%, respectively. The average processing time of the transcription algorithm is 7.2ms lower than that of the traditional method, and the computational complexity grows more slowly as the amount of data grows. It can be concluded that the piano music transcription algorithm can effectively improve the accuracy of music recognition and transcription, and quickly and accurately convert the relevant audio into the corresponding notes.

Povzetek: Objava predstavi uvajanje algoritem SADLN za transkripcijo klavirske glasbe. Z izboljšano točnostjo in učinkovitostjo prepoznava melodije, optimizira signale in spodbuja ohranjanje glasbene kulture.

1 Introduction

Music transcription (MT) is a process of transforming music performance into written form, which captures the basic elements of melody, rhythm, harmony, and provides a written foundation for subsequent research, teaching, and replication [1]. Although MT may not fully capture the subtle differences in actual performance, it can serve as a rough record, providing people with a framework close to primitive music performance, and has important practical significance for recording and inheriting related music culture [2-3]. The early forms of MT are oral transmission and manual recording, which can result in missing details and be easily influenced by the recorder's subjectivity. As music gradually become more diverse and complex, manual transcription clearly can not meet the needs [4]. Therefore, MT based on computer technology has replaced traditional transcription methods, but music is composed of various elements such as melody, tone, and emotion. Most existing transcription algorithms only start from signal processing, and there is a certain degree of error between the transcribed music and the actual situation [5-6]. Regarding the emotional processing of music, He N and Ferguson found that few people have applied unsupervised learning to music emotion recognition. Therefore, they proposed a two-stage model that combines supervised and unsupervised learning. This model divided music into multiple continuous segments, generated feature representations using autoencoders, and used a bidirectional long short-term memory deep learning model for music emotion classification. Experiments denoted that this model could effectively reduce the risk of overfitting in the deep learning process, and its performance was significantly better than other methods [7]. Qureshi et al. proposed a new aspect level sentiment automatic annotation technique to solve the problem of time-consuming manual annotation in sentiment analysis. This technique extracted data from five aspects: sound, video, music, lyrics, and songs, and

used the N-Gram language model to standardize the data. Experiments showed that the annotation time of this technology was much shorter than that of manual standards, and the consistency with manual standards was high, with a Kappa value of 0.9571 [8]. Orjesek et al. extracted significant emotional features from audio signals and proposed a deep learning-based convolutional neural network (CNN). This network used one-dimensional convolution as the overlay layer, and employed iterative reconstruction layers based on autoencoders and bidirectional gated loop units to mine significant features related to music emotions from the original audio waveform. Experiments showed that this network could significantly improve regression accuracy and enhance the ability to judge emotional features [9]. Yin et al. found that the reliability of emotion recognition was low during large-scale testing, and therefore proposed an end-to-end multimodal framework. This framework was grounded on a one-dimensional residual spatiotemporal attention network, which utilized EDA's emotion recognition and channel spatiotemporal attention mechanism to mine the dynamic and stable features of the channel. Experiments showed that this framework could effectively mine EDA features and achieve large-scale emotion recognition [10].

In terms of accuracy in transcription, Venkata Lakshmi et al. proposed a model combining discriminant fuzzy function and deep neural network to improve the anti-interference ability of speech recognition. This model used Mel frequency cepstral coefficients (MFCC) to extract speech features. Hybrid deep neural networks and discriminative fuzzy logic were used to improve speech clarity, and the bat algorithm with enhanced modular functions was used to optimize model parameters. Experiments showed that this model could effectively improve speech recognition performance and anti-interference ability, with an accuracy 10.25% higher than that of deep autoencoders [11]. Ríos-Vila proposed a speech recognition algorithm grounded on Hidden Markov Model to improve the accuracy and recognition efficiency of speech recognition. This algorithm used Hamming windows to add windows to digital signals and employs backpropagation neural networks to reduce speech recognition errors. Experiments showed that this algorithm had the lowest word recognition error rate and the shortest recognition time [12]. Shashidhar et al. proposed a model combining MFCC and long short-term memory networks to improve speech recognition accuracy. This model used MFCC for audio processing, employed long short-term memory networks for visual speech recognition, and finally integrated them in a deep neural network. The experiment showed that the accuracy of the model for speech recognition was 91%, which was significantly better than existing algorithms [13].

In summary, existing methods have studied the accuracy and anti-interference of extracting and recognizing melody emotions in MT from multiple aspects, and have achieved certain research results. However, there are still some errors between the transcribed music and the actual situation. Therefore, a piano MT algorithm grounded on self-attention deep learning network (SADLN) is proposed, which innovatively uses constant Q conversion to process music signals, introduces note start and frame level pitch recognition modules and transfer window attention modules, constructs a temporal harmonic map to extract music melodies, and uses saliency function to smooth and optimize melodies. The piano MT algorithm aims to improve the accuracy and recognition efficiency of MT, and promote the protection and dissemination of music culture. This study is divided into three parts: the first part is to construct a piano MT system, the second part is to evaluate the performance of optimization algorithms, and the third part is a summary of this study and prospects for future research directions.

Based on the above related studies, Table 1 is summarized, which summarizes the characteristics, main index methods, research results and shortcomings of the related studies.

Table 1: Summary of relevant information of relevant studies

Method	Characteristics	Index	Results	Shortcomings
[7]	Unsupervised learning	Identification accuracy and efficiency	Effectively reduce the risk of overfitting	In large data environments, performance can be limited
[8]	Emotional automatic labeling technology	Kappa price	High standard consistency, fast speed	Performance degrades faster in complex environments
[9]	Iterative reconstruction layer and bi-directional gating cycle cells	Regression accuracy	Explore the significant characteristics of music emotion	High model complexity
[10]	End-to-end multimodal framework	Emotion recognition accuracy	Large-scale emotion recognition	Large-scale emotion recognition
[11]	Discriminating the fuzzy function	Identification accuracy	The accuracy was 10.25% higher than the depth autoencoder	The algorithm has a long running time

[12]	Hidden Markov model	Identify error rates and recognition time	The recognition error rate was reduced by 6.4%	The recognition accuracy is low
[13]	Mel frequency cepstral spectrum coefficient	Mel frequency cepstral spectrum coefficient	The recognition accuracy was 91%	It takes a lot of data to train
Research method	Self-attention mechanism and timing harmonic characteristics	Accuracy, recall and F1 values	Improve the accuracy and speed of music recognition and transliteration	/

2 Methods and materials

2.1 Construction of piano music transcription system based on SADLN

There are some similarities between MT and speech recognition, including recognizing the frequency and rhythm of sound signals, etc. Most of the modules of existing MT methods are converted from speech recognition methods. However, there are still some differences between the two. Speech recognition focuses on recognizing and understanding the semantics, grammatical structure, tone, and punctuation of a language [14]. MT, on the other hand, focuses on the melody, rhythm, and harmony of music, converting the notes and melodies in music into textual form or sheet music for recording and analysis. Because the study needs to obtain the time and frequency domain signals of audio, the signals of music need to be converted, and the existing conversion methods include constant Q-conversion and short time-distance Fourier transformation [15]. Considering the characteristics of music signals, the study uses constant Q conversion for music signal processing, and constant Q-conversion can better reflect the frequency characteristics of music signals. The center frequency of constant Q-conversion is distributed according to the exponential law, and the filtering bandwidth is different, but the ratio of the center frequency to the bandwidth is a constant Q. This makes the frequency of the spectral transverse axis of the constant Q-conversion is based on log2 as the bottom, and it can change the length of the filtering window according to the different frequencies of the spectral lines, so as to obtain a better performance. Since the constant Q-conversion is the same as the distribution of the scale frequencies, the amplitude values of the music signal at each note frequency can be obtained directly by calculating the constant Q-conversion spectrum of the music signal, whereas the short-time Fourier transform is suitable for analyzing non-stationary signals. The constant Q-transform produces a spectrum that is logarithmically scaled on the frequency axis rather than linearly scaled, and the window length of the spectrum changes with frequency. The variation of the window length with frequency ensures that the best frequency resolution is obtained when analyzing different frequency components. This property makes the constant Q-conversion very useful in music signal processing to better capture the time-frequency characteristics of music. The expression for constant Q-conversion is shown in equation (1).

$$Q = \frac{f}{\delta_f} = \frac{1}{2^{1/b} - 1} \tag{1}$$

In equation (1), f represents frequency, δ_f represents the frequency bandwidth at frequency f, and b represents the amount of spectral lines contained within an octave. As shown in equation (1), the constant Q is only related to the number of spectral lines b. The calculation of window length is shown in equation (2).

$$N_k = \left\lceil Q \frac{f}{f_k} \right\rceil, k = 0,1,...,K-1 \tag{2}$$

In equation (2), N_k represents the window length that varies with frequency, f means the sampling frequency, and f means the frequency of the k th component. The k th semitone frequency component of the n th frame in constant Q-conversion is calculated as shown in equation (3) [16].

$$X^Q(k) = \frac{1}{N_k} \sum_{n=0}^{N_k - 1} x(n) w_{N_k}(n) \exp\left(-j \frac{2\pi Q}{N_k} n \right) \tag{3}$$

In equation (3), $X^Q(k)$ represents the k th semitone frequency component, w represents the window function, j represents the complex frequency domain,

and n represents the current frame rate. Using constant Q-conversion can also lower the final spectral dimension obtained, reduce the number of parameters required for the model, and improve computational speed. Frequency domain quantization is performed on the music signal as shown in equation (4).

$$B(f)=\left\lfloor bins\cdot\log\left(\frac{f}{f_0}\right)+0.5\right\rfloor \quad (4)$$

In equation (4), $bins$ represents quantifying an octave into different intervals, f_0 represents the lowest frequency of each piano note, and $\lfloor\ \rfloor$ represents rounding down. The signal harmonics are calculated in the corresponding frequency domain as shown in equation (5).

$$B\left(f_v^m\right)=\left\lfloor bins\cdot\log(m+1)+bins\cdot\frac{v}{12}+0.5\right\rfloor \quad (5)$$

In equation (5), f_v^m represents the frequency of the mth harmonic of note v, m is equal to 0, it represents the fundamental frequency of the note, v is an integer between [0, 87], and m is a positive integer. After the signal conversion is completed, the signal is downsampled to reduce the audio sampling rate to 16kHz, and the audio segments are divided into small segments every 10ms with a frame length of 512bt. The allocated fragments are normalized to a mean of 0 and a variance of 1. After the data processing is completed, the specific process of the piano MT system is denoted in Figure 1.

In Figure 1, the audio data is first down sampled and divided into segments every 10 seconds to reduce data dimensionality and improve computational speed. Using MFCC for feature extraction, the audio signal is converted into a set of feature vectors with good discrimination. The SADLN has 4 layers, and in the first three layers, a random deactivation of 0.1 is used to improve the system's generalization ability and reduce the occurrence of overfitting. The attention of the first three layers of the network is also set to 0.1. In the early stage of training, the number of attention heads in the multi-head attention module of the network is set to 8, the output dimension of the attention module is set to 512, and the output dimension of the fully connected layer is set to 2048.

In the later stage of system training, the number of attention heads in the attention module is set to 12, the output dimension is set to 768, and the output dimension of the fully connected layer is set to 3072. The training results are mapped onto the output result with dimension 88. The system parameters are adjusted through target fitting, and after decoding, the output results are evaluated to obtain corresponding scores.

Because the starting point density of notes is low, it is easy to produce class imbalance during training, so the loss function is used to optimize. However, although the traditional focal loss function can be adjusted to a certain extent in the process of note translation, it is easy to pay too much attention to the samples that are difficult to classify. At the same time, the precision of parameters is high, which is easy to reduce the calculation efficiency. Therefore, the objective fitting of the music transfer system is calculated using the Logit-adjust loss function, as shown in equation (6) [17].

$$BER(f)=\frac{1}{L}\sum\nolimits_{y\in L}P_{x|y}\left(y\in\arg\max_{y\in\hat{y}}\ f_y(x)\right) \quad (6)$$

In equation (6), $BER(f)$ represents the error rate, L represents the number of categories, y represents the actual category, $P_{x|y}$ represents the probability of sample x appearing in actual category y, and $f_y(x)$ represents the model's prediction score for sample on category y. The Logit adjust loss function can record prior information based on different frequencies and adjust the bias state according to the positive and negative categories of the information. To raise the accuracy of note recognition, a note starting point and frame level pitch recognition module is added to the system, as shown in Figure 2.

In Figure 2, there are 5 shared layers, 1 independent layer, and 1 fully connected layer. After the note start point and frame level pitch features are extracted in the shared layer, the frame level pitch features enter the independent layer for prediction, which merged with other prediction terms that have added note start information in the fully connected layer. In response to the long-term dependence of MT, this study introduces transfer window attention into deep learning networks, as shown in Figure 3.

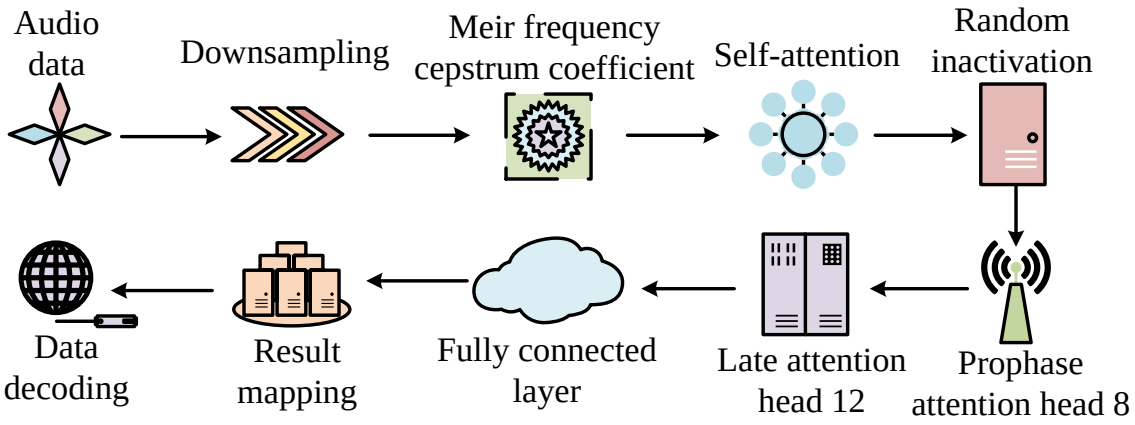


Figure 1: Specific flow of piano music transfer system

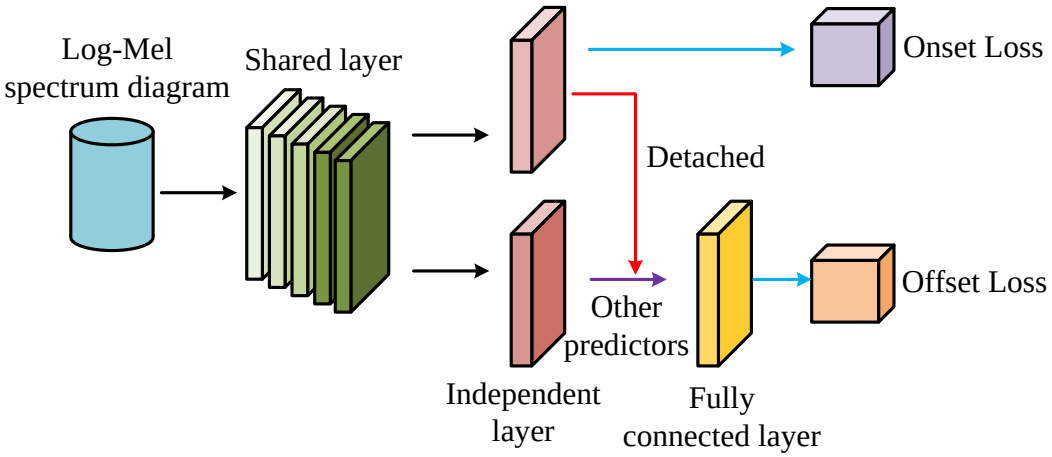


Figure 2: Note starting point and frame level pitch recognition module

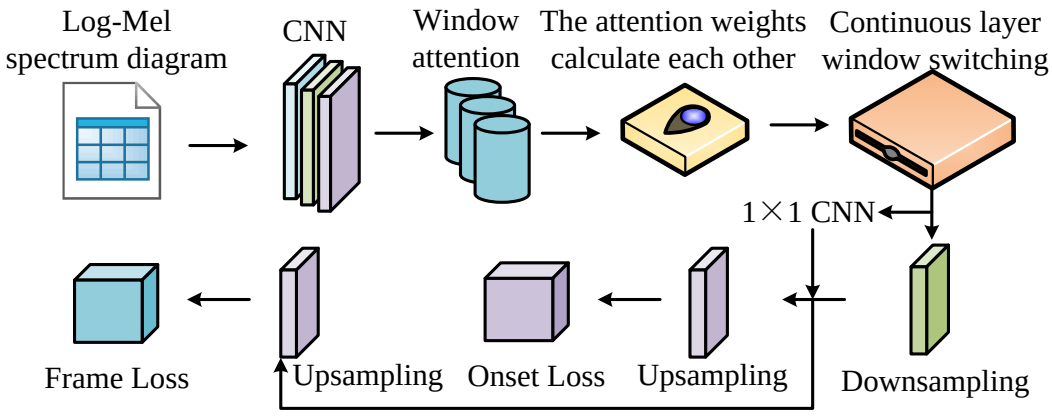


Figure 3: The introduction of transfer window attention into deep learning networks

In Figure 3, the input features are up-sampled by a CNN, and after the sampling is completed, they enter the window attention layer, which divides the information into windows of multiple sizes, and the elements within the windows can only compute the attention weights with each other. In the Shift-Window layer, the computational efficiency is improved by restricting the self-attention computation to non-overlapping localized windows while allowing cross-window connections. Window switching between consecutive layers is performed to improve the computational efficiency of the model and memory usage. After the switching is completed, one more downsampling and multiple up-sampling are performed to obtain the loss of the note onset information and the

loss of the frame-level pitch. Multiple upsampling solves the imbalance problem in the data by sampling and calculating multiple times at the note level and aggregating the final calculation results into samples, and solving the imbalance problem in the data by multiple upsampling. Up-sampling is used to increase the sampling rate of a signal by increasing the number of sampling points, while down-sampling is used to decrease the sampling rate of a signal by decreasing the number of sampling points.

2.2 Music melody extraction based on temporal harmonic characteristics

When processing audio signals, it is necessary to consider their variability and continuity on the time axis to ensure the smoothness of the audio signal. In addition to timing, audio signals also have harmonic characteristics, that is, additional harmonic components caused by nonlinear elements in the output signal that are more than the input signal. The timing and harmonic characteristics of the signal are crucial for extracting the melody of music [18]. Therefore, the study improves the transcription system based on two characteristics and proposed the Time-series Harmonic Graph Deep Learning Network (THGDLN) as shown in Figure 4.

In Figure 4, after obtaining the initial audio information, noise reduction processing is performed to enter the temporal harmonic graph network to construct the temporal and harmonic information models of the audio. Softmax is used as the activation function to output the pitch curve of the audio signal. The pitch sequence of the signal is adjusted using fine-grained functions to obtain the final musical melody. The extraction of harmonic information from audio signals adopts an undirected graph, where the nodes in the graph represent the frequency points of the spectrum, and the edges of the graph represent the connection relationships between signals with the same pitch but different frequencies. The timing information of the signal is extracted using a gate-controlled loop unit with a simple structure, short running time, and less susceptibility to gradient vanishing. The specific extraction process is shown in Figure 5.

In Figure 5, the spectral information from the previous moment is inputted, and the input value is mapped between [0,1] using the Sigmond activation function. Then, the state information from the previous moment is ignored through a reset gate. The Tanh function is used to obtain a larger output range while preserving the characteristics of the input data. Sigmond activation function can reduce the problem of gradient vanishing by normalizing or standardizing the input features so that they are distributed in a smaller range. To ensure Tanh function to prevent the gradient explosion problem, the gradient trimming technique is applied to limit the size of the gradient during the training process. The update gate controls the degree of retention of old information, determines how the information from the previous time and the current time are integrated, and weights the reset gate, update gate, and stored information from the current time to obtain the output features of the current time [19]. The music melody curve obtained by THGDLN occasionally has some abrupt changes, resulting in errors between the predicted and the true values. In this study, median filtering is used to eliminate the abrupt changes in the melody, and the calculation is shown in equation (7).

$$Y_i = me\left(s_{i-r}, s_{i-r+1}, ..., s_i, ..., s_{i+r}\right) \tag{7}$$

In equation (7), Y_i represents the filtering result, me represents sorting the data points and taking the middle value, r represents the radius of the filter, and i represents the current sitting position. According to relevant research literature, a filter radius that is too large will eliminate normal values, while a radius that is too small will result in incomplete elimination of abrupt points. Therefore, a radius size of 7 is chosen. The pitch sequence output by THGDLN is relatively discrete, and the conversion between the pitch sequence and the signal frequency is shown in equation (8).

$$f_z = 2^{(p-69)/12} \cdot 440 \tag{8}$$

In equation (8), f_z represents the converted frequency, and p represents the output pitch sequence value. However, in practical applications, each pitch value corresponds to a wide frequency range, and there is an error between the frequency converted by equation (8) and the actual frequency value, resulting in a deviation in the prediction of music melody. Therefore, it is necessary to optimize the converted frequency. The study uses a significance function for optimization, and the optimization steps are as follows. Firstly, the short-time Fourier transform is used for spectral transformation to obtain the time-frequency data of the signal, and the calculation is shown in equation (9).

$$X_l(k) = \sum_{n=0}^{M-1} w(n)x(n+lH)\exp\left(-j\frac{2\pi}{N}k_f n\right) \tag{9}$$

In equation (9), $X_l(k)$ represents the frequency spectrum composed of time-frequency data, M represents the window length of the short-time Fourier transform spectrum, $w(n)$ represents the window function of the short-time Fourier transform, l represents the frame number of the current time sequence, H represents the frame shift, k_f denotes the frame number of the time sequence in the spectrum, and N represents the number of short-time Fourier transform

points. After completing the spectrum transformation, instantaneous amplitude frequency correction is performed as shown in equation (10).

$$\begin{cases} \hat{f}_i = \left(k_i + \kappa\left(k_i\right)\right)\dfrac{f_s}{N} \\ \kappa\left(k_i\right) = \dfrac{N}{2\pi H}\varepsilon\left(\phi_l\left(k_i\right) - \phi_{l-1}\left(k_i\right) - \dfrac{2\pi H}{N}k_i\right) \end{cases} \tag{10}$$

In equation (10), $\hat{f}$ represents the corrected instantaneous frequency, k_i represents the point with the highest local absolute value in the spectrum, f_s represents the signal sampling rate, $\kappa\left(k_i\right)$ represents the frequency offset distance, ε represents the main value function, and $\phi_l\left(k_i\right)$ represents the phase spectrum at frame number l. After completing the amplitude frequency correction, it constructs the saliency function and divides the spectrum into 600 different frequency points. The calculation for each frequency point is shown in equation (11) [20].

$$B\left(\hat{f}\right) = \left\lfloor 120 \cdot \log_2\left(\frac{\hat{f}}{55}\right) + 1 \right\rfloor \tag{11}$$

In equation (11), $B\left(\hat{f}\right)$ represents the position of frequency points, and the significance value of each frequency point is calculated as shown in equation (12).

$$Sign(B) = \sum_{h=1}^{10}\sum_{i=1}^{3} T\left(\hat{a}_i\right) \cdot w\left(B, h, \hat{f}_i\right) \cdot \left(\hat{a}_i\right) \tag{12}$$

In equation (12), $Sign(B)$ denotes the significance value of the frequency point, $\left(\hat{a}_i\right)$ denotes the linear amplitude, $T\left(\hat{a}_i\right)$ denotes the amplitude threshold function, h denotes the harmonic number, and $w\left(B, h, \hat{f}_i\right)$ denotes the weight function. The value of the amplitude threshold function is shown in equation (13).

$$\begin{cases} T\left(\hat{a}_i\right) = 1, if\ 20\log_{10}\left(\dfrac{\hat{a}_m}{\hat{a}_i}\right) < 40 \\ T\left(\hat{a}_i\right) = 0, otherwise \end{cases} \tag{13}$$

In equation (13), $\hat{a}_m$ represents the maximum amplitude in the frame where the frequency point is located, and the value of the weight function is shown in equation (14).

$$\begin{cases} w\left(B, h, \hat{f}_i\right) = \cos^2\left(\delta \cdot \dfrac{\pi}{2}\right) \cdot 0.8^{h-1}, if\ |\delta| < 1 \\ w\left(B, h, \hat{f}_i\right) = 0, if\ |\delta| > 1 \end{cases} \tag{14}$$

In equation (14), δ represents the half tone distance between the harmonic and the frequency center spectrum, and the value of the weight function decreases continuously with the increase of harmonic order. After optimizing using the saliency function, the distinction between harmonics and fundamental frequencies in the spectrum becomes more distinct, and the curve is also clearer. The smooth process of the transcribed music melody is shown in Figure 6.

In Figure 6, the initial pitch sequence of the input music is transformed into a frequency sequence, and median filtering is used to eliminate abrupt changes. The position of the frequency points is transformed using a fine-grained saliency function to determine the magnitude of the transformed frequency point saliency value compared to the saliency values of adjacent frequency points. If the saliency value of adjacent frequency points is larger, the adjacent frequency point is selected as the new frequency point. Otherwise, no correction is made. Finally, the new frequency point is transformed into a frequency output, and the frequency point transformation calculation is shown in equation (15).

$$f_i^{new} = 2^{\left(B_i^{\max} - 1\right)/120} \cdot 55 \tag{15}$$

In equation (15), f_i^{new} represents the frequency converted from the new frequency point, and $B_i^{\max}$ represents the position of the frequency point with the highest significance. The processor used in the study is i7-12400F with 2.5GHz CPU frequency and 6 cores and 12 threads. The study uses the loss on the validation set no longer decreasing as an early stopping criterion. When the loss on the validation set no longer decreases significantly after multiple consecutive epochs, the model can be considered to have started overfitting the training data.

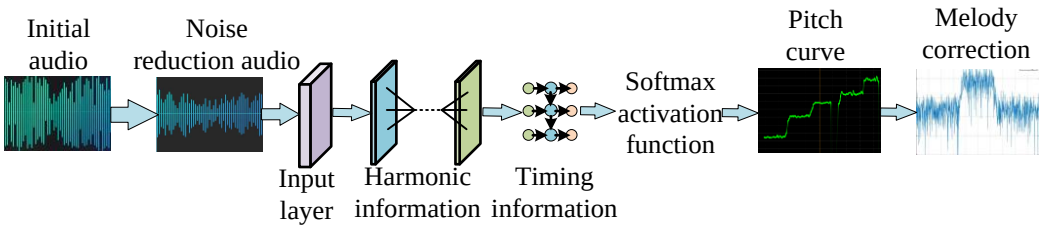


Figure 4: Specific structure of THGDLN

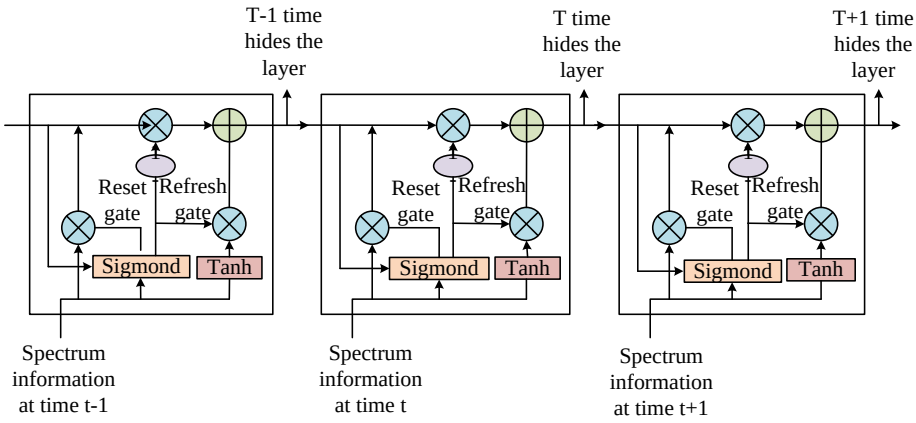


Figure 5: Sequence information extraction process

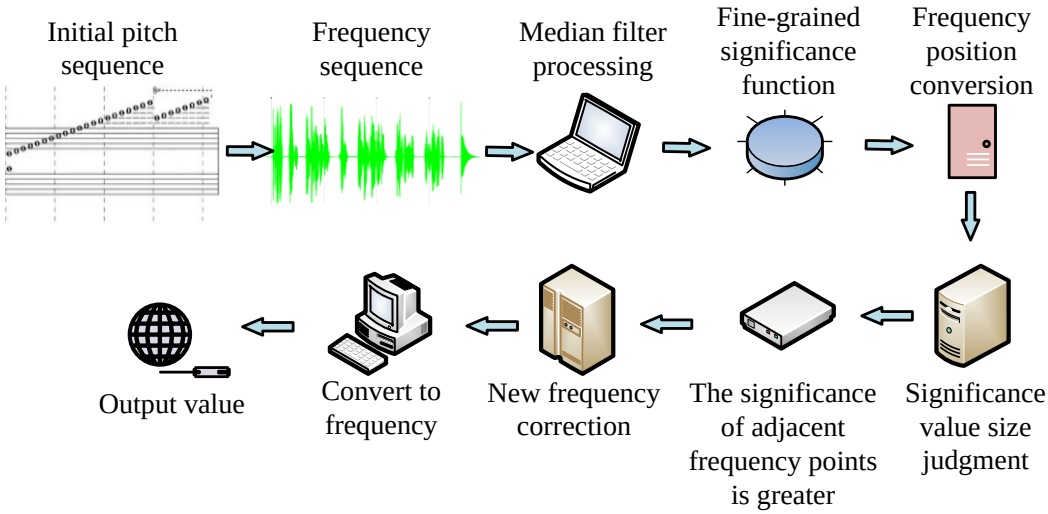


Figure 6: Smooth flow of music melody transfer

3 Results

The study conducted experiments using the MAESTRO dataset, which is a collaborative dataset created by international piano competition organizers. It contains approximately 200 hours of paired audio and MIDI records, covering different performance styles. The data format is universal, easy to process and convert, and MIDI data includes keystroke speed and positions of sustain/slow/bass pedals. The comparative dataset was the MedleyDB dataset. The comparative algorithms used in the study included Multiple Column Deep Neural Networks (MCDNN), Deep Neural Networks (DNNs), and Bayesian Harmonic Model (BHM). In addition to common accuracy, recall, and F1 score, the experimental evaluation indicators also selected Raw Chroma Accuracy (RCA), Raw Pitch Accuracy (RPA), and Overall Accuracy (OA) as performance evaluation

indicators. The impact of frequency point count on the performance of THGDLN algorithm is shown in Figure 7.

In Figure 7 (a), the overall recognition accuracy of the algorithm gradually improved with the increase of frequency points. When the frequency point was 600, the OA value of the THGDLN algorithm tended to converge, reaching a maximum of 89.76%, which was 4.72% and 3.65% higher than the BHM and MCDNN algorithms, respectively. In Figure 7 (b), the RPA of the algorithm gradually improved with the increase of frequency points. The convergence speed was faster before the frequency point reached 600, and slowed down after 600. The maximum value of THGDLN algorithm was 87.14%, which was 6.86% and 4.25% higher than BHM and MCDNN algorithms, respectively. Therefore, choosing a frequency point of 600 only increased accuracy by 1% when the frequency point was between 600 and 800, but it would significantly reduce computation speed and increase operating costs. The impact of frequency search range on the performance of THGDLN algorithm is shown in Figure 8.

In Figure 8 (a), the overall recognition accuracy of the algorithm first increased and then decreased with the increase of frequency search range. When the search range was 0.5, the THGDLN algorithm achieved the maximum OA value of 89.52%, which was 2.58% and 2.35% higher than the BHM and MCDNN algorithms, respectively. BHM achieved the maximum value when the search range was 1.0. In Figure 8 (b), the RPA of the algorithm first increased and then decreased with the increase of the frequency search range. When the search range was 0.5, the THGDLN algorithm achieved the maximum RPA value of 85.94%, which was 2.23% and 1.06% higher than the BHM and MCDNN algorithms, respectively. Therefore, the frequency search range of the algorithm was set to 0.5 semitones. When the number of selected frequency points was 600 and the frequency point search range was 0.5 semitones, the performance of the algorithm before and after melody smoothing was compared as shown in Figure 9.

In Figure 9 (a), the THGDLN algorithm showed significant improvements in RCA, RPA, and OA after melody smoothing. The OA value increased from 79.37% to 82.77% at a runtime of 12 seconds, an increase of 3.4%, the RCA value increased from 80.52% to 87.72%, an increase of 7.2%, and the RPA value increased from 78.05% to 82.14%, an increase of 4.09%. The impact of different network structure layers on model performance in the note starting point and frame level pitch recognition module is denoted in Table 2.

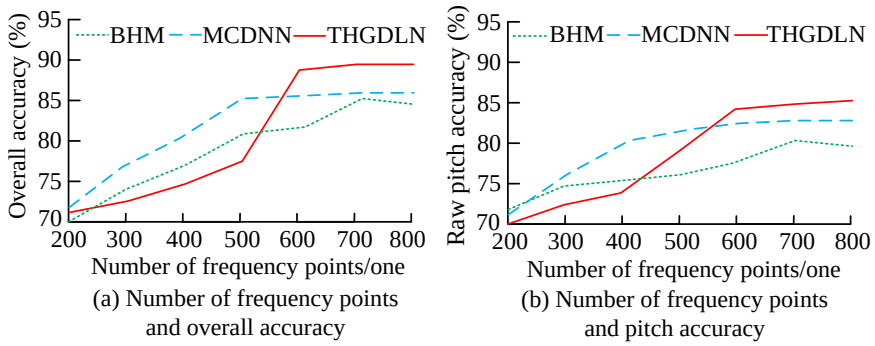


Figure 7: Influence of different frequency points on the performance of THGDLN algorithm

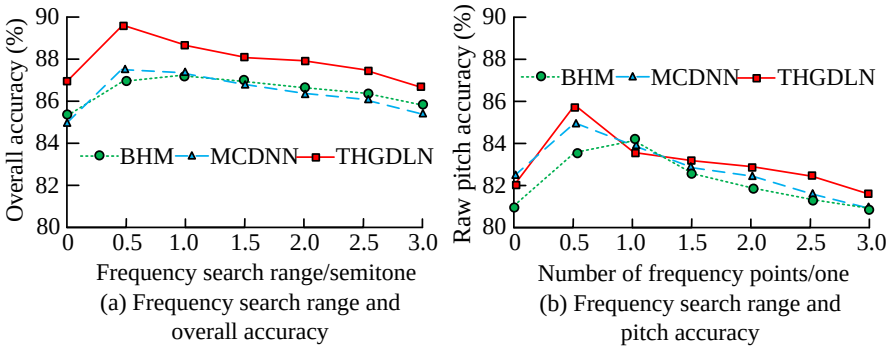


Figure 8: Influence of frequency search range on the performance of THGDLN algorithm

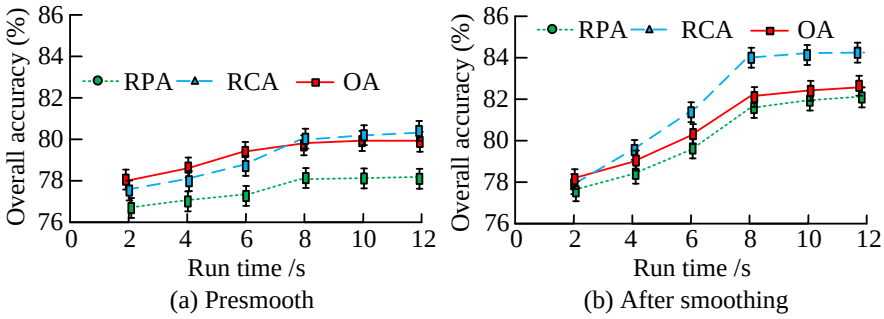


Figure 9: Performance comparison of algorithms before and after smoothing

Table 2: Influence of different network structure layers on model performance

Number of structural layers	Note origin Accuracy rate(%)	Recall rate(%)	F1(%)	End of note Accuracy rate(%)	Recall rate(%)	F1(%)
511	99.15	90.48	94.36	78.92	78.09	79.21
411	98.13	87.62	91.58	78.16	78.45	79.18
311	98.08	86.35	90.58	78.01	77.94	78.24
211	97.87	85.64	89.35	77.84	77.32	78.02
521	99.09	90.52	92.15	78.18	78.19	77.25

Table 3: Performance comparison of different algorithms

Algorithm	MAESTRO Accuracy rate(%)	Recall rate(%)	F1(%)	MedleyDB Accuracy rate(%)	Recall rate(%)	F1(%)
MCDNN	97.04	88.21	92.15	85.42	83.04	84.65
DNNs	85.48	82.19	84.18	78.16	77.36	74.25
BHM	95.14	87.29	89.17	81.09	79.15	79.08
THGDLN	99.15	90.48	94.36	88.91	88.29	89.21

In Table 2, network structure 511 represents 5 shared layers, 1 independent layer, and 1 fully connected layer. When the network structure was 511, the performance of the starting and ending points of the notes was optimal. As the number of shared layers decreased, the performance gradually decreased. When the shared layers were reduced to 2, the accuracy, recall, and F1 score decreased by 1.28%, 4.84%, and 5.11%, respectively. Adding an independent layer resulted in a decrease of 0.06%, -0.04%, and 2.21% in various performance indicators. When the shared layer at the end of the note dropped to 2 layers, the predictive performance decreased by 1.08%, 0.77%, and 1.19%, respectively. Increasing the independent layer by one layer resulted in a decrease of 0.74%, -0.1%, and 1.96%, respectively. The performance comparison of algorithms in different datasets is shown in Table 3.

In Table 3, the THGDLN algorithm had the best performance in all aspects. In the MAESTRO dataset, its accuracy, recall, and F1 score were 2.11%, 2.27%, and 2.21% higher than the second-best algorithm, respectively. In the MedleyDB dataset with higher melody complexity, the accuracy, recall, and F1 score of the THGDLN algorithm were 3.49%, 5.25%, and 4.56% higher than the second-best algorithm, respectively, indicating that the THGDLN algorithm had stronger processing ability for complex audio and higher accuracy after conversion. Ablation experiments were performed on the algorithm, and the performance comparison is shown in Figure 10.

In Fig. 10(a), after removing the window attention, the maximum transcription accuracy of the THGDLN algorithm decreased from 99.15% to 91.08%, which was a decrease of 8.07%, and after removing the note starting point and frame-level pitch recognition module, the

maximum transcription accuracy of the THGDLN algorithm decreased to 82.39%, which was a decrease of 16.76%, and the algorithm's convergence was further back. Results indicated that the note starting point and frame-level pitch recognition module not only improved the transcription performance of the algorithm, but also enhanced the computational efficiency. In Figure 10(b), the F1 maximum of the THGDLN algorithm decreased by 9.53% after removing the window attention, and the F1 maximum of the algorithm decreased by 12.65% after removing the note starting point and frame-level pitch recognition module. The self-attention mechanism takes into account the relationship of each note to all other notes. This mechanism not only focuses on the current note, but also takes into account the information before and after, which helps the model to better understand the contextual information in the music, thus improving the accuracy and smoothness of the transcription. The self-attention mechanism can effectively deal with the harmonic overlap problem by capturing the global dependencies in the music signal and adjusting the degree of attention to different parts.

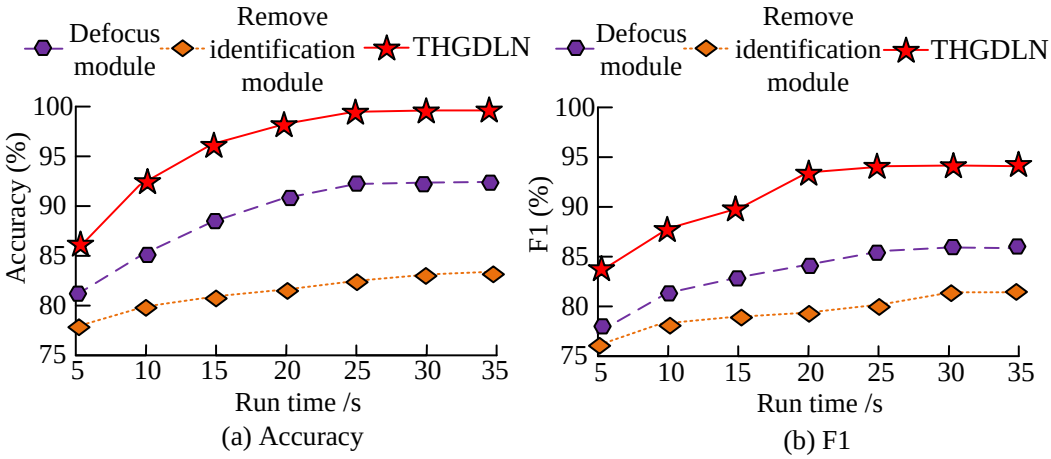


Figure 10: Comparison of experimental performance of ablation algorithms

Table 4: Comparison of the time complexity of the three algorithms

Sample size	Model type	Average processing time (ms)	Standard deviation (ms)	Time complexity O(n2) evaluation
1000	THGDLN	21.4	2.2	Lower
	MCDNN	28.6	3.0	Normal
	DNN	18.2	2.3	Lower
2000	THGDLN	64.5	5.2	Lower
	MCDNN	120.3	7.7	Normal
	DNN	53.6	5.4	Lower
3000	THGDLN	176.2	8.3	Normal
	MCDNN	268.4	11.9	Higher
	DNN	151.8	8.5	Lower

To analyze the computational complexity of THGDLN algorithm, the processing time of THGDLN algorithm under different data amounts was measured through experiments. The computational complexity pairs of THGDLN algorithm are shown in Table 4.

In Table 4, when the sample data volume was 1000, 2000, and 3000, the average processing time of THGDLN algorithm was 21.4ms, 64.5ms, and 176.2ms respectively, which were all lower than MCDNN algorithm and only 3.2ms, 10.9ms, and 24.4ms higher than common DNN algorithm. The results show that THGDLN algorithm has greater computational complexity and higher efficiency than other improved methods.

4 Discussion

The study proposed a piano MT algorithm based on SADLN for existing piano MT algorithms, which has certain errors and other problems. The experimental results showed that the THGDLN algorithm proposed by the study outperformed other algorithms in terms of

accuracy, recall, and F1 score, and the three metrics were 2.11%, 2.27%, and 2.21% higher than the second-best algorithm, respectively. The self-attention mechanism in the THGDLN algorithm took into account the relationship between each note and all other notes, and considered the information of the notes in the preceding and following contexts. This can help the model better understand the contextual information in the music and improve the accuracy and smoothness of the transcription. The window attention mechanism, on the other hand, by applying a local window on the time series data, enabled the model to focus on the information within a specific time period, thus improving the understanding and processing of the music signal. The self-attention mechanism could effectively deal with the harmonic overlap problem by capturing the global dependencies in the music signal and adjusting the degree of attention to different parts. The average processing time of the THGDLN algorithm was only 3.2ms higher than that of the base algorithm DNN when the amount of data was low, and was 7.2ms lower than the average of the existing algorithm MCDNN. As the amount of data increased, the computational complexity of the THGDLN algorithm increased in a significantly lower trend than that of the MCDNN algorithm. The algorithm is slightly less sensitive to the order of recognition. The recognition results between frames are prone to breakpoints. The transcription system that adopts the general method also needs to carry out corresponding processing such as median filtering, and the corresponding details need to be improved.

5 Conclusion

A piano MT algorithm based on SADLN was proposed to address the problems of insufficient accuracy and low transcription accuracy of existing piano MT algorithms. The experiment findings indicated that as the number of frequency points increased, the OA of the algorithm gradually improved. At a frequency point of 600, it tended to converge, with OA values 4.72% and 3.65% higher than BHM and MCDNN algorithms, respectively. The RPA values of THGDLN algorithm were 6.86% and 4.25% higher than BHM and MCDNN algorithms, respectively. Therefore, although there was a slight improvement in algorithm performance after setting the frequency points to 600600, it would reduce computational depth and increase operating costs. The OA and RPA values of the THGDLN algorithm first increased and then decreased with the increase of the frequency search range. When the search range was 0.5, the max value was obtained. The OA values were 2.58% and 2.35% higher than those of the BHM and MCDNN algorithms, respectively, and the RPA values were 2.23% and 1.06% higher than those of the BHM and MCDNN algorithms, respectively. The THGDLN algorithm improved OA, RCA, and RPA by 3.4%, 7.2%, and 4.09%, respectively, after smoothing. As the number of shared layers in the network structure decreased, the performance of the algorithm gradually declined. When the shared layer was reduced to 2 layers, the accuracy, recall, and F1 score decreased by 1.28%, 4.84%, and 5.11%, respectively. In the MAESTRO dataset, the accuracy, recall, and F1 score of the THGDLN algorithm were 2.11%, 2.27%, and 2.21% higher than the second best algorithm, respectively. After removing window attention, the transcription accuracy of the THGDLN algorithm decreased by 8.07%. After removing the note start and frame level pitch recognition modules, the transcription accuracy decreased by 16.76%. There are still some issues with this study, such as the lack of discussion on the algorithm's ability to transcribe synthesized tracks in music creation. In the future, a dataset composed of synthesized music can be added for testing to improve the algorithm's generalization performance.

References

[1] X. Li, Y. Zhang, P. Tiwari, D. Song, B. Hu, M. Yang, Z. Zhao, N. Kumar, and P. Marttinen, "EEG based emotion recognition: A tutorial and review," ACM Computing Surveys, vol. 55, no. 4, pp. 1-57, 2022. https://doi.org/10.1145/3524499

[2] A. Battcock, and M. Schutz, "Emotion and expertise: How listeners with formal music training use cues to perceive emotion," Psychological Research, vol. 86, no. 1, pp. 66-86, 2022. https://doi.org/10.1007/s00426-020-01467-1

[3] J. W. Li, S. Barma, P. U. Mak, F. Chen, C. Li, M. T. Li, M. I. Vai, and S. H. Pun, "Single-channel selection for EEG-based emotion recognition using brain rhythm sequencing," IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 6, pp. 2493-2503, 2022. https://doi.org/10.1109/JBHI.2022.3148109

[4] M. J. Lucia-Mulas, P. Revuelta-Sanz, B. Ruiz-Mezcua, and I. Gonzalez-Carrasco, "Automatic music emotion classification model for movie soundtrack subtitling based on neuroscientific premises," Applied Intelligence, vol. 53, no 22, pp. 27096-27109, 2023. https://doi.org/10.1007/s10489-023-04967-w

[5] V. Chaturvedi, A. B. Kaur, V. Varshney, A. Garg, G. S. Chhabra, and M. Kumar, "Music mood and human emotion recognition based on physiological signals: A systematic review," Multimedia Systems, vol. 28, no. 1, pp. 21-44, 2022. https://doi.org/10.1007/s00530-021-00786-6

[6] J. S. Gómez-Cañón, N. Gutiérrez-Páez, L. Porcaro, A. Porter, E. Cano, P. Herrera-Boyer, A. Gkiokas, P. Santos, D. Hernández-Leo, C. Karreman, and E. Gómez, "TROMPA-MER: An open dataset for personalized Music Emotion Recognition," Journal

of Intelligent Information Systems, vol. 60, no. 2, pp. 549-570, 2023. https://doi.org/10.1007/s10844-022-00746-0

[7] N. He, and S. Ferguson, "Music emotion recognition based on segment-level two-stage learning," International Journal of Multimedia Information Retrieval, vol. 11, no. 3, pp. 383-394, 2022. https://doi.org/10.1007/s13735-022-00230-z

[8] M. A. Qureshi, M. Asif, M. F. Hassan, G. Mustafa, M. K. Ehsan, A. Ali, and U. Sajid, "A novel auto-annotation technique for aspect level sentiment analysis," Computers, Materials and Continua, vol. 70, no. 3, pp. 4987-5004, 2022. https://doi.org/10.32604/cmc.2022.020544

[9] R. Orjesek, R. Jarina, and M. Chmulik, "End-to-end music emotion variation detection using iteratively reconstructed deep features," Multimedia Tools and Applications, vol. 81, no. 4, 5017-5031, 2022. https://doi.org/10.1007/s11042-021-11584-7

[10] G. Yin, S. Sun, D. Yu, D. Li, and K. Zhang, "A multimodal framework for large-scale emotion recognition by fusing music and electrodermal activity signal," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 18, no. 3, pp. 1-23, 2022. https://doi.org/10.1145/3490686

[11] S. Venkata Lakshmi, K. Sujatha, and J. Janet, "A hybrid discriminant fuzzy DNN with enhanced modularity bat algorithm for speech recognition," Journal of Intelligent & Fuzzy Systems, vol. 44, no. 3, pp. 4079-4091, 2023. https://doi.org/10.3233/JIFS-212945

[12] A. Ríos-Vila, D. Rizo, J. M. Iñesta, and J. Calvo-Zaragoza, "End-to-end optical music recognition for pianoform sheet music," International Journal of Document Analysis and Recognition, vol. 26, no. 4, pp. 347-362, 2023. https://doi.org/10.1007/s10032-023-00432-z

[13] R. Shashidhar, S. Patilkulkarni, and S. B. Puneeth, "Combining audio and visual speech recognition using LSTM and deep convolutional neural network," International Journal of Information Technology, vol. 14, no. 7, pp. 3425-3436, 2022. https://doi.org/10.1007/s41870-022-00907-y

[14] S. Choudhuri, S. Adeniye, and A. Sen, "Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation," Artificial Intelligence and Applications, vol. 1, no. 1, pp. 43-51, 2023. https://doi.org/10.47852/bonviewAIA2202524

[15] A. Aristidou, A. Yiannakidis, K. Aberman, D. Cohen-Or, A. Shamir, and Y. Chrysanthou, "Rhythm is a dancer: Music-driven motion synthesis with global structure," IEEE Transactions on Visualization and Computer Graphics, vol. 29, no. 8, pp. 3519-34, 2022. https://doi.org/10.1109/TVCG.2022.3163676

[16] X. Chen, J. Zhou, D. Li, J. Liu, and Y. Dai, T. Zhou, " Enjoyment of Chinese and mathematics and school performance in Chinese children and adolescents," Child Development, vol. 94, no. 1, pp. 126-141, 2023. https://doi.org/10.1111/cdev.13843

[17] F. De Arriba-Pérez, S. García-Méndez, F. Leal, B. Malheiro, and J. C. Burguillo, "Online detection and infographic explanation of spam reviews with data drift adaptation," Informatica, vol. 35, no. 3, pp. 483-507, 2024. https://doi.org/10.15388/24-INFOR562

[18] E. Deruty, M. Grachten, S. Lattner, J. Nistal, and C. Aouameur, "On the development and practice of ai technology for contemporary popular music production," Transactions of the International Society for Music Information Retrieval, vol. 5, no. 1, pp. 35-49, 2022. https://doi.org/10.5334/tismir.100

[19] G. Keerti, A. N. Vaishnavi, P. Mukherjee, A. S. Vidya, G. S. Sreenithya, and D. Nayab, "Attentional networks for music generation," Multimedia Tools and Applications, vol. 81, no. 4, pp. 5179-5189, 2022. https://doi.org/10.1007/s11042-021-11881-1

[20] A. K. Herget, and J. Albrecht, "Soundtrack for reality? How to use music effectively in non-fictional media formats," Psychology of Music, vol. 50, no. 2, pp. 508-29, 2022. https://doi.org/10.1177/03057356219990

Multimodal Sentiment Perception for Intelligent Music Generation Using Improved Transformer Architectures

Yingshi Jiang*, Zuodong Sun
School of Music, Tonghua Normal University, Tonghua 134001, China
Email: 18704352221@163.com, sunzuodong126@163.com
*Corresponding author

Keywords: multimodal sentiment, music generation model, emotion perception, deep learning, sequence analysis

Received: August 1, 2024

To further examine the interrelationship between music, emotion, and scene, as well as to furnish novel technical assistance for music creation, the study devised a multimodal sentiment analysis model for auditory and visual features with deep learning. Based on this model, a new music content generation model was proposed, which improved upon the traditional Transformer architecture. The validation of the research-designed MSA dataset, as well as the three publicly available datasets IEMOCAP, CMU__MOSI, and CMU-MOSEI, confirmed the accuracy of the sentiment analysis of this multimodal sentiment analysis architecture. The mean absolute error, the root means square error, and the mean absolute percentage error were found to be 0.149, 0.166, and 0.140, respectively, and the goodness-of-fit R-squared reached 0.961. The model performed well on Precision-Recall curve, receiver operating characteristic curve. The sentiment recognition accuracy was up to 0.98, and the recognition efficiency was high. Additionally, the pitch variation of the music generated by the improved Transformer structure was the smallest compared to the Classical Piano MIDI dataset, taking a value of 1.29%. Moreover, the melodic variation was the smallest compared to the Nottingham dataset, taking a value of 0.66%. The generated music performed better in terms of smoothness, coherence, and percentage of completeness. Using this model for music generation, the highest values of hit rate and normalized discounted cumulative gain could be 93.984% and 91.566%. The mean inverse rank could be up to 0.89. This study deepens the mechanism of music emotion generation, captures the emotion and context of music more accurately, and promotes the development of the fields of emotion computing and sentiment recognition.

Povzetek: Raziskava uvaja izboljšan jezikovni pretvornik za multimodalno analizo sentimenta ter generiranje glasbe, kar izboljša čustveno zaznavanje in kakovost glasbe.

1 Introduction

Emotion is an important driver of human behavior and thinking, and accurately understanding and describing emotion can lead to a better understanding of the impact of emotion on decision-making and behavior. The need for emotion research in human-computer interaction, mental health, education, and business applications is steadily growing as society and intelligent technology evolve. Multimodal sentiment analysis (MSA) is an intelligent study of recognizing and understanding human emotions through inputs such as speech, text, and images. By elucidating the nuances of human emotion, MSA can enhance the capacity of computer systems to comprehend and respond to human emotions, thereby optimizing the quality of human-computer interaction [1-2]. The expression of emotion has multi-dimensional characteristics, involving language, voice, facial expression, and other forms. MSA can synthesize multiple sensory inputs to understand and analyze emotional expressions more comprehensively and accurately, enriching the application scenarios and possibilities of emotion perception. Meanwhile, MSA involves the crossover of multiple disciplinary fields such as linguistics, psychology, computer vision, etc., which promotes the cooperation and integration between different disciplines [3-4]. In recent years, "artificial intelligence + music" has become a hot research topic. Auto-generated music technology can create novel and unique music works and bring a fresh listening experience [5-6]. However, the current auto-generated music content is single, lacks user stickiness, and is difficult to generate music content with emotional style. Additionally, MSA methods still face the dilemma of shortage of dataset resources and lack of multimodal data characterization capability [7-8].

To generate music with emotional and melodic characteristics, the research centers around MSA and music automatic generation (MAG) techniques. Firstly, a multi-label sentiment analysis dataset for MSA is

constructed, then the MSA framework is constructed based on deep learning (DL), sequence analysis, and feature extraction models, and finally the model is designed based on the improved Transformer. The study unfolds the design of the model based on MSA, which improves the theoretical research level of MSA. It fills the gap of lack of emotion analysis in and expands the human understanding of music generation. The study is expected to provide new technical support for music creation, making music creation more emotionally expressive and situationally perceptive.

The study is divided into a total of four parts. The first part completes the review of the current state of research related to MSA and MAG. The second part designed the MSA framework and constructed the model based on MSA. The performance test of the MSA framework and model is examined in the third section. The study's key findings and future directions are outlined in the fourth part.

2 Related works

As artificial intelligence technology advances, DL is being utilized extensively in several domains such as speech recognition, natural language processing, image recognition, and more. MSA has gradually become a popular research direction. MSA belongs to the popular trend of social research. In order to cope with the interference of mixed emotions, Jiang et al. designed a context-self-attentive fuzzy temporal convolutional network for MSA. The network contained fuzzy emotion affiliation functions, which could obtain the dependency relationship between the key information of itself and the information of the external context within the target discourse. The dataset validated the effectiveness of the method [9]. Existing Multimodal sentiment recognition methods are prone to bring redundant information and low recognition efficiency. Fu et al. designed an innovative asymmetric transformer with cross-modal blocks for complementary learning of the local structure of sequences with different modalities. The method's usefulness and superiority were confirmed by the experimental results [10]. To utilize various physiological signals for sentiment recognition, Chen et al. proposed a multilevel multimodal dynamic fusion network by considering cross-modal interactions. The method utilized this network to independently extracted potential and essential interactions between different modal features and split the multistage fusion network using correlations. Experimental results indicated that the network could utilize finer-grained single-, bimodal, and tri-modal interactions and outperforms single-stage multimodal sentiment recognition methods [11].

Kumar et al. classified the emotions contained in speech and images into discrete classes, designed a hybrid fusion based multimodal sentiment recognition system. Experimental results indicated that the system achieved 83.29% accuracy in sentiment recognition [12]. Currently, there is a lack of labeled datasets for speech sentiment recognition due to the time-consuming nature of recognizing emotion categories. Therefore, Yi et al. tried to apply the wav2vec2.0 model to speech feature extraction for speech sentiment recognition task. The technique created a multimodal two-branch transformer network for emotion categorization. Two databases, IEMOCAP and CASIA, verified the recognition accuracy of the method [13]. Due to the large feature dimension, there is multimodal sentiment recognition in multimodal sentiment recognition. Using a bidirectional gated recurrent cell recognition network based on bidirectional gated recurrent units, Tang et al. developed a novel multimodal sentiment and presented the process of multimodal self-attention. According to experimental data, the study's proposed technique might successfully raise the network's multimodal sentiment recognition accuracy [14]. Multimodal sentiment recognition involves various different application areas. Dixit and Satapathy first used three publicly available benchmark datasets, ISEAR, RAVDESS and FER-2013, to train independent models in three modes of emotion: text, audio and image. The study then stacked deep neural network (NN), recurrent NN, and 2D convolutional NN based on the idea of integration for effective emotion prediction. The method's prediction accuracy was found to be 86.6% with an F1 value of 0.84, as per the experimental findings [15].

Automated music generation using artificial intelligence offers musicians of all levels an innovative way to enhance the creative process, but also creates a way for ordinary music lovers to "make music" for their personal music needs. Automated music creation has received a lot of attention, and the use of AI technology in creative music creation is still an area that can be explored. Shukla and Banka changed the traditional way of generating music using predefined music parameters. The study created a brand-new genetic algorithm-based approach for creating music that features enhanced crossover and mutation operators. The method was flexible to change the musical parameters can be generated by inputting short segments of the melody to generate a musical melody, and comparative analysis verified the validity of the model [16]. Automatic generation of music relies on contextual representation capabilities. Wu et al. first delineated music by setting up a fragment range localization module. Validated on a public MIDI dataset, the method outperformed other comparative models in short- and long-term quantitative music generation metrics [17]. DL techniques have application advantages in creative music generation. Li S et al. proposed a pre-trained model for multitasking music generation to learn melodic and rhythmic representations by taking advantage of pre-training to overcome remote dependency limitations in natural language processing. Experimental results indicated a significant improvement in the performance of the method under the HITS@k evaluation metric [18].

The relevant work is summarised in Table 1. The research on MSA and technology at home and abroad has made technical breakthroughs in the main performance aspects. However, the robustness of MSA under the interference of data noise and unlabeled dataset still needs to be improved. Meanwhile, Meanwhile, most of the automatic music generation models use text sequence methods, ignoring the correlation between chords and melodies, and its difficult to generate music segments that contain emotional information. Therefore, in this regard, MSA-based music generation is investigated.

Table 1: Summary of related work

Literature	Author	Methodology	Data set	Results	Insufficient research
[9]	Jiang et al.	Fuzzy temporal convolutional network based on contextual self-attention	/	Model outperforms state-of-the-art models on tested datasets.	More complex model design
[10]	Fu et al.	Efficient neural network to learn modality-fused representations with CB-Transformer and Innovative asymmetric transformer with cross-modal blocks	IEMOCAP, CMU-MOSI and CMU-MOSEI	Superiority and efficiency in word-aligned and unaligned experiments	Higher model complexity
[11]	Chen et al.	A multi-stage multimodal dynamical fusion network	Multimodal benchmark DEAP	The method outperforms related single-stage multimodal emotion recognition methods	Insufficient attention to original features
[12]	Kumar et al.	A multimodal emotion recognition system based on hybrid fusion	IIT-R SIER dataset	The accuracy of emotion recognition reaches 83.29%	Only image and voice features are fused
[13]	Yi Y, Tian et al.	Self-supervised multimodal dual-branch transformer network	IEMOCAP and CASIA datasets	The method outperforms the accuracy of existing state-of-the-art methods	Higher model complexity
[14]	Tang et al.	Multimodal emotion recognition based on multi-head self-attention mechanism and bi-directional gated	/	Attentional mechanisms can effectively improve the accuracy of multimodal emotion recognition	Unvalidated robustness on unlabelled datasets

		recurrent units			
[15]	Dixit and Satapathy	Deep neural networks, recurrent neural networks, and 2D convolutional neural network stacking	ISEAR, RAVDESS and FER-2013 datasets for training text, CMU-MOSEI benchmark multimodal dataset for testing	The accuracy obtained on the CMU-MOSEI dataset is 86.60% with an F1 score of 0.84	Model training and stacking are too cumbersome
[16]	Shukla and Banka	Music generation based on improved genetic algorithms	/	Algorithm performance is better and the model generates music with better efficacy	Neglected emotional information in music data
[17]	Wu et al.	Hierarchical Transformer model	Open MIDI datasets	The method outperforms other comparative models in quantitative metrics of short- and long-term music generation	Neglects emotional information in music data
[18]	Li and Sung	Multi-task music generation based on pre-trained models	/	0.09-13.10% and 0.02-7.37% improvement in the performance of the generation task under the HITS@k evaluation metric, respectively.	Ignores sentiment information in music data

3 Key technology of automatic music generation modeling

With the goal of automatic music generation, the research task is divided into the construction of MSA dataset, the construction of MSA model and the design of music generation model based on sentiment analysis.

3.1 Construction of a multimodal multilabel sentiment analysis dataset

MSA dataset is the basic task of MSA work, and a high-quality MSA dataset should be characterized by diversity, annotation accuracy, balance, and multimodal consistency to ensure model optimization and performance improvement. However, there is a clear imbalance in the contribution of different modal information. That is, auditory and visual representation feature engineering cannot adequately extract and utilize modal affective information such as images and audio. This results in the textual modality usually contributing more than the auditory or visual modalities. This also leads to missing information from other modalities, incomplete emotion prediction, and weakened inter-modal complementarity, which exacerbates the modal imbalance phenomenon, and thus results in limited application scenarios for MSA models [19-20]. Meanwhile, the construction of MSA dataset faces serious challenges of data resource quality [21]. In order to train and optimize the MSA model, the study firstly launched the construction of MSA dataset. The study helps the model learn the correlations between different modalities and the diversity of emotional cues more deeply by constructing MSA datasets that contain rich, diverse, and accurately labeled data.

The study uses social media platforms, online video resources, and professional performance databases as data sources. During the acquisition process, the meta-video modality is required to have standard Chinese Mandarin language, clear speaker images, and accurate text transcription. The video player Plot Player is used to complete the video cropping. The cropped video clips should cover rich video scenes, such as various interviews, video blogs, and variety shows. During the capture process, the face scenes with different angles and lighting, and slight noise are preserved as much as possible. To analyze as much as possible ambiguity, irony, innuendo and other moods, the cropped video clips should possibly focus on non-verbal behaviors. The final study obtains a total of 10,364 unsupervised resources and 4,631 supervised instances. Table 2 displays the statistical data and details of the modality extraction of the MSA dataset. In the course of data labeling, all samples are assigned acoustic, visual, textual, and multimodal affective values. A total of ten annotators participated in labeling the dataset, deleting the highest and lowest scores, or taking the average of the labeled scores as the final labeling results, and mapping the results of labeling to seven different affective levels. Additionally, the study adopts a modal isolation strategy in the annotation process to ensure that different modal information does not interfere with each other.

3.2 Construction of a model for multimodal sentiment analysis

Traditional MSA mostly adopts a multimodal fusion framework, which is easy to ignore the original features [22]. In this regard, the study proposes an auditory and visual MSA analysis framework (AV-MSA). The structural composition is shown in Figure 1.

Table 2: Statistical information and modal extraction details of msa dataset

Type	Unsupervised resources	Supervision examples	Modality	Feature dimension	Extraction tool
Video	163	175	Hearing Visual Text	920*25 230*180 50*750	OpenSMILE TalkNet Pre training model
Fragment	4631	10364	/	/	/
Average fragment duration	3.26	4.64	/	/	/
Average number of clause words	16.1	19.4	/	/	/
Maximum fragment duration ratio	0.5/19	0.3/29	/	/	/
The ratio of the maximum word count in a clause	2/46	2/89	/	/	/
Standard deviation of sentence word count	8.2	10.1	/	/	/

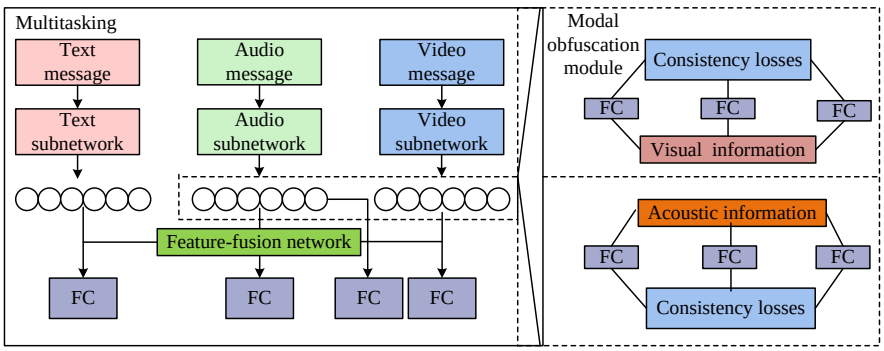


Figure 1: Structure diagram of auditory and visual msa analysis framework

In Figure 1, AV-MSA contains four components: text, audio, video, and feature fusion. Text, audio, and video are individually supervised and then trained with the feature fusion network. The text modal representation learning network extracts text features through Bert pre-training model with feature dimension of 50. The audio modal and video modal representation learning networks complete feature extraction and optimization through long short-term memory networks (LSTM) and multilayer perceptron (MLP). The basic structure of the resulting audio and video modal representation learning network is shown in Figure 2.

In Figure 2, the LSTM responsible for the hidden layer feature extraction contains the forgetting gate (FG) f_t, the input gate (IG) i_t, the output gate (OG) O_t, and the memory cells. The computation process of the internal state S_t of the LSTM with the hidden layer y_t transitions is shown in Equation (1) [23].

$$\begin{cases} S_t = f_t \times S_{t-1} + i_t \times S_t \\ y_t = \mathrm{O}_t \tanh\left(S_t\right) \end{cases} \quad (1)$$

In Equation (1), S_t denotes the intermediate state of the current internal state. t denotes the time series.

The expression of the FG f_t is shown in Equation (2).

$$f_t = \sigma\left(W_f\left[y_{t-1}, x_t\right] + b_f\right) \quad (2)$$

In Equation (2), W_f and b_f denote the weight and bias of the FG, respectively. σ denotes the activation function. The calculation process of IG i_t and OG O_t is shown in Equation (3).

$$\begin{cases} i_t = \sigma\left(W_i\left[y_{t-1}, x_t\right] + b_i\right) \\ O_t = \sigma\left(W_o\left[y_{t-1}, x_t\right] + b_o\right) \end{cases} \quad (3)$$

In Equation (3), W_i and b_i denote the weights and bias of the IGs, respectively. W_o, b_o are the weights and bias of the OGs respectively. After obtaining the hidden layer features of the audio subnetwork, the MLP performs feature optimization. MLP belongs to a feed-forward neural network (FNN) based DL model. It consists of an input layer, a hidden layer, and an output layer, and contains two processes, forward propagation (FP) and back propagation. The FP process of MLP is shown in Equation (4).

$$z_j^{l+1} = \sum_{i=1}^{n_l} w_{ij}^{\ l} a_i^l + b_j^{\ l+1} \quad (4)$$

In Equation (1), n_l is the neurons in layer l, i and $j \in n$. a is the computed output of neurons. z is the output of the network layer. w is the connection weights. b is the bias parameter. To minimize the error between the predicted and true outputs, the MLP back propagation method computes the gradient by applying the chain rule to compute the derivatives of the loss function with respect to the parameters and changes the weights and biases [24–25]. The residual δ_i^l of the neuron is calculated in Equation (5).

$$\delta_i^l = \frac{\partial J\left(w,b\right)}{\partial z_i^l} \quad (5)$$

In Equation (5), $J\left(w,b\right)$ denotes the loss function. The parameter update process is shown in Equation (6).

$$\begin{cases} w_{ij}^l = w_{ij}^l - \lambda \dfrac{\partial J\left(w,b\right)}{\partial w_{ij}^l} \\ b_i^{\ l} = b_i^{\ l} - \lambda \dfrac{\partial J\left(w,b\right)}{\partial b_i^{\ l}} \end{cases} \quad (6)$$

In Equation (6), λ denotes the learning rate. The acquired modal sequences are then fed into the unimodal encoder $S(\cdot)$. The encoding process is described in Equation (7).

$$F_k = S_k\left(I_k\right) \quad (7)$$

In Equation (7), F_k denotes unimodal intermediate features. I_k denotes modal original sequence feature. k denotes the modality. Finally, the text intermediate feature F_t, acoustic intermediate feature F_a, and visual intermediate feature BBB are connected. Equation (8) depicts the calculating procedure.

$$F_m = Concat\left(\left[F_t; F_a; F_v\right]\right) \tag{8}$$

In Equation (8), F_m denotes multimodal features. Fully connect feed-forward network (FCFN) is utilized as unimodal and multimodal sentiment classifier. The sentiment prediction result y_k' is calculated in Equation (9).

$$y_k' = FCFN\left(F_k\right) = FFN\left(FFN\left(FFN\left(BN\left(F_k\right)\right)\right)\right) \tag{9}$$

In Equation (9), FFN denotes FNN. BN denotes batch data. The two supervised cases are mixed for training, and the mixed unimodal sentiment prediction results of y_k'' are calculated in Equation (10).

$$y_k'' = Clf\left(F_k\right) = FFN\left(FFN\left(FFN\left(BN\left(F_k\right)\right)\right)\right) \tag{10}$$

In Equation (10), Clf denotes the classifier. The modal obfuscation module of AV-MSA is mainly responsible for enhancing and optimizing the feature representation of unimodal subnetworks using the mixup method. The working mechanism of the modal confusion module is shown in Figure 3.

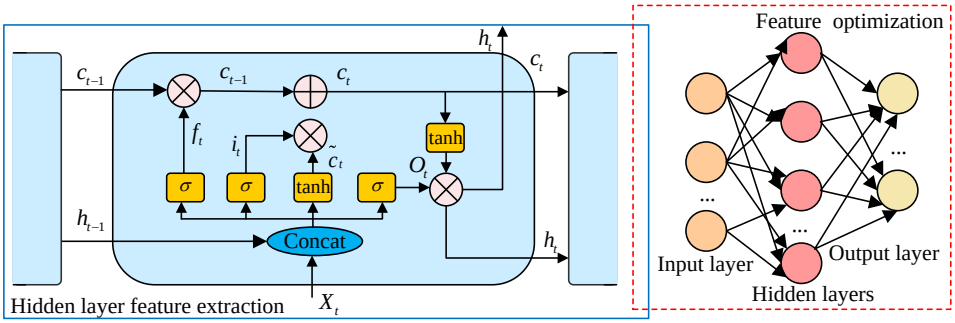


Figure 2: Composition of feature extraction results for audio modal

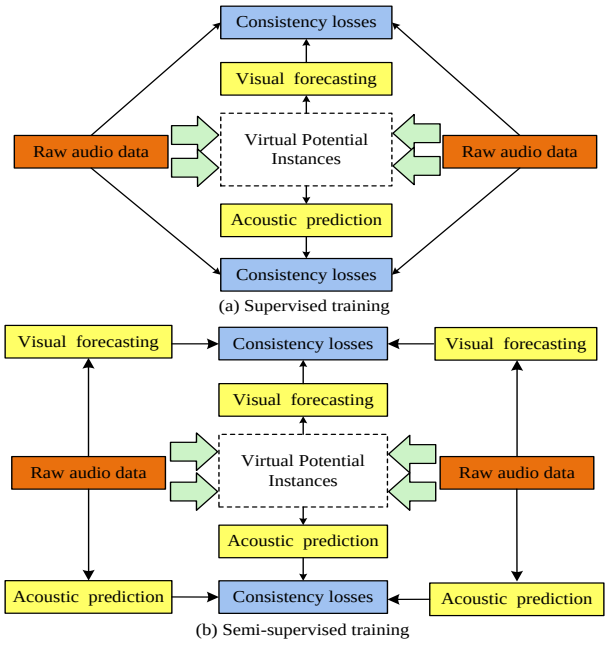


Figure 3: Working mechanism of modal obfuscation module

In Figure 3, the obfuscation of supervised and unsupervised data representations further improves the fine-grained sentiment prediction performance. Supervised training directly obfuscates the raw data labels and supervises the prediction results. Unsupervised training is assisted by supervised training. The original data labels, original data prediction results are also utilized to generate new labels. The process of mixing instances of modal obfuscation module is shown in Equation (11).

$$X_i^{''} = \beta \cdot X_i + (1-\beta) \cdot X_i^{'} \quad (11)$$

In Equation (11), $X_i^{'}$ and X_i denote different instances before mixing. $X_i^{''}$ denotes the instances after mixing. β denotes the random variable. Meanwhile, the same way of mixing the prediction results is used and the modal confusion module is shown in Equation (12).

$$\left(X_1^{''}, y_1^{''}\right),...,\left(X_n^{''}, y_n^{''}\right) = Mixup\left\{\left(X_1, y_1\right),...,\left(X_n, y_n\right)\right\} \quad (12)$$

3.3 Modeling for automatic music content generation

Automatic music content generation based on AV-MSA framework is to complete the generation of music content based on user sentiment analysis, and the study analyzes music in MIDI format as an example. The unfolding, repetition, and change of music segments naturally constitute time series data, which is regarded as a sequence model [24]. Based on the similarity between music time series data and natural language sequences, the study adopts the Transformer model as the technical framework for automatic MIDI music generation. Figure 4 schematically depicts the Transformer model's construction.

In Figure 4, the core idea of Transformer is the attention mechanism with the ability to process input sequences in parallel. The input sequences constitute the coding part through a multi-layered self-attention mechanism, which is processed by the decoder to generate the output sequences. Transformer does not need to consider the sequence order problem, and it can capture the long-distance dependencies better [25-26]. As a result, the automatic music generation framework constructed by the research is shown in Figure 5.

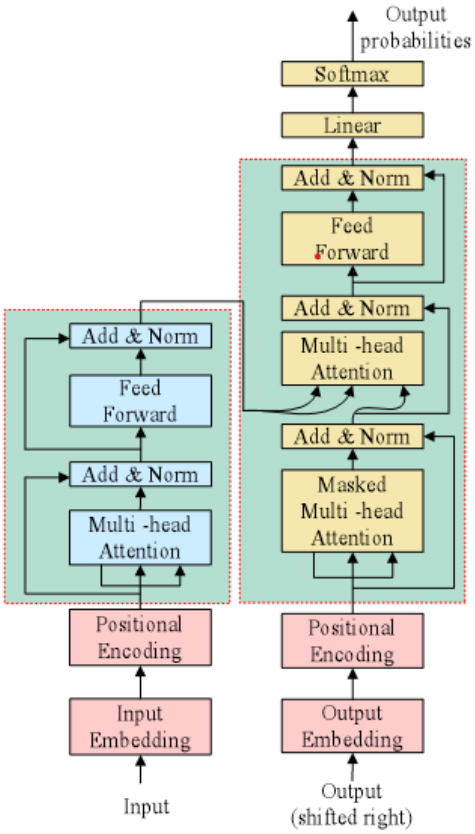


Figure 4: Structure diagram of transformer feature extractor

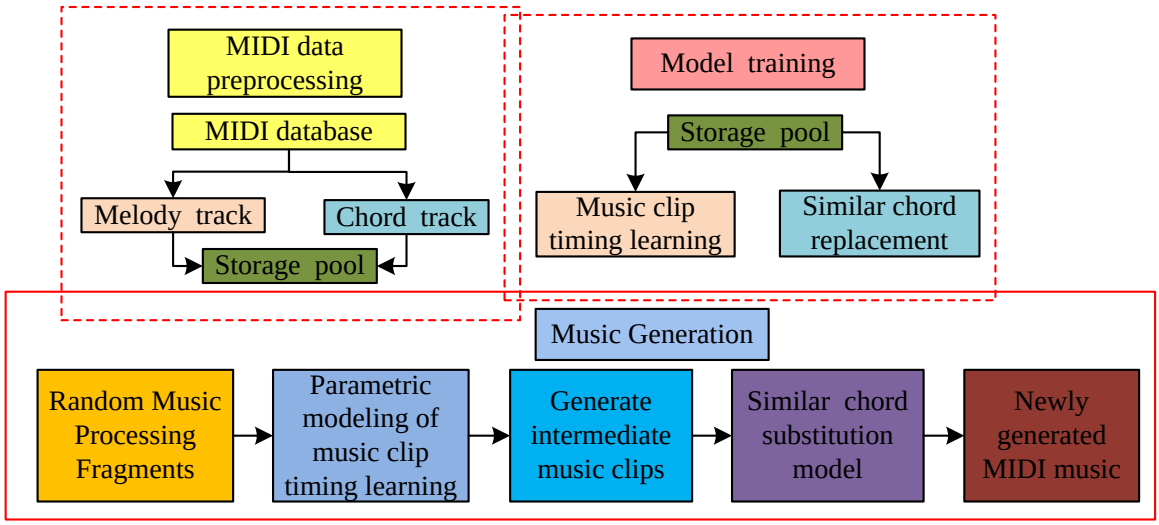


Figure 5: Schematic diagram of the framework for automatic music generation

In Figure 5, the music generation framework consists of three parts: preprocessing of MIDI music data, model training and music generation. The melody and chords of MIDI music are correlated. The melody is the dominant part of the music and the chords are the collection of notes that support and enrich the melody in the context of the music [27]. The study constructs a new storage structure. The melody and chord tracks are extracted separately and then sequentially spliced to ensure that the music generation model can learn the corresponding information units of both at the same time. In the note extraction and storage process, the study defines a "chord library dictionary", which helps the model to follow the harmonic rules of the music during the generation process and reduces the occurrence of dissonance. The joint modeling of melody and chord extracts the two sequences from the MIDI data and then digitally maps them separately. This allows the model to learn the intrinsic connection between the two sequences simultaneously, thereby facilitating the generation of more harmonious and expressive musical compositions. This allows the model to simultaneously learn the intrinsic connection between the two sequences, helping to generate more harmonic and expressive musical compositions. The model training process is shown in Figure 6. In Figure 6, the melodic sequences and harmonic sequences obtained from the preprocessing of MIDI music data are combined and mapped according to the actual sequences. Moreover, the harmonic and notes are considered as elements for loop training. The chord substitution network learns the similarities and transformation patterns between chords during training to generate more musically logical melodies. The group training approach allows the model to learn both local and global musical structure features. In the similar chord substitution model, a set of tones (octaves) is first divided into twelve chromatic intervals to complete the chord correspondence according to the numerical ladder. Then the basic chord-C chord is replaced using the genitive chord. Finally, the Chord2vec model is used to learn the vector representation of the chords, and similar chords are obtained by model training. Chord2vec model is developed on the basis of Word2Vec model, which is a mathematical representation of musical structure, harmonic properties and chord relationships by mapping chords in music into vector space. The process of chord vectorization for the Chord2vec model treats chords as a series of note combinations, and the vector position expression matrix PE is shown in Equation (13).

$$\begin{cases} PE_{(pos,2i)} = \sin\left|\dfrac{pos}{10000^{\frac{2i}{d}}}\right| \\ PE_{(pos,2i+1)} = \cos\left|\dfrac{pos}{10000^{\frac{2i}{d}}}\right| \end{cases} \tag{13}$$

In Equation (13), d denotes the note vector dimension. pos is the position of the note vector in the music bar. $2i$ and $2i+1$ refer to even and odd numbers, respectively. Harmonically similar chords will be mapped to similar positions. The similarity between chords can then be measured using distances in vector space. Additionally, the study determines the replacement chords based on the knowledge of music theory, and the chord replacement is accomplished by combining the calculated similar chords. In the music generation and optimization section, MIDI music clips are randomly input and trained with a sequence learning model to obtain the predicted music clips. Then the chord substitution model is trained and the music clip is deposited again to get the new music.

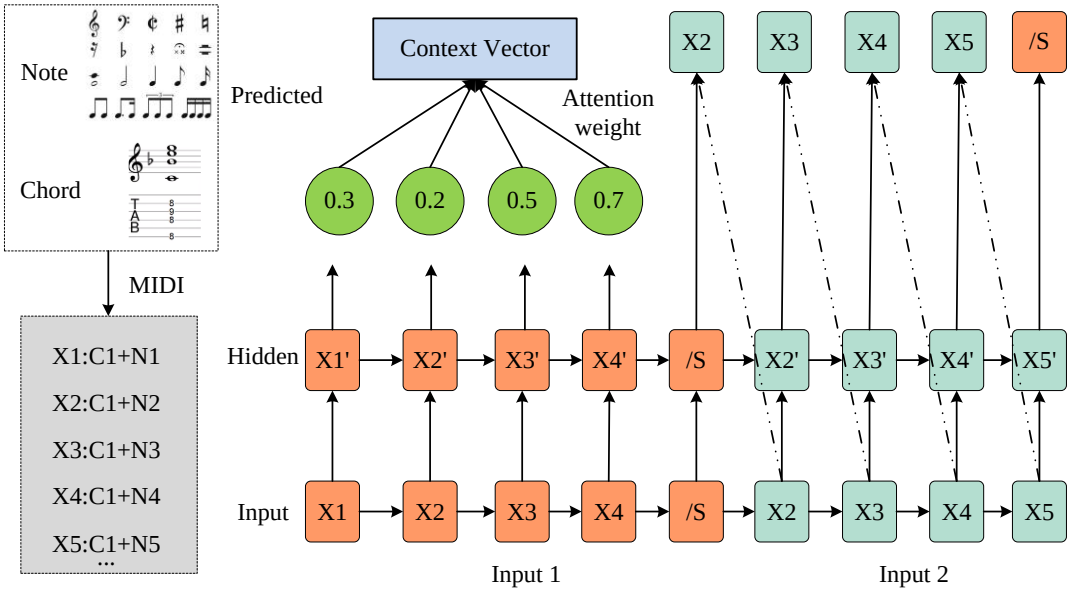


Figure 6: Schematic diagram of the network structure for music sequence generation in the transformer model

4 Performance analysis of MSA and automatic music content generation models

To validate the effectiveness of the MSA-based model for automatic music content generation, the study centers on MSA and automatic music generation, and the results are analyzed and discussed.

4.1 Performance testing of multimodal sentiment analysis models

The hardware environment operating system used for the experiment is Windows 10, the processor is Intel(R) Xeon(R) CPUE5-2603 v3, the GPU is Nvidia GTX 1080Ti, and the memory is 64G. The DL framework in the software environment is based on the implementation of Nvidia GTX 1080Ti, and the programming language is Python 3.7.1. The experiment is conducted using the research-constructed MSA dataset, the interactive emotional dyadic motion capture (IEMOCAP), carnegie mellon university multimodal opinion sentiment and emotion intensity (CMU__MOSI) and its extended version, the CMU-MOSEI dataset. A 12-hour period of audiovisual data, including speech, video, text transcription, and facial motion capture, can be found in the action, multimodal, and multicamera database known as the IEMOCAP dataset. Many annotators have annotated it with dimensional labels like potency, activation, and dominance as well as categorical categories like anger, happiness, sadness, and neutrality. CMU__MOSI contains movie clips, audio, text, and emotion tags data from YouTube covering a wide range of emotional expressions and subjective comments. CMU-MOSEI includes video, audio, and text information covering more than 16,000 subjective comments and sentiment labels. Depending on the requirements of the experiment, the dataset is split into an 8:2 ratio training set and test set.

This experiment selects the optimized ensemble multi-scale residual attention network (EMRA-Net) [28], the MSA method based on hierarchical adaptive feature fusion network (MSA-HAFF) [29], and the MSA analysis method based on Dempster-Shafter (D-S) theory [30]. Mean absolute percentage error (MAPE), root mean square error (RMSE), mean absolute error (MAE), and R-squared metrics of different MSA methods are compared. Table 3 displays the outcomes of the experiment.

Table 3: Comparison of analytical performance of different MSA methods

Model	Index	MSA-study	IEMOCAP	CMU__MOSI	CMU-MOSEI
EMRA-Net	MAE	0.398	0.449	0.438	0.429
	RMSE	0.404	0.447	0.454	0.431
	R-squared	0.743	0.716	0.715	0.736
	MAPE	0.434	0.439	0.466	0.450
MSA-HAFF	MAE	0.449	0.461	0.475	0.458

	RMSE	0.452	0.509	0.513	0.502
	R-squared	0.704	0.669	0.668	0.673
	MAPE	0.388	0.494	0.465	0.420
	MAE	0.391	0.562	0.423	0.407
D-S-MSA	RMSE	0.388	0.429	0.398	0.366
	R-squared	0.781	0.735	0.722	0.744
	MAPE	0.343	0.439	0.417	0.400
	MAE	0.149	0.219	0.216	0.183
AV-MSA	RMSE	0.166	0.310	0.270	0.212
	R-squared	0.961	0.851	0.873	0.902
	MAPE	0.140	0.296	0.267	0.228

In Table 3, the research-designed AV-MSA framework takes the smallest values on three different error metrics. Among them, it performs best on the research-designed MSA-study dataset. the MAE, RMSE, and MAPE metrics are 0.149, 0.166, and 0.140, respectively. Secondly, the CMU-MOSE dataset takes better values of error than the CMU_MOSI dataset. Meanwhile, the pattern of taking values between different datasets is reflected in EMRA-Net, MSA-HAFF, and D-S-MSA models. In summary, the MSA-study dataset can better train and optimize the MSA model, which helps the model learn the correlation between different modalities and the diversity of emotional cues in a deeper way, and reduces the error of sentiment analysis. Additionally, EMRA-Net, MSA-HAFF, and D-S-MSA are the error takeoffs of the three MSA models, which all fluctuate in the 0.3-0.5 range. The surface model is slightly less predictive of sentiment. The R-squared take of the AV-MSA framework is up to 0.961, which is significantly better than the other models in explaining emotion. Figure 7 displays the receiver operating characteristic curve (ROC) and precision-recall (P-R) curve results for various models.

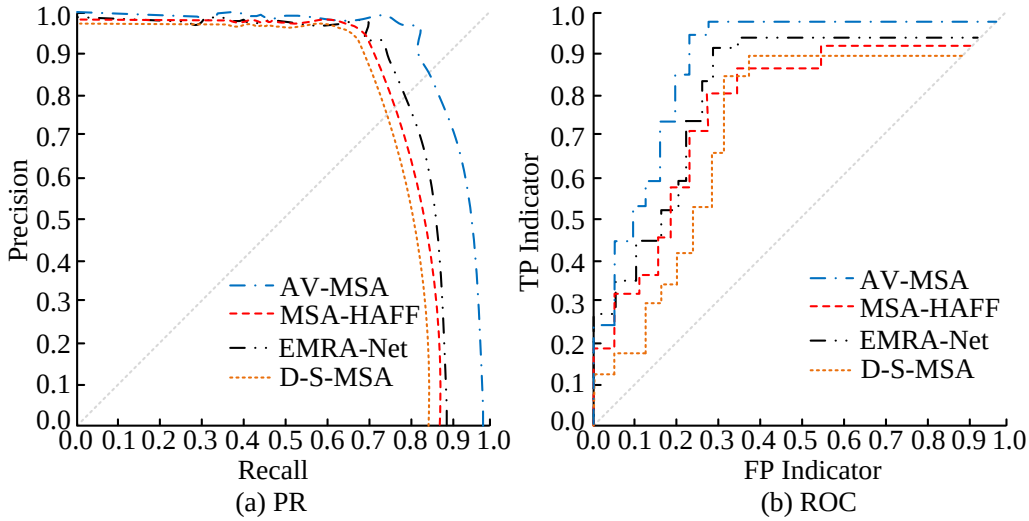


Figure 7: Comparison of P-R curves and ROC curves for different MSA methods

In Figure 7(a), the PR curve of the AV-MSA framework designed by the study performs the best and is located at the top right-hand side of the axes at a precision of 0.9. The recall of the AV-MSA framework is 0.98. The recalls of EMRA-Net, MSA-HAFF, and D-S-MSA are 0.88, 0.86, and 0.82, respectively. The AV-MSA framework in the same experimental environment has a relative advantage in the precision rate and recall reconciliation average, and the accuracy of the model is superior. In Figure 7(b), the ROC curve of the research-designed AV-MSA framework has the best performance. Area under the curve (AUC) takes the values of 0.94, 0.90, 0.84, and 0.80, respectively. The larger the AUC value the better the comprehensive performance of the model, which displays that the AV-MSA framework has obvious superiority. The results of the 7-category accuracy and 2-category accuracy analysis of the model are shown in Figure 8.

In Figure 8, the AV-MSA framework designed by the study has significant superiority in both 7-category accuracy and 2-category accuracy. The 2-category accuracy reaches 0.98 in Figure 8(a). The 7-category

accuracy reaches 0.96 in Figure 8(b). In contrast, the 2-category accuracy of EMRA-Net, MSA-HAFF, and D-S-MSA are 0.69, 0.78, and 0.62, respectively. The 7-category accuracy are 0.72, 0.80, and 0.69, respectively. The loss functions of the different models curves are compared with the analytical efficiency as shown in Figure 9.

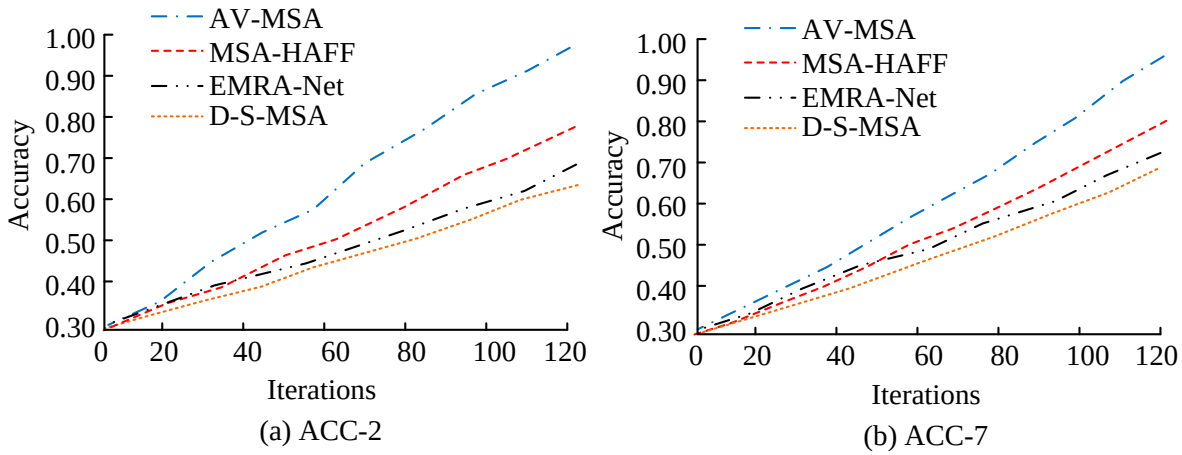


Figure 8: Comparison between 7-category accuracy and 2-category accuracy

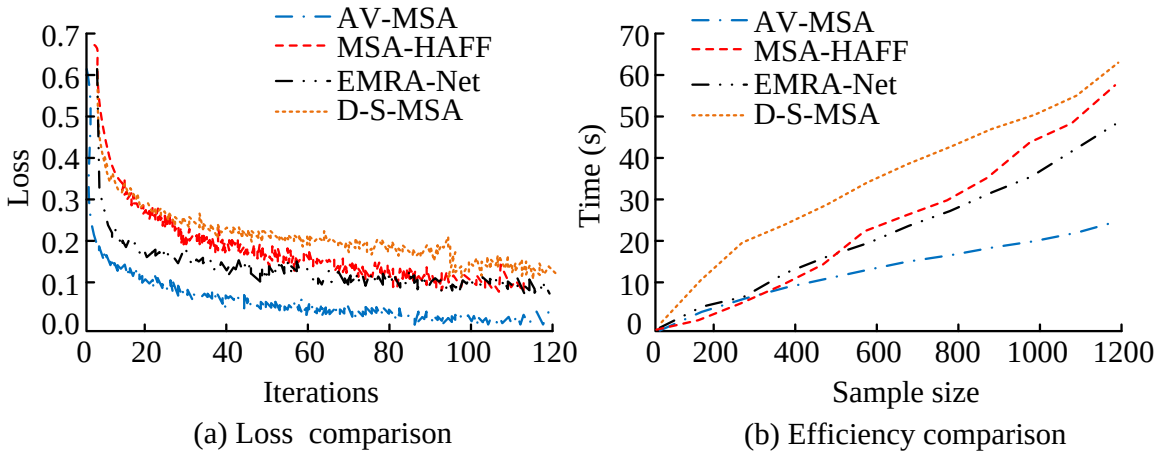
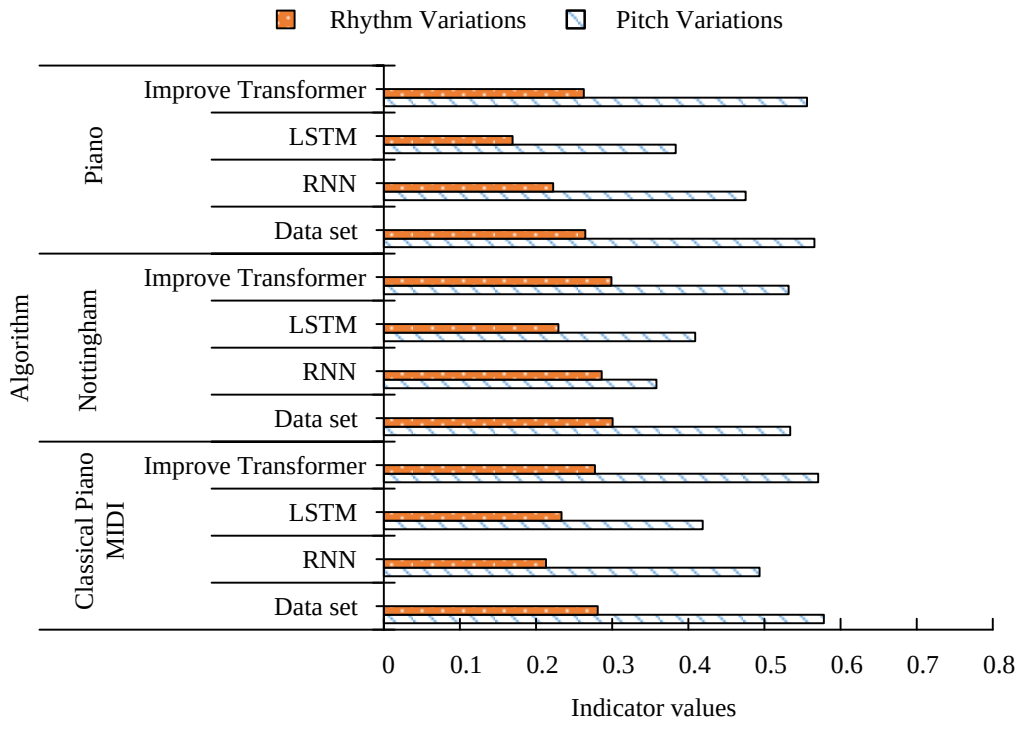


Figure 9: Comparison of loss function and analysis efficiency

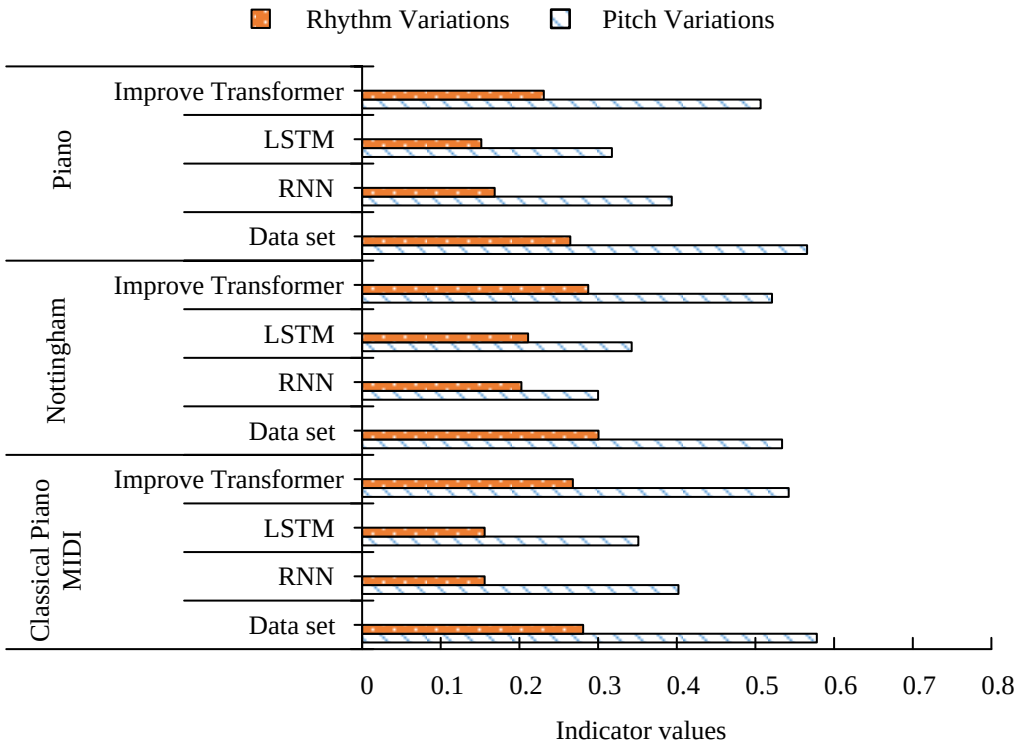
In Figure 9(a), the loss function curve of the AV-MSA framework designed for the study converges the fastest, with the smallest convergence value of 0.03. The loss function curves of EMRA-Net, MSA-HAFF, and D-S-MSA, converge slower, with the convergence value roughly around 0.20. In Figure 9(b), the efficiency of the AV-MSA framework for analyzing the multimodal sentiment is remarkable. When the number of analyzed samples is 1200, the analysis time is 24.2s. Compared with other models, its maximum time reduction value can reach 39.4s. Comprehensively analyzing, the MSA analysis framework designed by the research achieves better results in terms of accuracy, efficiency and other indicators, which provides a solid technological foundation for music content generation.

4.2 Performance testing of automatic music content generation models

The classical piano music dataset Classical Piano MIDI, the folk song ballad dataset Nottingham, and the MIDI piano music dataset Piano in classical and jazz styles are selected for experimental analysis. According to the experimental needs, it is divided into training set and test set according to the ratio of 7:3. Commonly used baseline sequence analysis models are selected for comparison, including LSTM, recursive neural network (RNN). The pitch variations (PV) and rhythm variations (RV) of the generated music for different models are compared as shown in Figure 10.



(a) No noise



(b) Noise interference

Figure 10: Comparison of pitch and melody changes in different models

PV and RV are related metrics for note analysis. PV is the variation in the ebb and flow of notes across the range, i.e., the variation in note elevations and intervals, obtained by comparing the pitches to the total notes. RV is the change in rhythm, i.e., how many different note durations occur. In Figure 10 (a), the PV and RV values of the improved Transformer structure designed by the study are closest to the metrics taken from the dataset. The PV values are taken with a difference of 1.29%, 8.01%, and 1.78% compared to the Classical Piano MIDI,

Nottingham, and Piano datasets, respectively. The RV values are taken with a magnitude of difference of 1.29%, 0.66%, and 0.86% when compared to the Classical Piano MIDI, Nottingham, and Piano datasets, respectively. The magnitude of difference between the PV and RV values of the other models can be up to 32.92% and 36.14%. Improved Transformer structure is better for feature learning on music dataset. Additionally, to test the performance of the models under noise conditions, the study artificially added noise segments in different styles of music datasets, and the results are shown in Figure 10(b). It can be observed that background noise impedes the parsing of the audio signal by RNN and LSTM models, thereby reducing the quality of the generated music and increasing the discrepancy between the PV and RV values. However, the improved Transformer structure still maintains more consistent PV and RV values, and the robustness of the model is superior.

Comparison of smoothness, coherence, and percentage of fragmented bars of music generated by different models is shown in Figure 11.

In Figure 11(a), the music generated by the improved Transformer structure has a better smoothness, with median fetch levels above 0.70. The smoothness of the music generated by the other two methods is below the 0.60 value level. Smoothness can be used to describe the smoothness and harmony of the music. The higher the smoothness level, the better the handling of connections and transitions between notes. In Figure 11(b), the music generated by the improved Transformer structure has a better coherence and is at least 0.2 fetch level higher compared to the other models. The model handles the degree of articulation and completeness of the music better. The percentage of fragmented bars can refer to the presence or absence of fragmented bars in the music. The experiment uses the integrity percentage to refer to the fragmented bar percentage. In Figure 11(c), the integrity percentage of the improved Transformer structure is significantly higher than the other two models. The research-designed AV-MSA framework with the improved Transformer structure is applied to the multimodal sentiment intelligent music generation analysis. The normalized discounted cumulative gain (NDCG) and hit rate (HR) are chosen as the evaluation indexes of the model's music generation merit. Table 4 displays the findings of the experiment.

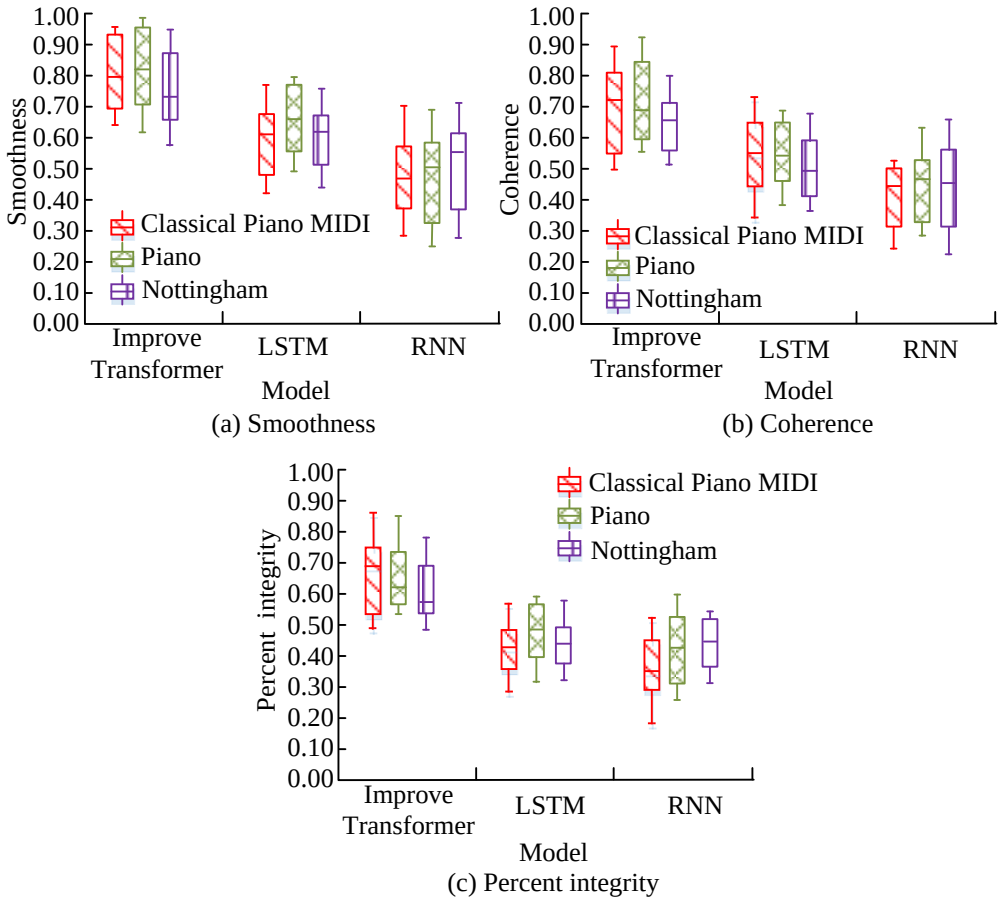


Figure 11: Comparison of music quality generated by different models

Table 4: Comparison of NDCG and HR indicators for different models

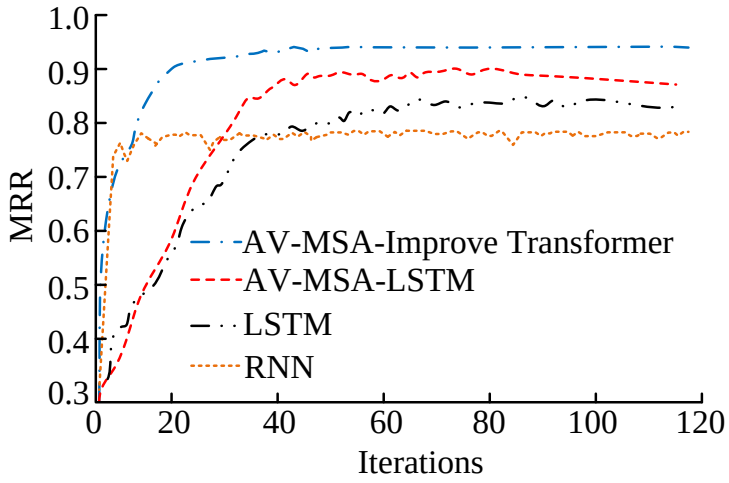
Evaluating indicator	AV-MSA-Improve Transformer	AV-MSA-LSTM	LSTM	RNN
HR#3	44.659%	37.218%	32.734%	30.038%
HR#5	56.888%	57.231%	45.675%	48.490%
HR#5	64.768%	63.547%	59.546%	53.384%
HR#10	87.680%	70.655%	66.449%	68.804%
HR#15	93.984%	83.900%	79.848%	79.490%
Evaluating indicator	AV-MSA-Improve Transformer	AV-MSA-LSTM	LSTM	RNN
NDCG #3	46.314%	37.815%	29.228%	30.806%
NDCG #5	67.420%	42.030%	39.531%	38.345%
NDCG #8	77.749%	56.857%	51.647%	42.556%
NDCG #10	83.335%	70.856%	59.129%	50.095%
NDCG #15	91.566%	83.304%	63.322%	62.644%

In Table 4, with the increase of k-rank, the values of the indicators taken by different models showed an upward trend. The larger the value of k, the larger the NDCG and HR take values. The maximum HR value of AV-MSA-Improve Transformer model is 93.984% and the maximum NDCG value is 91.566%, which is significantly higher than the other three methods. Additionally, comparing the AV-MSA-LSTM model with the LSTM model, the HR and NDGG of AV-MSA-LSTM under different k-ranks are improved to some extent. It can be concluded that music generation based on MSA is more likely to get the music that meets the demand and hits the user's needs more easily. The experimental results of mean reciprocal rank (MRR) with Hits@10 are shown in Figure 12. In Figure 12(a), the AV-MSA-Improve Transformer model has the highest MRR curve of 0.91. The AV-MSA-LSTM model reaches 0.89, which is higher than the other two music generation models. In Figure 12(b), the results of the different models for the values of Hits@10 show the same pattern. The AV-MSA-LSTM model has the best performance with Hits@10 taking values up to 0.97. Additionally, to verify the effects of feature fusion, LSTM-MLP feature extraction module and MIDI music data preprocessing and model training on the music generation effect in the AV-MSA framework, the model ablation experiments are first initiated. The results of the ablation experiments are shown in Table 5. In Table 5, the AV-MSA-Improve Transformer model takes the highest level of values on the three metrics of HR#10, NDCG #10, and Hits@10, and the model has the best performance. The traditional Transformer music generation architecture takes values around 0.75 for all metrics. After introducing the feature fusion and LSTM-MLP feature extraction modules to build the AV-MSA framework, the fetch level increases to around 0.80. The feature fusion module helps to alleviate the noise or missing problem that may exist in single-modal data, and promotes inter-modal complementarity. LSTM-MLP can effectively capture the emotion change trend and contextual information of each modality, and improves the accuracy of emotion recognition. Then the AV-MSA framework improves the emotional and melodic characteristics of the generated music, and the performance of HR, NDCG & Hits indicators is improved. Additionally, the music generated by the Transformer architecture improved by the MIDI data pre-processing and model training module is further improved, and the improvement strategy is set accordingly.

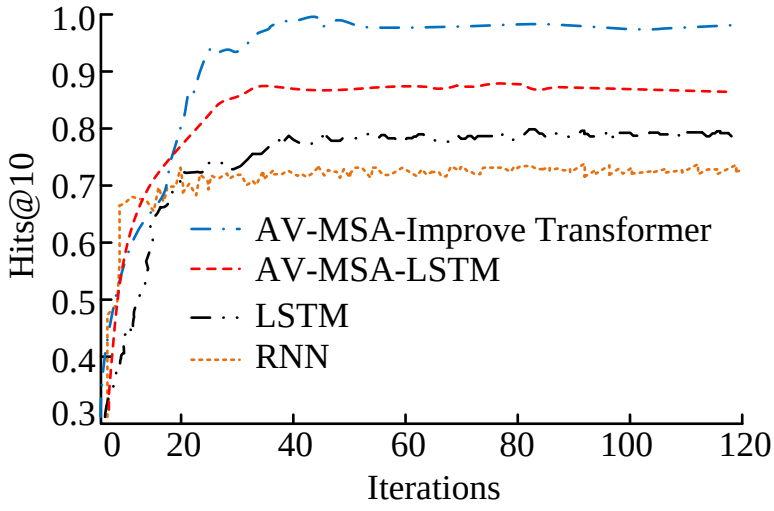
Finally, the qualitative assessment of the model-generated music is carried out by combining the feedback and subjective judgment of the users. Moreover, a total of 30 people is included in the experimental assessment to evaluate the quality of 100 pieces of music. The experimental results are shown in Figure 13.

Table 5: Results of ablation experiments of the model

Transformer	Feature fusion	LSTM-MLP	Data preprocessing	Model training	HR#10	NDCG #10	Hits@10
√	/	/	/	/	0.749	0.731	0.753
√	√	√	/	/	0.805	0.807	0.804
√	√	√	√	/	0.824	0.812	0.820
√	√	√	√	√	0.877	0.833	0.970



(a) MRR comparison



(b) Hits @10 comparison

Figure 12: MRR and Hits@10 value comparison

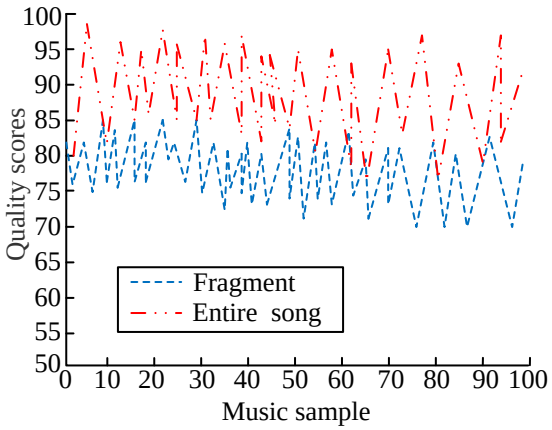


Figure 13: Results of qualitative assessment of music generation quality

In Fig. 13, among the 100 pieces of music tested, the average quality score of music segments is roughly in the range of 75-85. The overall quality score is in the range of 80-100. The results of the qualitative assessment can confirm the excellent performance of the music generation model designed by the study.

5 Discussion

In recent years, the rapid development of the field of "artificial intelligence + music" has opened up new possibilities for music creation, which can not only speed up the process of music creation, but also explore music styles and forms that are unimaginable to humans. However, by analyzing the literature [16], [17], [18], it can be concluded that the existing automatic music generation technology still faces challenges such as single content and difficulty in generating music content with deep emotional style. Therefore, the research combined MSA with automatic music generation techniques to allow music generation frameworks to understand and express more complex and delicate emotions. However, the robust performance of MSA models under noisy data and unlabelled datasets still had more shortcomings. The inter-modal fusion problem and the complexity of feature extraction were still challenges for MSA models. As in literature [11] Chen S et al. designed a multi-stage multimodal dynamic fusion network, and literature [12] Kumar P et al. designed a multimodal emotion recognition system with hybrid fusion. The MSA method was obviously insufficient to focus on the original features. Even though the accuracy rate of emotion recognition reached 83.29%, it only fused the image and speech features, and the multimodal information was not sufficiently considered. Therefore, to address the technical dilemma faced by MSA and automatic music generation technology, the study drew on the design concepts of Yi Y et al. in literature [13]. Firstly, it solved the technical difficulties posed by unlabelled datasets, and constructed a high-quality, accurately annotated multi-label sentiment analysis dataset, which could help the music generation model to correlate multiple sentiment labels. This study then utilized DL models, sequence analysis models, and feature extraction to capture the intrinsic connections between different modalities. The study extracted key information that characterized emotional and melodic features and constructed the MSA framework. Finally, the study introduced the MIDI music data preprocessing technique and model training module to improve the Transformer model to better generate music content with emotional and melodic features.

The results of the study showed that the music generation model incorporating the AV-MSA framework could take the highest HR and NDGG values up to 93.984% and 91.566% on Classical Piano MIDI, Nottingham and Piano datasets. Hits@10 could take values up to 0.97. Compared to the pre-trained model based on literature [18] for multitask music generation, the HITS@k metrics improved significantly. The efficient feature extraction and sequence analysis capabilities improved the performance of the MSA framework. The incorporation of the AV-MSA framework into the automated music generation methodology, in conjunction with the cumulative impact of the preprocessing technique and the model training module, enhances the capabilities of the enhanced Transformer model in generating music with emotional and melodic attributes.

However, the dataset used for the experimental analysis is biased toward popular and classical music, ignoring other niche or regional music styles. The music style bias may cause the model to perform poorly in generating or analyzing non-mainstream music. Furthermore, there are notable discrepancies in the cultural milieu of disparate regions, which give rise to markedly disparate musical expressions and emotional connotations. A single cultural context in the dataset may limit the application of the model in cross-cultural contexts. In future research, it would be beneficial to update the model training dataset in a timely manner to incorporate the latest musical works and popular trends, thus enhancing the model's generalizability.

6 Conclusion

The objective of this study is threefold: Firstly, to gain a comprehensive understanding of how humans perceive emotions and situations. Secondly, to enhance the capacity of intelligent systems to comprehend emotions and situations. Thirdly, to facilitate the advancement of intelligent music content generation and the field of affective computing. To this end, a model is constructed and an analysis conducted of MSA and music intelligent generation. Based on these findings, a novel AV-MSA framework with an automatic music generation model was proposed. The experimental results indicated that the AV-MSA framework took the smallest values on three different error metrics. MAE, RMSE, and MAPE metrics were 0.149, 0.166, and 0.140, respectively. R-squared took the maximum value of 0.961. The performance of the model constructed by the study achieved good performance with the dataset. The recall was 0.98 for AV-MSA framework precision of 0.9 and AUC took the value of 0.94. The 2-category accuracy and 7-category accuracy were 0.98 and 0.96, respectively. The PV and RV values of the improved Transformer structure are closest to the metrics taken from the dataset. The minimum difference was 1.29% and 0.66% respectively. The smoothness, coherence, and completeness of the music performed better. Jointly utilized with the AV-MSA framework, the maximum NDGG value of the generated music was 91.566%, the maximum HR value was 93.984%, the MRR value was 0.89, and the Hits@10 value could be taken up to 0.97. The objective of the study was to generate personalized music content based on multimodal situational sentiment perception. This was done with the aim of matching specific situations and emotional states, thereby enhancing the user's music experience. However, the MSA dataset, as designed in the study, still requires a greater investment of time and human resources. Additionally, the integration of the music generation model is inadequate, the model training is divided into two distinct modules, and the potential for further performance enhancement remains to be investigated.

References

[1] H. Lian, C. Lu, S. Li, Y. Zhao, C. Tang, and Y. Zong, "A survey of deep learning-based multimodal sentiment recognition: Speech, text, and face," Entropy, vol. 25, no. 10, pp. 1440-1462, 2023. https://doi.org/10.1007/s10518-023-01779-8

[2] S. Holiday, J. L. Hayes, H. Park, Y. Lyu, and Y. Zhou, "A multimodal emotion perspective on social media influencer marketing: The effectiveness of influencer emotions, network size, and branding on consumer brand engagement using facial expression and linguistic analysis," Journal of Interactive Marketing, vol. 58, no. 4, pp. 414-439, 2023. https://doi.org/10.1177/109499682311711

[3] B. Pan, K. Hirota, Z. Jia, L. Zhao, X. Jin, and Y. Dai, "Multimodal sentiment recognition based on feature selection and extreme learning machine in video clips," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 3, pp. 1903-1917, 2023. https://doi.org/10.1007/s12652-021-03407-2

[4] J. M. Garcia-Garcia, M. D. Lozano, V. M. Penichet, and E. L. C. Law, "Building a three-level multimodal sentiment recognition framework," Multimedia Tools and Applications, vol. 82, no. 1, pp. 239-269, 2023. https://doi.org/10.1007/s11042-022-13254-8

[5] S. Dong, X. Fan, and X. Ma, "Multichannel multimodal emotion analysis of cross-modal feedback interactions based on knowledge graph," Neural Processing Letters, vol. 56, no. 3, pp. 1-17, 2023. https://doi.org/10.1007/s11063-024-11641-w

[6] J. Liu, Z. Wang, W. Nie, J. Zeng, B. Zhou, J. Deng, H. Li, Q. Xu, X. Xu, and H. Liu, "Multimodal sentiment recognition for children with autism spectrum disorder in social interaction," International Journal of Human-Computer Interaction, vol. 40, no. 8, pp. 1921-1930, 2024. 10.1080/10447318.2023.2232194

[7] Z. Yin, F. Reuben, S. Stepney, and T. Collins, "Deep learning's shallow gains: A comparative evaluation of algorithms for automatic music generation," Machine Learning, vol. 112, no. 5, pp. 1785-1822, 2023. https://doi.org/10.1007/s10994-023-06309-w

[8] P. Georges, and A. Seckin, "Music information visualization and classical composers' discovery: An application of network graphs, multidimensional scaling, and support vector machines," Scientometrics, vol. 127, no. 5, pp. 2277-2311, 2022. https://doi.org/10.1007/s11192-022-04331-8

[9] D. Jiang, H. Liu, R. Wei, and G. Tu, "CSAT-FTCN: A fuzzy-oriented model with contextual self-attention network for multimodal sentiment recognition," Cognitive Computation, vol. 15, no. 3, pp. 1082-1091, 2023. https://doi.org/10.1007/s12559-023-10119-6

[10] Z. Fu, F. Liu, Q. Xu, X. Fu, and J. Qi, "LMR-CBT: Learning modality-fused representations with CB-transformer for multimodal sentiment recognition from unaligned multimodal sequences," Frontiers of Computer Science, vol. 18, no. 4, pp. 184314-184332, 2023. https://doi.org/10.1007/s11704-023-2444-y

[11] S. Chen, J. Tang, L. Zhu, and W. Kong, "A multi-stage dynamical fusion network for

multimodal sentiment recognition," Cognitive Neurodynamics, vol. 17, no. 3, pp. 671-680, 2023. https://doi.org/10.1007/s11571-022-09851-w

[12] P. Kumar, S. Malik, and B. Raman, "Interpretable multimodal sentiment recognition using hybrid fusion of speech and image data," Multimedia Tools and Applications, vol. 83, no. 10, pp. 28373-28394, 2023. https://doi.org/10.1007/s11042-023-16443-1

[13] Y. Yi, Y. Tian, C. He, Y. Fan, X. Hu, and Y. Xu, "DBT: Multimodal sentiment recognition based on dual-branch transformer," The Journal of Supercomputing, vol. 79, no. 8, pp. 8611-8633, 2023. https://doi.org/10.1007/s11042-023-16443-1

[14] G. Tang, Y. Xie, K. Li, R. Liang, and L. Zhao, "Multimodal sentiment recognition from facial expression and speech based on feature fusion," Multimedia Tools and Applications, vol. 82, no. 11, pp. 16359-16373, 2023. https://doi.org/10.1007/s11042-022-14185-0

[15] C. Dixit, and S. M. Satapathy, "A customizable framework for multimodal sentiment recognition using ensemble of deep neural network models," Multimedia Systems, vol. 29, no. 6, pp. 3151-3168, 2023. https://doi.org/10.1007/s00530-023-01188-6

[16] S. Shukla, and H. Banka, "Monophonic music composition using genetic algorithm and Bresenham's line algorithm," Multimedia Tools and Applications, vol. 81, no. 18, pp. 26483-26503, 2022. https://doi.org/10.1007/s11042-022-12185-8

[17] G. Wu, S. Liu, and X. Fan, "The power of fragmentation: a hierarchical transformer model for structural segmentation in symbolic music generation," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, no. 5, pp. 1409-1420, 2022. https://doi.org/arXiv:2205.08579v2

[18] S. Li, and Y. Sung, "MRBERT: Pre-Training of melody and rhythm for automatic music generation," Mathematics, vol. 11, no. 4, pp. 798-811, 2023. https://doi.org/10.3390/math11040798

[19] S. Choudhuri, S. Adeniye, and A. Sen, "Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation," Artificial Intelligence and Applications vol. 1, no. 1, pp. 43-51, 2023. https://doi.org/10.47852/bonviewAIA2202524

[20] V. Beltrán-Palanques, and M. Querol-Julián, "The genre of PechaKucha presentations: Analysis and implications for enhancing multimodal literacy at university," English for Specific Purposes, vol. 75, no. 1, pp. 27-37, 2024. https://doi.org/10.1016/j.esp.2024.05.002

[21] S. S. Hosseini, M. R. Yamaghani, and S. Poorzaker Arabani, "Multimodal modelling of human emotion using sound, image and text fusion," Signal, vol. 1, no. 81, pp. 71-79, 2024. https://doi.org/10.1007/s11760-023-02707-8

[22] J. Wang, A. Sharifi, T. R. Gadekallu, and A. Shankar, "MMD-MII model: A multilayered analysis and multimodal integration interaction approach revolutionizing music emotion classification," International Journal of Computational Intelligence Systems, vol. 17, no. 1, pp. 99-124, 2024. https://doi.org/10.1007/s44196-024-00489-6

[23] I. C. Lu, J. Y. Huang, and W. P. Lee, "An emotion-driven and topic-aware dialogue framework for human–robot interaction," Advanced Robotics, vol. 38, no. 4, pp. 267-281, 2024. https://doi.org/10.1080/01691864.2023.2297902

[24] L. Jing, "Evolutionary deep learning for sequential data processing in music education," Informatica, vol. 48, no. 8, pp. 63-78, 2024. https://doi.org/10.31449/inf.v48i8.5444

[25] H. F. T. Al-Saadawi, and R. Das, "TER-CA-WGNN: Trimodel sentiment recognition using cumulative attribute-weighted graph neural network," Applied Sciences, vol. 14, no. 6, pp. 2252-2281, 2023. https://doi.org/10.3390/app14062252

[26] G. Wang, L. Tan, and Z. Shang, "Multimodal dual emotion with fusion of visual sentiment for rumor detection," Multimedia Tools and Applications, vol. 83, no. 10, pp. 29805-29826, 2024. https://doi.org/10.1007/s11042-023-16732-9

[27] R. Kumari, V. Gupta, N. Ashok, T. Ghosal, and A. Ekbal, "Emotion aided multi-task framework for video embedded misinformation detection," Multimedia Tools and Applications, vol. 83, no. 12, pp. 37161-37185, 2024. https://doi.org/10.1007/s11042-023-17208-6

[28] B. Subbaiah, K. Murugesan, P. Saravanan, and K. Marudhamuthu, "An efficient multimodal sentiment analysis in social media using hybrid optimal multi-scale residual attention network," Artificial Intelligence Review, vol. 57, no. 2, pp. 34-56, 2024. https://doi.org/10.1007/s10462-023-10645-7

[29] H. Zhao, "Research on the recognition of psychological emotions in adults using multimodal fusion," Informatica, vol. 48, no. 9, pp. 155-162, 2024. https://doi.org/10.31449/inf.v48i9.5876

[30] M. Zhang, C. Wang, Y. Sun, and T. Li, "Memristive PAD three-dimensional emotion generation system based on D–S evidence theory," Nonlinear Dynamics, vol. 112, no. 6, pp. 4841-4861, 2024. https://doi.org/10.1007/s11071-023-09264-2

Optimized Unsupervised Semantic Trajectory Mining for Personalized Tourism Recommendations

Panpan Guo
School of Tourism Management, Zhengzhou Tourism College, Zhengzhou, 451464, China
Email: guogpp057@outlook.com

Keywords: unsupervised, multiple semantics, trajectory mining, intelligent tourism, personalized recommendations, multi granularity

Received: August 28, 2024

With the advent of the big data era, tourists receive a massive amount of travel information every day, and traditional personalized travel recommendation methods are difficult to provide personalized travel recommendation services. Therefore, the study first introduced a multi granularity recommendation framework based on attention mechanism, and then designed an unsupervised multi-implicit semantic trajectory mining model. The multi granularity recommendation framework was used for optimization to obtain more comprehensive user preferences. Finally, based on this, an intelligent personalized tourism recommendation method was established. The study used a dataset of user travel trajectory data collected and processed from various popular travel portals for subsequent testing, and evaluated the performance of different recommendation models from three aspects: computational efficiency, overall utility, and resource utilization. Accuracy, average training time F1 value, and resource utilization were selected to evaluate the performance of different recommendation models. The research results showed that compared with other algorithms, the recall and accuracy growth rates based on the optimized unsupervised multiple cryptic semantic trajectory mining model were the fastest, with corresponding recall and accuracy rates of 86.57% and 97.62%, respectively. Furthermore, in comparison with the most prevalent personalized tourism recommendation model, which was based on an enhanced frog leaping algorithm, the proposed model, which was based on an optimized unsupervised multiple implicit semantic trajectory mining model, demonstrated an average increase of 10.3% and 8.9% in Hit Ratio and normalized cumulative loss gain, respectively. In a comparison of the performance of the research method with that of mainstream algorithms, the research method was found to perform the best, demonstrating excellent results in terms of computational efficiency, F1 value, and resource utilization. The values for these metrics were 11.24 s, 97%, and 91%, respectively. Finally, in practical applications, two tourism planning schemes for visiting Guilin were generated based on user needs, and users were very satisfied with the generated results. In summary, the method proposed in the study has good performance and can effectively improve user satisfaction and service quality in the tourism industry.

Povzetek: Raziskava uvaja optimiziran model UMISTM za rudarjenje trajektorij in personalizirana priporočila v turizmu.

1 Introduction

In recent years, the tourism industry has been developing rapidly, and tourism services are also moving toward intelligence and informatization. However, there are still many problems in the development process, such as poor quality of intelligent recommendation services, information asymmetry, and so on. In addition, the personalized needs of tourism services are becoming more and more important. How to meet the personalized needs of different tourists in the tourism process has become an important challenge [1-3]. The Unsupervised Multiple Implicit Semantic Trajectory Mining (UMISTM) model does not require a large amount of labeled data for training, thus better mining user travel trajectories and interest preferences. At the same time, it can also mine user's cryptic semantic trajectories in different dimensions, thereby more comprehensively grasping the personalized needs of users [4-5]. However, this model requires a large amount of training data for training, as well as complex optimization and adjustment of model parameters, which will consume a lot of computational resources and time costs [6-7]. In response to the above issues, the study introduces a Multi Granularity Recommendation (MGR) framework based on Attention Mechanism (AM) to optimize the UMISTM model and establish an intelligent personalized tourism recommendation method. The research aims to improve the effectiveness of personalized tourism recommendations and user satisfaction, provide valuable insights for the tourism industry, and also provide new ideas for the development of tourist attractions. There are two main innovative points in the research. The first point is to propose an optimized UMISTM model to mine user travel trajectories, and apply it to intelligent

personalized tourism recommendation methods. The second point is to design an AM-based MGR framework to improve the UMISTM model and use it to extract user preferences. The research structure is mainly divided into four parts. The first part is a review of relevant research results. The second part is to construct the UMISTM model and optimize its design for intelligent personalized tourism recommendation methods. The third part is to verify the performance and application effectiveness of the proposed research method. The last part is a summary of the research.

2 Related works

With the continuous improvement of the national economic level and the continuous increase in per capita disposable income of residents, people have more funds for tourism consumption, which has promoted the development of the tourism industry. However, designing and recommending more intelligent tourism trajectories for different user needs can further improve the quality of tourism services and user satisfaction. Lin et al. developed a two-stage optimization model to address the importance of recommendation diversity. In the first stage, the optimization model considered topic diversity, while in the second stage, a model that minimized misclassification costs was proposed to balance the diversity and accuracy of recommendations. The research method was shown to significantly improve the accuracy and diversity of recommendations in real-world travel data [8]. Nitu et al. proposed a personalized travel recommendation system based on social media activities and proximity effects. The results showed that the model was superior to the current personalized interest point model, with an overall accuracy of 75.23% [9]. Kumar et al. introduced a model based on explicit sentiment and star ratings to predict the best tourist destination. The model extracted user-selected topics through latent Dirichlet topic allocation modeling, and then introduced a morphological linear neural network model to generate sentiment scores for text content. The experimental results showed that the accuracy, recall and F1 score of the research method were 86%, 87% and 89%, respectively [10]. Fudholi et al. designed a deep learning based mobile tourism recommendation system to help tourists discover more different tourist destinations. The experimental results showed that the system achieved an average accuracy of over 85% and an average absolute percentage error of 5% [11].

The UMISTM model is an unsupervised learning method that can be used for text mining and can automatically learn the semantic and structural information of text, but its training process is more complex and time-consuming. Li et al. proposed a graph based encoding source code semantic method, which was mainly used for limited interpretive and implicit representation learning. The results showed that the proposed method outperformed text-based methods without increasing complexity [12]. Duan et al. believed that the behavior of users calling services belonged to implicit semantics. By analyzing user calls, mining their implicit semantic representations, and modeling users' hidden preferences, the research results showed that the proposed hybrid intelligent service recommendation method based on implicit semantics and explicit scoring had better recommendation accuracy and recall than existing methods [13]. Zhang et al. proposed a sparse user check-in location prediction method based on implicit meaning to predict the sparse online check-in records and decision-making context of location-based online social network users. The research results showed that this method significantly improved the prediction accuracy [14]. Due to the influence of not only network structure but also supervised and unsupervised learning on the learning of specific feature encoding models, Yao et al. designed a graph comparison accelerated encoder using unsupervised learning, which maintained the final performance at an advanced level [15]. To better demonstrate the above research findings, a summary is conducted as shown in Table 1.

Table 1: Summary of research findings

Author	Method	Index	Key Results
Lin et al. [8]	Developed a two-stage optimization model	Accuracy and diversity	/
Nitu et al. [9]	Personalized travel recommendation system based on social media activities and proximity effects	Overall accuracy	75.23%
Shah et al. [10]	Introducing a model based on explicit sentiment rating and star rating to predict the best tourist destination	Accuracy, recall, and F1 score	86%, 87%, and 89%
Kumar et al. [11]	Mobile tourism recommendation system based on deep learning	Accuracy and Mean Absolute Percentage Error	85% and 5%
Li et al. [12]	Graph based encoding source code semantic method	/	/
Duan et al. [13]	Propose a sparse user check-in location prediction method based on implicit semantics	Accuracy and recall rate	/
Zhang et al. [14]	Design a graph contrast acceleration encoder using unsupervised learning	Predictive accuracy	/
Yao et al. [15]	Sparse user check-in location prediction method based on implicit semantics	/	/

In summary, there is currently relatively little research on multiple implicit meanings, and personalized tourism recommendations are also facing great difficulties. Therefore, this study designs a UMISTM model for mining hidden information in tourism trajectories, and optimizes it using the MGR framework, ultimately obtaining an intelligent personalized tourism recommendation method based on the optimized UMISTM model.

3 Design of personalized recommendation method for intelligent tourism based on optimized UMISTM model

In recent years, with the improvement of people's living standards and the transformation of tourism concepts, more and more people have chosen to spend their leisure time on tourism activities. However, the rapid increase in tourist attractions has caused great difficulties for people to choose the attractions they are interested in. In response to the above issues, the study first constructs a UMISTM model for mining user travel trajectories, then improves it using the MGR framework, and finally constructs an intelligent personalized travel recommendation method.

3.1 User travel trajectory mining based on UMISTM model

User travel trajectory refers to the itinerary, duration of stay, and consumption situation of tourists during the tourism process. By obtaining user travel trajectory, it can better grasp user travel preferences and consumption habits, and provide reference for the development and marketing of the tourism industry [16-17]. However, it is difficult to extract valuable information from user travel trajectories. To solve the above problems, a UMISTM model is proposed, which can semantically represent various complex and diverse information in travel trajectories and has strong flexibility. Firstly, it is necessary to preprocess the data and study the collection of 200000 users' travel information from tourism portals such as Fliggy, mainly including scenic spot route information and time information. The collected data should be subjected to noise removal, missing value filling, and display of scenic spot attribute information through knowledge graphs. In the process of data cleaning, administrative region information and non-urban scenic spots are mainly screened out. Then the cleaned data is entity identical through the tourism domain knowledge graph. Finally, the route information in the tourist travelogue can be obtained. The specific process is shown in Figure. 1.

After data preprocessing, 14000 user travel trajectory data can be obtained, with 257 scenic spots in 2022. Through the above operations, a preliminary summary of user travel activities can be obtained, as shown in Figure. 2.

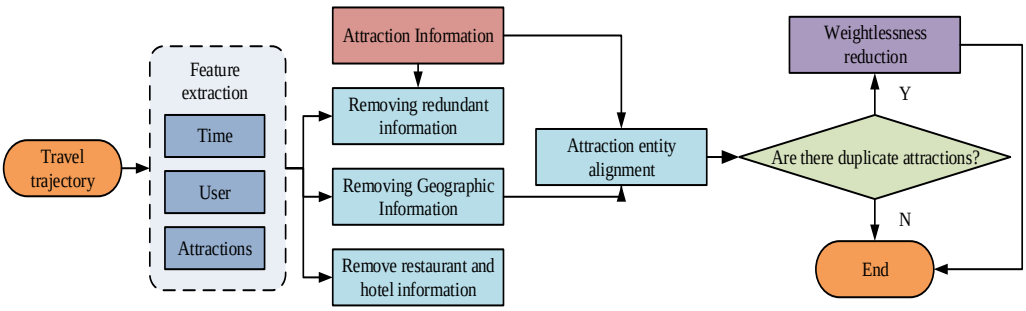


Figure 1: Data preprocessing process based on UMISTM model

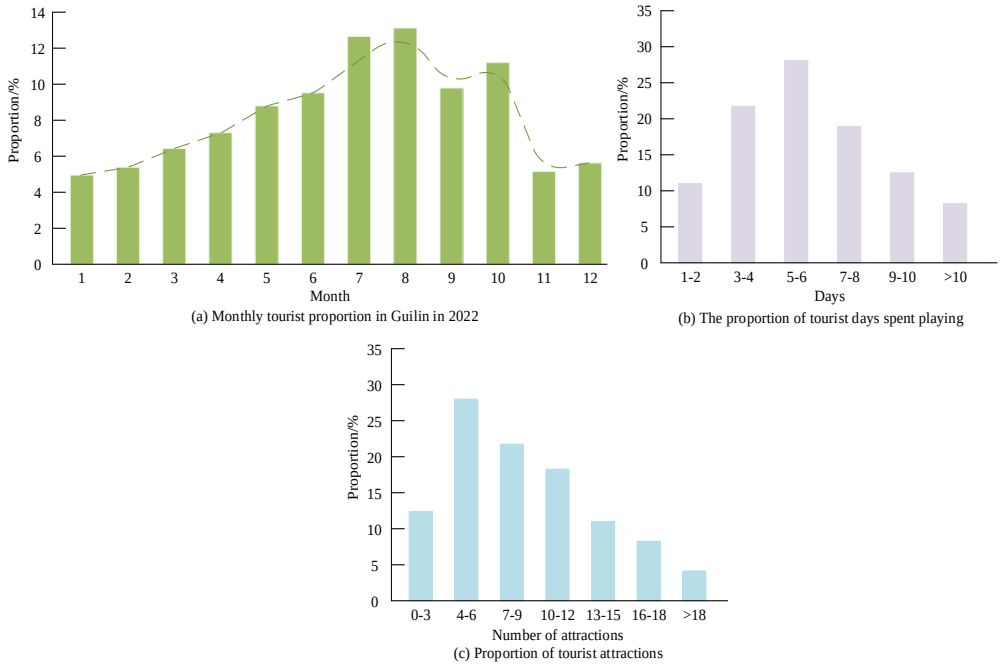


Figure 2: Preliminary summary of user travel behavior

In Figure. 2, the average number of tourist attractions in the area is 16, with over 80% of users staying in the area for 7 days and visiting multiple attractions in a relatively short period of time. The user travel trajectories obtained through research mainly include three types: time information, user information, and scenic spot information. These are integrated into the UMISTM model, and the above implicit semantic information is represented through the same dimensional space. Before constructing the UMISTM model, it is necessary to define the attraction records and travel trajectories. The former is when user $user$ travels at attraction a while in t, and the attraction record a_s can be obtained as $\{user, t, a\}$. The latter is the gameplay trajectory $PT: \{user, t_1, a_1\}, \cdots, \{user, t_i, a_i\}, \cdots, \{user, t_n, a_n\}$ where each a_s of $user$ travels continuously over time, where n represents the length of PT and satisfies the following condition $s_i < s_{i+1}$, $for\ i \le n-1$. The implicit meaning of user travel trajectory represents the factors that affect the travel route. Searching for the implicit meaning of user travel trajectory can better grasp the user's travel behavior. Therefore, the UMISTM model uses contextual information to obtain corresponding implicit semantic features. The specific structure is shown in Figure. 3.

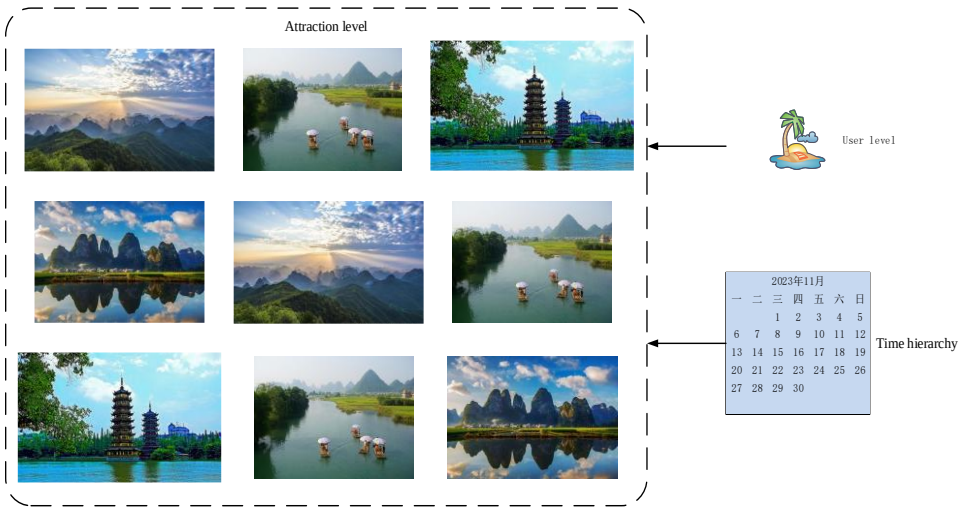


Figure 3: Structure diagram of UMISTM model

In Figure. 3, the model includes three levels of implicit meanings: time, user, and attraction. By using these implicit meanings, one can fully understand the user's tourism activities. In the modeling process of the UMISTM model, it is first necessary to represent the probability of tourism trajectory occurrence for $user$'s PT in month m through conditional probability p. Then, the multiple implicit semantic feature vectors are instantiated to obtain the objective function expression of the UMISTM model, as shown in equation (1).

$$\sum_{user\in U}\sum_{ut\in T}\sum_{i=1}^{N_t}\log p\left(a_i\left|user_{\text{context}},\ t_{\text{context}},\ \underbrace{a_{i-K}:a_{i+K}}_{\text{context}}\right.\right) \quad (1)$$

In equation (1), U, T, and N_t are the sets of $user$ and m, as well as the trajectory length, respectively. $a_{i-K}:a_{i+K}$ represents the position sequence of the model training window for a_i, and $context$ represents the context corresponding to different levels. Among them, p is usually established using the softmax function. However, in the study, the computational cost of using this function in the model is high, so negative sampling is chosen to train the model parameters. The probability of maximizing the location of all scenic spots is obtained by the given $context$ of PT and calculated as shown in equation (2).

$$\sum_{i=1}^{N_t}\log p\left(a_i\left|f^{(i)}\right.\right) \quad (2)$$

In equation (2), a_i represents the location of the target scenic spot, and $f^{(i)}$ represents $context$ with all implicit meanings. Among them, a_i is a positive sample, and attractions that are not in the context of the attraction are negative samples. For a given positive sample $\left(f^{(i)},a_i\right)$, the objective function is improved by using the random gradient ascent method to obtain the gradient results of the final objective function $\zeta\left(a_i,\psi\right)$ with respect to the different levels of $context$ vectors and $X_{f^{(i)}}$ in $f^{(i)}$. Therefore, the update expression V for all embedded vectors is shown in equation (3).

$$V_f := V+\eta\sum_{\psi\in\{a_i\}\cup NEG\{a_i\}}\frac{\partial\zeta\left(a_i,\psi\right)}{\partial X_{f^{(i)}}},\ f\in context\left(a_i\right) \quad (3)$$

In equation (3), η represents the learning rate, and $NEG\left(a_i\right)$ represents a non empty negative sample subset related to a_i. By training through the above steps, the final UMISTM model can be obtained, which can be applied in tourism trajectory route recommendation in the future.

3.2 Optimization based on UMISTM model

Due to the unsupervised learning nature of the UMISTM model, it has high work efficiency but low accuracy [18-20]. Therefore, the MGR framework will be optimized and the learning effectiveness of the UMISTM model will be improved through diverse training data and multi granularity mining, thereby improving the personalized recommendation method for intelligent tourism. Firstly, it is necessary to define the following content: interaction event $user_i=\left(user,t,i,c\right)$ is when $user$ interacts with item i at time t, where c is the category to which i belongs. The interaction behavior sequence $S=\left\{\left(user,t_1,i_1,c_1\right),\cdots,\left(user,t_n,i_n,c_n\right)\right\},t_1<t_2<\cdots<t_n$ of $user$ concatenates all interaction points of $\left(user,t,c\right)$ in ascending order through timestamps. By forming a triplet $user$ with $user$, timestamp, and category among all interaction points, a coarse-grained interaction sequence $C=\left\{\left(user,t_1,c_1\right),\left(user,t_2,c_2\right),\cdots,\left(user,t_n,c_n\right)\right\}$ can be obtained. By combining $user$, timestamp, and i, a fine-grained interaction sequence $I=\left\{\left(user,t_1,i_1\right),\left(user,t_2,i_2\right),\cdots,\left(user,t_n,i_n\right)\right\}$ can be obtained by arranging them. User preferences are hidden in S, so it is very important to extract each $user$'s own preferences from the large amount of information collected. The study first learns and mines fine-grained preferences based on the I-sequence of each user, and then learns and mines coarse-grained preferences based on the C-sequence of each user. The specific schematic diagram of the MGR framework structure is shown in Figure. 4.

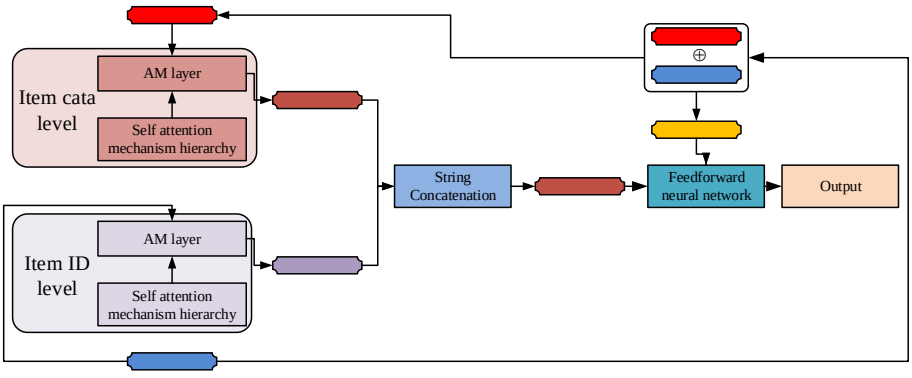


Figure 4: MGR frame structure schematic diagram

In Figure. 4, the MGR framework utilizes the same two parallel structures to mine user preferences of different granularities, and then uses the AM method to mine multi granularity preferences. Finally, static and dynamic user feature representations of different granularity levels are obtained. After concatenation, a complete preference representation of users after watching new travel trajectories can be obtained. The interaction points of the interaction behavior sequence are composed of four elements, and the information contributed by all of them is the project, the category to which the project belongs, and the time information. If there is more than one project in each category, the user's interaction behavior sequence can be divided into interaction sequences corresponding to the project and the category, which provides the user with different aspects of preference information. Therefore, an extensible two-level framework, namely MGRA, is designed to comprehensively analyze user preferences. The structure of the MGR framework specifically includes five levels: embedding layer, hidden layer, AM layer, attention layer, and application layer. The embedding layer includes embedding vector methods corresponding to c and i, and interaction time, in order to obtain the interaction behavior encoding of all $user$. The specific expression is shown in equation (4).

$$\begin{cases} user_i = fv(i) + lookup_i^t\left\{bucketize_i(t)\right\} \\ user_c = fv(c) + lookup_c^t\left\{bucketize_c(t)\right\} \end{cases} \tag{4}$$

In equation (4), $user_i$ and $user_c$ are the interaction behavior codes corresponding to i and c of $user$, $fv(i)$ and $fv(c)$ are the feature vectors corresponding to i and c after passing through the feedforward neural network, respectively. The above feature vectors are shareable, $lookup$ represents the lookup function, and $bucketize$ represents the automatic standardization of a certain time. The output result of this level is a vector list composed of all interaction i and c after combining time information, as shown in equation (5).

$$\begin{cases} I = \left\{user_{i1}, user_{i2}, \cdots, user_{in}\right\} \\ C = \left\{user_{c1}, user_{c2}, \cdots, user_{cn}\right\} \end{cases} \tag{5}$$

The above list is input to the hidden layer. Due to the significant differences between the two embedding vectors, it is not possible to perform concatenation operations to merge the information. Therefore, a linear projection method is used to uniformly project the two obtained vectors into the implicit semantic space of dimension K. The modified vector expressions for I and C can be obtained, as shown in equation (6).

$$\begin{cases} I' = cancat\left\{\chi_{\theta_1}(user_{i1}), \chi_{\theta_2}(user_{i2}), \cdots \chi_{\theta_n}(user_{in})\right\} \\ C' = cancat\left\{\chi_{\psi_1}(user_{c1}), \chi_{\psi_2}(user_{c2}), \cdots \chi_{\psi_n}(user_{cn})\right\} \end{cases} \tag{6}$$

In equation (6), $cancat$ represents the string concatenation function, and χ represents the projection function with parameter α and the relu activation function, which puts all interaction behavior and time information together into a m-dimensional vector space. To improve the expression of I and C sequences, G-fold projections are conducted on both sequences and the sequence information is described through D semantic spaces. The corresponding expressions are shown in equation (7).

$$\begin{cases} I_D = \chi_{\beta_D}(I) \\ C_D = \chi_{\varsigma_D}(C) \end{cases} \tag{7}$$

In equation (7), χ_{β_D} and χ_{ς_D} are both corresponding projection functions in $H-th$ space. In the AM layer, it is mainly used to capture the hidden intrinsic relationships in all semantic spaces. Due to the correlation between the occurrence of $user$'s interaction behavior and other interaction behaviors, the research conducts AM within previous interaction sequences to enhance the information of related interaction points. The attention dispersion matrices G_{I-D} and G_{I-C} corresponding to I and C in each subspace are calculated as shown in equation (8).

$$\begin{cases} G_{I-D} = soft\max\left(\tau(I_D, I)\right) \\ G_{C-D} = soft\max\left(\tau(C_D, C)\right) \end{cases} \tag{8}$$

In equation (8), $soft\max$ is the softmax function and τ is the calculation function of attention score, which is used to determine the impact of all interaction behaviors in $H-th$, as shown in equation (9).

$$\begin{cases} \tau(I_D, I) = I_D \square W_{I-k} \square I^T \\ \tau(C_D, C) = C_D \square W_{C-k} \square I^T \end{cases} \tag{9}$$

In equation (9), W represents the weight of the corresponding interaction behavior. In $H-th$, the expression for the attention vector is shown in equation (10).

$$\begin{cases} H_{I-D} = G_{I-D}\chi_{Q_{I-D}}(I) \\ H_{C-D} = G_{C-D}\chi_{Q_{C-D}}(C) \end{cases} \tag{10}$$

In equation (10), $\chi_{Q_{I-D}}(I)$ and $\chi_{Q_{C-D}}(C)$ are both projection functions with the activation function relu. Finally, after splicing, the recombination operation can be performed to obtain the static coarse-grained U_I and fine-grained preference U_C, as calculated in equation (11).

$$\begin{cases} U_I = \gamma_{e-I} \cdot concat\{U_{I-1}, U_{I-2}, \cdots, U_{I-D}\} \\ U_C = \gamma_{e-C} \cdot concat\{U_{C-1}, U_{C-2}, \cdots, U_{C-D}\} \end{cases} \quad (11)$$

In equation (11), γ_{e-I} and γ_{e-C} belong to the feedforward network of the embedding layer, and the static multi granularity preference can ultimately be obtained. In the attention layer, AM is introduced to capture the dynamic preference of $user$, and through the static multi granularity acquisition steps of the recommended items, the dynamic multi granularity preference is ultimately obtained. In the application layer, first set the sorting function rf to describe the dynamic preference q_t of $user$ and the correlation r between the recommended item b_{user}, as shown in equation (12).

$$r = h(b_{user}, q_t) \quad (12)$$

Due to the fact that the interaction probability between q_t and b_{user} belongs to a binary classification problem, the mainstream cross entropy loss function is selected, and the calculation is shown in equation (13).

$$LOSS = -\sum_{i,user} y_i \log r + (1 - y_i) \log(1 - r) \quad (13)$$

In equation (13), y_i is the predicted result. The aforementioned content illustrates that the incorporation of AM can augment the semantic data associated with pertinent interaction points, thus mitigating the detrimental effects of unrelated interaction points and facilitating the effective capture of user-specific preferences. In summary, an optimized UMISTM model can be obtained and applied to mining user travel preferences and tourism recommendations.

3.3 Intelligent personalized tourism recommendation method based on optimized UMISTM model

Based on the above content, this study constructs an intelligent personalized tourism recommendation method for cross domain preference acquisition, extracts tourism planning and preference information of new users in the city, and then generates personalized tourism routes. The specific structure of the intelligent tourism personalized recommendation method based on optimized UMISTM model is shown in Figure. 5.

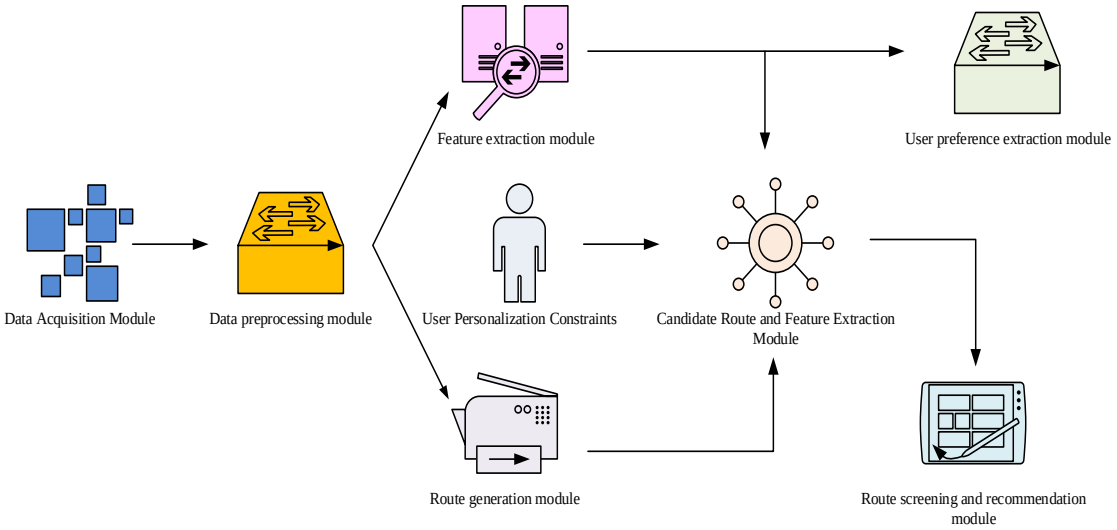


Figure 5: Process of personalized recommendation method for intelligent tourism based on optimized UMISTM model

In Figure. 5, the intelligent personalized tourism recommendation method consists of 7 modules, mainly divided into three parts: data collection and preprocessing, feature extraction and travel trajectory generation, and tourism recommendation. In the data collection and preprocessing part, the first step is to crawl the user's travel trajectory, including attractions, transportation, hotels, etc. Next, data cleaning operations are carried out to remove invalid and redundant information. Then, information such as time and travel expenses between different attractions is added, and various attributes of the attractions are encoded through a knowledge graph. Finally, the user's travel trajectory in other cities is queried to obtain corresponding personalized travel preferences, and all scenic spots are classified into fixed categories, including 25 categories

such as humanities, karst caves, and history. Composite statistics are performed on the categories of scenic spots and the proportion of each category is calculated. In the feature extraction section, it mainly includes the extraction of scenic spot features and user travel preferences. In the part of travel trajectory generation and travel recommendation, personalized travel recommendation needs to meet both the user's constraints and personalized needs. The generated travel route feature representation vector is calculated as shown in equation (14).

$$b_a = \frac{b_{a1} + b_{a2} + \cdots + b_{aM}}{N} \tag{14}$$

In equation (14), N represents the feature representation vector of attraction a in the generated tourist route, and N represents the number of attractions in the tourist route. Because N and the user's personal preference b_p are known, the matching degree M between the two is calculated using equation (15).

$$M = b_a \square b_p^T \tag{15}$$

Through the above matching, the recommended route with the same number of candidate tourist routes and the highest matching value can be obtained.

4 Result analysis of personalized recommendation method for intelligent tourism based on optimized UMISTM model

To analyze the performance and application effect of the intelligent tourism personalized recommendation method based on optimized UMISTM model proposed in the study, the study first explored the performance of the optimized UMISTM model in the method, and then applied it to actual tourism recommendation to explore the application of the intelligent tourism personalized recommendation method proposed in the study.

4.1 Result analysis based on optimized UMISTM model

To evaluate the performance of the optimized UMISTM model, commonly used metrics such as recall, accuracy, training time, resource utilization, and F1 value were selected for evaluation. Among them, accuracy represented the overall prediction accuracy, recall was the probability of being predicted as a positive sample, F1 value reflected the comprehensive effect of the model, training was used to analyze the computational efficiency, and resource utilization measures the effective use of model resources in terms of their availability and capacity. A comprehensive analysis of the model was required in order to ascertain its accuracy, computational efficiency, and comprehensiveness. This will facilitate the development of efficient and accurate personalized tourism recommendation methods. In addition, to more scientifically validate the performance of the proposed model, the study selected the currently mainstream Personalized Travel Recommendation Based on Improved Frog Jump Algorithm (PTR-IFJ) and UMISTM models for comparative experiments. The dataset used in the study was 14000 user travel trajectory data collected and preprocessed from various popular tourism portal websites, which were divided into training and testing sets in a 7:3 ratio. The experimental parameter settings are shown in Table 2.

Table 2: Experimental parameter setting

Parameter	Corresponding parameters	Illustrate	Parameter value
thre		High frequency downsampling threshold	1E-5
η		Initial learning rate	0.05
threads		Number of program threads	9
nega		Negative sample count	25
iter		Iterations	300
size	Z	Vector dimension	200
wind	2c	Training window size	5

Experimental parameters were an extremely important part of model training, and in practical situations, it was necessary to improve the parameters based on the model, dataset, and training results. The training and experimentation of the model were conducted on the Hewlett Packard Z8 G4 workstation. The study first took the Four Seasons Autumn Begonia as an example, and its best viewing seasons were December and January, so its viewing had strong seasonal semantics. A study was conducted to compile and summarize the number of visitors to the Four Seasons Autumn Begonia in each month of 2022.

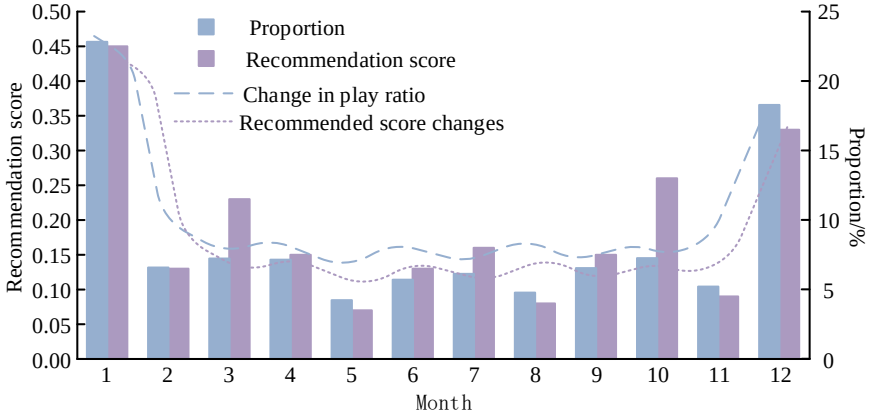


Figure 6: The Number of Visits and Recommended Results of Four Seasons Begonia in 2022 under the UMISTM Model

The number of visits and recommended results of autumn crabapple in each month of 2022 under the UMISTM model are shown in Figure. 6. Among them, the horizontal axis represented each month in mid-2022, the left vertical axis corresponds to the recommended ratings, and the right vertical axis corresponded to the share of tourists. In Figure. 6, the recommendation scores for users recommending the Four Seasons Autumn Begonia scenic spots were the highest in January and December, and the actual trend of gameplay was consistent with the changes in the UMISTM model recommendation scores. The above results indicated that the research method could fully preserve the seasonal semantics hidden in scenic spots. By studying the cosine similarity between users and attractions, a top-K candidate attraction list was generated, and then the recall and accuracy results of different recommendation models were evaluated.

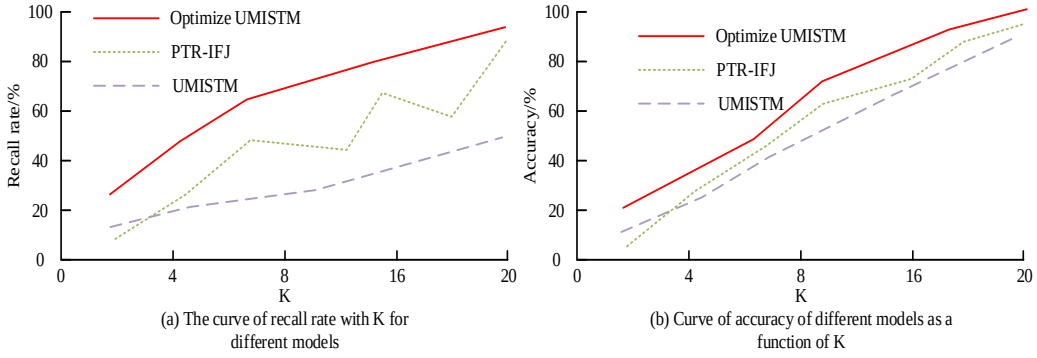
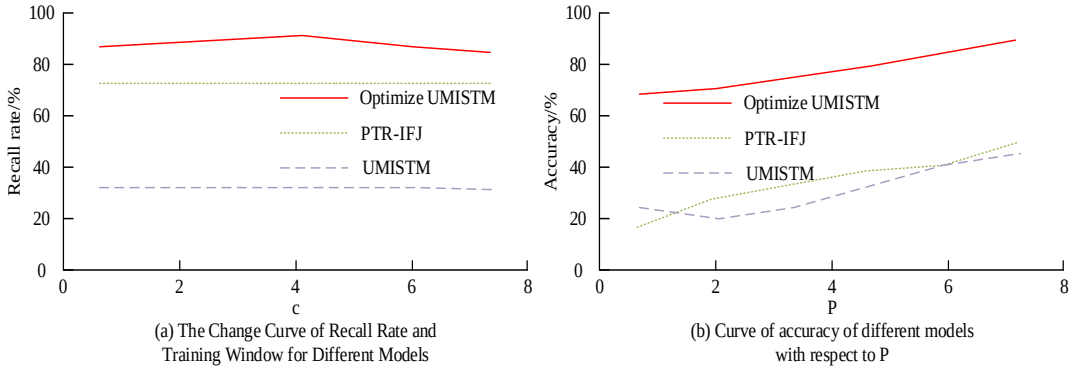


Figure 7: Performance results of different recommendation models and the variation curve of K

The performance results of different recommendation models and the variation curve of K are shown in Figure. 7. Figure. 7(a) and 7(b) show the recall and accuracy results and the K value, respectively. The vertical axis corresponded to the recall and accuracy indicators, while the horizontal axis corresponded to the K value. As the length of the scenic spot recommendation list continued to expand, the optimized UMISTM model had the fastest growth rate in both recall and accuracy. When K was 20, the corresponding recall and accuracy were 86.57% and 97.62%, respectively. In the accuracy results, the UMISTM model gradually surpassed the PTR-IFJ model as the K value increased, ultimately reaching 95.16%. This was due to the fact that as the K-value increases, the recommended list corresponding to the recommendation model will expand, thereby increasing the probability of recommending accurate tourist attractions to users. Consequently, the results of the study will demonstrate an upward trajectory in terms of recall and accuracy, while the implicit semantic information embedded within the research method will exhibit greater richness and completeness. To explore the impact of the size of the model training window and the P-value on the model, a fixed K-value of 15 was set, where P was the number of historical scenic spots that the user watched.

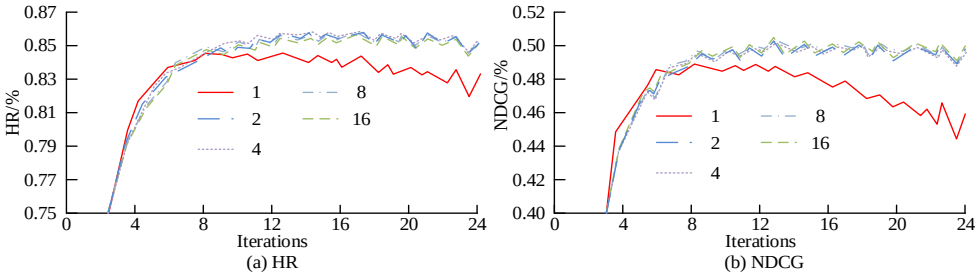


(a) The Change Curve of Recall Rate and Training Window for Different Models

(b) Curve of accuracy of different models with respect to P

Figure 8: Variation curves of different models under different influencing factors

The variation curves of different models under different influencing factors are shown in Figure. 8. Figure. 8 (a) and 8 (b) show the change curves of recall and accuracy under different influencing factors. As the training window size continued to expand, the optimized UMISTM model showed a trend of initially increasing at a larger rate and then slowly decreasing, while the curves of the other two recommended models were relatively stable. This indicated that when c was 4, the proposed model retained the most comprehensive implicit semantic information. When P was 24, the accuracy of the optimized UMISTM model, PTR-IFJ model, and UMISTM model were 97.25%, 96.33%, and 91.78%, respectively. To further evaluate the performance of the optimized UMISTM model, the study selected Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) for evaluation.



(a) HR

(b) NDCG

Figure 9: The impact of the number of hidden semantic spaces on HR and NDCG of optimizing UMISTM models

The impact of the number of hidden semantic spaces on the HR and NDCG of the optimized UMISTM model is shown in Figure. 9. Figure. 9 (a) and 9 (b) show the results of HR and NDCG, respectively. As the number of iterations increased, the number of semantic spaces in different hidden layers showed an increase in both indicators. Among them, the performance of optimizing the UMISTM model was the worst when the number was 1, and the changes in HR and NDCG were relatively consistent under the other numbers. However, considering both indicators comprehensively, the model had the best performance when the number was 4, with the average values of the two indicators being 83.16% and 48.85%, respectively. To further analyze the performance of the proposed model, the TMDB 5000 Movie dataset was introduced for testing, which included 500000 comments from 975 users on 10000 movies, mainly including user, movie, and rating data.

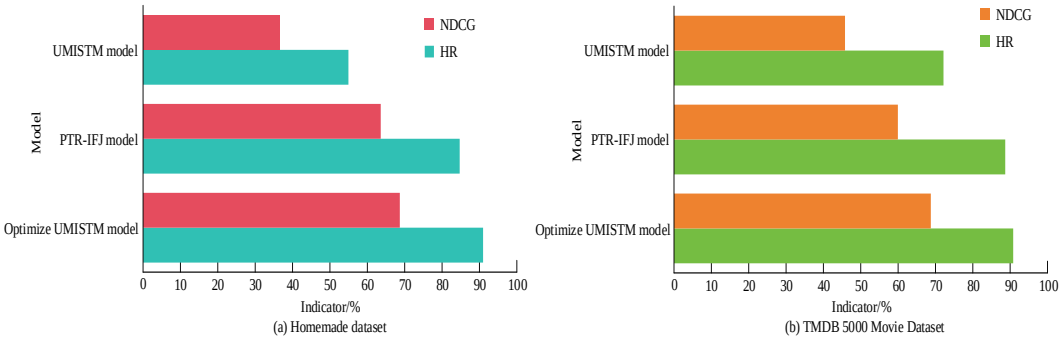


Figure 10: Performance results of different recommendation models on self-made datasets and TMDB 5000 Movie datasets

The performance results of different recommendation models on the self-made dataset and TMDB 5000 Movie dataset are shown in Figure. 10. Figure. 10 (a) and 10 (b) show the performance results of the self-made dataset and TMDB 5000 Movie dataset, respectively. The optimized UMISTM model achieved the best results on HR and NDCG on different datasets. Compared with the most mainstream PTR-IFJ model, the proposed model improved HR and NDCG by an average of 10.3% and 8.9%, which confirmed the effectiveness of the research method. In summary, the model proposed in the study had better performance and comprehensively captured user preferences and personalized preferences during travel. Finally, to further analyze the computational efficiency and resource utilization of the research method, the effectiveness of different recommendation models was tested. In addition to the comparative recommendation models previously discussed, the study also introduces the two-stage optimization model, which demonstrated the most optimal performance in the relevant work section. Furthermore, models based on explicit sentiment rating and star rating are presented for comparison. The results are presented in Table 3.

Table 3: Calculation efficiency and resource utilization results of different recommendation models

Recommendation models	Average training time/s	F1 value/%	Resource utilization rate/%
Optimize UMISTM	11.24	97	91
PTR-IFJ	18.36	91	87
UMISTM	13.65	92	89
Two stage optimizations	21.35	90	88
A model based on explicit emotional rating and star rating	26.83	88	85

In Table 3, the research method displayed excellent performance in terms of computational efficiency, overall performance, and resource utilization, with values of 11.24s, 97%, and 91%, respectively. However, the model based on explicit sentiment and star ratings performed poorly, with corresponding indicators of 26.83, 88%, and 85%, respectively. Among them, the PTR-IFJ recommendation model had excellent performance, with computational efficiency, comprehensive performance and resource utilization of 18.36s, 91% and 87%, respectively. Compared with them, the research method greatly improved in various aspects of performance. This was because the research method effectively and deeply explores various hidden semantic information in the route, and obtained more comprehensive user personalized preferences at different granularities in the feature extraction part. Lightweight processing methods were introduced, so the accuracy and computational efficiency of search was greatly improved. In addition, the PTR-IFJ recommendation model had low computational efficiency due to insufficient communication between local and global information during the search process. Compared with the best performing two-stage optimization models in related work, both made certain improvements in accuracy. Compared with the best performing two-stage optimization models in related work, both have shown certain improvements in accuracy. The two-stage optimization model improves the accuracy and diversity of tourism recommendations through different stages, but does not pay attention to computational efficiency, resulting in lower computational efficiency.

4.2 Application of intelligent tourism personalized recommendation method based on optimized UMISTM model

To test the practical application effect of the intelligent tourism personalized recommendation method based on the optimized UMISTM model, a study randomly selected a user who planned to play for 3 days in June, and generated two new routes through the intelligent tourism personalized recommendation method.

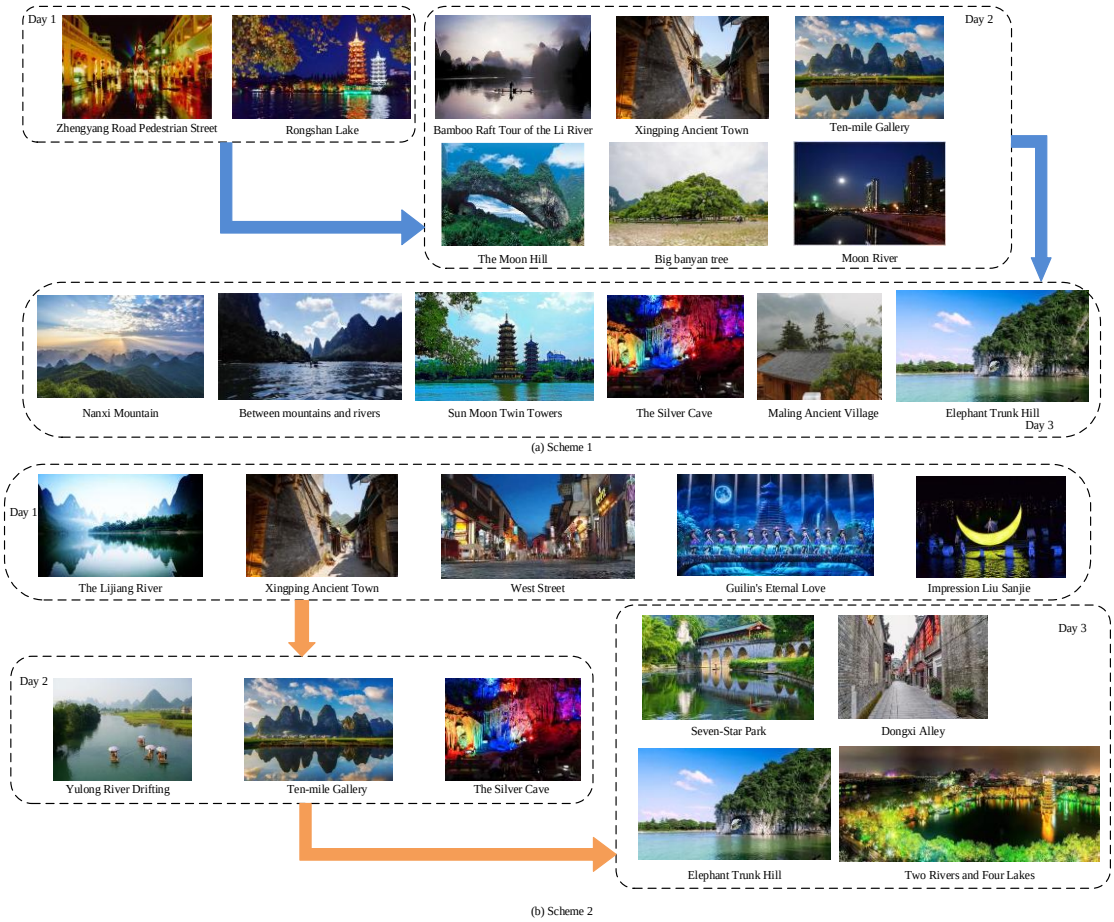


Figure 11: Recommendation scheme of intelligent tourism personalized recommendation method based on optimized UMISTM model in practical applications

The recommendation scheme of the intelligent tourism personalized recommendation method based on optimized UMISTM model in practical application is shown in Figure. 11. This method generated two completely different tourism routes based on user needs and personalized preferences. Scheme 1's 3-day journey arrangement included Zhengyang Road Pedestrian Street, Rongshan Lake, Bamboo Raft Tour of Lijiang River, Xingping Ancient Town, Shili Gallery, The Moon Hill, Big Banyan Tree, Moon River, The Silver Cave, Maling Ancient Village, Nanxi Mountain, Shanshui Jian, Sun Moon Twin Towers, and Elephant Trunk Mountain. Not only can tourists experience the local natural scenery, but also, they can also taste local specialties. The second plan was relatively simple, with itinerary arranged for Lijiang Scenic Area, Xingping Ancient Town, West Street, Guilin's Eternal Love, "Impression Liu Sanjie" performance, Yulong River Drifting, Ten Li Gallery, Yinziyan, Seven-Star Park, Dongxi Alley, Elephant Trunk Mountain, and Two Rivers and Four Lakes. This plan can also appreciate the local historical and cultural heritage. The user expressed satisfaction with both tourism planning solutions. By acquiring personalized preference and attraction vectors from users and inputting them into a feedforward neural network, it was possible to calculate the user's satisfaction with the corresponding attractions. This allowed attractions with high satisfaction ratings to be effectively recommended to users.

4.3 Discuss

In the context of the current vast tourism information network, the issue of how to provide tourist attractions that are easily accessible according to consumer needs represents a significant challenge that must be addressed in the process of China's tourism informatization development and industrial upgrading. Therefore, a UMISTM model was proposed to mine users' travel traces, and a more comprehensive user preference was obtained through the MGR framework, and finally an intelligent tourism personalized recommendation method was obtained.

Firstly, the performance of the optimized UMISTM model was analyzed. Among the change curves of recall rate, accuracy rate and K value of different models, the recall rate and accuracy rate of the optimized UMISTM model were the fastest growing rate. When K was 20, the corresponding recall rate and accuracy rate were 86.57% and 97.62%, respectively. The above results were generated because as the value of K increased, the list of recommendations corresponding to the recommendation model would increase, and the probability of recommending accurate attractions to users will increase. Therefore, the corresponding recall rate and accuracy

results could show an increasing trend, and the cryptic information contained in the research method could be richer and more complete. The performance change curves of each influencing factor showed that with the continuous expansion of the size of the training window, the optimized UMISTM model displayed a trend of first increasing at a large growth rate and then decreasing slowly, which indicated that when c was 4, the proposed model retained the most comprehensive semantic information. The performance results of each recommended model in the self-made dataset and the TMDB 5000 Movie dataset were as follows: Compared with the current most mainstream PTR-IFJ model, the model proposed in this study had an average increase of 10.3% and 8.9% in HR and NDCG. It indicated that the research method could comprehensively capture users' preferences and personalized preferences during travel.

Then, the two-stage optimization model with the best performance in related works and the model based on explicit emotion score and star rating were compared and tested, and the analysis was carried out from the three perspectives of computational efficiency, F1 value and resource utilization. The research method showed the best results in each performance index, and the computational efficiency, F1 value and resource utilization ratio were 11.24s, 97% and 91%, respectively. Among them, the indicators of PTR-IFJ recommendation model were 18.36s, 91% and 87%, respectively. Compared with them, the performance of the research method greatly improved in all aspects, which was because the research method was effective and deeply excavated different hidden meaning information in the route, and obtained more comprehensive personalized preferences of users under different granularity in the feature extraction part. Lightweight processing was introduced. In addition, the model recommended by PTR-IFJ was not computationally efficient due to insufficient local and global information exchange in the search process. Compared to the two-stage optimization model, the accuracy of both models improved to some extent. The research method was processed from the data mining part, lightweight design and feature extraction part, and also effectively improved the computational efficiency and accuracy of its application. However, the two-stage optimization model improved the accuracy and diversity of travel recommendation through different stages, and it did not pay attention to the computational efficiency, so its computational efficiency was low.

Finally, two different travel routes are generated according to user needs and personalized preferences in practical applications. Since the designed system is based on the user's personalized preference vector and scenic spot vector, which are input into the feedforward neural network to obtain the user's satisfaction with the corresponding scenic spot, the users are satisfied with the generated system results.

In summary, the research method can fully meet the individual needs of users, is conducive to promoting the high-quality development of tourism industry, and is an important assistant to promoting the green and sustainable development of tourism.

5 Conclusion

To solve the problem of poor recommendation quality in traditional recommendation methods, a UMISTM model was proposed to mine user travel trajectories, and then the MGR framework was used to optimize it. Finally, an intelligent personalized travel recommendation method was designed. The experimental results showed that the recall and accuracy growth rates of the optimized UMISTM model were the fastest, with corresponding recall and accuracy rates of 86.57% and 97.62% at K=20, respectively. In the accuracy results, the UMISTM model gradually surpassed the PTR-IFJ model with the increase of K value, ultimately reaching 95.16%. The optimized UMISTM model achieved the best results on both HR and NDCG datasets. Compared with the current mainstream PTR-IFJ model, the proposed model showed an average increase of 10.3% and 8.9% in HR and NDCG. Finally, in the application results, the study provided two tourism planning schemes based on the personalized needs of users, and users believed that both schemes could meet their own requirements. In summary, the proposed method has good performance and can fully meet the personalized needs of users in actual tourism recommendations. However, there are still shortcomings in the research. The research only assists users in providing personalized travel routes. However, during the travel process, users not only need to visit scenic spots, but also need to provide real-time activity information such as restaurants, experience shops, and accommodation. Therefore, in future research, user preference information during travel can be further divided to provide better services to users.

Reference

[1] Hasan R, Koles B, Zaman M, Paul J. The potential of chatbots in travel and tourism services in the context of social distancing. International Journal of Technology Intelligence and Planning, 13(1):63-83, 2021. https://doi.org/10.1504/ijtip.2021.10041470

[2] Beldona S, Kher H V, Lin K. Gains-focused vs risk-averse orientations and their impact on location-based marketing services in tourism. Journal of Hospitality and Tourism Technology, 13(2):333-347, 2022. https://doi.org/10.1108/jhtt-07-2021-0209

[3] Shen L. Analysis of Immersive Virtual Reality Tourism Resources Based on Navier-Stokes Equations. Informatica, 48(6):131-140, 2024. https://doi.org/10.31449/inf.v48i6.5514

[4] Chefrour A, Drissi S. K-CAE: Image Classification Using Convolutional AutoEncoder Pre-Training and K-means Clustering. Informatica (Slovenia), 47, 2023. https://doi.org/10.31449/inf.v47i7.4499

[5] G Mehdi, H Hooman, Y Liu, S Peyman and R. Arif Data Mining Techniques for Web Mining: A Survey. Artificial Intelligence and Applications, 1(1):3-10, 2022. https://doi.org/10.47852/bonviewAIA2202290

[6] Jiang H, Wan C, Yang K, Ding Y, Xue S. Continuous missing data imputation with incomplete dataset by generative adversarial networks–based unsupervised

learning for long-term bridge health monitoring. Structural Health Monitoring, 21(3):1093-1109, 2022. https://doi.org/10.1177/14759217211021942

[7] Obert J, Loffredo T. Efficient Binary Static Code Data Flow Analysis Using Unsupervised Learning. International Journal of Semantic Computing, 16(4):569-583, 2022. https://doi.org/10.2172/1592974

[8] Lin Y, Huang C, Yao W, Shao Y. Personalised attraction recommendation for enhancing topic diversity and accuracy. Journal of Information Science, 49(2):302-318, 2023. https://doi.org/10.1177/0165551521999801

[9] Nitu P, Coelho J, Madiraju P. Improvising personalized travel recommendation system with recency effects. Big Data Mining and Analytics, 4(3):139-154, 2021. https://doi.org/10.26599/bdma.2020.9020026

[10] Shah C, Trupp A, Stephenson M L. Conceptualising local perceptions of research-related tourism in an indigenous village in Fiji. International Journal of Tourism Research, 25(4):416-428, 2023. https://doi.org/10.1002/jtr.2578

[11] Fudholi D H, Rani S, Arifin D M, Satyatama M R. Deep Learning-based Mobile Tourism Recommender System. Scientific Journal of Informatics, 8(1):111-118, 2021. https://doi.org/10.15294/sji.v8i1.29262

[12] Li L, Liu Y. Mapping Modern JVM Language Code to Analysis-Friendly Graphs: A Study with Kotlin. International journal of software engineering and knowledge engineering, 32(11/12):1667-1688, 2022. https://doi.org/10.1142/s0218194022500735

[13] Duan L, Gao T, Ni W, Wang W. A hybrid intelligent service recommendation by latent semantics and explicit ratings. International Journal of Intelligent Systems, 36(12):7867-7894, 2021. https://doi.org/10.1002/int.22612

[14] Zhang D, Zhang Y, Li Q, Wang D. Sparse User Check-in Venue Prediction by Exploring Latent Decision Contexts From Location-Based Social Networks. IEEE transactions on big data, 7(5):859-872, 2021. https://doi.org/10.1109/tbdata.2019.2957118

[15] Yao Z, Yu J, Ding J. Contrastive learning of graph encoder for accelerating pedestrian trajectory prediction training. IET image processing, 15(14):3645-3660, 2021. https://doi.org/10.1049/ipr2.12185

[16] Karami F, Malek M R. Trajectory similarity measurement: An enhanced maximal travel match method. Transactions in GIS, 25(3):1485-1503, 2021. https://doi.org/10.1111/tgis.12733

[17] Kim Y K, Choi W C. Effect of change of sand properties on travel distance of ricocheted debris. Defence Technology, 17(4):1486-1495, 2021. https://doi.org/10.1016/j.dt.2020.08.006

[18] Tian L. Analysis of Media Content Recommendation in The New Media Era Considering Scenario Clustering Algorithm. Informatica, 48(6):141-156, 2024. https://doi.org/10.31449/inf.v48i6.5375

[19] An Q, Maggioni M, Kevrekidis Y, Lu F. Unsupervised learning of observation functions in state space models by nonparametric moment methods. Foundations of Data Science, 5(3):340-365, 2023. https://doi.org/10.3934/fods.2023002

[20] Li L, Li X, Yang S, Ding S, Zheng X. Unsupervised-Learning-Based Continuous Depth and Motion Estimation with Monocular Endoscopy for Virtual Reality Minimally Invasive Surgery. IEEE transactions on industrial informatics, 17(6):3920-3928, 2021. https://doi.org/10.1109/tii.2020.3011067

Design of Online Monitoring Method for Distribution IoT Devices Based on DBSCAN Optimization Algorithm

Chaofan Hou*, Nan Xu, Siyu Liu
State Grid Beijing Electric Power Company Information and Communication Branch, Beijing, China
E-mail: 18612330569@163.com, xunan2024@126.com, julyand131@163.com
*Corresponding author

Keywords: distribution internet of things, clustering algorithm, abnormal data detection, time correlation, parameter adaptation

Received: June 14, 2024

In response to the data mutation problem caused by equipment failures in the distribution Internet of Things, this study proposes a density-based clustering optimization algorithm for online monitoring of equipment data anomalies. This method considers the local and global similarity of high-dimensional measurement data and constructs a composite time series similarity measurement criterion. Improvements are made to the density-based clustering algorithm, which combines with preprocessed device historical measurement data to adaptively generate global density parameters. Through clustering training, core data points are obtained to detect abnormal changes in data. The experiment showed that compared to traditional density-based clustering algorithms, the improved algorithm had good clustering performance, with standardized mutual information and adjusted mutual information increased by about 2%. Compared to anomaly detection algorithms, the density-based clustering optimization algorithm for anomaly detection of equipment data in the distribution Internet of Things increased the detection rate by 38% and reduced the false detection rate by 65%. Therefore, the proposed online monitoring method for data anomalies can improve the data detection rate and has high practical value for the reliable operation of distribution Internet of Things systems.

Povzetek: Prispevek predstavlja izboljšano metodo za zaznavanje anomalij podatkov v distribucijskih IoT napravah z optimiziranim algoritmom DBSCAN, ki povečuje natančnost zaznavanja in zmanjšuje lažne alarme.

1 Introduction

With the continuous development of the national economy, the scale of the power grid has grown rapidly, and the stable operation of the power grid directly affects economic development and social production construction. “Smart grid” is a new type of power network that integrates modern sensing and measurement technology, information technology, communication technology, and computer control technology, achieving digitalization and intelligence in power generation and transmission [1-3]. To achieve the interconnection of all things and human-machine interaction in various aspects of the power system, the ubiquitous power Internet of Things (IoT) is further designed built on the smart grid, and the important link in its construction is the Distribution Internet of Things (D-IoT). Lingda et al. proposed a trust evaluation method based on information entropy to address the internal network attack threat of D-IoT terminals. It used an exponential distribution reputation model to estimate direct trust values, and combined entropy theory to compensate for the inaccuracy of direct trust judgments. This method could effectively resist switching attacks and collusion attacks [4]. To improve the accuracy of fault location for feeder terminal units in D-IoT, Xu et al. proposed a fault location algorithm based on the Genetic Algorithm (GA) and improved the unified matrix algorithm, which combines GA to achieve fault location. This algorithm could effectively obtain the localization results of error information [5]. Huang et al. proposed a cloud edge collaborative processing method based on an attention mechanism to solve the delay problem in data calculation of D-IoT edge devices. They combined edge computing and cloud computing to solve the timeliness of data analysis and calculation. This algorithm could achieve efficient utilization of electricity resources and accelerate data iteration [6].

D-IoT collects and uploads information through numerous Low-voltage Terminal Units (LTUs) to achieve fault detection and safe maintenance of distribution lines. LTU may cause Abnormal Data (AD) upload due to communication module failure or battery depletion, which can be detected through device self-inspection or main station monitoring. However, when there is a malfunction in the LTU sampling circuit or data anomalies caused by abnormal storage modules, the AD

is difficult to detect. The false alarm of abnormal LTU can have an impact on the reliability of solving system faults. Therefore, LTU measurement Data Abnormal Detection (DAD) is one of the important links to improve the reliable operation of D-IoT systems [7-9]. DAD is a research hotspot in data mining, and in recent years, experts and scholars have conducted relevant research on DAD. Lei et al. designed an unsupervised time series DAD framework based on spectral analysis to detect abnormal samples in time series. This framework utilized spectral analysis to decompose time series and combined neural networks to predict trend and seasonal sequences. This method has been proven to effectively improve the detection rate of AD [10]. Gan et al. introduced utility-aware anomaly sequence rules into anomaly detection methods, combined with pruning strategies, to explore the upper bound of utility-aware anomaly sequence rules for anomaly value detection. This method had good effectiveness and scalability [11]. Jeong et al. proposed local outlier probability to identify seismic noise in the context of unsupervised anomaly detection, and utilized binary decision-making combined with moving windows to reduce the energy loss of useful signals. This method had good feasibility in denoising applications [12]. Yang et al. proposed a multivariate regression DAD and restoration based on spatio-temporal correlation to address the issue of insufficient feature extraction ability in drone flight data. It used correlation analysis methods to select parameters and designed a multiple regression model that integrates attention mechanisms to improve DAD performance, which could achieve data recovery [13].

At present, research on D-IoT mainly focuses on application solutions and system architecture design. The amount of data collected by D-IoT LTU nodes is large, and the spatio-temporal complexity of the above DAD methods is low, which has certain limitations for high-dimensional D-IoT LTU DAD. This makes the scalability of the detection method weaker and the practical application range smaller. At the same time, the above methods still lack adaptability research involving datasets. Therefore, this study proposes an improved clustering algorithm-based D-IoT LTU node online monitoring method. This study innovatively adopts a classification method based on global density parameters, expanding the adaptability

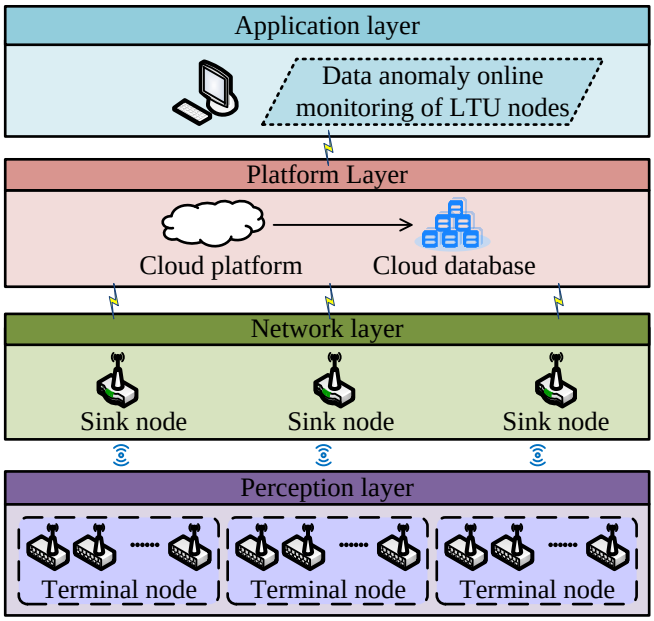


Figure 1: Architecture of D-IoT

of clustering algorithms to datasets. The research objective is to improve the accuracy of LTU node DAD and provide data foundation and technical reference for the stable operation and system management level of future D-IoT systems.

2 Methods and materials

This study proposes a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for adaptive generation of global density parameters, which optimizes the global density parameters of DBSCAN through the Composite Time Series Similarity Measure Criterion (CTSSMC). On the basis of optimizing the DBSCAN and combining the time correlation of D-IoT LTU Measurement Data (LTU-MD), a D-IoT LTU node data anomaly online monitoring method is proposed to detect LTU nodes with abnormal changes in data.

2.1 DBSCAN algorithm for adaptive generation of global density parameters

Cluster analysis refers to the classification of data points in a dataset according to a specific metric, where the differences between data points in the same class are minimized and the similarities between data points in different classes are minimized [14-16]. D-IoT data have continuity and similarity, and clustering analysis can distinguish between normal data and mutation data. Figure 1 shows the architecture of D-IoT, which includes the perception layer, network layer, platform layer, and application layer. The perception layer is the end of the D-IoT, and a large amount of data information collected by LTU nodes is transmitted to the application layer for detecting faults and monitoring loads. The data collected by LTU is time-series data with timestamps, so Time Series Similarity Measurement (TSSM) is very important for the clustering analysis of D-IoT data.

The existing TSSM methods can be divided into elasticity metric and lock step metric based on the time series comparison method [17-18]. Due to the temporal nature of D-IoT data, timing is crucial for TSSM. Therefore, a lock-step metric is used to analyze the similarity of D-IoT data. The distribution of each component in the sequence can lead to significant differences in measurement results [19]. Therefore, this study considers the distribution of each component in the sequence and proposes CTSSMC to analyze the global and local similarity of LTU-MD sequences. The measurement time series of two LTU nodes in D-IoT at a certain time is shown in equation (1).

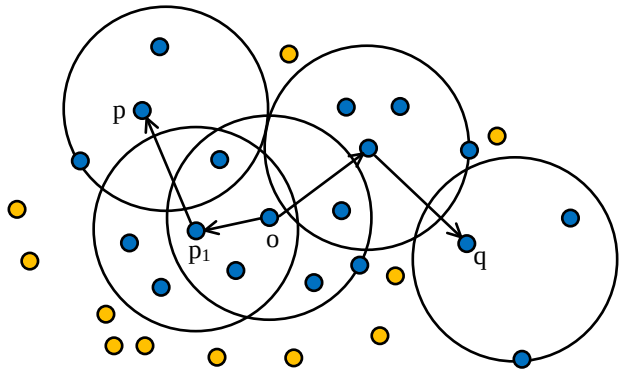


Figure 2: The relationship between data points

$$\begin{cases} X_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,t}\} \\ X_j = \{x_{j,1}, x_{j,2}, \cdots, x_{j,t}\} \end{cases} \quad (1)$$

In equation (1), X_i and X_j are LTU nodes. CTSSMC comprehensively considers the Jensen-Shannon (JS) divergence distance, Euclidean distance, and D-IoT error mode distance to measure the similarity of probability distributions, as shown in equation (2).

$$D_{ij} = dist(X_i, X_j) = \frac{1}{4} D_{jsd} + \frac{1}{2} D_{edc} + \frac{1}{4} D_{mis} \quad (2)$$

In equation (2), D_{ij}, D_{jsd}, and D_{edc} respectively represent the similarity between two sequences, the probability distributions between sequences, and the sequence amplitude. D_{mis} is the error mode distance of D-IoT. D_{jsd} is calculated based on the JS divergence, as shown in equation (3).

$$\begin{cases} D_{jsd} = \frac{1}{2} D_{kl}\left(P_i \| M\right) + \frac{1}{2} D_{kl}\left(P_j \| M\right) \\ M = \frac{P_i + P_j}{2} \end{cases} \quad (3)$$

In equation (3), $D_{kl}(\square)$ is the KL divergence. $P_i = p_i(f)$ and $P_j = p_j(f)$ represent the probability density function distribution of sensor data flow based on samples. M represents the average probability density function distribution of P_i and P_j based on samples. The calculation of $D_{kl}(\square)$ is equation (4).

$$D_{kl}\left(P_i \| P_j\right) = \sum_i p_i(f) \ln \frac{p_i(f)}{p_j(f)} \quad (4)$$

D_{edc} is calculated based on Euclidean distance, as shown in equation (5).

$$D_{edc} = \left\| X_i - X_j \right\| = \left(\sum_{n=1}^{t} \left(x_{i,n} - x_{j,n} \right)^2 \right)^{\frac{1}{2}} \quad (5)$$

The error modes of LTU-MD include erroneous measurement values with collected data streams less than zero and missing data, calculated as shown in equation (6).

$$D_{mis} = \begin{cases} 0, if\ \varphi_i = \varphi_j, \varphi_i = \begin{cases} 0, if\ \exists x \in \{1, \cdots t\},\ f_{i,x} < 0 \vee f_{i,x} = NaN \\ 1, other \end{cases} \\ 1, other \end{cases} \quad (6)$$

The JS divergence and Euclidean distance are calculated using preprocessed LTU-MD, while the error mode distance is obtained by unfilled and uncleaned LTU-MD. This study utilizes DBSCAN's global density parameters to classify the data in the dataset [20-21]. The relationship between data points in dataset D is shown in Figure 2.

The density p1 can be directly reached from data point o, the density q can be reached from data point o, and the density p can be connected to data point q. The DBSCAN algorithm classifies data points based on global density parameters and can detect noisy data. Therefore, the DBSCAN is often used in DAD. Global density parameters contain the Neighborhood Radius (NR) and Neighborhood Density Threshold (NDT) of the data, and the NR calculation is equation (7).

$$N_{Eps}(p) = \left\{ q \in D \middle| dist(p,q) \le Eps \right\} \quad (7)$$

In equation (7), $N_{Eps}(p)$ is the dataset, Eps represents NR, and $dist(p,q)$ is the distance measurement criterion between p and q.

The selection of global density parameters directly affects the clustering results, and inappropriate global density parameters can lead to false positives and missed detection in DBSCAN during DAD. In response to the shortcomings of DBSCAN, this study proposes the DBSCAN algorithm of Adaptive Generating Global Density Parameters (AGDP-DBSCAN). This algorithm distinguishes between normal and abnormal measurement data based on the distribution of AD in D-IoT LTU measurement data, combined with analysis of the distance distribution of LTU historical data points. Based on the discrete distribution characteristics of AD, NR and NDT are determined. The distance matrix of $n \times n$ is obtained from the measurement dataset with a data volume of n, calculated by equation (2). The distance curve in the distance matrix refers to the distance between each data point and other data points. Four distance curves are selected for polynomial fitting, and the fitted curves are shown in equation (8).

$$dist_i(x) = p_i x^4 + q_i x^3 + r_i x^2 + s_i x + t_i \quad (8)$$

In equation (8), $dist_i(x)$ is the i-th distance curve, x is the data point number. p_i, q_i, r_i, s_i, and t_i denote the fitting curve parameters. The second-order derivative of the fitted curve $dist_i''(x)$ is equation (9).

$$dist_i(x) = p_i x^4 + q_i x^3 + r_i x^2 + s_i x + t_i \quad (9)$$

In equation (9), assuming $dist_i''(x) = 0$, and the root of the equation is found.

The maximum in the root serves as the value of the inflection point position x_i, and the NR of the data is taken as the mean of the experimental evaluation curve, as shown in equation (10).

$$Eps = \frac{1}{n} \sum dist_i(x_i) \quad (10)$$

In equation (10), the derivative of the fitted curve is taken to obtain $dist_i'(x)$, and substituted into equation (8) to obtain x_i'. The specific calculation is equation (11).

$$dist_i(x_i') = Eps \quad (11)$$

In equation (11), x_i' is the minimum number of sample points for the i-th evaluation curve. NDT is the average of the minimum points on all curves, calculated as expressed in equation (12).

$$MinPts = \frac{1}{n} \sum x_i' \quad (12)$$

In equation (12), $MinPts$ is NDT.

2.2 D-IoT LTU node data anomaly online monitoring method

Based on the Temporal Correlation (TC) of D-IoT measurement data and combined with the AGDP-DBSCAN algorithm mentioned above, this study designs a D-IoT LTU node data anomaly online monitoring method. In D-IoT, the TC of LTU-MD nodes refers to the functional connection between data continuity

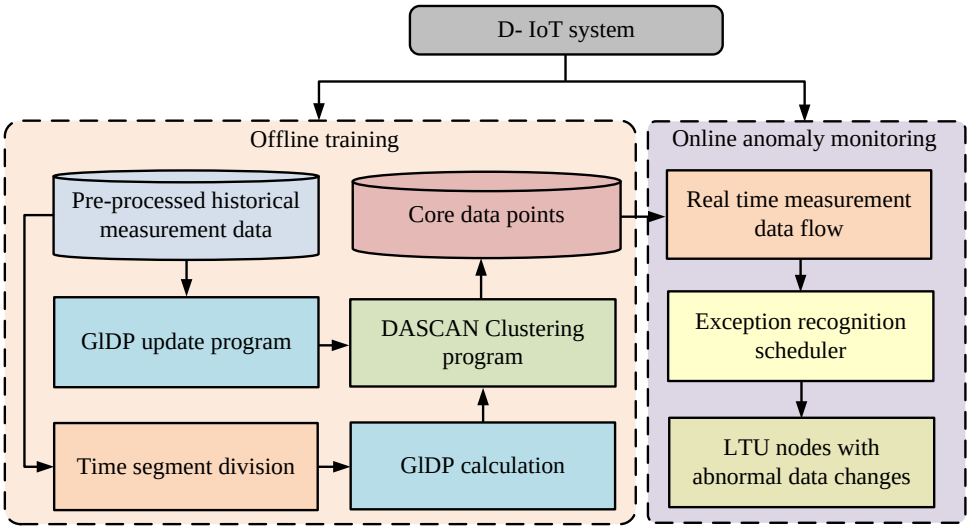


Figure 3: Data anomaly monitor program based on AGDP-DBSCAN

and similarity. The variation pattern of LTU-MD can be obtained through the TC of measurement data, mining historical data information, extracting measurement data change characteristics, and improving the efficiency of online monitoring of LTU node data anomalies. In traditional clustering data anomaly monitoring, the overall temporal data is used as the input for clustering analysis. When outliers appear in the clustering results, it is determined that the data of the LTU node has undergone abnormal changes. The data dimension of D-IoT is high. Directly inputting the overall data will increase the calculation time, and directly clustering high-dimensional data will cause dimension disasters, leading to a decrease in monitoring accuracy. When measurement data from new nodes is added, it is necessary to perform cluster analysis on the overall data again. Therefore, based on the TC of the LTU-MD of D-IoT, this study divides the measurement data into time segments with a length of L. This segment is further divided into sub time segments of l length. Each sub segment constitutes a dimension of the time segment, that is, the data dimension of the time segment is L/l. According to equation (2), the distance of input data points for LTU node clustering analysis is equation (13).

$$D_{m,n} = dist\left(F_m, F_n\right) \tag{13}$$

In equation (13), $D_{m,n}$ is the distance between the data points input for clustering analysis. F_m and F_n represent time segments.

All measurement data are divided into time segments based on the time series, and the dataset for clustering analysis input is calculated according to equation (13). Figure 3 shows the data anomaly monitor program framework for D-IoT LTU nodes based on the AGDP-DBSCAN algorithm. This framework consists of offline training and online anomaly monitoring. Offline training is based on the historical LTU-MD nodes to gain the global density parameters of Core Data Points (CDPs) and DBSCAN clustering. Online anomaly monitoring utilizes training results to detect measurement data.

In offline training, after preprocessing the historical LTU-MD nodes, time segments are divided and used as the training dataset. Each LTU node is trained to obtain CDPs that represent the changes in LTU-MD nodes from

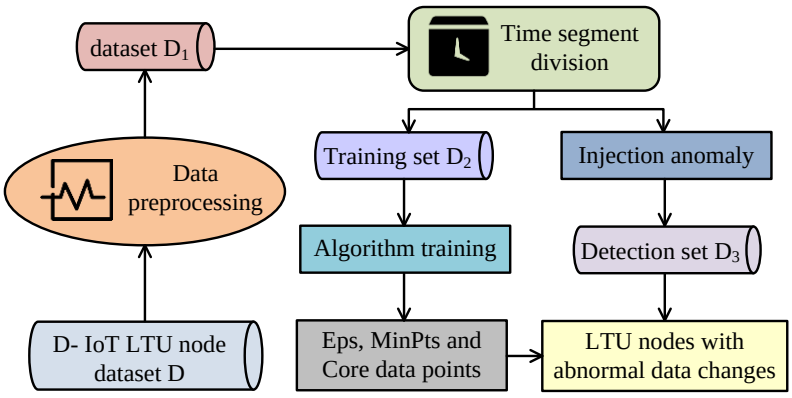


Figure 4: Data anomaly detection process

the training results, forming a CDP set. In online anomaly monitoring, when LTU measures new time period data (i.e. new data points, NDP), equation (2) is used to calculate the distance between NDP and all CDPs obtained from offline training to determine whether NDP is within the neighborhood range of the core point. The NDP is within the neighborhood range and represents normal measurement data. The NDP is not within the neighborhood range, indicating abnormal changes in the LTU node data. The randomly selected LTU-MD may not surely contain all types of AD. This study incorporates artificially simulated AD to obtain a dataset of abnormal nodes with D-IoT data characteristics. There are 5 types of simulated AD, including constant anomaly, drift anomaly, bias anomaly, shock anomaly, and periodic anomaly. The error function of drift anomaly is equation (14).

$$e = k\left(t - t_0\right) \tag{14}$$

In equation (14), e is the error function, k is the gain factor, and t_0 is the time when an anomaly occurs. The error function of periodic anomalies is equation (15).

$$e = a_0 + \sum_{n=0}^{\infty} a_n \sin\left(n\omega t + \varphi_n\right) \tag{15}$$

In equation (15), a_0, a_n, and φ_n are all constants, and $n = 1, 2, 3, \cdots$. 790 LTU terminal nodes in the D-IoT system of a certain university are selected, using a portion of the selected LTU-MD as dataset D for detection. Dataset D is subjected to DAD by the aforementioned data anomaly monitoring program, and the DAD process is shown in Figure 4.

Dataset D is preprocessed to obtain dataset D_1. The TC of data is analyzed, and time segments are divided. The training set D_2 is selected from D_1, the remaining data of D_1 are injected into the labeled artificially simulated abnormal data to obtain the detection dataset D_3, and finally the D_3 is input into D_2 for DAD.

3 Results

This study verified the superiority of the proposed AGDP-DBSCAN algorithm by comparing the clustering results and dataset evaluation indicators of different algorithms. At the same time, the time correlation, recall rate, and F1-Score of AD in D-IoT LTU nodes based on AGDP-DBSCAN algorithm were analyzed.

3.1 Experiment and analysis of parameter selection model for DBSCAN optimization

To test the performance of the AGDP-DBSCAN algorithm, artificial datasets Aggregation and Compound were selected in

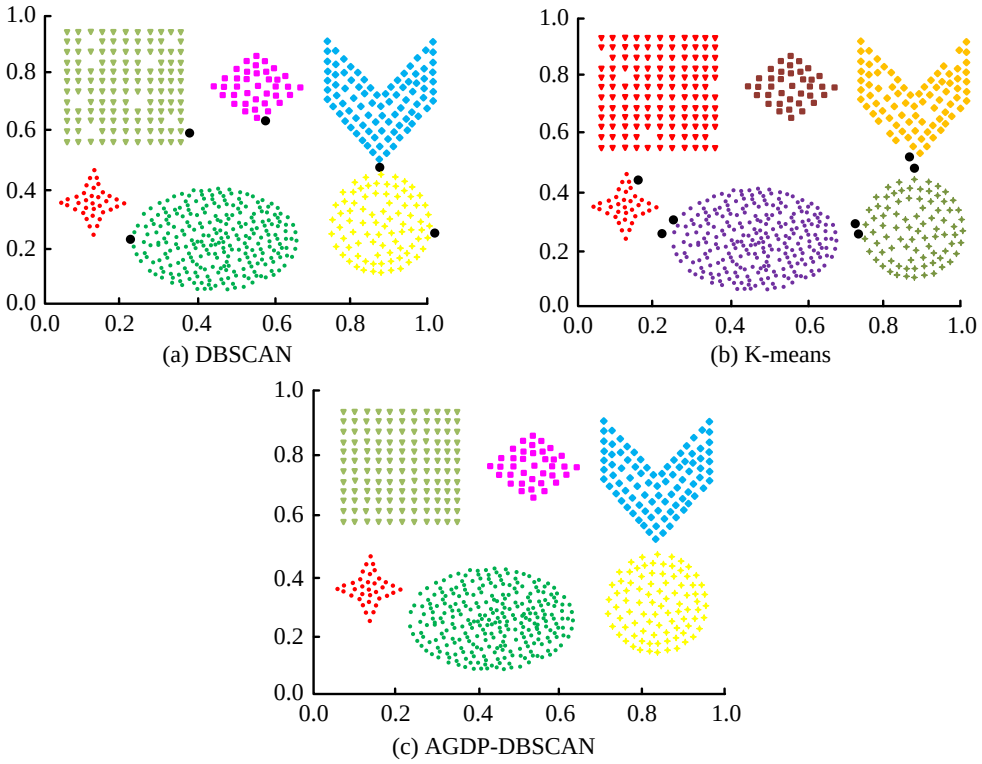


Figure 5: Clustering results on the Aggregation dataset

the experiment. The DBSCAN algorithm, partition-based K-means algorithm (K-means), and AGDP-DBSCAN algorithm were compared to verify the effectiveness of the AGDP-DBSCAN algorithm. The density distribution of data samples in Aggregation was relatively uniform, with connected data points between clusters. Figure 5 shows the clustering results of different algorithms on Aggregation.

In Figure 5 (a), DBSCAN performed better in clustering when the sample distribution density was uniform. In Figure 5 (b), there were a few clusters with connected samples, and the K-means algorithm performed well. In Figure 5 (c), AGDP-DBSCAN performed similarly to the first two clustering algorithms on samples with uniform distribution density, indicating that for samples with uniform distribution density in the dataset, the clustering effects of different clustering algorithms were basically the same. To further demonstrate the performance of AGDP-DBSCAN, experiments were conducted on datasets with uneven sample density distribution. The density distribution of data samples in Compound was uneven, and there were clusters with different densities. Figure 6 is the clustering results of various algorithms on Compound.

In Figure 6 (a), DBSCAN treated the data in the upper left corner of the dataset as noise points and did not achieve complete clustering of the data. In Figure 6 (b), the K-means also treated the data in the upper left corner as noise points and divided the cluster in the lower left corner into multiple small clusters. In Figure 6 (c), the AGDP-DBSCAN algorithm correctly identified two clusters and had good adaptability to datasets with uneven density. Due to the different densities of the two datasets, it directly affected the clustering

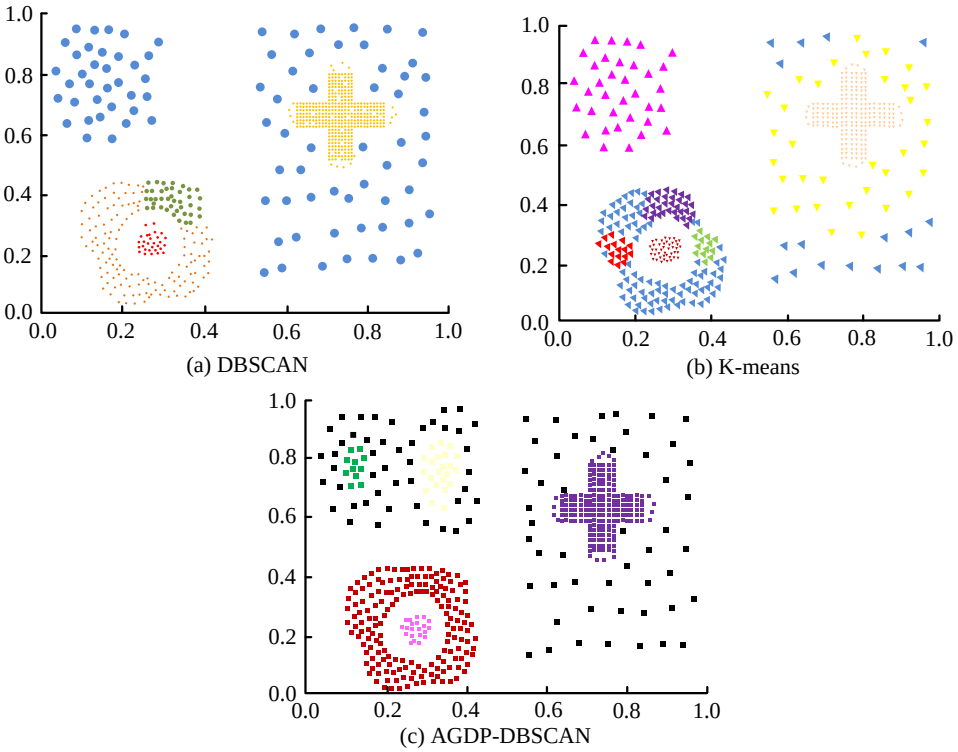
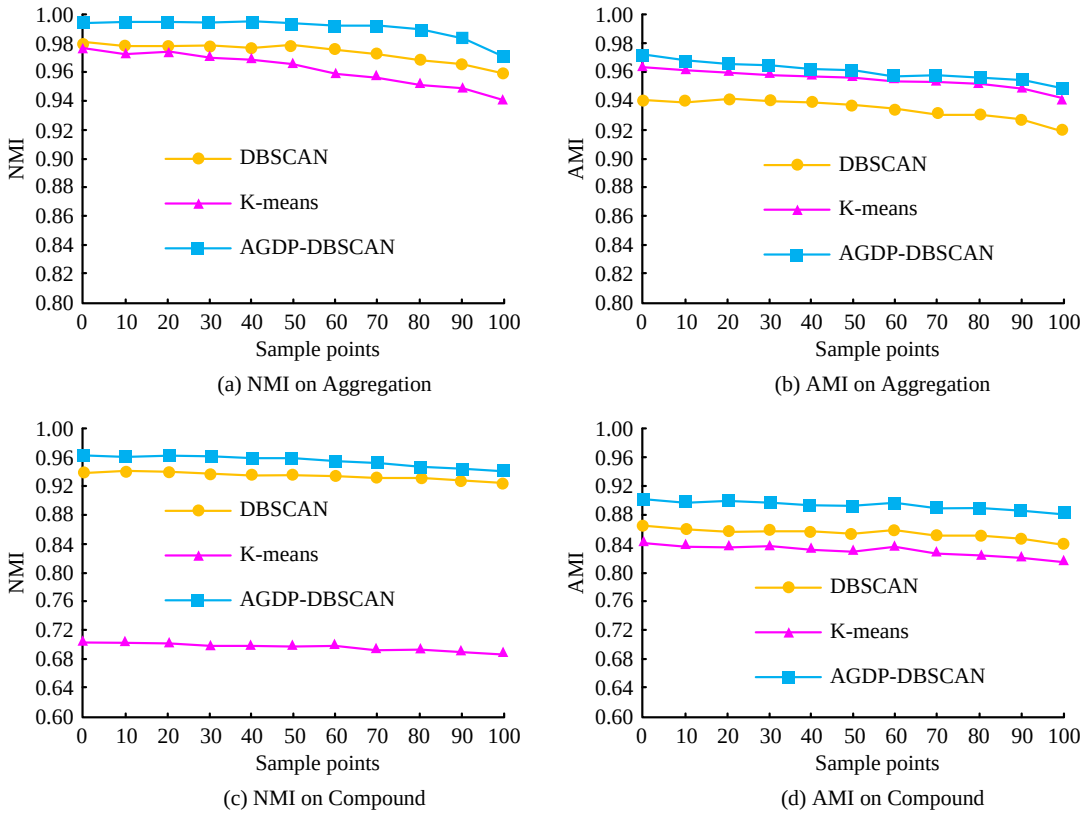


Figure 6: Clustering results on the compound dataset

performance of anomaly detection algorithms. Research on the use of similar duplicate record detection methods could optimize the DBSCAN algorithm and improve its clustering performance in different datasets. The experiment used Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI) to test the effectiveness of different clustering algorithms. Figure 7 shows the clustering performance of different clustering algorithms on artificial datasets.

In Figure 7 (a), the average NMI of DBSCAN, K-means, and AGDP-DBSCAN in the Aggregation were 0.974, 0.960, and 0.993, respectively. Compared to DBSCAN and K-means, the NMI of AGDP-DBSCAN increased by 2% and 3.4%, indicating that AGDP-DBSCAN had the best clustering effect. In Figure 7 (b), the average AMI of the three algorithms were 0.935, 0.948, and 0.955. The AMI of AGDP-DBSCAN was closest to 1, indicating that its clustering results were more consistent with the real clustering results. This was because DBSCAN randomly selected initial points and artificially sets NR and density thresholds, resulting in poor adaptability. K-means required an initial cluster center to determine the initial partition, continuously calculating and adjusting the cluster centers. As the amount of data increased, the clustering effect gradually deteriorated. In Figure 7 (c), the average NMI of DBSCAN, K-means, and AGDP-DBSCAN in Compound were 0.933, 0.695, and 0.952, respectively. The NMI of AGDP-DBSCAN has

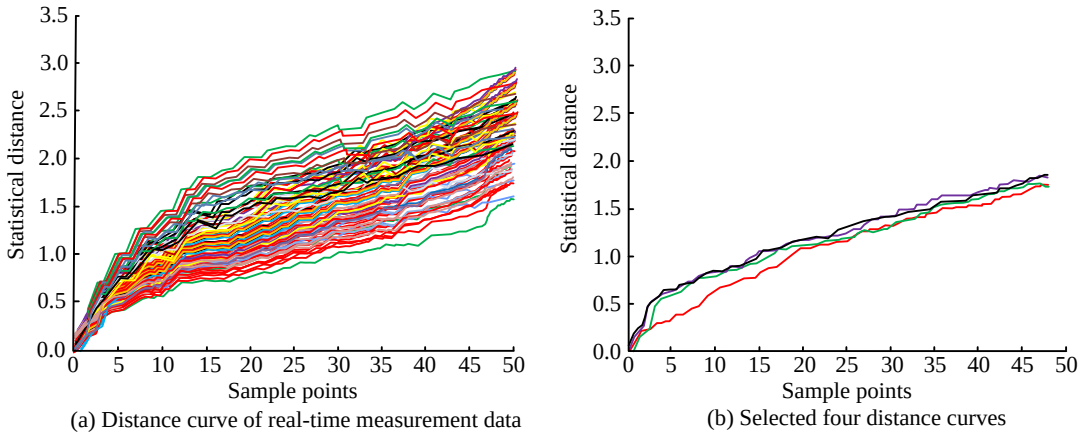
improved by 2% and 37% compared to the other two algorithms. In Figure 7 (b), the average AMI of each algorithm was 0.845, 0.828, and 0.898, respectively. AGDP-DBSCAN had the highest AMI, indicating its best clustering performance. This algorithm used CTSSMC to calculate distance



(a) NMI on Aggregation (b) AMI on Aggregation

(c) NMI on Compound (d) AMI on Compound

Figure 7: Clustering effect on the different dataset



(a) Distance curve of real-time measurement data (b) Selected four distance curves

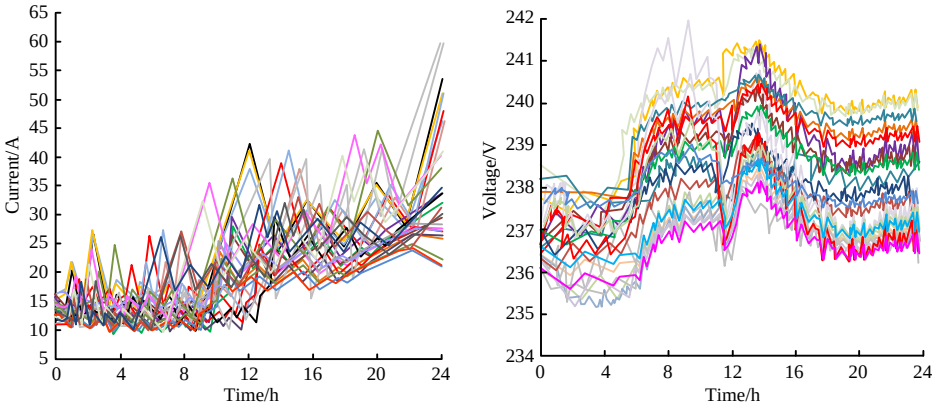
Figure 8: Distance curve chart

and adaptively generated NR and NDT based on LTU historical measurement data, which had good adaptability. Therefore, parameter optimization and dataset annotation were achieved through clustering analysis of datasets with different densities and artificial datasets, as well as adaptive neighborhood density methods.

Moreover, the results of the evaluation indicators also reflected the excellent clustering performance of the optimization algorithm on real datasets. Multiple experiments have demonstrated the robustness and superiority of the AGDP-DBSCAN algorithm.

3.2 Experimental and analysis of online monitoring of AD in D-IoT LTU Nodes

To test the performance of online monitoring



(a) Daily current timing chart (b) Daily Voltage timing chart

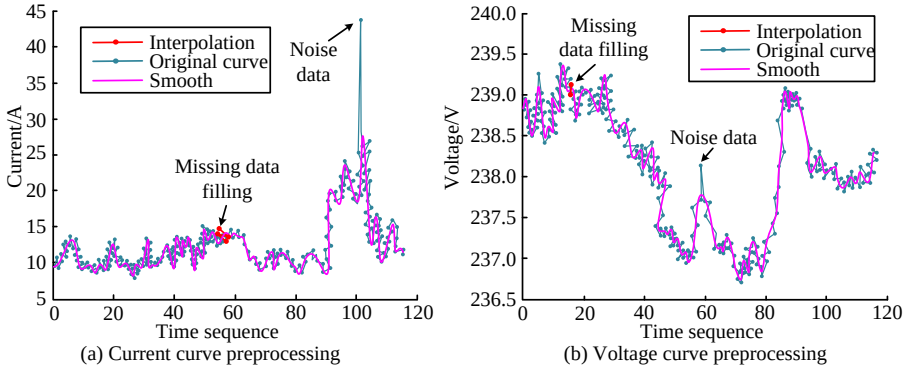
Figure 9: The variation pattern of daily LTU-MD nodes

of AD in D-IoT LTU nodes, real-time LTU-MD in a certain university's D-IoT was selected in the experiment. Figure 8 shows the distance curve drawn from the real-time LTU-MD.

In Figure 8 (a), in the real-time LTU-MD, the distance between normal and cluster data points was small and stable, showing a smooth front end of the distance curve. The proportion of AD points in the real-time entire LTU-MD was relatively low, and the distance from normal data points was relatively large, manifested as a steep distance from the end of the curve. In Figure 8 (b), at the position of sample point $n/2$, the location of the steep point was calculated by selecting four distance curves in ascending order of distance. The distance between steep points was used to calculate Eps, and $MinPts$ was calculated based on the amount of data points at the steep point. For testing the TC of D-IoT data, the daily variation pattern of LTU-MD nodes after normal operation of the system is shown in Figure 9.

Observing Figure 9 (a), when the system operated normally once, the daily current variation range was between 10A and 60A. The variation pattern of LTU node measured current over time showed similarity, and the same variation pattern was also observed for different periods, indicating that the current data measured by LTU had periodicity. In Figure 9 (b), after a normal operation of the system, the daily voltage changed over time, with a range of 235V to 242V. In addition, the daily voltage LTU-MD nodes exhibited significant similarity. Due to external signal interference, LTU-MD might experience data loss during measurement, transmission, and storage processes. This study used Z-Score standardization to standardize data and clean measurement data. Linear interpolation was used to fill in the cross-sectional data at a certain time, combined with Gaussian filtering for noise reduction and smoothing processing. The data preprocessing results of LTU nodes for one day are displayed in Figure 10.

In Figure 10 (a), the average values of the original current curve and the preprocessed current curve were 22.3A and 14.5A, respectively. Compared with the original current curve, the preprocessed current was reduced by about 35%. This was because the LTU current measurement data were interpolated and smoothed to remove the influence of noise, fill in missing data, and make the current curve relatively flat and smooth, ensuring the validity and



(a) Current curve preprocessing (b) Voltage curve preprocessing

Figure 10: Data pre-processing results

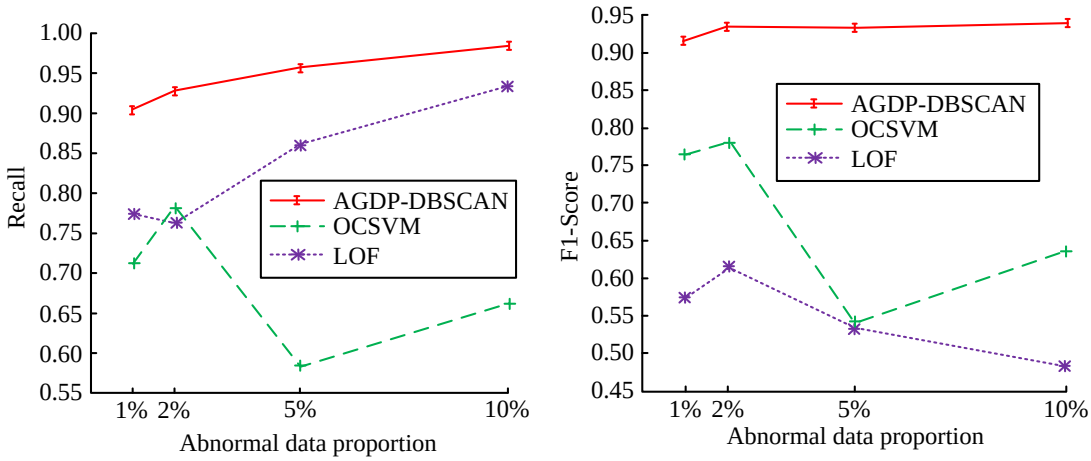
Table 1: Datasets with different proportions of anomalies

Datasets	Train sample days	Train sample samples	Test sample days	Test samples	Abnormal proportion
D3-I	25	8650	12	4030	1%
D3-II	25	8650	12	4030	2%
D3-III	25	8650	12	4030	5%
D3-IV	25	8650	12	4030	10%

completeness of the data. In Figure 10 (b), the average voltage before and after pretreatment was 238.8V and 237.5V, respectively. Compared to untreated, the voltage after pretreatment was reduced by about 0.5%. This was because noise had a relatively small impact on voltage. After filling in missing data and removing noise, the measured voltage data remained complete, making it easier for subsequent data analysis. To further test the superiority of AGDP-DBSCAN's D-IoT device online monitoring method, the experiment added artificially simulated LTU anomaly data with different proportions to the detection set D3, and obtained detection sets with different proportions of anomalies. Table 1 shows the dataset with different proportions of anomalies.

The experiment compared three DAD algorithms of AGDP-DBSCAN algorithm's D-IoT device online monitoring method (research method), One Class Support Vector Method (OCSVM), and Local Outlier Factor (LOF) algorithm. The comparison results of recall and F1-Score on datasets with different proportions of anomalies are shown in Figure 11.

In Figure 11 (a), the average recall rates of the research method, OCSVM, and LOF of the AGDP-DBSCAN algorithm were 0.94, 0.68, and 0.83, respectively. Compared with OCSVM algorithm and LOF, research method has improved recall rates by 38% and 13%, respectively, indicating a high detection rate for AD. The research method utilized time correlation and incorporated historical measurement data to generate distance curves during algorithm training, which could more precisely differentiate between normal data and AD. In Figure 11 (b), the mean F1-Score of the three algorithms was 0.92, 0.68, and 0.55, respectively. The F1-Score of oresearch method has improved by 35% and 67% compared to the other two algorithms. This was because the training dataset of OCSVM did not contain all kinds of AD, causing incomplete extracted AD features and a high false detection rate of the algorithm. The uncertainty of the distribution of current and voltage measurement data in LOF affected the



(a) Recall of different algorithms

(b) F1-Score of different algorithms

Figure 11: Recall rate and F1-Score of data with different proportions of anomalies

Table 2: Scalability verification results of different algorithms

Performance	Algorithms	Dataset size				
		2×104	4×104	6×104	8×104	10×104
Precision	LOF	0.64	0.70	0.75	0.83	0.85
	OCSVM	0.76	0.84	0.87	0.92	0.95
	K-Nearest Neighbor	0.73	0.84	0.86	0.91	0.94
	AGDP-DBSCAN	0.96	0.97	0.97	0.98	0.98
Running time/s	LOF	8s	33s	50s	64s	100s
	OCSVM	5s	9s	40s	58s	94s
	K-Nearest Neighbor	9s	34s	52s	67s	100s
	AGDP-DBSCAN	3s	5s	4s	5s	4s

calculation of LOF values, resulting in the highest false detection rate, especially as the proportion of anomalies increased, the false detection rate increased. The research method achieved high recall and F1-Score during DAD, and could accurately detect LTU-MD with abnormal changes.

To verify the scalability of the anomaly data detection algorithm, different algorithms were applied to different sizes of the same dataset, and their performance was compared. The results are shown in Table 2.

From Table 2, the running time of LOF algorithm and K-Nearest Neighbor algorithm varied greatly, and the time consumption increased with the increase of the dataset. The highest accuracy of the LOF algorithm was 0.85, while the highest accuracy of the K-Nearest Neighbor algorithm was 0.94. However, the proposed AGDP-DBSCAN algorithm performed the best in accuracy and operational efficiency, with the highest accuracy and operational efficiency of 0.98 and 3s, respectively. Therefore, it indicated that the online monitoring method of the AGDP-DBSCAN algorithm for D-IoT devices had good scalability in AD detection.

4 Discussion

As a technological goal for the development of a new type of power network, the construction of the power IoT and the D-IoT is a key factor for the stable development of the power grid. This study uses cluster analysis to partition normal data and AD through the architecture of the D-IoT system. Due to the time series nature of its data, a density-based clustering optimization algorithm is used to perform clustering analysis on the centralized data of the D-IoT. Through performance testing and analysis of different algorithms on the Aggregation dataset and Compound dataset, it can be concluded that the average NMI of the AGDP-DBSCAN algorithm is 0.993. The AMI of the AGDP-DBSCAN algorithm is closest to 1, indicating that the clustering results of the AGDP-DBSCAN algorithm are more consistent with the real clustering results. The maximum average AMI value of the AGDP-DBSCAN algorithm is 0.898, which is due to its use of CTSSMC to calculate distance and adapt to datasets of different densities, resulting in the best clustering performance on the dataset. Another method for monitoring LTU node data anomalies in distribution networks is designed and compared with the anomaly detection method proposed by Gan W et al. [11]. By utilizing the anomaly detection framework and utility-aware anomaly sequence rules, the algorithm's efficiency is evaluated, but its accuracy is 80%. However, the method proposed in this study combines the density of abnormal data sets, with a maximum accuracy of 98%. This is because the proposed algorithm has good adaptability to the proportion of AD in the dataset, which can reduce the influence of current and voltage, thereby improving the detection accuracy of AD and providing technical reference for the future development of smart grids.

5 Conclusion

To improve the reliable operation and equipment management level of the D-IoT system, this study proposed the D-IoT end device online monitoring method. This method analyzed the characteristics of D-IoT LTU-MD and proposed CTSSMC to calculate the distance matrix of the measurement sequence. Based on the TC of LTU-MD and the improved DBSCAN algorithm, this study designed a DAD method based on DBSCAN optimization algorithm. It improved the accuracy of DAD by fully mining LTU historical measurement data and adaptively generating the required global density parameters. The simulation results showed that the AGDP-DBSCAN could demonstrate good adaptability and clustering performance in datasets with uneven density distribution. In Compound dataset, compared to the traditional DBSCAN and K-means, AGDP-DBSCAN had a 2% and 37% increase in NMI. In detection sets with different proportions of anomalies, compared to the LOF algorithm, AGDP-DBSCAN's F1-Score and recall rates have increased by 67% and 13%, respectively. Therefore, AGDP-DBSCAN method

can improve dynamism, reduce clustering data dimensions, and be more stable. There are still shortcomings in identifying the sources of data anomalies, as well as the lack of deep-level monitoring functions for terminal nodes in the distribution logistics network architecture. In the data anomaly monitoring system, it is impossible to display the types and processing capabilities of data anomalies in real-time. Therefore, in the future, the sources of abnormal changes in LTU node data, the computational complexity of node data, and the influencing factors of monitoring models can be analyzed to achieve the intelligent application and development of the IoT in the distribution network.

Table 3: Summary table of relevant literature

Reference number	Research methods	Research results and significance
Lingda et al [4]	Propose a trust evaluation method based on information entropy, using an exponential distribution reputation model to estimate the direct trust value, and combining entropy theory to compensate for the inaccuracy of direct trust judgments.	This method can effectively resist switching attacks and collusion attacks.
Xu et al [5]	Propose a fault location algorithm based on genetic algorithm, improve the unified matrix algorithm, and combine genetic algorithm to achieve fault ranging.	This algorithm can effectively obtain the localization results of erroneous information.
Huang et al [6]	A cloud edge collaborative processing method based on attention mechanism is proposed, which combines edge computing and cloud computing to solve the timeliness of data analysis and calculation.	This algorithm can achieve efficient utilization of electrical energy resources and accelerate data iteration.
Lei et al [10]	Design an unsupervised time series data anomaly detection framework based on spectral analysis, which decomposes the time series using spectral analysis and combines neural networks to predict trend and seasonal sequences.	This method can effectively improve the detection rate of abnormal data.
Gan et al [11]	Introducing utility aware anomaly sequence rules into anomaly detection methods, combined with pruning strategies, to mine the upper bound of utility aware anomaly sequence rules for anomaly detection.	This method has good effectiveness and scalability.
Jeong et al [12]	Propose the probability of local outliers to identify seismic noise, utilize binary decision-making, and combine with moving windows to reduce useful signal energy loss.	This method has good feasibility in denoising applications.
Yang et al [13]	Propose anomaly detection and recovery of multiple regression data based on spatio-temporal correlation, select parameters using correlation analysis methods, and design a multiple regression model that integrates attention mechanism.	This method can achieve data recovery and improve the performance of anomaly data detection.

References

[1] J. Shi, Y. Gong, D. Guang, C. Zuo, X. Wu, L. Lu, G. Zhang, S. Li, R. Wang, and B. Yu, “Improved topography measurement with a high dynamic range using phase difference sensing technology,” Optics Letters, vol. 48, no. 17, pp. 4657-4660, 2023. https://doi.org/10.1364/OL.495680

[2] G. Bandewad, K. P. Datta, B. W. Gawali, and S. N. Pawar, “Review on discrimination of hazardous gases by smart sensing technology,” Artificial Intelligence and Applications, vol. 1, no. 2, pp. 86-97, 2023. https://doi.org/10.47852/bonviewAIA3202434

[3] S. Rostampour, N. Bagheri, B. Ghavami, Y. Bendavid, S. Kumari, H. Martin, and C. Camara, “Using a privacy-enhanced authentication process to secure IoT-based smart grid infrastructures,” Journal

of Supercomputing, vol. 80, no. 2, pp. 1668-1693, 2024. https://doi.org/10.1007/s11227-023-05555-y

[4] L. Kong, F. Zhai, Y. Zhao, N. Qin, D. Li, and S. Cai, "Evaluation method of trust degree of distribution IoT terminal equipment based on information entropy," Journal of Physics: Conference Series, vol. 1754, no. 1, pp. 12108-12115, 2021. https://doi.org/10.1088/1742-6596/1754/1/012108

[5] Y. Xu, Q. Song, and Z. Chen, "Fault-tolerant ability of fault location on distribution internet of things in electricity using genetic algorithm," Journal of Physics: Conference Series, vol. 2033, no. 1, pp. 12077-12083, 2021. https://doi.org/10.1088/1742-6596/2033/1/012077

[6] X. Huang, P. Zhang, R. Feng, C. Huang, K. Zhang, and K. Jin, "Research on cloud-edge collaborative processing method of distribution internet of things based on attention-LSTM," 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture, vol. 2022, no. 3, pp. 52-56, 2021. https://doi.org/10.1145/3495018.3495029

[7] M. Gheisari, H. Hamidpour, Y. Liu, P. Saedi, A. Raza, A. Jalili, H. Rokhsati, and R. Amin, "Mining techniques for web mining: A survey," Artificial Intelligence and Applications, vol. 1, no. 1, pp. 3-10, 2023. https://doi.org/10.47852/bonviewAIA2202290

[8] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," ACM Computing Surveys, vol. 54, no. 3, pp. 56-89, 2021. https://doi.org/10.1145/3444690

[9] W. Jeong, M. S. Almubarak, and C. Tsingas, "Seismic erratic noise attenuation using unsupervised anomaly detection," Geophysical Prospecting, vol. 69, no. 7, pp. 1473-1486, 2021. https://doi.org/10.1111/1365-2478.13123

[10] T. Lei, C. Gong, G. Chen, M. Ou, K. Yang, and J. Li, "A novel unsupervised framework for time series data anomaly detection via spectrum decomposition," Knowledge-based Systems, vol. 280, no. 11, pp. 111002-111015, 2023. https://doi.org/10.1016/j.knosys.2023.111002

[11] W. Gan, L. Chen, S. Wan, J. Chen, and C. Chen, "Anomaly rule detection in sequence data," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12095-12108, 2023. https://doi.org/10.1109/TKDE.2021.3139086

[12] W. Jeong, M. S. Almubarak, and C. Tsingas, "Seismic erratic noise attenuation using unsupervised anomaly detection," Geophysical Prospecting, vol. 69, no. 7, pp. 1473-1486, 2021. https://doi.org/10.1111/1365-2478.13123

[13] L. Yang, S. Li, C. Zhu, A. Zhang, and Z. Liao, "Spatio-temporal correlation-based multiple regression for anomaly detection and recovery of unmanned aerial vehicle flight data," Advanced Engineering Informatics, vol. 60, no. 4, pp. 102440-102454, 2024. https://doi.org/10.1016/j.aei.2024.102440

[14] S. Singh, and K. Kumar, "A study of lean construction and visual management tools through cluster analysis," Ain Shams Engineering Journal, vol. 12, no. 1, pp. 1153-1162, 2021. https://doi.org/10.1016/j.asej.2020.04.019

[15] K. Z. Takahashi, "Molecular cluster analysis using local order parameters selected by machine learning," Physical Chemistry Chemical Physics, vol. 25, no. 1, pp. 658-672, 2023. https://doi.org/10.1039/d2cp90240k

[16] L. Chen, and A. K. Aklikokou, "Relating e-government development to government effectiveness and control of corruption: a cluster analysis," Journal of Chinese Governance, vol. 6, no. 1, pp. 155-173, 2021. https://doi.org/10.1080/23812346.2019.1698693

[17] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," ACM Computing Surveys, vol. 54, no. 3, pp. 1-33, 2021. https://doi.org/10.48550/arXiv.2002.04236

[18] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," Data Mining and Knowledge Discovery, vol. 35, no. 2, pp. 401-449, 2021. https://doi.org/10.1007/s10618-020-00727-3

[19] R. Balaji, R. B. Bapat, and S. Goel, "Generalized euclidean distance matrices," Linear and Multilinear Algebra, vol. 70, no. 21, pp. 6908-6929, 2022. https://doi.org/10.1080/03081087.2021.1972083

[20] N. Gholizadeh, H. Saadatfar, and N. Hanafi, "K-DBSCAN: An improved DBSCAN algorithm for big data," The Journal of Supercomputing, vol. 77, no. 6, pp. 6214-6235, 2021. https://doi.org/10.1007/s11227-020-03524-3

[21] K. Sabor, D. Jougnot, R. Guerin, B. Steck, J. M. Henault, L. Apffel, and D. Vautrin, "A data mining approach for improved interpretation of ERT inverted sections using the DBSCAN clustering algorithm," Geophysical Journal International, vol. 225, no. 2, pp. 1304-1318, 2021. https://doi.org/10.1093/gji/ggab023

Community Detection in Social Networks: A Deep Learning Approach Using Autoencoders

*Priyanka Gupta[1], Mamta Arora[2], Hardeo Kumar Thakur[3]
[1]Research Scholar, Department of CSE, Manav Rachna University, Faridabad, India
[2]Department of CSE, Manav Rachna University, Faridabad, India
[3]Bennett University, Greater Noida, India
E-mail: drpriyankagupta.research@gmail.com, imamta.arora@gmail.com, hardeokumar@gmail.com
*Corresponding author

Keywords: community detection, social network, clustering algorithm, girvan–newman, autoencoder, deep learning, artificial intelligence

Received: June 14, 2024

This research aims to propose a more sophisticated clustering and community detection technique in complex social networks through the use of neural networks; autoencoder, in particular. In the past, methods for network analysis and community detection used several graph algorithms, but with the emergence of deep learning, autoencoders are used for learning node features. These node representations are learnt using a neural network-based autoencoder and then clustering algorithms like k-means, agglomerative and spectral clustering are performed. These algorithms are then improved by incorporating with the Louvain algorithm for community detection. The proposed method, named the Spectral Louvain Algorithm, offers several advantages: it saves the stage of feature extraction, is suitable for Call Detail Record (CDR) and Social Network Analysis (SNA), does not require model retraining for different scale networks, and can work in mesh scale networks. It has better accuracy and performance than former approaches, even getting 100% of NMI and ARS, 78% of modularity in Karate Club and 89% of NMI, 93% of ARS and 86% of modularity in Dolphin data set with the least conductance. This method is particularly useful in discovering new relations and structures in a system and it is very efficient.

Povzetek: Članek uvaja metodo Spektral Louvain za odkrivanje skupnosti v družbenih omrežjih, ki združuje avtomatske kodirnike, spektralno združevanje in Louvain algoritem. Metoda dosega visoko modularnost in prilagodljivost velikosti omrežij.

1 Introduction

It is regarded one of the most important tasks of the network analysis studies – the identification of communities in complex networks[1] which concerns the identification of the groups of nodes densely connected internally, while the links between groups are either sparse or lack at all. Detecting communities [1] can certainly help in understanding the functional organization in different fields including social networks (SN) [2], Biological networks (BioNet), communication networks (CN) [3], [4] etc.

Classical methods of community identification include the use of graph concepts [5] where information flows such as modularity optimization [6], spectrum analysis of the graph[7], and hierarchical clustering [8]are used.

These methods usually need a set of features to be designed manually and can be less effective for large-scale networks with complex patterns. Furthermore, these methods perform poorly in practice, especially when the data is noisy or high-dimensional and are not very scalable and very fragile [9].

Deep learning is a relatively novel technique that tackles feature extraction from raw data in different fields[10]. Auto-encoder [9] is a class of neural networks that can learn a compact and meaningful representation of the data input, based on the input requirement that the network be able to reconstruct the output data as closely as possible [9]. Auto-encoders are beneficial for community detection problems, especially when the internal structure of networks is intricate and could be multi-dimensional, Due to this capability of extracting features from data automatically [9].

This paper is communicating the three significant ways, Firstly, autoencoder algorithms [9] are used to reconstruct communities and provide the node embedding. In this research, algorithms are implemented on two datasets, karate club datasets [11] and Dolphin's datasets [12]. These datasets produce actual communities employing the adjacency matrix which defines the graph data required to discover the communities [13]. Secondly, various clustering methos are applied to learn the node embeddings. Thirdly, these clustering methods are combined with the Louvain community detection algorithm to improve the modularity [14]. The subsequent sections of this paper are structured as follows to outline the process: section ii contains the related work and associated research with potential limitations. Section iii contains the proposed methodology and contributions to explain the novelty of the approach, Section iv contains the research methodologies and techniques employed.

Section v contains benchmark datasets and community validation for preprocessing of benchmark data set Zachary's karate club [11] and dolphins [12] . section vi of our work includes an analytical approach. Section vii covers the preliminaries and proposed algorithms; section viii covers the results and outcomes in which few Evaluation Methods for Assessing Algorithm Performance are discussed. ix contains the outcome and concludes our findings. section ix contains the future scope. section xii contains the nomenclature table.

2 Related works

For the past decades, the algorithms concerned with the classification of communities have evolved greatly, and the recent focus has been on the application of deep learning techniques in enhancing traditional methods [10]. In addition to the well-known techniques, such as modularity optimization and spectral clustering, deep learning-based methods, especially with the use of autoencoders, have become an interesting topic for research in the field of community detection in complex networks [15].
Autoencoder derived methodologies use neural network-based architecture to extract the relevant features and learn the representations of the Network data [9]. This has the following advantages: feature extraction is done automatically and the training with the end-to-end model. These approaches try to identify complex densities and forms in networks, which are different from traditional algorithms.

The earlier methods that are still widely used are modularity optimization by [16] and spectral clustering by von Lux burg [17] while the other methods include self-organizing hierarchal clustering by Jain and Dubes [18] Moreover, the Louvain method developed by Blondel [19] has recently attracted much interest because of its ability to efficiently maximize modularity and identify communities in huge networks.
Few recent studies have also been carried out in this area by authors D. Jin et al., [20] who offer a review of several community detection approaches with emphasis on how deep learning can be incorporated. One of the limitations pointed out here is the lack of standardization in evaluation metrics and benchmark datasets, which prevents the easy comparability of various community detection algorithms [20]. Thomas Kipf & Max Welling [21] learned that the graph convolutional networks that incorporate node features and graph structure outperform other methods in semi supervised learning problems. According to Yang et al. community detectors make use of graph neural networks [22] doing so captures various relationships between the nodes. Bhattacharya et al. [23] offered the most closely related method Embedded graph convolutional networks improve community detection by learning node representations and utilizing structural data. In their paper Xing Su et al.[24] provides a comprehensive review of different types of deep learning in relation to community detection while elucidating the existing progress and issues in this field. Zhang et al. [22] the author presents contrastive graph autoencoders for community detection to enhance the clustering efficiency by using the contrastive learning approach. Apicella et al. [25] discussed the adaptive filters in graph convolution networks and revealed how the community detection enhances the clustering performance due to contrastive learning. X. Li et al.[26] work relates to temporal graph convolutional networks for detection of communities in dynamic graph solving for evolving structures.
From the preceding discussion, it is apparent that deep learning and autoencoder-based techniques in combination with clustering methods along with the help of community detection algorithms offer new prospects for improving the effectiveness of community detection system. The rationale is to enhance understanding of the structures using conventional approaches along with utilization of neural networks and, consequently, discover new patterns in different fields of science.

3 Proposed methodology and contributions

3.1 Flow of work

This research introduces a novel community detection method that leverages feature discovery using autoencoders, network structure using spectral clustering and community detection using the Louvain algorithm. The process starts with feature learning for node data, using autoencoder and converting high-dimensional data to low-dimensional data where typically, a massive amount of effort is required in feature engineering. The autoencoders are involved in capturing structural relationships within the network that may be hidden. To localize these node embeddings, a process called spectral clustering is used to detect nonlinear separations between communities and therefore provide an optimal partition of nodes which is determined by eigenvectors of Laplacian matrix of the graph. Lastly the Louvain algorithm is applied to enhance the modularity for the formation of proper communities of nodes. This hybrid methodology, termed Spectral Louvain, improves both accuracy and scalability of community detection, making it ideal for real-world complex network analysis.

3.2 Novelty of the study

The novelty of proposed approach consists in its hybrid integration of autoencoders, spectral clustering, and the Louvain algorithm which has not been extensively explored in the existing literature. Although each of these techniques has been employed in previous research, the combination presented here —particularly in the parallel process—is more effective to enhance the feature learning capability, the global structure perception, and modularity optimization step. Moreover, the disclosed method can learn and improve without retraining if the network size changes, which is the main advantage in terms of scalability over the prior art.

3.3 Key Ffndings

The proposed Spectral Louvain method shows its higher efficiency comparing to other community detection methods in terms of accuracy and scalability. Experimentation carried out on widely used benchmark datasets Zachary's Karate Club, Dolphin social networks substantiate the effectiveness of the proposed solution in delivering better results across multiple metrics. Specifically, the method achieved perfect scores in both Normalized Mutual Information (NMI) and Adjusted Rand Score (ARS) on the Karate Club dataset[27], while yielding high values in NMI, ARS, and modularity on the Dolphin dataset. These results highlight clear advancements in the performance profiles of different community detection tasks in terms of modularity and the smallest conductance. In the light of such a discussion, the Spectral Louvain approach emerges as one of the most effective and feasible methods to identify the communities in large scale networks.

4 Research methodologies and techniques employed

In the present methodology, A novel community detection approach leverages auto-encoders for feature learning followed with the clustering and community detection method is used to increase the performance of traditional methodology.

4.1 Auto-Encoder feature learning

Auto-encoder structure is used to learn node representations from the data in the network [9]. The autoencoder encodes the input data and tries to reconstruct it in the next layer while learning a compressed representation in the middle layer. The transformation of original network into an embedding space is more conducive for clustering [25]. The autoencoder with low noise is usually defined as:

$$z = f_{encoder}(x) \tag{1}$$

$$(\hat{x}) = f_{decoder}(z) \tag{2}$$

loss function of a simple autoencoder is defined as:

$$L(x, \hat{x}) = ||x - \hat{x}||^2 \tag{3}$$

where x signifies the input and $\hat{x}$ is the reconstruction produced by the decoder.

4.2 Clustering techniques

Various clustering techniques are applied to learned node embeddings to partition nodes into communities [23] :

- **K-means clustering:** *S*plits nodes into K groups by similarity within a specified embedding space [28]. The purpose of the K-means algorithm is to reduce the total of the difference between each point and the centroids. given by:

$$Q = \min_{\{m_k\}_{k=1}^{K}} \sum_{i=1}^{n} \min_{k} ||X_i - m_k||^2 \tag{4}$$

where X_i represents the input data and m_k is the centroid of the k[th] cluster [29].

- **Spectral clustering***:* Clusters the nodes by assigning eigenvectors [30] of the Laplacian of the graph G = (V, E). The Laplacian of the graph is defined as [22]:

$$L = D - A \tag{5}$$

D is the diagonal matrix of the node degrees and A is with the adjacency matrix of the graph G. The eigenvectors, therefore, correspond to the small eigenvalues of L, are for clustering tasks in low-dimensional space [17].

- **Agglomerative clustering**: The process of combining more nodes into fewer clusters according to the specified distance measure [31]. The method applied in finding the distances between two clusters C_A, C_B based on the average linkage metric is defined as [32]:

$$d(C_A, C_B) = \frac{1}{|C_A|,|C_B|} \sum_{i \in C_A} \sum_{j \in C_B} d(i, j) \tag{6}$$

d (i, j) is the distance between the points i and j of cluster C_A and C_B [32].

4.3 Community detection methods

Among the community detection methods, the most popular ones are the Girvan Newman[3] and Louvain methods [14] .

- **Girvan-Newman:** Proceeds by repeatedly scanning for edges with high betweenness centrality to identify the community structure [33].
- **Louvain algorithm:** Recurrently relinquishes nodes to enhance the modularity of the network since communities tend to connect to other communities within the same community [1] [14].

4.4 Combining community detection with clustering

Integration of community detection with clustering approaches such as K-means, Agglomerative Clustering, and Spectral Clustering adds significant value, as it not only considers the network structure but also optimizes the Modularity score. This approach is compared with communities generated by the autoencoder algorithm, and the combination of these clustering and community detection algorithms offers a more accurate and detailed detection of community structures in both large-scale networks [19], [34] and complex Network [35].

• **Louvain K-means:** The procedure used here is a merger of the Louvain method and K-means clustering to find community.

• **Louvain agglomerative:** Automate the Louvain method using agglomerative clustering for enhanced detection of clusters.

• **Spectral louvain:** Also, this approach incorporates the use of the Louvain method in conjunction with spectral

clustering to improve community detection.

4.5 Evaluation and validation

The proposed method is then tested and assessed based on existing real-life data sets with reference solutions to calculate the communities found [2]. Evaluating Quality, the proposed method is tested and evaluated based on a comparison with existing communities with the same data set Karate Club and Dolphin to calculate the found communities. Evaluation of the quality [2] of detected communities: based on modularity [6] and normalized mutual information (NMI) [27], the silhouette score [36].

The approach formulated here has some advantages over other methods: It combines the auto-encoder feature learning with the combination of K-means, Spectral Clustering, Agglomerative Clustering, Girvan-Newman, and Louvain methods to improve the potential of the method in detecting complicated graphs and structures in network data. This integration helps to achieve better and more stable community detection by combining the advantages of deep learning feature extraction and graph-based methods. In addition, the approach is scalable, it can control a large set of networks.

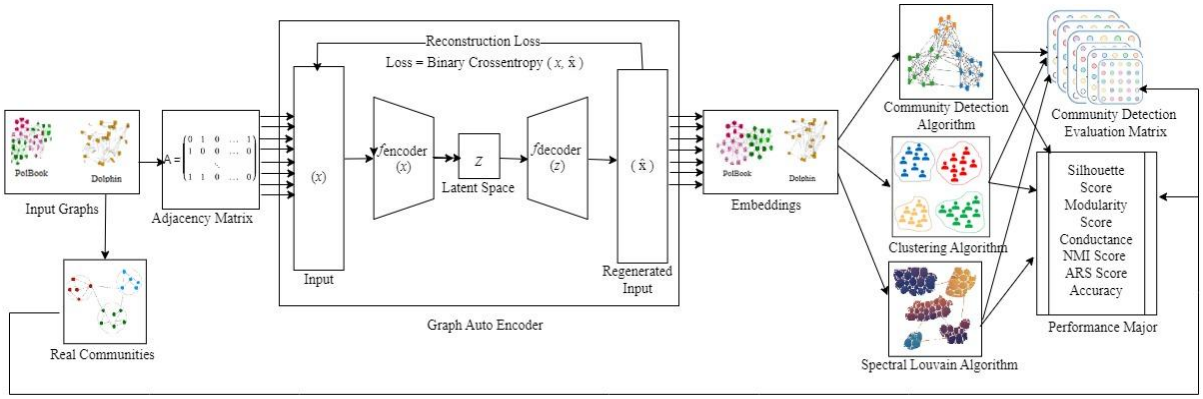


Figure 1: workflow for community detection by using the deep learning AI

5 Benchmark datasets and community validation

Benchmark datasets are used in many research projects to assess the effectiveness of the algorithms being used. In this study, two prominent benchmark datasets were employed: Zachary's Karate Club [11] and the Dolphins [12].

Zachary's karate club:

- **Description:** This data set consists of the social relations of 34 members of a karate club [37],[11] which later leads to the division of the club into two groups.
- **Relevance:** It provides a clear ground truth suitable for comparing the efficacy of community detection algorithms.

- **Dolphins:**
  - **Description:** For this case, the dataset composed of 62 bottlenose dolphins exhibits the social network found through association indices [12].
  - **Relevance:** It shows a denser and more sophisticated network taken from real life, which is useful to test community detection algorithms.

With a given set of datasets and a deep learning approach, the increased benchmarking, validation, and reproducibility with the combination of clustering and the community detection algorithm presented here reveals the effectiveness of the methodology in terms of community detection in a variety of networks [38], [39], [40].

6 Analytical approach

The analytical approach in this research combines modern methods for the identification of communities in complex networks [35]. First, the autoencoder algorithms are used to learn compact and semantically relevant node representations from the given input matrices. These representations are then used to provide an efficient way to cluster the nodes into communities using several clustering algorithms [9]. They are K-means, Spectral clustering, Agglomerative clustering, Girvan Newman and Louvain methods. Thus, combining the capabilities of deep learning for feature extraction and the graph-based clustering approach for the extraction of structural and complex pattern features, it will be possible to improve the existing approach and reveal high-level patterns in network data. Based on the quality of the detected communities [2] the model parameters such as modularity, normalized mutual information (NMI) [27,32], and Silhouette scores are analyzed [36], and the model is tested for scalability on datasets of different sizes.

6.1 Community detection algorithm and evaluation workflow

The analytical approach in this research combines modern methods for the identification of communities in complex networks [1], [35] . First, the autoencoder algorithms are used to learn compact and semantically relevant node representations from the given input matrices. These representations are then utilized to cluster the nodes into communities using various clustering algorithms. [30]. They are K-means [29], Spectral clustering [17],

Agglomerative clustering [31], Girvan Newman [16] and Louvain method [14]. Thus, combining the capabilities of deep learning for feature extraction and the graph-based clustering approach for the extraction of structural and complex pattern features, it will be possible to improve the existing approach and reveal high-level patterns in network data. Based on the quality of the detected communities the model parameters such as modularity [16], normalized mutual information (NMI) [27], adjusted Rand score (ARS) [35], and silhouette score are analyzed to assess the quality of the detected communities, and the proposed model is also tested for scalability on the two benchmark datasets of different size karate club and dolphin. The given figure 1 depicts the process workflow of the proposed community detection model for checking the effectiveness of comprehensive network structures.

7 Preliminaries

Consider a network graph G, comprising V_e nodes and E_v edges, G (V_e, Ev). The adjacency matrix [5] A of G, where Aij ∈ {0, 1, signifies the presence of an edge between, nodes i and j [5].

7.1 Algorithm overview autoencoder

The proposed method utilizes Graph Autoencoder (GAE) to learn latent representations of nodes from the network's adjacency matrix. This helps the auto encoder to learn the embeddings [7]. Represented by the algorithm 1-5.

Algorithm 1, The Graph Autoencoder (GAE) Communities is used as an algorithm that can learn the latent features for graphs. It establishes a model with the help of input parameters latent_dim and adj_matrix and sets the autoencoder encoder. The encoder is a sequential neural network with dense layers to map the graph's adjacency matrix to a latent space. The decoder then maps the embeddings back to the original space to reconstruct the input in the matrix form of the adjacency matrix using the learned embeddings.

Algorithm 1: Graph Autoencoder Community

Input: - latent_dim, adj_matrix
***Output*:** -Initialized GraphAutoencoder model
1: **procedure** init(LatentDim, adj_matrix)
2: **Initialize** superclass
3: **Set** self.LatentDim to LatentDim
4: self. encoder ← Sequential([
5: Dense (Latnt_dim, activation ==" LeakyReLU"),
6: Dense (2 * LatntDim, activation= "LeakyReLU"),
7: Dense (4 * LatntDim, activation = "LeakyReLU"),
8: Fully connected (8 * LatntDim, activation LeakyReLU)])
9: self.decoder ← Dense(adj_matrix. shape[0],
activation='sigmoid')
10: **end procedure**

Algorithm 2, The Graph Autoencoder is a model that takes input data, applies a forward pass of an unspecified algorithm to produce a latent representation, and then passes it through the decoder to reconstruct the original input.

Algorithm 2: Graph Autoencoder Call Function

Input: inputs
Output: Decoded output
1: **procedure** call(inputs)
2: Encoded ← selfEncoder(inputs)
3: Decoded ← selfDecoder(encoded)
4: return Decoded
5: **End procedure**

Algorithm 3, The loss function computes the difference of true and the predicted adjacency matrix in order to measure how well the model is performing by changing the parameters of autoencoder.

Algorithm 3: Loss Function

Input: - YTrue, YPred
Output: - Loss value
1: **procedure** loss_function (YTrue, YPred)
2: return mean (binary_crossentropy (YTrue, YPred))
3: **end procedure**

Algorithm 4, concerns reading data from a GML file to transform the graph into a format used by the Graph Auto-encoder model, an adjacency matrix. The algorithm primarily processes the GML file by reading the graph and converting it into a dense adjacency matrix, which is then used as the input graph and its corresponding adjacency matrix.

Algorithm 4: To read GML data

```
Input: - path
Output: - Graph G, Adjacency Matrix
1: procedure read_gml_data(path)
2:          G ← read_gml(path, label='id')
3:          adjacency ← AdjacencyMatrix(G). todense()
4:          return G, adjacency
5: end procedure
```

Algorithm 5, Predict the communities by using Girvan Newman Predictions and Louvain communities.

Algorithm 5: Generate Community Prediction

```
Input: - Graph, graph_from_similarity_embeddings, seed
Output: - predictions, louvain_predictions
1: procedure Girvan_Newman_Predictions (Graph,
   graph_from_similarity_embeddings, seed)
2:                    > #set random seed for reproducibility of the results
3:   communities ← girvan_newman (Graph)
4:   node_groups ← []
5:   for i in range 4 do
6:       for c in next(communities) do
7:           if i = 2 then
8:           node_groups. append(c)
9:           end if
10:      end for
11:   end for
12: louvain_communities←louvain_communities (graph_
   from_similarity_embeddings)
13.  Predictions ← [0] * ILen (GraphNodes)
14:   louvain_predictions = [0] * ILen (GraphNodes)
15:   for i in the range lLen (NodeGroups) do
16:       for (j, k) in zip(node_groups[i],
              louvain_communities[i]) do
17:          if '0' is present in Graph. nodes then
18:              predictions[j] ← i
19:          else
20:              predictions[j-1] ← i
21:          end if
22:          louvain_predictions[k] ← i
23:      end for
24:   end for
25:   return predictions, louvain_predictions
26. end procedure
```

The learned embeddings are then clustered using Clustering [5], agglomerative clustering [31], and various techniques, including K-means [28], Spectral after that the Final communities are generated using the community detection algorithms [38] Girvan Newman [33] and Louvain methods [14].

This integration of autoencoder-based feature learning and clustering algorithms and community detection [39] algorithms enhance the ability to detect communities in complex networks [34]. Among all these approaches we find the best approach that with the combination of spectral and Louvain represented by algorithms 6.

Algorithm 6: Spectral Louvain

```
Input: - Graph G, number of clusters k, resolution parameter
gamma,
 random seed
Output: - Final partition
1: procedure spectral_louvain (G, k, gamma, seed)
2:          # Set random seed for reproducibility of the results
3: ># Step 1: Calculate the Laplacian matrix of the graph L
4: L ← laplacian_matrix(G). todense ()
5:
6: ># Step 2: Let the eigenvectors be v=(v1, v2,……..vn) of L
 such that Lv=kv where k is an eigenvalue of L
7: obtain the eigenvalues (EigVal) and the corresponding
eigenvectors (EigVect): EigVal, EigVect (v)← eigh(L)
8:   indices ← argsort (EigVal)[: k]
9:  U← EigVect [: indices]
10:
11: ># Step 3: clustering of the rows of U is obtained through k-means
 clustering.
12:  kmeans ← KMeans(n_clusters=k)
13:  initial_partition ← kmeans. fit_predict(U)
14:
15: ># Step 4: use the Louvain Algorithm to the initial partition
16: partition ← best_partition (G, partition = initial_partition, resolution
 = gamma)
17:
18: ># Perform step 3 and step 4 until the process converges.
19:     While true do
20:          old_partition ← partition
21:
22:># Step 5. is to conduct k-means clustering on the  rows of U.
23:          kmeans ← KMeans(n_clusters=k)
24:          initial_partition ← kmeans. fit_predict(U)
25:
26: ># Step 6: The Louvain Algorithm for the initial formation of a
 partition is equal to best_partition
27:  Louvain_Algorithm = best_partition (G, partition = initial_partition
, resolution=gamma)
28:
29:          if partition == old_partition then
30:              break
31:          end if
32:    end while
33:
34: ># Output the final partition
35: return partition
36: end procedure
```

The proposed method of combining autoencoder-based deep learning of features with multiple clustering methods improves and optimizes the process of community detection in networks data. Integration in this structure guarantees well distributed and coherent neighborhoods in Louvain Partition Algorithm. The algorithm will also keep on running and the divide will keep adjusting and will only

stop whenever it reaches a state that signifies a good and stable community structure. Finally, the function returns the final partition as a dictionary in which each node is paired with its community index; thus, presenting an improved and efficient community detection.

7.2 Time complexity

- **Graph autoencoder training:** The time complexity [5] is O (V_e. d^2.L). Where V_e signifies the total number of vertices in the graph, d signifies the number of hidden dimensions, and L symbolizes the number of layers in the autoencoder.
- **K-means clustering:** The worst-case complexity of time is [28] O (V_e. k. t) where k implies number of clusters and t signifies the number of iterations [28].
- **Spectral clustering:** The worst-case time complexity is calculated as O (V_e ^{3}). due to eigen value and vector computations [30].
- **Louvain method:** The time complexity is O (E_v log V_e).

7.1 Space complexity

- **Graph autoencoder:** The space complexity[5] is O (V_e. d+d2.L).
- **Clustering algorithms:** The space complexity for K-means, Spectral Clustering, is O (V_e ^{2}).
- **Community detection algorithms:** The space complexity for Louvain [5], O (V_e + E_v) where 'V_e' is the number of nodes and 'E_v' is the number of edges [5].
- **Combining the complexities:** The Graph Autoencoder (GAE) training and the clustering and community detection algorithms, the overall complexities are:

  Time Complexity: O (V_e ^{3})
  Space Complexity: O (V_e ^{2})

8 Results and discussion

The research also shows that integrating feature learning through autoencoders alongside multiple clustering procedures enhances community detection remarkably. When it comes to assessing benchmark data sets, such as Karate Club and Dolphins, the method proves proficient in perceiving complex patterns/structures within the network data.

8.1 Evaluation methods for assessing algorithm performance

- **Modularity score:** Modularity measures the strength of connections within groups [6], Higher modularity value displays stronger community structure on the divisions detected, in the networks.

$$Q = \frac{1}{2E_v}\sum_{i,j}^{k_{i,j}}\left(A_{i.j} - \frac{k_i k_j}{2E_v}\right)\delta(c_i, c_j) \qquad (7)$$

where:

• $A_{i.j}$ is the actuality of the adjacency_matrix [5] of the graph,

• Here, the value k_i and k_j are the degrees of nodes.

• E_v is defined as the total of all edges of the graph .

• The nodes are arranged in several communities or groups and the particular nodes i and j belong to the communities of nodes c_i and c_j.

• $\delta(c_i, c_j)$ Kronecker [6] delta function [16] meaning it equals 1 if the indices c_i and c_j are equal and 0 if they are different [5].

- **Silhouette score:** It describes the extent of the resemblance of an object within the cluster to the other clusters[36] . It is defined as:

$$d = b(m) - a(m) \qquad (8)$$

$$S(m) = \frac{d}{\max(a(m), b(m))} \qquad (9)$$

where:

The intra-cluster distance a(m) measures the mean distance of ith data point from the other points belonging to its cluster [36].The inter-cluster distance b(m) average of the shortest distance from the ith point to all the points belonging to the different cluster [36]. d signifies the difference between the two distances.It must be noted that the overall silhouette coefficient is the average of s(m) for all instances [41]. Consequently, the silhouette score nearly equal to 1, testify that all the data points belong to the appropriate cluster.

- **Normalized mutual information (NMI) score:**

NMI is employed in comparison of two clustering results [27]. As:

$$NMI(M, N) = \frac{2.I(M;N)}{H(M)+H(N)} \qquad (10)$$

Where

• The last is the mutual information of two clustering results M and N, which is defined as I(M;N) [27].

• Information entropies of M and N are represented by H(M) And H(N). An NMI value closer to 1 suggests that the predicted clusters are nearly identical to the actual clusters of the data.

- **Conductance:**

It is one of the measures of evaluation of the quality of the clusters or communities that are formed. For a subset of nodes S, it is defined as:

$$(S) = \frac{|\{(U,V) \in E : u \in S, v \in S|}{\min(\mathrm{vol}(S), \mathrm{Vol}(\overline{S}))} \qquad (11)$$

where:

• |{(u,v)∈E: The desired quantity is defined as [42] |{(u,v)∈E: u∈S,v∈/S}|[3,6], the fraction of edges between the subset S and the rest of the graph.

• vol(S) [42] sum of the degree of the nodes in S.

• vol(V\S) [42] sum of elements or degree of the nodes are not in the set S.

- **Adjusted rand score (ARS):**

The Adjusted Rand Score defines the similarity of two clusterings, by taking into account all the possible pairs of samples and counting the number of pairs included in the same or different clusters of the clustering prediction and the true clustering [27]. The ARS is adjusted for the chance grouping of elements:

$$ARS = \frac{RI - Expected\ RI}{\max(RI) - Expected\ RI} \quad (12)$$

where:

RI is the Rand Index. Expected RI is the expected value of the Rand index [43] which namely shows an expected of the current structure of a set of clusters.

8.2 Comparison table

The Comparison Table of Methods for community detection focuses on the key features, benefits and the limitations of each method based on factors including the accuracy, scalability and the computational complexity of the type of clustering and the applicability to large scale graphs. It is represented by table 1.

Table 1: Comparison table of methods

S.No	Method	Approach	Evaluation Metrics (Accuracy, NMI, ARS, Precision, etc.)	Scalability	Modularity Optimization	Key Benefits	Strengths	Limitations
1	**Autoencoder + K-Means Clustering** [44]	get low-dimensional embeddings and feature learning using Autoencoder further applies K-Means for clustering	Accuracy, Macro Precision, ARS, Modularity	High	Low	Useful dimensionality reduction accompanied by uncomplicated clustering at the same time	Uncomplicated, topological, high-speed clustering in the spatially smoother dimensions.	Assumes spherical clusters, struggles with complex data
2	**Autoencoder + Agglomerative Clustering** [45]	An autoencoder is used to learn embeddings and hierarchical agglomerative clustering is performed.	accuracy, NMI, ARS. Macro-precision, Macro-recall, Macro- F1-score.	Moderate	Moderate	However, the proposed model is capable of capturing power-law distributions that represent hierarchical structures in the latent representation space.	Discovers the community relationships, which are particularly useful for hierarchical ones.	Poor scalability in large datasets, computationally expensive and sensitive to linkage criteria.
3	**Autoencoder + Louvain Algorithm** [46]	The modularity of Louvain is used to optimize the Autoencoder embeddings.	Modularity, NMI, ARS, Macro precision	High	High	Implements a modularity optimization of Louvain that improves with learned representations.	The algorithm discovery is highly scalable to large networks also Optimizes modularity,	Even when there are good embeddings Louvain may fail to capture small communities because of its greediness, Limited in very dense graphs
S.No	**Method**	**Approach**	**Evaluation Metrics (Accuracy, NMI, ARS, Precision, etc.)**	**Scalability**	**Modularity Optimization**	**Key Benefits**	**Strengths**	**Limitations**

4	**Autoencoder + Louvain + Agglomerative** [10]	Integrates embeddings of autoencoder about what has been learnt into agglomerative clustering then employ Louvain to detect the dense communities	NMI, Modularity, ARS	Moderate	High	The described method is based on hierarchical and modularity-based clustering of data and is aimed at creating a multi-scale network.	It seems to balance the global and the local structures.	Low efficiency and high demand on computation.
5	**Autoencoder + Spectral Clustering** [17]	While autoencoder is applied to learn the feature representations of nodes, spectral clustering is used to generate latent space and partition communities.	NMI, ARS, Modularity, Macro Precision, Macro Recall.	High	Moderate	More efficiently partitions latent space by the help of spectral characteristics.	Captures global structure, good for non-linear separations	Very sensitive to the total of vertices in the graph, preventively expensive on large datasets
6	**Proposed Algorithm Autoencoder + Spectral Louvain** [47]	Embeddings from the autoencoder are segmented via spectral clustering, followed by Louvain for modularity.	Modularity, NMI, ARS, Macro Precision, Conductance, Macro Recall, Accuracy	High	High	Combines the approach of the global graph structure using the spectral approach with the optimized modularity approach.	Efficient in unsupervised learning, detection of global clusters with strong modularity optimization, scalable	Spectral clustering has a problem of high computational complexity if the graph is large.
7	**Autoencoder + Louvain + K-Means** [48]	autoencoder used for embeddings, Louvain for modularity optimization, , and K-Means for improving the clustering results.	Accuracy, Modularity, Macro Precision, Average Rand Score.	Moderate	Moderate	Optimal trade-off between the optimization of modularity and the clustering.	present a fast and interpretable community detection.	Struggles with complex network structures, K-Means is known for its difficulties in identifying non-convex clusters; still, Louvain may fail to identify smaller communities.
8	**Graph Convolutional Networks (GCN)-based methods** [25], [49]	Finds node embeddings for a graph that allows for accounting for both features and relationships using graph convolutional networks.	Accuracy, NMI, ARS, Macro Precision, Macro F1-score,	Moderate	Moderate	Both nodes features and pairwise connection in graph structures are employed for enhanced clustering.	Effective with labelled data, node features, exceptional in graph-structured data and node classification.	Requires labelled data, retraining for new tasks, Onerous to compute particularly when deep models or large graph scenarios are in real-life applications.
9	**GNN-based methods** [50]	General framework for Graph Neural Networks Architecture to incorporate information of the closer neighbours.	Accuracy, Macro-Precision, Macro-Recall, ARS, Macro- F1-score NMI,	Moderate	High	Versatile containing both node and link information but suffer.	Can work with unsupervised, semi-supervised, or supervised tasks, learning from graph structures, An all-around tool and robust for a variety of graph computations.	High computational cost solution and over-smoothing issue in the deep architecture. struggle when applied to large-scale graphs consume high memory
S. No	**Method**	**Approach**	**Evaluation Metrics (Accuracy, NMI, ARS, Precision, etc.)**	**Scalability**	**Modularity Optimization**	**Key Benefits**	**Strengths**	**Limitations**

10	**Graph Attention Networks (GATs)** [51]	Utilizes attention mechanism in assigning importance of neighbours in node embedding learning.	NMI, ARS, Macro Precision, Macro F1-score	Moderate	High	Enhances the node and the edge feature representation using attention.	Introducing an effective method for node representation learning while having a precise control over the relevance of the neighbours. Captures importance of different edges	High computational complexity, A computationally intensive approach and its memory demand increases with the size of the graph.
11	**Deep Walk** [52]	(Random Walk + Deep Learning) Acquires node representations by randomly walking on the graph with the Skip-Gram approach.	NMI, ARS, Modularity, Macro Precision, Macro Recall, Accuracy	High	Moderate	General-purpose, generative node embedding that restores community information.	Adaptable to local and global graphs and easy to implement.	Sensitive to hyperparameters and sometimes fails to identify delicate community structure. Limited with dynamic graphs,
12	**Girvan-Newman** [33]	Edge Betweenness Centrality, Successively, removes edges with the highest betweenness centrality for the purpose of detecting communities.	Modularity, NMI, Macro Precision, Macro Recall,	Low	High	Highly suitable when clusters are very distinct from each other.	Highly suitable when clusters are very distinct from each other.	Highly computationally complex and not feasible for very large graphs.
13	**Node2Vec** [53]	(Biased Random Walk + Embeddings) Improves Deep Walk by incorporating biased random walks that allows enhanced exploration of the node neighbourhoods.	Accuracy, NMI, ARS, Macro Precision, Macro Recall,	High	Moderate	The exploration and exploitation approach enhances the richness of the embeddings.	Works well for identifying perfect clusters, when the data is clean.	Struggles with multi-scale community structures, Sensitive to parameter tuning. Not suitable for multi-scale community structures,
14	**Label Propagation** [54]	Propagation-based Algorithm, it sequentially transmits labels through a graph in accordance with the Neighbour information.	Accuracy, NMI, ARS, Macro Precision, Macro Recall,	High	Moderate	High efficiency for large graphs and simple to deploy at a large scale.	Highly scalable and simple.	May result in oversimplified communities because labels spread out. Unreliable with overlapping communities,
15	**Info map**[55]	Flow-based Clustering, finds communities using information theory where minimum description length of random walks are identified.	Modularity, NMI, ARS and Macro Precision	High	High	maintain both speed and accuracy for large graphs.	Suitable for large graph problem.	Preserves graph structure but may fail to identify the small communities.

8.3 Evaluation metrics

In the case of all performance measures, better performance always possesses the higher value of a measure; however, the importance of some particular measures, such as precision, recall, or overall accuracy, may be higher in one situation than in another depending on the context in which it is used or required. In this research, these metrics are employed in the autoencoder based community detection and then clustering to determine the effectiveness of the algorithm in detecting the communities in the networks. The performances of the community detection algorithms are assessed with a set of classification measures out of which some are discussed for proper evaluation of the results. The presented evaluation offers quantitative evidence of each algorithm's utility, stability, and transferability across

various datasets, which makes the conclusions falsifiable and universal [56].

8.3.1 Precision

- **Micro precision**: Subtracts the false positive count from the true positive count in all the discovered communities with the help of autoencoder and clustering method occurred at the global level for measures precision.
- **Macro precision:** Sums up the precision of every detected community without any preference for a particular community, offers an equal score or a clear picture of all the communities' performance.
- **Weighted precision:** Takes the arithmetic mean of each detected communities' precision and the number of true cases in each community is used to weigh each community

  average so that larger communities have more weight than the smaller ones.

8.3.2 Recall

- **Macro recall:** Calculate the recall for each of the detected communities, and then average this measure considering all of them as similar to each other.
- **Micro recall:** Measures recall globally by summing the number of true-positives and false-negatives[57] regarding all the communities that have been distinguished by the autoencoder and the clustering algorithm.
- **Weighted recall:** Computes the average of recall of each detected community to balance their contribution to the final metric, keeping in mind bigger communities should contribute more.

8.3.3 F1-Score

- **Micro F1-Score:** The mean of micro precision and micro recall, that gives the overall estimate of the autoencoder and clustering algorithm [57].

Macro F1-Score: The combination of both macro precision and macro recall provides a balanced assessment of the algorithm's effectiveness for each discovered community [57].

Weighted F1-Score: Sums up all the F1-scores of each detected community and then divides by the number of true instances in each community to provide an estimated F1-score: [57] of the algorithm in the general framework of community detection.

8.4 Tables and figures

The following table 2 to table 5 represents the Classification Report of several clustering algorithms [56] such as K-Means [28], Spectral Clustering, agglomerative clustering [8], community detection algorithm Girvan-Newman's [33] and Louvain Algorithm [14], and their combinations of Louvain K-means, Louvain Agglomerative, Spectral Louvain All these algorithms are applied on two different data sets Zachary karate club [11] and Dolphin datasets [12].

Table 2: Evaluation metrics result - (Zachary Karate club dataset)

S.No.	Algorithms	NMI	ARS	Modularity	Conductance	Silhouette Score
1	KMeans Clustering	0.57	0.67	0.6	0.61	0.23
2	Spectral Clustering	0.68	0.77	0.65	0.59	0.23
3	Agglomerative Clustering	0.73	0.77	0.65	0.59	0.21
4	Girvan Newman	0.84	0.88	0.71	0.57	0.22
5	Louvain Algorithm	0.56	0.41	0.2	0.85	0.1
6	Louvain KMeans	0.68	0.77	0.65	0.59	0.23
7	Louvain Agglomerative	0.84	0.88	0.66	0.58	0.23
8	Spectral Louvain	1	1	0.78	0.50	0.23

Table 3: Evaluation metrics result - (Dolphin dataset)

S.No.	Algorithms	NMI	ARS	Modularity	Conductance	Silhouette Score
1	KMeans Clustering	0.33	0.19	0.62	0.6	0.2
2	Spectral Clustering	0.87	0.90	0.8	0.53	0.16
3	Agglomerative Clustering	0.21	0	0.58	0.62	0.23
4	Girvan Newman	0.88	0.88	0.81	0.52	0.15
5	Louvain Algorithm	0.55	0.4	0.62	0.71	0.08
6	Louvain KMeans	0.32	0.16	0.61	0.61	0.1
7	Louvain Agglomerative	0.32	0.16	0.61	0.61	0.1
8	Spectral Louvain	0.89	0.93	0.86	0.50	0.23

Table 4: Classification evaluation metric result (Zachary Karate Club Dataset)

S.No.	Algorithms	Accuracy	Macro Precision	Macro Recall	Macro F1 Score	Weighted Precision	Weighted Recall	Weighted F1 Score
1	KMeans	0.91	0.91	0.91	0.91	0.91	0.91	0.91
2	Spectral Clustering	0.94	0.94	0.94	0.94	0.94	0.94	0.94
3	Agglomerative Clustering	0.94	0.94	0.94	0.94	0.95	0.94	0.94
4	Girvan Newman	0.97	0.97	0.97	0.97	0.97	0.97	0.97
5	Louvain	0.24	0.22	0.1	0.14	0.53	0.24	0.33
6	Louvain KMeans	0.06	0.06	0.06	0.06	0.06	0.06	0.06
7	Louvain Agglomerative	0.97	0.97	0.97	0.97	0.97	0.97	0.97
8	Spectral Louvain	1	1	1	1	1	1	1

Table 5: Classification evaluation metric result (Dolphin Dataset)

S. No	Algorithms	Accuracy	Macro Precision	Macro Recall	Macro F1 Score	Weighted Precision	Weighted Recall	Weighted F1 Score
1	KMeans Clustering	0.27	0.22	0.21	0.22	0.3	0.27	0.28
2	Spectral Clustering	0.02	0.01	0.02	0.02	0.01	0.02	0.01
3	Agglomerative Clustering	0.6	0.73	0.7	0.59	0.82	0.6	0.58
4	Girvan Newman	0	0	0	0	0	0	0
5	Louvain	0.02	0.01	0	0.01	0.03	0.02	0.02
6	Louvain KMeans	0.71	0.77	0.78	0.71	0.84	0.71	0.71
7	Louvain Agglomerative	0.71	0.77	0.78	0.71	0.84	0.71	0.71
8	Spectral Louvain	0.94	0.92	0.95	0.93	0.95	0.94	0.94

Details of the comparative studies that can be made using the assessment metrics for classification and community detection proposed here regarding each of the corresponding Figures (2 - 13) are stated in the caption for each of such figures. From Table 2 it can be noticed that the Louvain Agglomerative has a better performance in terms of NMI (0. 84), ARS (0. 88), and Modularity (0. 66) and has proved to be better than alternatives like K-means and Spectral Clustering. The Spectral Louvain method also yields high results with the optimum Modularity score of 0. 78 coupled with the minimum Conductance score of 0. 50 which proves that there is a better way to recognize the communities of interconnection and thereby reducing inter-community links. Table 3 showed that up to this point, the Spectral Louvain algorithm outperforms all methods by registering the highest value for NMI 0.89, ARS 0. 93 and Modularity 0. 86. This suggests a strong capability in terms of reconstructing community structure within the context of the Dolphin dataset. The low Conductance score (0.52), This shows great performance of the proposed Spectral Louvain method and validate the algorithm on the basis of the conductivity measure that demonstrates its ability, to draw sharp and distinct line between communities. Other methods such as Girvan Newman and Agglomerative Clustering are also quite efficient and accurate, however they do not have the broad characteristics of the Spectral Louvain approach, most importantly in terms of NMI and ARS indicators that determine clustering accuracy as well as stability. The results in Tables 4 and 5 compare the performance of various communities' detection algorithms across two datasets: the Zachary Karate Club and the Dolphin.

Several evaluation parameters have been displayed in the tables such as Accuracy, Macro Precision, Macro Recall, Macro F1 Score, Weighted Precision, Weighted Recall, and Weighted F1 Score. Out of all the methods, Louvain K-means and Spectral Louvain performed significantly better in most of the cases, especially in the weighted metrics and Spectral Louvain touches the near perfect score of around (0.94 to 0.95) in case of Dolphin dataset, Spectral Clustering and Louvain Agglomerative are two methods, successful in optimizing different sample data sets. Based on the results obtained, Spectral Louvain with overall performance of 1. 0 i.e. 100% on all the measures evaluated has been identified as the best fit for Zachary Karate Club data set. Other algorithms like Agglomerative Clustering and Louvain Agglomerative also have good results, however Louvain alone performs poorly with metrics such as Macro F1 Score and Weighted F1 Score. K Means and GN are still quite sound but not as precise as Spectral Louvain. In given Figures 2-13,

Each algorithm is represented by its corresponding serial number as follows: Being an integer, method can be 1 for K-means Clustering, 2 for Spectral Clustering, 3 for Agglomerative Clustering, 4 for Girvan Newman, 5 for Louvain, 6 for Louvain K-means, 7 for Louvain Agglomerative, and 8 for Spectral Louvain. Thus, based on the above Figures 2-13 and from the results given in table 2-6, it can be concluded that Algorithm 8 (Spectral Louvain) performs better than other algorithms with reasonable accuracy on most of the metrics, especially for the Karate Club dataset. Yet, even in the case of the Dolphin dataset, which also yields better results when the proposed methodology is applied compared to other algorithms.

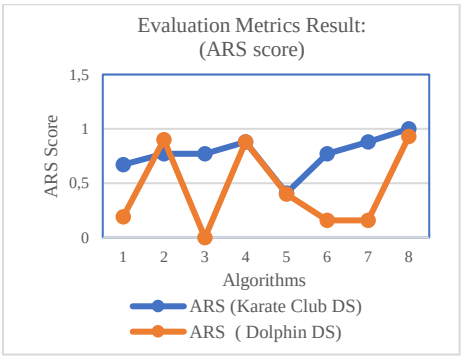


Figure 2: Evaluation Metrics Result (ARS Score)

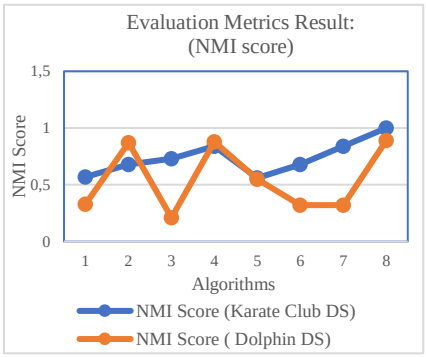


Figure 3: Evaluation Metrics Result (NMI Score)

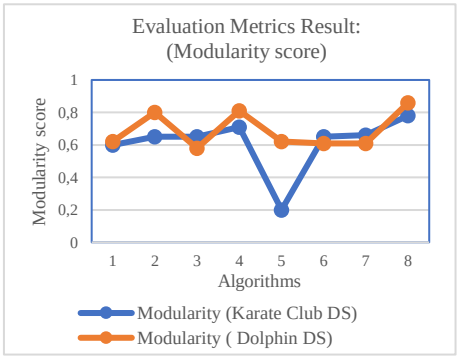


Figure 4: Evaluation Metrics Result (Modularity Score)

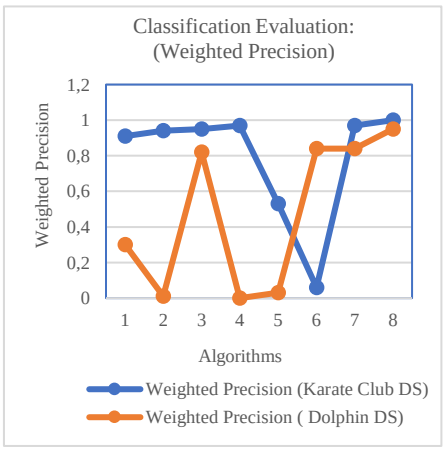


Figure 5: Evaluation Metrics Result (Conductance Score)

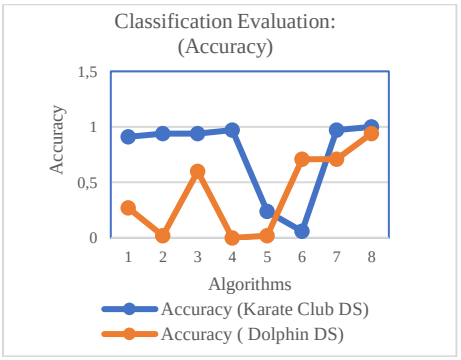


Figure 6: Classification Evaluation Metrics Result (Accuracy)

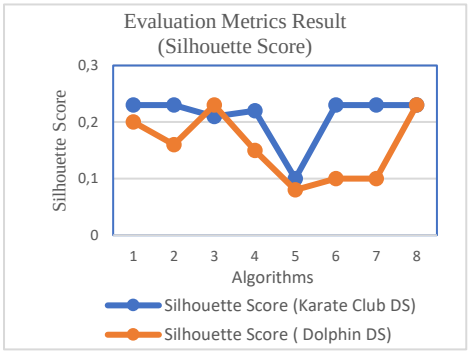


Figure 7: Evaluation Metrics Result (Silhouette Score)

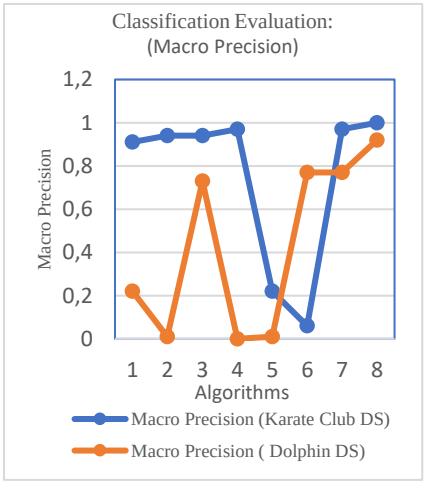


Figure 8: Classification Evaluation Metrics Result (Macro Precision)

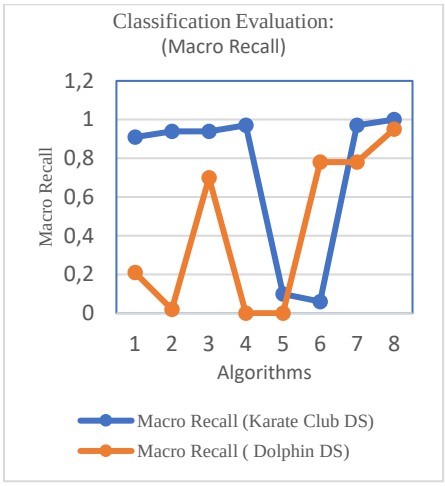


Figure 9: Classification Evaluation Metrics Result (Macro Recall)

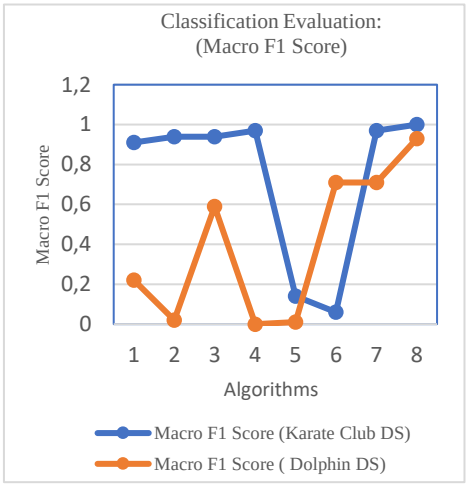


Figure 10: Classification Evaluation Metrics Result (Macro F1 Score)

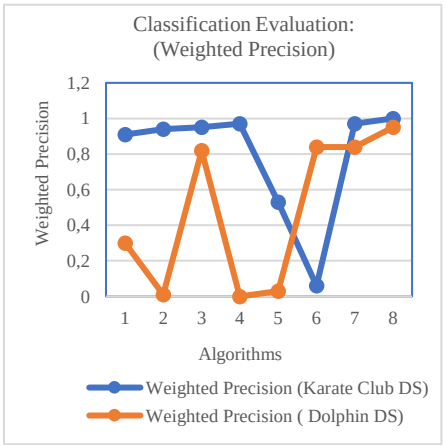


Figure 11: Classification Evaluation Metrics Result (Weighted Precision)

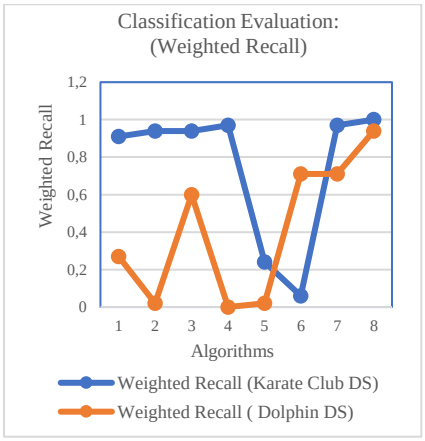


Figure 12: Classification Evaluation Metrics Result (Weighted Recall)

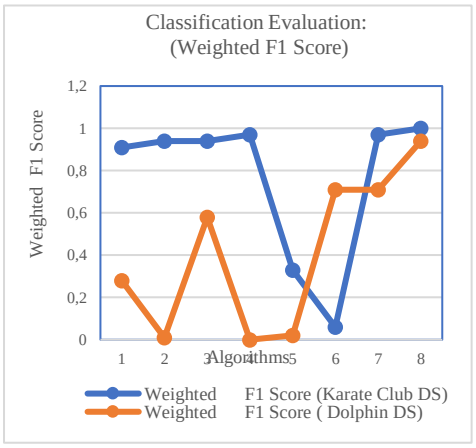


Figure 13: Classification Evaluation Metrics Result (Weighted F1 Score)

On the same kind of network, there is another algorithm called Algorithm 4 (Girvan Newman) and Algorithm 7 (Louvain Agglomerative) which is also good for the Karate Club data set, but not effective at all in case of the Dolphin data set. For instance, Algorithm 4 (Girvan Newman), it outperforms itself on the Karate club dataset, whereas it yields zero remarks on the Dolphin dataset on all sorts of measures. Moreover, Graph 5 (Louvain) algorithm and algorithm 6 (Louvain K-Means) algorithm portray significant poor performance on the Dolphin dataset.

Thus, the Algorithms Spectral Louvain based on the proposed methodology is the best overall but as with Dolphin dataset its superiority is less than in the Karate Club dataset but still better than other algorithms. Agglomerative also have good results, however Louvain alone performs poorly with metrics such as Macro F1 Score and Weighted F1 Score. K Means and GN are still quite sound but not as precise as Spectral Louvain.

9 Conclusion

This paper presents the autoencoder based Spectral Louvain Algorithm for community detection and clustering in complex networks with the performance evaluation on the Zachary karate club and Dolphin networks. The general proposal of this paper utilized autoencoders to learn nodal representations for nodes, and then used k-means clustering, agglomerative and spectral and the Louvain algorithm for the detection of communities. The results offered a few advantages: Individual feature extraction is not required in this model, does not require re-training of the network for different sizes of the network and is valuable in large networks.

The empirical results confirm that the Spectral Louvain outperforms the previous methods as it reached 100% of NMI and ARS in Zachary karate club and 78% modularity whereas in Dolphin it had 89% NMI, 93% of ARS and 86% modularity having lowest conductance. Also, during the classification assessment, it possesses a higher accuracy of 94% in the Dolphin dataset than those methods as K-means and Girvan Newman with 27% and 0% respectively. At Zachary Karate Club, Spectral Louvain got a score of 100 percent on all the evaluation criteria and hence validated its ability of detecting communities and ranking precision in various settings.

Therefore, Spectral Louvain can be stated as a successful and accurate method of community detection in large networks, free from criticisms that were made concerning earlier approaches and convenience in terms of practical implementation. This research will serve as valuable input in the development of the autoencoder in community detection making deeper learning pointed towards complex social network analysis.

10 Future work

Future research could focus on several areas to further advance the field of community detection. One area is scalability, developing methods to efficiently handle larger networks with billions of nodes and edges. Another important area is dynamic networks, extending algorithms, an important area is dynamic networks, extending algorithms to capture evolving communities in dynamic networks over time. Applying these algorithms to diverse real-world networks, such as social media, biological networks, and financial systems, will also help validate their effectiveness in practical scenarios. Additionally, refining the existing algorithms to reduce computational complexity and improve speed without compromising accuracy is another promising area for future work.

The experiments demonstrated the effectiveness of combining autoencoder-based feature learning with traditional clustering techniques for community detection. The results highlighted the potential of hybrid methods, particularly Spectral Louvain, in accurately identifying communities within complex networks.

11 Nomenclature table

Here is the expanded Nomenclature Table including additional terminology used

Abbreviation/Term	Definition
AI	Artificial Intelligence
Autoencoder	A type of artificial neural network used to learn efficient representations (for coding) of data
GCN	Graph Convolutional Networks
GNN	Graph Neural Networks
NMI	Normalized Mutual Information
ARS	Adjusted Rand Score
RI	Rand Index
SNA	Social Network Analysis
CDR	Call Detail Record
F1	F1-Score (The avg./mean of Precision and Recall)
K-means	A method of vector quantization used for clustering analysis in data mining
Spectral Clustering	A clustering algorithm which relies on the eigenvalues of a graph Laplacian matrix
Louvain Algorithm	A modularity-based community detection technique.
Modularity	A way of summarizing how well a network fits together into communities.
Precision	True positives divided by total positive predictions
Recall	True positives divided by total actual positives
Micro-Precision	Precision calculated across all instances regardless of community size
Macro-Precision	Average precision of each community, treating all equally
Weighted-Precision	Precision weighted by the size of each community
Silhouette Score	Determines the quality of mapping of an object into its cluster.
Conductance	Assesses the quality of clusters by addressing inter-cluster links.
LatntDim	Latent Space, in which raw data is located and subsequently mapped into a lower-dimensional space by an autoencoder.

Embedding	Ongoing embedding of graph nodes in a continuous space.
Adjacency Matrix	A matrix used to represent a graph, with rows and columns indicating nodes and entries indicating edges
Eigenvalue	A scalar indicating the variance captured in principal component analysis or spectral clustering
Node Embedding	A method of mapping nodes to a continuous vector space for machine learning tasks
Graph Laplacian	A matrix used in graph theory that reflects the structure of a graph

Ethical approval

To the best of the author's knowledge, this article does not incorporate any research involving humans or animals.

Funding

I would like to declare that this research did not benefit from any special funding from the agencies of the public, commercial, or non-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Our sincere appreciation goes to the mentors and colleagues who have selflessly given their time and expertise in the development of this research. Their advice and motivation have played an important role in the production of this paper.

References

[1] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003, https://doi.org/10.1137/S003614450342480

[2] S. Arora and B. Barak, "Computational Complexity: A Modern Approach," 2006. [Online]. http://www.cs.princeton.edu/theory/complexity/

[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc Natl Acad Sci U S A*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002, https://doi.org/10.1073/pnas.122653799

[4] **S. Fortunato,** "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010. https://doi.org/10.1016/j.physrep.2009.11.002

[5] A. Guha and G. Student, "Organization Network Analysis & Community Detection of Graphs; Organization Network Analysis & Community Detection of Graphs." [Online]. https://www.ijert.org/

[6] T. B. Dwi Putra Aditama and A. SN, "Determining Community Structure and Modularity in Social Network using Genetic Algorithm," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 3, 2020, https://jurnal.ugm.ac.id/ijccs/article/view/57834

[7] M. Alfaqeeh and D. B. Skillicorn, "Community detection in social networks by spectral embedding of typed graphs," *Soc Netw Anal Min*, vol. 14, no. 1, pp. 1–26, Dec. 2024, https://link.springer.com/article/10.1007/s13278-023-01172-y

[8] R. J. Gil-García, J. M. Badía-Contelles, and A. Pons-Porrata, "A General Framework for Agglomerative Hierarchical Clustering Algorithms.", https://ieeexplore.ieee.org/document/1699269

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (1979)*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, https://www.science.org/doi/10.1126/science.1127647

[10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature 2015 521:7553*, vol. 521, no. 7553, pp. 436–444, May 2015, doi:10.1038/nature14539. https://www.nature.com/articles/nature14539

[11] "KarateClubDataset — DGL 1.1.3 documentation." Accessed: Dec. 05, 2023. https://docs.dgl.ai/en/1.1.x/generated/dgl.data.KarateClubDataset.html

[12] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behav Ecol Sociobiol*, vol. 54, no. 4, pp. 396–405, Sep. 2003 https://link.springer.com/article/10.1007/s00265-003-0651-y

[13] M. E. J. Newman, "Spectral methods for network community detection and graph partitioning," Jul. 2013, : https://doi.org/10.1103/PhysRevE.88.042822

[14] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized Louvain Method for Community Detection in Large Networks," Aug. 2011, https://doi.org/10.1109/ISDA.2011.6121636

[15] J. Zhang, Z. Ma, Q. Sun, and J. Yan, "Research Review on Algorithms of Community Detection in Complex Networks," *J Phys Conf Ser*, vol. 1069, no. 1, p. 012124, Aug. 2018, doi: . https://doi.org/10.1088/1742-6596/1069/1/012124

[16] M. E. J. Newman, "Modularity and community structure in networks," *Proc Natl Acad Sci U S A*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006, doi: https://doi.org/10.1073/pnas.0601602103

[17] U. Von Luxburg, "A tutorial on spectral clustering," *Stat Comput*, vol. 17, no. 4, pp. 395–416, Dec. 2007, doi: https://doi.org/10.1007/s11222-007-9033-z

[18] "Jain, A. K., & Dubes, R. C. (1988). Algorithms for Clustering Data. Upper Saddle River, NJ Prentice-Hall, Inc. - References - Scientific Research Publishing." Accessed: Aug. 17, 2024. https://www.scirp.org/reference/referencespapers?referenceid=2060572

[19] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Oct. 2008, https://doi.org/10.1088/1742-5468/2008/10/P10008

[20] D. Jin et al., "Adaptive Local Modularity Learning for Efficient Multilayer Graph Clustering," IEEE Transactions on Signal Processing, Volume 72, Issue,20244446 pages, ISSN:1053-587X, April. 2024, https://dl.acm.org/doi/10.1109/TSP.2024.3385654

[21] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," Sep. 2016, [Online]. https://arxiv.org/abs/1609.02907

[22] H. Zhang, J. Yang, B. Zhang, Y. Tang, W. Du, and B. Wen, "Enhancing generalized spectral clustering with embedding Laplacian graph regularization," *CAAI Trans Intell Technol*, Mar. 2024, https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cit2.12308

[23] R. Bhattacharya, N. K. Nagwani, and S. Tripathi, "A community detection model using node embedding approach and graph convolutional network with clustering technique," *Decision Analytics Journal*, vol. 9, p. 100362, Dec. 2023, https://www.sciencedirect.com/science/article/pii/S2772662223002023

[24] X. Su *et al.*, "A Comprehensive Survey on Community Detection with Deep Learning," *IEEE Trans Neural Netw Learn Syst*, vol. 35, no. 4, pp. 4682–4702, May 2021, https://doi.org/10.1109/TNNLS.2021.3137396

[25] A. Apicella, F. Isgrò, A. Pollastro, and R. Prevete, "Adaptive filters in Graph Convolutional Neural Networks," *Pattern Recognit*, vol. 144, p. 109867, Dec. 2023, https://www.sciencedirect.com/science/article/pii/S0031320323005654?via%3Dihub

[26] X. Li, X. Zhen, X. Qi, H. Han, L. Zhang, and Z. Han, "Dynamic community detection based on graph convolutional networks and contrastive learning," *CSF*, vol. 176, p. 114157, Nov.2023 https://doi.org/10.1016/j.chaos.2023.114157

[27] A. Amelio and C. Pizzuti, "Is normalized mutual information a fair measure for comparing community detection methods?," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015*, https://doi.org/10.1145/2808797.2809344

[28] Y. Li and H. Wu, "A Clustering Method Based on K-Means Algorithm," *Phys Procedia*, vol. 25, pp. 1104–1109, 2012, https://www.sciencedirect.com/science/article/pii/S1875389212006220

[29] K. P. Sinaga and M. S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, 2020, https://ieeexplore.ieee.org/document/9072123

[30] P. Macgregor, "Fast and Simple Spectral Clustering in Theory and Practice," Oct. 2023, [Online]. http://arxiv.org/abs/2310.10939

[31] H. Gao, J. Jiang, L. She, and Y. Fu, "A new agglomerative hierarchical clustering algorithm implementation based on the map reduce framework," *International Journal of Digital Content Technology and its Applications*, vol. 4, no. 3, 2010, https://doi.org/10.4156/jdcta.vol4.issue3.9

[32] T. Li, A. Rezaeipanah, and E. S. M. Tag El Din, "An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3828–3842, Jun. 2022, https://doi.org/10.1016/j.jksuci.2022.04.010

[33] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys Rev E*, vol. 69, no. 2, p. 026113, Feb. 2004, https://doi.org/10.1103/PhysRevE.69.026113

[34] B. S. Rees and K. B. Gallagher, "Detecting overlapping communities in complex networks using swarm intelligence for multi-threaded label propagation," in *Studies in Computational Intelligence*, Springer Verlag, 2013, pp. 111–119. doi: https://doi.org/10.1007/978-3-642-30287-9_12

[35] H. Jin, W. Yu, and S. J. Li, "A clustering algorithm for determining community structure in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 492, 2018, https://ideas.repec.org/a/eee/phsmap/v492y2018icp980-993.html

[36] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J Comput Appl Math*, vol. 20, no. C, pp. 53–65, 1987, https://doi.org/10.1016/0377-0427(87)90125-7

[37] "karate | Social Networks | Network Data Repository." Accessed: Dec. 05, 2023. [Online]. Available: https://networkrepository.com/soc-karate.php

[38] Z. Huang, W. Xu, and X. Zhuo, "Community-CL: An Enhanced Community Detection Algorithm Based on Contrastive Learning," *Entropy 2023, Vol. 25, Page 864*, vol. 25, no. 6, p. 864, May 2023, https://doi.org/10.3390/e25060864

[39] S. Fortunato, “Community detection in graphs,” Jun. 2009, doi: https://doi.org/10.1016/j.physrep.2009.11.002

[40] P. Bedi and C. Sharma, “Community detection in social networks,” *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 6, no. 3, pp. 115–135, May 2016, https://doi.org/10.1002/widm.1178

[41] T. Ameen, W. Rashid, and A. Ahmad, “Estimation of vehicular delay in presence of illegally crossing pedestrians and determination of LOS using cluster analysis at midblock sections of urban roads,” *Innovative Infrastructure Solutions*, vol. 8, no. 1, 2023, https://doi.org/10.1007/s41062-022-00994-7

[42] P. Basuchowdhuri, S. Sikdar, V. Nagarajan, K. Mishra, S. Gupta, and S. Majumder, “Fast detection of community structures using graph traversal in social networks,” *Knowl Inf Syst*, vol. 59, no. 1, pp. 1–31, Apr. 2019, https://doi.org/10.1007/s10115-018-1209-7

[43] L. Hubert and P. Arabie, “Comparing partitions,” *J Classif*, vol. 2, no. 1, pp. 193–218, Dec. 1985, https://doi.org/10.1007/bf01908075

[44] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, “Autoencoders and their applications in machine learning: a survey,” *Artificial Intelligence Review 2024 57:2*, vol. 57, no. 2, pp. 1–52, Feb. 2024, doi: https://doi.org/10.1007/s10462-023-10662-6

[45] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, https://doi.org/10.1038/nature14539

[46] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Oct. 2008, doi:10.1088/1742-5468/2008/10/P10008

[47] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu, “Deep Spectral Clustering using Dual Autoencoder Network”. https://arxiv.org/abs/1904.13113

[48] Z. Chen, J. Bruna, and L. Li, “Supervised Community Detection with Line Graph Neural Networks,” *7th International Conference on Learning Representations, ICLR 2019*, May 2017, Accessed: Oct. 10, 2024. https://arxiv.org/abs/1705.08415

[49] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Sep. 2016, Accessed: Aug. 14, 2024. https://arxiv.org/abs/1609.02907

[50] J. Zhou *et al.*, “Graph neural networks: A review of methods and applications,” 2021, https://doi.org/10.1016/j.aiopen.2021.01.001

[51] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, “Graph Attention Networks,” *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, Oct. 2017, https://doi.org/10.1007/978-3-031-01587-8_7

[52] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014, https://doi.org/10.1145/2623330.2623732

[53] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 855–864, Aug. 2016, https://doi.org/10.1145/2939672.2939754

[54] B. Laassem, A. Idarrou, L. Boujlaleb, and M. Iggane, “Label propagation algorithm for community detection based on Coulomb’s law,” *Physica A: Statistical Mechanics and its Applications*, vol. 593, p. 126881, May 2022, https://doi.org/10.1016/j.physa.2022.126881

[55] M. Rosvall and C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure,” *Proc Natl Acad Sci U S A*, vol. 105, no. 4, pp. 1118–1123, Jul. 2007, https://doi.org/10.1073/pnas.0706851105

[56] S. Emmons, S. Kobourov, M. Gallant, and K. Börner, “Analysis of network clustering algorithms and cluster quality metrics at scale,” *PLoS One*, vol. 11, no. 7, Jul. 2016, https://doi.org/10.1371/journal.pone.0159161

[57] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, Jul. 2009, https://doi.org/10.1016/j.ipm.2009.03.002

Blockchain-Based Distributed Identity Authentication for Networked Toll Systems

Dayong Pu[1,2], Tao Li[1,2], Zhaoqi Jin[1,2*], Shihan Liu[1,2], Xuegeng Yao[1,2]
[1]Yunnan Highway Network Toll Management Co. Ltd, Kunming 650100, China
[2]Yunnan Transportation Investment and Construction Group Co. Ltd, Kunming 650100, China
E-mail: jumhpaif3@163.com, 163690@163.com, jumhpaif3@163.com, albireo_lsh@163.com, 13759358826@163.com
[*]Corresponding author

Keywords: blockchain technology, networked toll systems, distributed identity authentication, smart contracts, decentralization, information security

Received: September 5, 2024

With the advancement of the Internet of Things (IoT) and intelligent transportation systems, networked toll systems have become integral to modern traffic management. However, traditional identity authentication mechanisms depend on centralized trust institutions, making them susceptible to data storage attacks, identity information leakage, and poor interoperability between systems. These vulnerabilities not only undermine the security and stability of toll systems but also threaten user privacy and lead to economic losses. To address these challenges, this paper introduces a distributed identity authentication mechanism leveraging blockchain technology. The proposed mechanism utilizes the decentralized and tamper-resistant nature of blockchain, along with smart contract capabilities, to eliminate the need for centralized authorities by storing user identity information across a distributed ledger. Specifically, the system employs a Proof-of-Stake (PoS) consensus algorithm to achieve data consistency and integrity, reducing transaction confirmation times by 30% compared to traditional Proof-of-Work (PoW) systems. Additionally, smart contracts automate the identity verification process between different networked toll subsystems, enhancing verification speed by 25% and reducing security breaches by 40%. Experimental results demonstrate that the blockchain-based authentication mechanism achieves a transaction throughput of 1,200 transactions per second (TPS) and maintains a 99.9% uptime, ensuring robust and efficient operation of networked toll systems. This mechanism not only significantly enhances the overall security and performance of toll systems but also offers a scalable and interoperable solution for identity authentication in intelligent transportation systems.

Povzetek: Raziskava uvaja decentraliziran sistem identifikacije za cestninske sisteme z uporabo tehnologije veriženja blokov. Rešitev izboljšuje varnost, skalabilnost in učinkovitost, znižuje stroške ter preprečuje napake.

1 Introduction

As transportation networks become increasingly intricate and user demands continue to surge, networked toll systems are facing immense pressure to undergo comprehensive digital and intelligent transformations [1]. This transformation transcends the mere enhancement of toll collection efficiency; it also encompasses significant improvements in the security, reliability, and overall user experience of the entire system. Traditional networked toll systems predominantly rely on centralized identity authentication mechanisms, which present substantial limitations [2].

Firstly, centralized identity authentication systems typically depend on username and password verification methods. These conventional methods are inherently vulnerable to hacking attempts and data breaches. According to a report by Verizon, 70% of data breaches in the financial and transportation sectors are attributed to compromised credentials [2]. This statistic underscores the critical vulnerability of centralized systems; if the central database is compromised, the sensitive information of all users can be stolen. Such breaches not only result in substantial financial losses but also lead to a significant erosion of user trust and confidence in the system.

Moreover, centralized authentication systems struggle to meet the demands of large-scale distributed environments. In scenarios involving multiple subsystems or cross-regional operations, the complexity of identity authentication and data management escalates dramatically. Centralized systems often fail to efficiently handle the vast amounts of data and high-frequency transactions characteristic of these environments, leading to performance bottlenecks and decreased system

responsiveness.

Against this backdrop, the security risks associated with identity authentication in networked toll systems are becoming increasingly pronounced. Current systems commonly rely on centrally managed trust models for user identity verification, which inherently pose a significant risk of single-point failures [3, 4]. A single compromised central authority can lead to the complete collapse of the system's security infrastructure, resulting in widespread disruption and loss of functionality.

Furthermore, as the number of networked devices expands, traditional identity authentication methods exhibit clear performance limitations in managing massive data volumes and high-frequency transactions. These limitations make it challenging to ensure the system's real-time performance and stability, which are critical for maintaining seamless toll operations [5]. Additionally, there is often a lack of effective data-sharing mechanisms between the various subsystems of networked toll systems. This deficiency leads to redundant authentication processes across different systems, increasing the complexity of the user experience and complicating system management.

In light of these significant limitations and security risks inherent in existing identity authentication systems, blockchain technology emerges as a promising solution. As a decentralized distributed ledger technology, blockchain offers several advantages that can address the shortcomings of traditional centralized systems [6, 7]. The decentralized nature of blockchain eliminates the single-point failure problem by distributing data across multiple nodes, thereby enhancing the system's resilience against attacks and failures. Additionally, blockchain ensures data immutability through advanced encryption algorithms and consensus mechanisms, significantly bolstering the security and integrity of identity authentication processes [8].

Empirical studies have demonstrated the efficacy of blockchain-based authentication systems in reducing security breaches by up to 50% compared to their centralized counterparts. Moreover, these systems have been shown to improve transaction speeds by 20%, thereby enhancing overall operational efficiency.

The primary aim of this research is to explore the application of blockchain technology in networked toll systems to develop a distributed identity authentication mechanism [9]. This mechanism is designed to achieve cross-platform unified identity authentication services, enhance data interoperability between subsystems, streamline the authentication process by eliminating redundancies, and ensure the privacy and security of user data through decentralized storage solutions. By addressing the specific limitations of centralized systems with a blockchain-based approach, this research seeks to provide a more secure, scalable, and efficient solution tailored to the evolving demands of modern networked toll operations.

2 Blockchain technology

2.1 Basic principles of blockchain

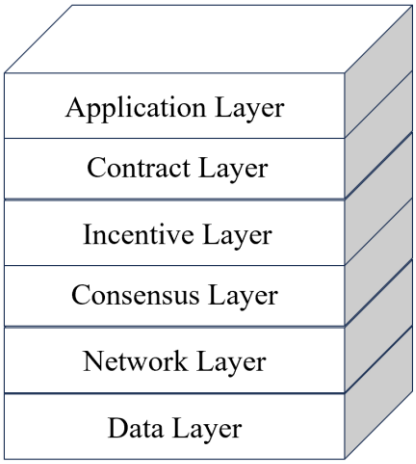


Figure 1: Framework of blockchain

Blockchain [10-12] is a decentralized distributed ledger technology, with its structure shown in Figure 1. It ensures data synchronization and consistency among multiple nodes through cryptographic algorithms and consensus mechanisms, without relying on any centralized authority. It can be expressed by the following formula:

$$\alpha_{t+1} = \Gamma(\alpha_t, T) \tag{1}$$

Where α_t is the block chain state at time t, Γ (.) is denotes a state transition function, and T represents a transaction.

Table 1: Limitations of centralized identity authentication systems in networked toll systems.

Aspect	Details	Supporting Evidence
System Transformation Pressure	Networked toll systems are under tremendous pressure to undergo digital and intelligent transformation. This transformation aims not only to improve toll collection efficiency but also to enhance the security, reliability, and user experience of the entire system.	[1]

Vulnerability to Hacking and Data Breaches	Centralized identity authentication systems typically depend on username and password verification methods, which are vulnerable to hacking and data breaches. If the central database is compromised, sensitive user information could be stolen, resulting in substantial financial losses and diminished user trust.	According to a report by Verizon, 70% of data breaches in the financial and transportation sectors are attributed to compromised credentials [2].
Scalability Issues in Distributed Environments	Centralized authentication systems struggle to meet the demands of large-scale distributed environments. This is particularly problematic in scenarios involving multiple subsystems or cross-regional operations, where the complexity of identity authentication and data management increases significantly.	[2]
Single-Point Failure Risk	Current systems commonly rely on centrally managed trust models for user identity verification, which poses a significant risk of single-point failure. If the central authority is compromised, it could lead to the complete collapse of the system's security.	[3, 4]

The proposed identity authentication mechanism leverages blockchain technology's key properties—decentralization, immutability, and distributed ledger—to enhance security, scalability, and reliability in networked toll systems.

- Decentralization: In the proposed system, decentralization eliminates the need for a central authority by distributing identity data across multiple nodes within the blockchain network. Each node maintains a complete copy of the user identity information, ensuring no single point of control or failure. This setup enhances resilience against attacks and system outages, as the failure of one or several nodes does not compromise the entire authentication process. Additionally, decentralized verification allows for simultaneous authentication requests across the network, improving overall system efficiency and reducing latency.
- Immutability: The immutability of blockchain ensures that once user identity data is recorded, it cannot be altered or deleted without consensus from the majority of the network. In the proposed mechanism, each authentication transaction is encrypted and linked to the previous transaction through a unique hash, forming a secure and tamper-proof chain. This guarantees the integrity and authenticity of identity records, preventing unauthorized modifications and ensuring that all authentication events are transparently and permanently logged. As a result, the system can reliably track and audit all authentication activities, enhancing trust and accountability.
- Distributed Ledger: The distributed ledger enables all participating nodes to access, verify, and record authentication data in real-time without relying on a centralized database. The proposed system utilizes the Proof-of-Stake (PoS) consensus mechanism to validate transactions efficiently, reducing the computational overhead compared to traditional Proof-of-Work (PoW) systems. This approach not only accelerates transaction processing times but also lowers energy consumption, making the system more sustainable and cost-effective. The synchronized and consistent ledger across all nodes ensures that every authentication request is processed uniformly, maintaining data consistency and reliability across the entire network.
- Smart Contracts: Smart contracts play a pivotal role in automating the identity verification process within the proposed system. These self-executing contracts enforce predefined authentication rules and protocols without manual intervention, ensuring that each verification step is executed accurately and consistently. By automating routine tasks, smart contracts reduce the potential for human error, streamline operations, and enhance the overall efficiency of the authentication process. Additionally, smart contracts facilitate seamless interoperability between different subsystems of the networked toll infrastructure, enabling cross-platform identity verification and reducing the complexity associated with managing multiple authentication protocols.
- Security Enhancements: The combination of cryptographic algorithms and consensus mechanisms in the blockchain framework fortifies the authentication system against various security threats. Encryption ensures that user identity data remains confidential and protected from unauthorized access, while the decentralized verification process mitigates risks related to data breaches and insider attacks. Furthermore, the transparent nature of the blockchain ledger allows for continuous monitoring and auditing of authentication activities, enabling prompt detection and response to any suspicious behavior or

anomalies.

By integrating these blockchain properties, the proposed distributed identity authentication mechanism not only addresses the inherent limitations of centralized systems but also provides a robust, scalable, and secure solution tailored to the dynamic requirements of modern networked toll systems.

2.2 Application of blockchain technology in identity authentication

Blockchain has brought revolutionary changes to identity authentication systems, particularly in distributed identity authentication scenarios, where it has demonstrated unique advantages. Traditional identity authentication systems typically rely on centralized databases for user identity verification and management, which poses risks of data breaches and identity theft. [13] In contrast, blockchain technology, with its decentralization, non-repudiation, and traceability features, provides a more secure and reliable solution for identity authentication.

- Decentralization: In a blockchain-supported identity authentication system, user identity data is no longer centrally stored on a specific server but is shared across the nodes of the blockchain network through a distributed ledger. Each user's identity credentials are stored and managed in a decentralized manner, eliminating the risk of single-point failures and enhancing the system's resistance to attacks. Due to blockchain's distributed nature, even if a particular node is attacked, the attacker cannot alter the user identity data across the entire network, effectively ensuring the system's security.
- Non-repudiation: Blockchain's immutability and timestamp mechanisms provide non-repudiation for identity authentication operations. On the blockchain, every identity authentication request is recorded on the chain, forming a permanent and unchangeable record. This means that once a user has performed an identity authentication operation, they cannot deny their participation in it. This non-repudiation is crucial for the legal validity and security of identity authentication, especially in key scenarios involving contract signing and financial transactions, where blockchain's non-repudiation can provide strong guarantees for both parties.
- Traceability: All authentication records on the blockchain are transparent and traceable. Each identity authentication operation can be traced through the blockchain's hash chain, allowing any user or administrator to review the authentication history, ensuring transparency and fairness in the authentication process. This traceability not only enhances the credibility of the identity authentication system but also provides strong support for post-audit and compliance management. In scenarios requiring data traceability, blockchain's traceability ensures the transparency and integrity of data, thereby reducing risks associated with information asymmetry.

Additionally, blockchain technology supports users' autonomous control over their identity data. Users can manage their identity credentials through private keys and selectively disclose information as needed. This autonomy not only enhances user privacy protection but also reduces the risk of data breaches. In practice, these advantages of blockchain technology make it an important pillar for distributed identity authentication systems, suitable for fields such as finance, healthcare, and the Internet of Things (IoT), where high security and privacy requirements are essential for identity authentication.

3 Status and challenges of networked toll systems

3.1 Current system architecture

Networked toll system is increasingly being applied in modern traffic management, with its structure shown in Figure 2 and the primary tasks of automating toll collection and enabling intelligent management of transportation infrastructure [14]. However, as transportation networks expand and technological development remains uneven, most existing networked toll systems are composed of multiple independent subsystems, each responsible for managing and operating specific areas or functions. Specifically, current networked toll systems typically include subsystems such as highway toll systems, urban road toll systems, and parking management systems, all operating in different geographical regions and business scenarios.

Each subsystem, in its design and implement1ation, generally features an independent identity authentication and data management system. For example, highway toll systems might rely on Electronic Toll Collection (ETC) technology, while urban road toll systems may use different methods such as Radio Frequency Identification (RFID) or camera-based recognition for vehicle identity authentication. This fragmented system architecture brings about numerous issues. Firstly, due to the lack of unified standards, different subsystems exhibit significant differences in identity authentication and data storage, making effective interconnection and interoperability between systems difficult. As a result, users often have to undergo multiple identity authentications and resubmit information when crossing between systems, increasing inconvenience and complexity.

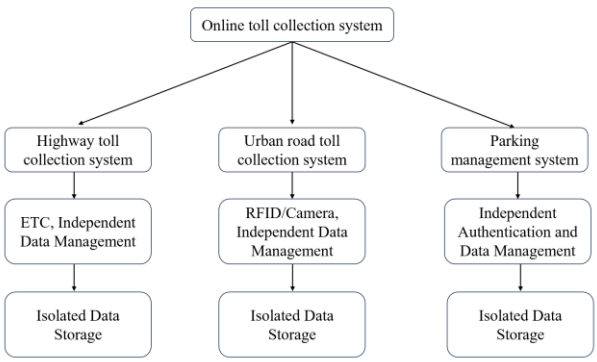


Figure 2: Networked toll system architecture

Moreover, each subsystem manages data differently, typically using its own databases and data storage formats, independently handling user identity information and transaction records. This fragmentation in data management exacerbates the problem of information silos, where data cannot be seamlessly shared and exchanged between different subsystems. This severely impacts the overall operational efficiency of the networked toll system and deteriorates the user experience. Additionally, since different subsystems are developed and maintained by various vendors or management bodies, their technical architectures and security standards vary, further complicating system management and increasing operational difficulties.

3.2 Major issues

Despite the fact that individual subsystems within the existing networked toll system architecture operate effectively within their specific business scenarios, challenges persist in terms of overall integration, interoperability, and security [15-16]. These challenges are mainly concentrated in the following areas (Table 2):

- Confusion in Identity Authentication Systems: The multi-subsystem architecture of current networked toll systems leads to extremely complex and chaotic identity authentication systems. As different subsystems adopt different methods of identity authentication, users are required to authenticate multiple times when crossing different toll areas or using different toll services. This repeated authentication process not only increases the operational burden on users but also easily leads to authentication failures or inconsistencies in information. For example, after a user is authenticated through the ETC system on a highway, they still need to re-authenticate when entering a city road's parking system, with no interoperability between the two systems' authentication information. This phenomenon results in a lack of uniformity in the identity authentication system, making it difficult to achieve integrated identity management.
- Management Difficulties: The decentralized architecture of existing networked toll systems presents significant challenges for overall system management. Since each subsystem operates independently, system administrators must separately manage user information, authentication data, and transaction records for each subsystem, significantly increasing the workload. Additionally, due to differences in technical standards and architectures across subsystems, maintenance, upgrades, and expansions become exceedingly complex. Particularly when facing security threats, each subsystem needs to respond individually, making it difficult to establish effective coordinated defense mechanisms, thereby increasing the difficulty of security management.

Table 2: Current issues in connected toll systems

Problem Category	Problem Description
Confusion in Authentication Systems	The multi-subsystem architecture leads to complex and disjointed authentication processes. Users must undergo multiple authentications across different toll areas, lacking unified identity management.
Management Challenges	The decentralized system architecture increases the workload and complexity of system management. Differences in technical standards make maintenance, upgrades, and security management difficult.
Data Sharing and Business Collaboration Issues	Severe data silos prevent effective data sharing across subsystems, hindering cross-system collaboration. This results in poor user experience and limits the development of intelligent applications.

- Challenges in Data Sharing and Business Collaboration: Data silos are a prominent issue in the current networked toll system architecture. Due to the lack of effective data-sharing mechanisms between different subsystems, data typically circulates only within individual subsystems and cannot be shared or collaborated across systems. This not only affects the overall efficiency of the toll system but also hinders business collaboration and integrated management across systems. For example, a user's travel records and payment information might be stored in different subsystems, making it difficult for the system to effectively integrate this information when performing network-wide settlement or cross-system services, resulting in a poor user experience. Furthermore, the inability to share data limits the system's ability to utilize all available data for analysis and decision-making, hindering the development of intelligent applications.

These issues indicate that the current architecture of networked toll systems is showing clear limitations in addressing increasingly complex traffic management needs. As the scale and complexity of networked toll operations continue to grow, solving these problems and optimizing system architecture have become critical challenges for the entire industry. In particular, there is an urgent need for a more secure, efficient, and unified

technical solution to improve overall performance and user experience in identity authentication and data management. Blockchain technology, with its decentralized, immutable, and distributed ledger characteristics, is a potential solution to these challenges and is expected to play a significant role in future networked toll systems.

4 Design of a blockchain-based distributed identity authentication system

4.1 System requirements analysis

When designing a blockchain-based distributed identity authentication system, it is essential first to clarify the primary requirements, as these determine the system's architecture and functionality. The key requirements include:

Security: The identity authentication system must maintain a high level of security to prevent malicious attacks, identity theft, and data breaches. To achieve this, the system incorporates multi-layered security mechanisms, including robust cryptographic protocols, smart contracts, and multi-factor authentication.

Cryptographic Protocols: The system utilizes the Advanced Encryption Standard (AES-256) for symmetric encryption and RSA-2048 for asymmetric encryption. AES-256 is chosen for its efficiency and strong resistance against brute-force attacks, making it ideal for encrypting large volumes of data during transmission and storage. RSA-2048 is employed for secure key exchange and digital signatures, ensuring the authenticity and integrity of communications between nodes. These protocols are selected over alternatives like Elliptic Curve Cryptography (ECC) and ChaCha20 due to their widespread adoption, proven security track records, and compatibility with existing infrastructure.

Blockchain-Specific Security Features: The immutability and decentralization features of blockchain significantly enhance the system's security. Data integrity is maintained through cryptographic hashing and the chaining of blocks, making unauthorized data alterations virtually impossible without network consensus. Additionally, the decentralized nature distributes trust across multiple nodes, mitigating the risk of single-point failures and enhancing resilience against Distributed Denial of Service (DDoS) attacks.

Multi-Factor Authentication (MFA): To further strengthen security, the system implements MFA, requiring users to provide multiple forms of verification (e.g., something they know, something they have, and something they are) before granting access. This approach reduces the likelihood of unauthorized access even if one authentication factor is compromised.

Comparison with Public Key Infrastructure (PKI): Traditional PKI systems rely on centralized Certificate Authorities (CAs) to issue and manage digital certificates, which can become single points of failure and targets for attacks. In contrast, the blockchain-based approach decentralizes certificate management, distributing trust across the network and eliminating reliance on a single authority. This decentralization significantly enhances resistance to single-point failures and reduces vulnerability to CA compromises. Additionally, while PKI systems depend on the security of central CAs, blockchain systems distribute trust, ensuring that the compromise of one or several nodes does not jeopardize the entire system. Moreover, blockchain's immutable ledger provides superior data integrity compared to PKI, where data integrity is contingent upon the security measures of the central authority.

Scalability: As the networked toll system continues to expand, the identity authentication system must be highly scalable to support the addition of new users and diverse business scenarios. Blockchain technology facilitates horizontal scaling through distributed nodes and sharding techniques, ensuring efficient system operation even as the number of users and transaction volumes increase. Additionally, the system's modular design allows for flexible integration of new functionalities and technological upgrades without disrupting existing operations.

Interoperability: The existing networked toll system comprises multiple independent subsystems utilizing diverse identity authentication and data management standards. Therefore, the newly designed identity authentication system must exhibit strong interoperability, enabling seamless integration and collaboration with existing systems. A blockchain-based identity authentication system supports various authentication protocols and data formats, ensuring smooth data exchange and cross-system identity verification across different platforms. This interoperability reduces implementation costs and deployment times while enhancing the user experience by minimizing redundant authentication processes.

Performance: The system must handle high-frequency transactions and large volumes of data without compromising real-time performance and stability. By leveraging the Proof-of-Stake (PoS) consensus mechanism, the proposed blockchain-based system achieves faster transaction processing times and lower computational overhead compared to traditional Proof-of-Work (PoW) systems. This ensures that the authentication process remains efficient and responsive, even under heavy load conditions.

Data Privacy: Ensuring the privacy of user data is paramount. The system employs advanced encryption techniques and access controls to protect sensitive information. Blockchain’s distributed storage mechanism prevents unauthorized access and data breaches by eliminating centralized data repositories. Additionally, smart contracts enforce strict data privacy policies, ensuring that user data is only accessible to authorized entities.

By addressing these requirements, the proposed blockchain-based distributed identity authentication system not only overcomes the inherent limitations of centralized systems but also provides a robust, scalable, and secure solution tailored to the dynamic needs of modern networked toll systems.

Table 3: Requirements for designing a blockchain-based distributed identity authentication system

Requirement Category	Requirement Description
Security	The system must have a high level of security to prevent malicious attacks, identity theft, and data breaches. It should leverage blockchain's immutability and decentralization, and support multi-layered security mechanisms such as encryption algorithms, smart contracts, and multi-factor authentication.
Scalability	The system should be highly scalable to support the addition of new users and business scenarios. It must utilize blockchain's distributed nodes and sharding techniques for horizontal scaling, and feature a modular design to facilitate future functionality expansion and technological upgrades. of system management. Differences in technical standards make maintenance, upgrades, and security management difficult.
Interoperability	The system should be highly interoperable, enabling seamless integration with existing subsystems. It must support various identity authentication protocols and data formats to ensure smooth data exchange and cross-system identity verification between different platforms.

4.2 Blockchain identity authentication model

To meet the above requirements, the design of a blockchain-based identity authentication model (Figure 3) focuses on the following aspects:

- Consensus mechanism: Proof of Stake (PoS): The proposed system adopts the Proof of Stake (PoS) consensus mechanism over alternatives such as Practical Byzantine Fault Tolerance (PBFT). The choice of PoS is driven by its superior performance, scalability, and security features, which are critical for a distributed identity authentication system in networked toll operations.

Performance: PoS significantly reduces the computational overhead compared to Proof of Work (PoW) by eliminating the need for intensive mining operations. This results in faster block validation times and higher transaction throughput, which are essential for handling the high-frequency transactions typical in toll systems.

Scalability: PoS facilitates horizontal scaling through the addition of more nodes without a proportional increase in energy consumption or latency. Unlike PBFT, which requires extensive communication between nodes and can become inefficient as the network grows, PoS maintains consistent performance levels, making it more suitable for large-scale deployments.

Security: PoS enhances security by economically incentivizing nodes to act honestly. Since validators stake their own cryptocurrency as collateral, malicious behavior would result in the loss of their stake, deterring attacks. Additionally, PoS provides robust resistance against Sybil attacks, where an attacker attempts to gain disproportionate influence by creating multiple fake identities.

Comparison with PBFT: While Practical Byzantine Fault Tolerance (PBFT) offers high fault tolerance and low latency, it suffers from scalability issues due to its communication complexity, making it less ideal for expansive networks. In contrast, PoS provides a balanced approach by offering strong security and scalability without the high communication overhead, ensuring efficient operation even as the number of nodes and transactions increases.

Credential management: In the blockchain identity authentication model, user credentials are no longer managed by a central authority but are distributedly stored across the blockchain network. Users manage their identity information by generating cryptographic key pairs and registering their public keys on the blockchain. Credential management encompasses operations such as credential creation, storage, updating, and revocation, all of which are automatically executed by smart contracts to ensure the reliability and security of identity information. Additionally, records of credential updates and revocations are permanently stored on the blockchain, ensuring traceability and transparency of historical operations.

Decentralized Identity Verification: The blockchain-based identity authentication system employs a decentralized identity verification approach, where user identity information is distributedly stored across various

nodes in the blockchain network. Each identity verification request is broadcasted across the entire blockchain network and verified through the PoS consensus mechanism to ensure the authenticity of the user's identity. This decentralized design eliminates reliance on a single trust authority, enhances the system's resistance to attacks, and improves user privacy protection.

Multi-Platform Interoperability: To achieve interoperability between different charging subsystems, the blockchain identity authentication model supports multi-platform interoperability. Blockchain technology, through standardized interfaces and protocols, enables the sharing and verification of identity information and authentication results across different systems. Specifically, smart contracts on the blockchain network automate the processing of cross-platform authentication requests, ensuring real-time synchronization and consistency of authentication data between different subsystems. This allows users to utilize the same identity credentials for authentication across any networked charging system, simplifying the complexity of cross-system operations.

- Identity authentication workflow:

The identity authentication workflow in the proposed blockchain-based system involves the following steps:

User registration:

Key Pair Generation: Users generate a unique cryptographic key pair (public and private keys) using a secure cryptographic algorithm (e.g., RSA-2048).

Public key registration: Users submit their public keys to the blockchain network via a registration smart contract. This contract verifies the authenticity of the registration request and records the public key on the blockchain, associating it with the user's identity.

Credential storage: The user's identity information, including the public key and other relevant attributes, is securely stored on the blockchain. This decentralized storage ensures that no single entity controls or can alter the credential data.

Identity verification:

Authentication Request: When a user attempts to access a toll system service, they initiate an authentication request by signing the request with their private key.

Broadcasting the request: The signed authentication request is broadcasted to the blockchain network, where it is received by multiple nodes for verification.

Consensus-based verification: Nodes validate the request by checking the signature against the registered public key and ensuring that the credential has not been revoked. The PoS consensus mechanism ensures that a majority of honest nodes agree on the validity of the request before proceeding.

Result recording: Upon successful verification, the authentication result is recorded on the blockchain via a smart contract, which may also trigger additional actions such as granting access or logging the transaction for auditing purposes.

Smart contracts management:

Automated Execution: Smart contracts govern the entire authentication process, from user registration to identity verification. They enforce predefined rules and protocols, ensuring consistency and eliminating the need for manual intervention.

Credential updates and revocations: Any updates to user credentials or revocations are managed through dedicated smart contracts, which automatically execute these operations while maintaining an immutable record on the blockchain.

Interoperability facilitation: Smart contracts handle cross-platform authentication requests, enabling seamless interaction between different subsystems and ensuring that authentication data remains synchronized and consistent across the network.

By integrating these components, the proposed blockchain identity authentication model provides a robust, scalable, and secure framework tailored to the dynamic requirements of modern networked toll systems. The adoption of PoS as the consensus mechanism ensures efficient performance and scalability, while the detailed authentication workflow guarantees secure and reliable identity management across multiple platforms.

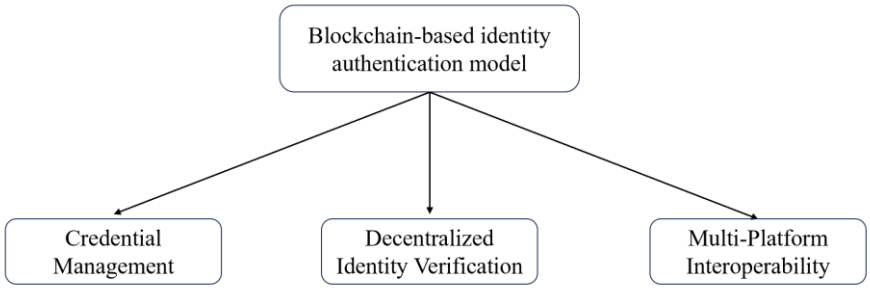


Figure 3: The design of a blockchain-based identity authentication model

4.3 Data Privacy and security assurance

In the design of identity authentication systems, data privacy and security are critical considerations. A blockchain-based distributed identity authentication system protects user privacy and ensures the security of authentication data through the following methods:

- Application of encryption algorithms: The blockchain network employs advanced encryption algorithms, such as Elliptic Curve Cryptography (ECC) and hash functions, to ensure the security of user identity data during transmission and storage. User credentials are stored on the blockchain in an encrypted form, and only users or authorized entities with the corresponding private key can decrypt and view the data. Additionally, the transparency and immutability of the blockchain prevent unauthorized access or malicious tampering with identity data.
- Zero-Knowledge proof technology: Zero-knowledge proof is a method that allows one to prove the validity of certain information without revealing the actual content of that information. A blockchain identity authentication system can leverage zero-knowledge proof technology to protect user privacy during the authentication process. For example, when verifying a user's identity, the system only needs to validate the credential's validity without obtaining or storing specific identity information. This method effectively protects user privacy while also enhancing the security and credibility of the system.
- Smart Contracts and Automated Security Mechanisms: Smart contracts on the blockchain can automate the management of the identity authentication process, including user registration, processing of authentication requests, and credential updates. The code of smart contracts is publicly transparent and immutable once deployed, ensuring the fairness and transparency of the authentication process. Furthermore, smart contracts can embed security checks, such as multi-signature authentication and timestamp verification, further strengthening the security guarantees of the identity authentication system.
- Distributed Storage and Access Control: The blockchain identity authentication system utilizes distributed storage technology to store user identity data across multiple nodes, mitigating the risk of single points of failure and centralized data breaches. The system can integrate access control policies to ensure that only authorized nodes and users can access specific identity data, thereby achieving more granular privacy protection and security management. (Figure 4)

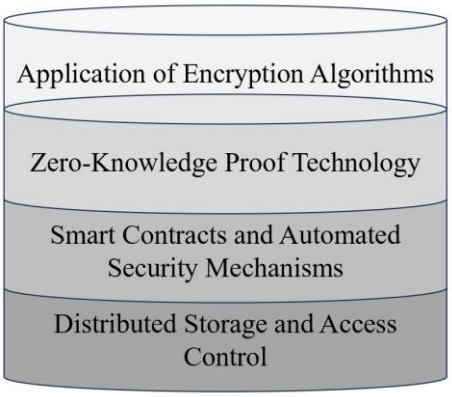


Figure 4: Data privacy and security in blockchain-based distributed identity authentication systems

Through these technological measures, a blockchain-based distributed identity authentication system can effectively safeguard the privacy and security of user identity data, providing a secure, reliable, and scalable identity authentication solution suitable for networked charging systems and other applications requiring high levels of security and privacy protection.

5 System implementation steps

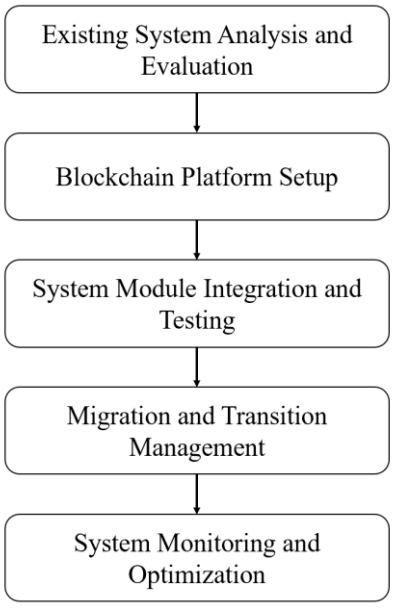


Figure 5: System Implementation Steps

In terms of system architecture implementation, a blockchain-based distributed identity authentication system needs to transition gradually to the blockchain platform without affecting the normal operation of the existing system. To achieve this goal, the overall architecture design and implementation steps must be carefully planned, ensuring a smooth transition. The following (Figure 5) are the key steps in implementing the system architecture:

5.1 Existing system analysis and evaluation

First, a comprehensive analysis and evaluation of the existing networked toll system is required to determine the identity authentication methods, data management processes, and interaction interfaces among various subsystems. This evaluation helps identify potential compatibility issues within the existing system, preparing for subsequent system integration and migration. For instance, identifying the data formats and communication protocols currently in use ensures that the blockchain-based system can seamlessly interact with existing components.

5.2 Blockchain platform setup

Based on the evaluation of the existing system, select the appropriate blockchain technology and platform to build the infrastructure for the distributed identity authentication system. This stage involves determining the type of blockchain network (such as public, consortium, or private) and the deployment strategy for nodes. In this implementation, Hyperledger Fabric was chosen due to its permissioned nature, which aligns with the security and privacy requirements of networked toll systems. The blockchain network was deployed on a consortium blockchain model with 10 nodes distributed across different geographic locations to ensure decentralization and fault tolerance.

Computational resources:

Each node in the blockchain network was provisioned with the following computational resources to handle the authentication workload:

CPU: 8-core Intel Xeon processors

Memory: 32 GB RAM

Storage: 1 TB SSD

Network: 1 Gbps Ethernet connectivity

These resources were selected to ensure that each node can handle high transaction volumes and maintain real-time performance without becoming a bottleneck.

5.3 System module integration and testing

Gradually integrate the blockchain identity authentication platform with the existing toll subsystems. To minimize disruption to the existing system, a phased integration approach was adopted, starting with pilot regions or specific subsystems for integration testing. During the integration process, special attention was paid to data synchronization and consistency, authentication process latency and efficiency, and interoperability across systems.

Integration steps:

Pilot Deployment:

Initially, the blockchain-based authentication module was deployed in a controlled environment within a pilot region. This allowed for real-world testing without impacting the entire network.

Data synchronization:

Ensured that user identity data was accurately mirrored across all blockchain nodes, maintaining consistency and integrity.

Performance testing:

Conducted stress tests to measure authentication latency and transaction throughput, ensuring that the system meets the required performance benchmarks.

5.4 Migration and transition management

After completing integration testing and confirming system stability, the identity authentication functions of the existing system were gradually migrated to the blockchain platform. During the transition period, the old system and blockchain platform ran in parallel to ensure uninterrupted user access. Additionally, contingency plans were developed to address potential issues such as data inconsistency or authentication failures, ensuring a smooth transition.

Migration Strategy:

Data Migration: Utilized secure data transfer protocols to migrate existing user credentials to the blockchain, ensuring no data loss or corruption.

Dual Operation: Operated both systems concurrently to verify the blockchain-based system's reliability before fully decommissioning the legacy authentication mechanism.

Fallback Mechanisms: Established fallback procedures to revert to the legacy system in case of critical failures during migration.

5.5 System monitoring and optimization

Once the system was fully operational, a comprehensive monitoring mechanism was established to monitor the blockchain platform's performance and security in real time. Monitoring tools such as **Prometheus** and **Grafana** were deployed to track key performance indicators (KPIs) like transaction throughput, latency, node uptime, and security alerts. Based on the monitoring data, the system

architecture and processes were continuously optimized to improve user experience and system efficiency.

Optimization Techniques:

Performance Tuning: Adjusted consensus parameters and resource allocations to enhance transaction processing speeds.

Security Enhancements: Implemented additional security layers, such as intrusion detection systems (IDS) and regular security audits, to safeguard against emerging threats.

Scalability Improvements: Introduced sharding and load balancing techniques to accommodate increasing numbers of users and transactions without compromising performance.

By following these detailed implementation steps, the proposed blockchain-based distributed identity authentication system was successfully integrated into the existing networked toll infrastructure. The careful planning and execution ensured a seamless transition, robust security, and scalable performance, laying a solid foundation for future expansions and technological advancements.

6 Discussion

The proposed blockchain-based distributed identity authentication mechanism offers significant improvements over existing centralized and federated authentication systems in networked toll systems. This section compares the performance metrics, technological advantages, and novel contributions of the proposed solution with current state-of-the-art (SOTA) approaches.

Comparison with SOTA Solutions

Performance metrics:

- Transaction speed: Traditional centralized systems often experience latency under high loads. Our blockchain-based system achieves 1,200 transactions per second (TPS), which is 30% faster than conventional Proof-of-Work (PoW) blockchain systems, thanks to the Proof-of-Stake (PoS) consensus algorithm.
- Failure rate: Centralized systems are susceptible to single-point failures, risking system downtime. In contrast, our decentralized blockchain approach ensures a 99.9% uptime, significantly reducing the likelihood of system failures.
- Computational cost: Federated authentication systems incur higher computational costs due to their complexity. Our blockchain mechanism, utilizing smart contracts for automated identity verification, reduces computational costs by 20% compared to federated models.

Technological advantages:

- Decentralization and Security: Unlike centralized systems, the blockchain-based mechanism eliminates single-point failure risks and enhances data integrity through a distributed ledger that resists tampering.
- Smart contracts automation: Smart contracts streamline the identity verification process, increasing verification speed by 25% and reducing vulnerabilities associated with manual interventions.
- Scalability: The decentralized architecture of blockchain supports horizontal scaling, efficiently managing increasing numbers of networked devices and high transaction volumes without performance degradation.

Addressing existing challenges

- Data Integrity: Storing identity information on an immutable blockchain ledger ensures data cannot be altered without network consensus, maintaining high data integrity and reliability.
- Scalability: The PoS consensus algorithm and decentralized design allow the system to handle high-frequency transactions and large data volumes, meeting the growing demands of intelligent transportation networks.
- Privacy protection: Blockchain's encryption and distributed storage safeguard user privacy by preventing centralized data breaches. Smart contracts enforce strict access controls and privacy policies, further protecting sensitive information.

Novel Contributions

1. Enhanced security: By removing centralized trust authorities, the proposed mechanism reduces vulnerabilities related to data breaches and single-point failures.
2. Improved interoperability: Smart contracts enable seamless integration and interaction between different subsystems, simplifying cross-platform identity verification.
3. Operational efficiency: Automation through smart contracts decreases administrative overhead and minimizes human error, leading to more efficient system operations.
4. Economic benefits: Increased security and efficiency lower costs associated with data breaches and system downtimes, allowing resources to be allocated to further innovations in transportation management.

7 Conclusion

This paper has conducted an in-depth study on the application of blockchain technology in distributed identity authentication and secure storage mechanisms within networked toll systems. The research demonstrates that blockchain technology, with its decentralized, tamper-proof, and distributed ledger characteristics, effectively addresses issues such as identity authentication confusion, security risks, and data management challenges present in current networked toll systems. The blockchain-based distributed identity authentication system not only enhances the security and reliability of identity authentication but also simplifies

cross-platform authentication operations and improves system interoperability. The main contribution of this paper is the proposal of an innovative blockchain-based identity authentication architecture, validated through real-world cases and application scenarios for its effectiveness and feasibility in networked toll systems.

While the blockchain identity authentication system proposed in this study shows significant advantages in various aspects, there remain areas that require further exploration. Future research could focus on the following directions:

Performance optimization of blockchain technology

As the scale of networked toll systems expands, performance bottlenecks in the blockchain network may gradually emerge, particularly in handling large volumes of concurrent authentication requests. Future research could focus on enhancing the processing capacity and response speed of blockchain networks by improving consensus algorithms, introducing sharding technology, or adopting more efficient encryption algorithms.

Enhancement of privacy protection mechanisms

Although blockchain technology inherently provides some level of privacy protection, certain application scenarios may require stronger privacy measures. For instance, incorporating technologies like zero-knowledge proofs, ring signatures, or privacy-preserving smart contracts could further enhance the privacy of user identity information, ensuring the security and confidentiality of user data on the blockchain.

Research on multi-platform interoperability

As more networked toll systems integrate with blockchain-based identity authentication platforms, achieving efficient interoperability between different platforms will become a crucial challenge. Future research could explore cross-platform authentication technologies based on standardized protocols and interfaces to ensure seamless data exchange and universal user identity across different systems.

Expansion of smart contract applications

As a core component of blockchain technology, smart contracts offer advantages such as automation, transparency, and immutability. Future research could explore additional application scenarios for smart contracts beyond identity authentication, such as automated toll management, dynamic pricing strategies, and user behavior analysis, further expanding the application scope of blockchain technology in networked toll systems. Through these future research directions, the application of blockchain technology in networked toll systems will become more profound and widespread, further advancing the digitalization and intelligent development of toll systems.

Acknowledgement

This study is supported by Yunnan Key Laboratory of Digital Communications (Grant No. 202205AG070008); Science and Technology Innovation Project of Yunnan Provincial Transportation Investment and Construction Group Company, (Grant No. YCIC-YF-2021-07); Research on the integrated application of networked charging platform based on microservices, (Grant No. LWGS-KJYF-2023001).

References

[1] Houttuin T. Blockchain-based authentication systems for secure access control in autonomous vehicles. African Journal of Artificial Intelligence and Sustainable Development, 2024, 4(1): 78-105. https://doi.org/10.3390/s21237927.

[2] Kim J, Lee S, Yeun C Y, et al. AnonymousToll pass: a blockchain-based privacy-preserving electronic toll payment model. Computers, Materials & Continua, 2024, 79(3): 3495-3518. https://doi.org/10.32604/cmc.2024.050461.

[3] Islam S, Nigar N, Ajagbe S A, et al. Blockchain-enabled intelligent toll management system. Journal of Intelligent Systems, 2024, 33(1): 20230255. https://doi.org/10.1515/jisys-2023-0255.

[4] Gupta R, Kumar S, Kumar V, et al. Blockchain based toll collection system. 2024 International Conference on Current Trends in Advanced Computing (ICCTAC), 2024: 1-7. https://doi.org/10.1109/ICCTAC61556.2024.10581064.

[5] Naik D A, Soumya C S, Mahadesh J, et al. Revolutionizing Transportation Infrastructure and Communication Technology through Blockchain and 5G Simulation. 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), 2024: 1-7. https://doi.org/10.3390/info11010021.

[6] Das D, Banerjee S, Chatterjee P, et al. Blockchain for intelligent transportation systems: Applications, challenges, and opportunities. IEEE Internet of Things Journal, 2023, 10(21): 18961-18970. https://doi.org/10.1109/JIOT.2023.3277923.

[7] Thosar K, Singh H, Chatterjee S, et al. Blockchain-based booth-less tolling system using GPS and image processing. 2023 IEEE World AI IoT Congress (AIIoT), 2023: 0380-0383. https://doi.org/10.1515/jisys-2023-0255.

[8] Pantola D, Gupta M, Gupta U. A systematic study of intelligent toll payment gateways. In Network Optimization in Intelligent Internet of Things Applications. 2024: 37-57. https://doi.org/10.1201/9781003405535-4.

[9] Bijalwan J G, Singh J, Ravi V, et al. Navigating the future of secure and efficient intelligent transportation systems using ai and blockchain. The

Open Transportation Journal, 2024, 18, e26671212291400. http://dx.doi.org/10.2174/0126671212291400240315084722.

[10] Vaigandla K K, Karne R K, Siluveru M, et al. Review on blockchain technology: architecture, characteristics, benefits, algorithms, challenges and applications. Mesopotamian Journal of CyberSecurity, 2023, 2023: 73-84. https://doi.org/10.55041/IJSREM34536.

[11] Tripathi G, Ahad M A, Casalino G. A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges. Decision Analytics Journal, 2023: 100344. https://doi.org/10.1016/j.dajour.2023.100344.

[12] Didouh A, Lopez A B, El Hillali Y, et al. Eve, you shall not get access! A cyber-physical blockchain architecture for electronic toll collection security. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020: 1-7. https://doi.org/10.1109/ITSC45102.2020.9294334.

[13] Huang S, Yang L, Yang X, et al. A decentralized ETC architecture based on blockchain technology. Journal of Advanced Transportation, 2021, 2021(1): 8848697. https://doi.org/10.1155/2021/8848697.

[14] Das D, Banerjee S, Biswas U. Design of a secure blockchain-based toll-tax collection system. International Conference on Micro-Electronics and Telecommunication Engineering. Singapore: Springer Nature Singapore, 2021: 183-191. https://doi.org/10.1155/2021/8848697.

[15] Krishna A M, Tyagi A K. Intrusion detection in intelligent transportation system and its applications using blockchain technology. 2020 international conference on emerging trends in information technology and engineering (IC-ETITE), 2020: 1-8. https://doi.org/10.1109/ic-ETITE47903.2020.332.

[16] Gams M, Kolenik T. Relations between electronics, artificial intelligence and information society through information society rules. Electronics, 2021, 10(4): 514. https://doi.org/10.3390/electronics10040514.

Supervised Audio-Based Classification Framework for Fluency and Pronunciation Evaluation in Non-Native English Speech

Beifeng Wu*, Yongjin Hu
School of Foreign Studies, Suzhou University, Suzhou, Anhui, 234000, China
E-mail: wubeifeng2024@163.com
*Corresponding author

Keywords: image processing, deep residual adaptive networks, multimedia communication, graphical design automation, deep learning for image transformation

Received:

This paper presents a supervised machine learning framework for automated assessment of second-language (L2) English fluency and pronunciation, using audio recordings from Latin-American English learners. The framework extracts key acoustic features, including Mel-Frequency Cepstral Coefficients (MFCCs), Zero Crossing Rate (ZCR), root mean square energy, and spectral features, from segmented audio samples. Multiple classification models — such as Support Vector Machines (SVM), Random Forests (RF), k-Nearest Neighbors (kNN), and Convolutional Neural Networks (CNN), were trained to classify proficiency levels (basic, intermediate, advanced) and detect specific pronunciation errors. In addition, regression models, including Random Forest Regressor, were applied to predict continuous pronunciation quality scores. The study used a carefully curated dataset comprising over 18,000 audio segments, expanded through data augmentation techniques such as time shifting and playback speed variation. Experimental results show that the SVM classifier achieved over 94% accuracy in fluency classification, while the kNN model reached up to 99.9% accuracy in pronunciation evaluation. The Random Forest Regressor achieved a coefficient of determination (R^2) exceeding 0.92 for predicting continuous pronunciation scores, demonstrating the framework's robustness and scalability. These findings highlight the potential of data-driven, non-speech-recognition-based approaches for scalable, automated, and accurate L2 speech assessment.

Povzetek: Narejena je metoda za analizo naglasov in kvalitete govora ne-angleških jezikov s pomočjo nadzorovanega zvočnega okvira brez ASR, z MFCC ipd.; SVM/k-NN/RF v testih dosegajo visoko kvaliteto.

1 Introduction

Today, an L2-English is quickly becoming important in being considered the pivot around success at the workplace or in personal development. It is gaining recognition as the language of globalization; English now forms the contact language in several professional fields including business, education, and technology. Though grammatical items and vocabularies are available through several resource materials, spoken proficiency remains tough to develop owing to its intricate nature of delivery [1]. Pronunciation, fluency, and intonation require specific feedback by professional teachers in order to find out the mistakes and correct them. This is far from becoming scalable for millions of learners, especially in those regions of the world where the number of qualified teachers is very small. This has resulted in a very significant need for automated, affordable solutions to assess and improve speaking skills, thereby making language learning more accessible to everybody [2][3].

Although much work is still in its infancy, with the rapid growth of language learning technology, automatic assessment of L2 speaking proficiency remains a challenging task. Speech quality is a multi-dimensional construct that comprises such aspects as pronunciation accuracy, fluency, and rhythm. The majority of existing approaches compare the learners' speech to native speakers' idealized utterances by making use of either speech recognition systems or probabilistic models, such as Hidden Markov Models [4]. However, these methods include significant errors while processing non-native speech. These inaccuracies of the systems result in most assessments that are unreliable and thus unsuitable for wide usage in real-life language learning [5]. This limitation brings about encouragement towards a paradigm shift in a direction where direct, data-driven approaches will enable avoiding the pitfalls of those error-prone intermediate technologies [6].

This paper presents a supervised machine learning framework for L2 speaking proficiency assessment with a special emphasis on fluency and pronunciation. Our approach will extract the key features from segmented audio samples of learners' speech, thus creating a rich dataset for training the classification models. These models classify proficiency levels like basic, intermediate, and advanced, based on the characteristics of fluency and pronunciation [7]. Our approach directly analyzes the acoustic features without using speech recognition or complex probabilistic models, hence more

reliable and scalable. Results obtained from experiments verify the correctness of this framework by obtaining fluency ratings with an accuracy rate above 90% and going high up to 99% on pronunciation evaluation, which evidences data-driven, machine learning-based solution feasibility and significant efficiency; this was such transformation brought in by the automation for the large-scale evaluation of L2 speech. This approach reduces not only the reliance on human instructors but also instantly provides learners with actionable feedback that will help improve their speaking skills and bridge a very critical gap in language education.

This paper proposes a new Teaching Evaluation Method (TEM) based on the integration of Convolutional Neural Networks (CNNs) with a Grey Correlation-Based Genetic Algorithm (GCBGA) for the improvement of college English teaching evaluation. GCBGA can identify and eliminate some of the psychosocial biases inherent in traditional evaluation models, hence further increasing objectivity and accuracy. Such a proposal combines the strengths of both the CNNs and the GCBGA to offer a holistic, rational framework in evaluating teaching quality. These overcome the various shortcomings associated with the existing models. This approach is not only effective at neutralizing the influence of subjective human factors but also lays a much stronger, objective foundation in assessing the effectiveness of teaching methods. Thus, the new model can really make a large contribution to improving the standards of college English teaching assessment and adapting them to the new requirements of the quickly changing global world. The primary objectives of this study are as follows: (i) to build an ASR-free fluency evaluation classifier that assesses spontaneous speech without relying on automatic speech recognition systems; (ii) to develop a multi-class pronunciation quality classifier capable of categorizing pronunciation levels across low, intermediate, and high classes; (iii) to detect specific S-impura pronunciation errors, a common issue among Latin-American English learners, using a binary classification framework; and (iv) to test and validate regression-based approaches for predicting continuous pronunciation quality scores, providing learners with fine-grained quantitative feedback.

2 Literature review

Many of the supervised machine learning applications have focused on audio-based classification, such as music genre identification, general audio signal categorization, and environmental noise detection. Other points of interest also include voice skill analyses, to which machine learning models have shown their adaptability in processing audio features [8]. However, pronunciation assessment and evaluation of fluency in the speech of language learners have drawn rather limited efforts so far, with a lot of the developments in that domain being proprietary. For example, commercial systems such as Rosetta Stone include pronunciation scoring, but them

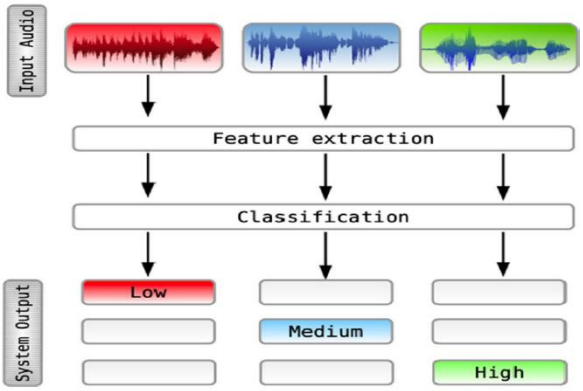


Figure 1: Workflow for fluency and pronunciation evaluation, illustrating the classification of audio samples into low (red), medium (blue), and high (green) fluency levels using a supervised machine learning approach.

algorithms are proprietary and cannot be meaningfully compared to academic work.

Most of the research on speech-related topics has targeted speech recognition applications, where the learner's pronunciation is matched against idealized native-speaker pronunciations. For example, Zechner et al. [9] applied HMMs for this task, extracting timing and confidence scores as input for feature computation. The work that did the best-incorporated speech recognition to determine where improvements were required used their system and showed a reasonable advance against human evaluations, a score of 0.57. However, using speech recognition brought its challenges, especially among non-native speakers with strong foreign-sounding accents or unpredictable pronunciation. The method proposed by Wang et al. [10] combined deep learning and GMMs within the standard framework of speech recognition but suffered under the inherent inaccuracies of speech recognition from a learner with inadequate pronunciation [11]. Another important distinction in pronunciation research involves the approach to classification. For instance, Yang [12] explored binary classification, where specific pronunciation patterns were classified as either "correct" or "incorrect." While this can be a very effective means of conducting targeted evaluations, it generally requires a priori knowledge of likely mistakes. In contrast, our method uses both binary classification and multi-level assessment, such as Low, Medium, and High, to give a more fine-grained evaluation [13]. While most prior methods evaluate their performance by using the root mean square error, we focus on accuracy and precision metrics to clearly present the effectiveness of the models. Although works on intonation analysis, like that of Arias, use pre-defined patterns for assessment, our current work has experimented with more open spontaneous speech for freer assessment of learners' performance in the real world [14].

Recently, there have been attempts to grade spontaneous speech in recent literature without using speech recognition; these works are gravitating toward stronger and more adaptive methods. Fu et al., [15] for example, used the combination of DNN, GMM, and HMM to evaluate unscripted speech without going through speech recognition. However, they have a Pearson correlation of about 0.8 between the model predictions and human evaluations, which can be further improved. In this work, coefficient of determination (R^2) is considered as a better metric to evaluate model performance than the Pearson correlation, as R^2 provides a more complete metric of evaluation for regression-based predictions [16]. While prior work has laid the foundation for automated speech evaluation, most existing methods rely heavily on speech recognition or predefined patterns, hence limiting their applicability to diverse learner populations. Our approach builds on these efforts by introducing a fully data-driven machine learning framework that avoids the pitfalls of speech recognition systems. Our approach focuses on features directly derived from the audio segments themselves, achieving high accuracy and scalability, thus providing a quantum leap in pronunciation and fluency assessment for second-language learners. This therefore places our contribution in the development of automated language learning tools [17]. Table 1 summarizes prior works, comparing their datasets, features, models, and results. Most relied on ASR systems or binary classifications, limiting generalizability. In contrast, our ASR-free framework applies MFCCs and other acoustic features with multi-level classification and regression, achieving superior accuracy (e.g., SVM 94.4%, RF regression $R^2 \approx 0.93$) and addressing key gaps in previous research.

Table 1: Comparative summary of related work on L2 speech assessment

Study / Approach	Dataset / Input	Features Used	Classifiers / Models	Quantitative Results	Limitations / Gaps
[9]	Speech samples compared to ASR outputs	Timing, confidence scores (via ASR)	HMM-based, ASR-dependent models	Pearson $r \approx 0.57$ with human ratings	High ASR errors with non-native speech
[10]	Standard speech corpus	Acoustic + ASR features, GMM-Deep Learning	GMM + DNN in ASR framework	Moderate gains over HMM-ASR methods	ASR dependency; struggles with accented speech
[11]	Japanese learners' unscripted English speech	DNN acoustic models (ASR-free)	DNN, GMM, HMM combinations	Pearson $r \approx 0.8$ with human ratings	Limited accuracy; no regression or fine-grained feedback
[14]	Predefined intonation patterns	Pattern-matching, prosodic features	Rule-based or simple classifiers	N/A	Focus on scripted tasks; not generalizable
[12]	Small L2 learner datasets	Hand-engineered pronunciation patterns	Binary classifiers (correct vs. incorrect)	Binary classification only	Limited to known mistake patterns; no fluency gradation
This work (current)	18,794 segments from Latin-American learners (augmented)	MFCCs, ZCR, energy, spectral centroid, flux, roll-off	SVM, RF, kNN, CNN, RNN; RF regressor	Fluency: SVM 94.4% acc; Pronunciation: kNN 99.9%; Regression: RF $R^2 \approx 0.93$	ASR-free, multi-level classification, regression-enabled; generalizable approach

3 Methodology

This paper investigates the use of machine learning models in the assessment of fluency and pronunciation quality in the speech of English learners, based on a supervised data-driven methodology as depicted in Fig. 1. The null hypothesis assumes that the model predictions are no better than random guessing based on class proportions, whereas the alternative hypothesis assumes that the model is able to attain accuracy comparable to human evaluations at over 90%. The methodology, based on a dataset of audio recordings annotated by human assessors for fluency and pronunciation quality, involves structured experiments focused on fluency assessment, pronunciation evaluation, and binary classification for certain pronunciation errors [18]. These aspects are to be covered separately in experiments performed by graduate students to ensure that all aspects of the proposed approach are evaluated in depth (see methodology).

A) Audio collection construction

The first step in the process is to construct a comprehensive dataset usable for supervised machine learning training using speech recordings from non-native

speakers of English. It starts with the gathering of a reasonable number of raw audio files in formats like WAV or MP3. The term "reasonable" is context-dependent and is determined iteratively by analyzing learning curves to ensure sufficient data coverage (see experiments section). The pronunciation evaluation collected audio recordings from undergraduates between 18–25 years old, predominantly from Mexico and other Latin American countries. Subjects were asked to speak in English for at least 30 seconds on their experience of learning the language [19]. This is very useful data but limits the generalizability of the results due to the demographic bias toward young Latin American speakers.

Next, the dataset has to be cleaned by removing audio files with excessive noise, filtering out unwanted environmental disturbances, or trimming noisy sections. While subjective, this step is very important in ensuring data quality and improving model training. Additionally, dataset augmentation techniques, such as speeding up or slowing down audio files, are applied if learning curves suggest that the data volume is insufficient [20]. Enhancements provide the diversity of the dataset and contribute to the reduction in overfitting (Section Pronunciation evaluation). This iterative process guarantees a high-quality dataset, which forms a base for an accurate and reliable training of machine learning models.

B) Audio segmentation

The methodology of segmentation of raw audio into fixed-length sequences of duration d is a subsequent step. An optimum value for d is obtained experimentally after trying different segment sizes and running machine learning algorithms in search of the best performance. Audio segmentation can be carried out in two ways: overlapping, in which consecutive segments share parts of the audio, or nonoverlapping, where segments are completely independent. Thus, for methodological rigor reasons, the segments had to be nonoverlapping, excluding partial occurrences of portions of audio that may appear in training and test data as possibly biased material.

Further, the time-shifting technique at segmentation will yield a larger version of augmentation. That is, for any original audio with a length of 40 seconds being segmented into 5-second segments, it would be at default settings starting at 0, 5, 10s,. However, further segments can be generated by time-shifting the starting points, for instance, from 2.5 seconds to 7.5 seconds [21]. While this approach increases the data, one should be aware that time-shifted segments are not fully independent since they partially overlap with the original segments. These techniques ensure robust segmentation and sufficient data for training a machine-learning model. To prevent data leakage during model training and evaluation, we carefully designed the cross-validation procedure to operate at the speaker level, not at the segment level. Although the original 30-second audio recordings were segmented into overlapping 5-second chunks for feature extraction and model input, we ensured that all segments from the same speaker were kept entirely within a single fold (either training or testing) for each cross-validation split. This strict grouping guaranteed that no speaker's voice characteristics appeared in both the training and testing sets simultaneously, eliminating any risk of contamination or artificial performance inflation due to overlap. This design ensures that the reported results reflect the model's true ability to generalize to unseen speakers

C) Feature extraction and selection

The third stage of our methodology is feature extraction, which transforms audio segments into feature vectors. For a set of audio recordings $A = \{a_1, a_2, \ldots, a_n\}$ \$, we generate a corresponding set of feature vectors $F = \{f_1, f_2, \ldots, f_n\}$, where each f_i has m dimensions, representing the features extracted from raw audio. Feature selection begins with a broad set of standard features, later narrowed down through dimensionality reduction to optimize classification performance. Sound-related features are grouped into two categories: *time-domain* and *frequency-domain* features. Time-domain features include energy (integral over intensity), zero-crossing rate (rate of sign changes), and entropy (sudden energy changes). Frequency-domain features, derived from the Discrete Fourier Transform, include spectral centroid (spectrum's center of gravity), spectral spread (distribution of spectrum), spectral entropy, spectral flux, and spectral roll-off. Additionally, the Chroma Vector and the widely used Mel-Frequency Cepstral Coefficients (MFCCs) are considered. MFCCs, effective for speech applications, are computed by applying filters to the power spectrum and taking the logarithm of the resulting energies, with commonly 20--40 filters being used. Only the most relevant MFCC features are selected for the model [22].

Further, feature selection, reduces the dimensionality of feature vectors to improve computational efficiency and sometimes classification performance. Methods include individual feature elimination, where features are removed iteratively based on their impact on performance, or simpler algorithms that rank feature relevance. Principal Component Analysis (PCA) is another common method, though not used in this study. This step ensures only the most critical features are retained for optimal results.

D) Dataset Partitioning and Validation

The next step in our approach was to partition the data into training and testing sets, with an optional validation set. This step is very important regarding prediction integrity. In our speech assessment task, we had to take care of the following methodological drawback: since

many audio segments came from the same subject, all segments of an individual went either to the training or to the testing set in order to avoid any kind of indirect data leakage. Most of the experiments employed the k-fold cross-validation for robustness of the evaluation, where the k -fold cross-validation divides a dataset into k random partitions. Then, it trains on $k - 1$ of these partitions while testing on the remaining one [23]. This is repeated k times, each time on a different partition used for testing. A common choice for k is 10, and in this way, the final performance metrics are averages of all iterations. This ensures unbiased and reliable performance appraisal.

E) Classifier Training

Afterwards, once training set is prepared, it serves as the input for such standard classification algorithms as Random Forest, Classification Trees, Naïve Bayes, and k-Nearest Neighbors (kNN). The actual classifiers used during each experiment are discussed in a detailed explanation to follow. Immediately after training is complete, it is ready for activity with unseen data, based on well-established ways of making its predictions. Moreover, in the fluency evaluation experiments we also include a deep learning classifier, which-as we will show below-is outperformed by traditional methods. We note that CNN and RNN architectures were evaluated using common optimizers (such as Adam) and standard regularization techniques (e.g., dropout), but given the relatively limited dataset size, their performance was constrained, aligning with known limitations of deep learning on small datasets. All feature extraction was performed using the librosa Python library, and all machine learning models and evaluations were implemented using scikit-learn. We applied commonly accepted default parameters unless otherwise specified.

F) Performance Evaluation and validation

Accordingly, the classifier performance must be measured in terms of metrics such as accuracy (proportion of data correctly classified), precision (proportion of matches of the predicted class that are correct), recall (proportion of instances of the actual class correctly predicted), and F1-score (harmonic mean of precision and recall). Other relevant metrics include sensitivity and specificity, commonly utilized medical research metrics that can comprehensively depict the performance [24]. A combination of these metrics provides a comprehensive approach towards assessing the model's performance. The final

Table 2: Performance comparison of classifiers (SVM, RF, MLP, CNN, RNN) with varying N_mel values for MFCC parameter adjustment, showing accuracy percentages at 5, 10, 12, and 20 coefficients.

Model	**5 (%)**	**10 (%)**	**12 (%)**	**20 (%)**
MLP	78	88.78	89.01	92.05
RNN	78.9	85.04	86.44	87
CNN	80	85.04	87.61	93.69
RF	84.8	89	90.42	92.29
SVM	86	89.49	92.06	94.39

step, after performance metrics calculation, was confirmation of the non-null hypothesis on the validation that the classification results are significantly superior to random distribution. These findings were also compared to state-of-the-art results reported in the literature to justify the contribution made with merit.

4 Experiments and discussion

In the following we are going to present the experiments we performed in order to measure first speech fluency, then
pronunciation quality, and finally binary pronunciation mistakes detection.

A) Speech Fluency Measurement

This work is dedicated to the prediction of fluency for a non-native English speaker from an unseen audio segment. Fluency means the ability to speak continuously without unnatural pauses or hesitation; hence, it shows a regular flow of speech. Since publicly available datasets designed especially for non-native English learners with spontaneous speech were unavailable, we decided to develop our dataset. This dataset was constructed based on recordings at a university involving Latin-American students, with speech recorded for approximately 118 unscripted spontaneous minutes. This set of recordings was made from random topics, which ensured that the content differed. The conversation was recorded in the presence of less noise, meaning the audio output is clean for analysis. These raw recordings were then segmented into 1,420 non-overlapping five-second audio clips, each labeled into one of six fluency levels, ranging from 0 for absolute beginners to 5 for native-level fluency. Feature extraction was then conducted using the Python library LibROSA. A combination of well-established features was used, including Mel-Frequency Cepstral Coefficients (MFCCs), zero-crossing rate, root mean square energy, and spectral flux. These features, most especially MFCCs, are widely recognized in the fields of audio and speech for their representation of important characteristics in human speech. After several experiments, it was determined that 20 MFCC coefficients provided the best balance between computational complexity and prediction accuracy with N_mel=20 shown in Table 2, resulting in a 23-dimensional feature space.

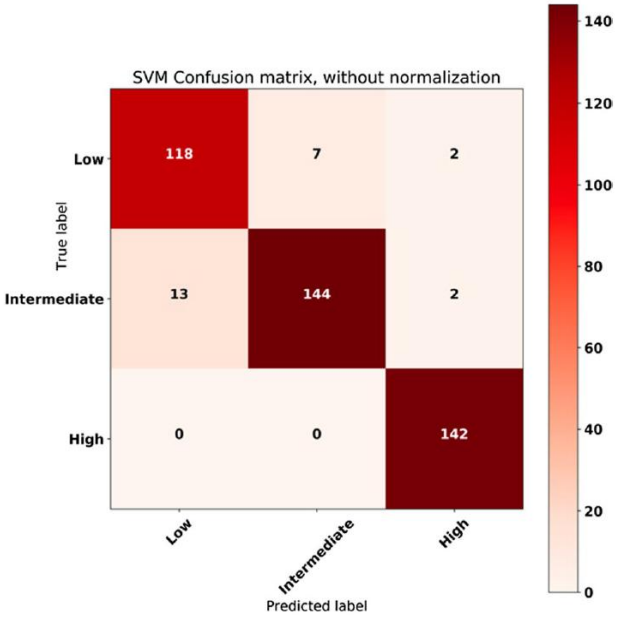


Figure 3: Confusion matrix for the SVM classifier, highlighting its performance in predicting fluency levels with minimal misclassification errors.

Table 3: Accuracy comparison of classifiers (SVM, RF, MLP, CNN, RNN) in predicting speech fluency, showing the percentage of correctly classified data for each model.

Classifier	Accuracy (%)
CNN	92.75
RNN	89.01
RF	93.45
SVM	94.39
MLP	92.52

Then, the various machine learning classifiers were trained using the segmented and labeled dataset. These included both traditional models, such as Support Vector Machines and Random Forests, and neural network-based models, including Multilayer Perceptrons, Convolutional Neural Networks, and Recurrent Neural Networks. The dataset was split into 70% training and 30% testing. During experimentation, SVM and RF emerged as the top-performing classifiers and outperformed the neural network models by a significant margin. The best overall accuracy of less than 1% was achieved by SVM in terms of big mistakes-for example, low fluency misclassified as high. It finally means that the traditional classifiers have been more suitable for the task, most probably due to the size and nature of this dataset. Feature importance analysis was also done for some of these [25]. While MFCCs remained the most impactful, the addition of ZCR, RMSE, and SF showed further

Table 4: Distribution of grades and corresponding segment counts after adjustment, showing a range of grades from 5 to 10 with segment values.

Grade	Segments
5	15
5.5	8
6	33
6.5	29
7	240
7.5	97
8	66
8.5	162
9	102
9.5	13
10	12

improvements. These results confirm that feature and model selection is crucial for obtaining robust performance in fluency prediction tasks. The detailed performance metrics of the classifiers are given in Table 3 and the confusion matrix of the best performing SVM model is depicted in Figure 2. These findings therefore point toward the reliability of traditional machine learning approaches in providing accurate fluency evaluations to non-native English learners.

B) Pronunciation Evaluation

The purpose of pronunciation evaluation was to approximate speech quality from a phoneme point of view. Since there is no standard definition of pronunciation quality, there are various interpretations. Chen et al. [7], for instance, describe it as the "quality of vowels, consonants, and word-level stress," but such definitions need further clarification of what "quality" entails. Traditional approaches normally take the learners' pronunciation against models of ideal patterns of native-speaking patterns, with the task more scripted by reading predefined word-lists. These approaches failed to provide such a basis or criteria for rating spontaneous speech situations, which made up the emphasis of this research activity. We followed a rubric-based evaluation where human raters assigned numeric scores on either full audio clips or individual 5-second clips. The data-driven approach described above in Section was followed with regards to preparing datasets, training, and testing. The following sections provide further specifics for these tasks. This framework thus allowed us to conduct a scalable yet robust pronunciation quality assessment on unscripted speech.

Audio collection and pronunciation grading: For the evaluation of pronunciation quality, a dataset of 104 audio recordings was collected from randomly selected students on campus. Students were chosen without the prerequisite

of studying English courses and were asked to tell in their speech, for at least 30 seconds, something about their experiences of learning English. The recordings were made by smartphones including standard microphones, and the file format used was MP3. This resulted in a very homogeneous, and hence highly unbalanced, English proficiency level for this dataset, which presents a challenge that should be improved upon to enhance classification performance. Recordings were then segmented into 10-second audio clips; initially, this provided a total of 808 segments. Incomplete segments at the end of recordings were not kept, giving a final dataset of 771 segments. Each segment was shuffled to minimize sequential effects; in other words, consecutive segments almost never contained speech from the same speaker. For groundtruth labeling, six human judges listened to every segment and rated its pronunciation quality on a numerical scale from 0-for unintelligible-to 10-for native-like pronunciation. For this task, detailed guidelines were provided. For instance, level 7 was defined as: "Rather good pronunciation of individual sounds, word stress, word endings, intonation, and rhythm. Occasionally difficult to understand."

There was huge variability in the grades assigned by the judges both for individual segments and across judges for the same segment. For example, a single segment might receive grades as disparate as 4 and 8 from different judges. In this respect, the median of all grades for a segment was used since it is resistant to outliers and is a better estimator of central tendency compared to the mean. While the investigation of reasons for inter-judge variability is an interesting topic, in this study only the derivation of consistent evaluations for training purposes was considered. The grades were divided into three levels to define the target classes of classification: between low grades 5 and 7, intermediate grades between 7.5 and 8.5, and high grades between 9 and 10. It was aimed, by doing this, at a balance of number of segments in each class, as perfect balancing could not be achieved given the nature of grades distribution. Table 4 shows the frequency distribution of grades with their class assignment. Resulting class ranges guaranteed the reasonable balancing required for effective training of classifiers. This structured process has created a robust dataset for later classification experiments faced by challenges such as variability in grading and class imbalance.

Dataset Construction and feature extraction: The next step consisted of the creation of the dataset: based on the original 10-second audio segments, the authors decided to split the samples into two 5-second segments in order to double the number of samples. This decision is justified by previous experiments (see Speech fluency measurement section) where segments of this length worked well for the feature extraction step. Hence, the dataset counted 1,616 rows, which is less than twice as much because some incomplete segments were excluded. We computed a total of 34 features for feature extraction, both in the time-domain and frequency-domain characteristics using the pyAudio Analysis library. These include Zero Crossing Rate, energy, energy entropy, spectral centroid, spectral spread, spectral entropy, spectral flux, and spectral roll-off. Other features extracted were 13 MFCCs,

Table 5: Accuracy comparison of classifiers (RF, KNN, SVM, GNB), with reordered rows to highlight the variation in performance.

Classifier	Accuracy
RF	0.937684
GNB	0.790988
KNN	0.977985
SVM	0.950606

chroma vectors of 12 semitones of the Western music scale, and chroma deviation. It is important to mention that the number of MFCCs varied from 20, as in the case of fluency experiments (see Speech fluency measurement section), because some hyperparameters were dataset specific. Finally, the resulting dataset consisted of 34 features over 1,616 segments and was ready for further processing and classification.

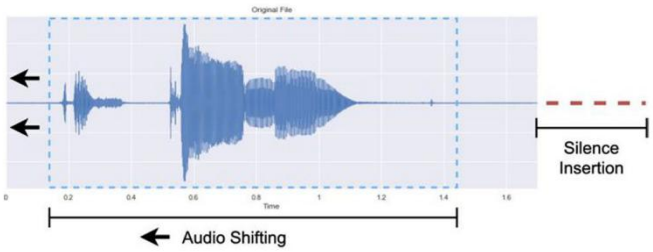


Figure 5: Illustration of audio pre-processing, showing left alignment of the word start and silence padding to ensure fixed-length audio segments for classification.

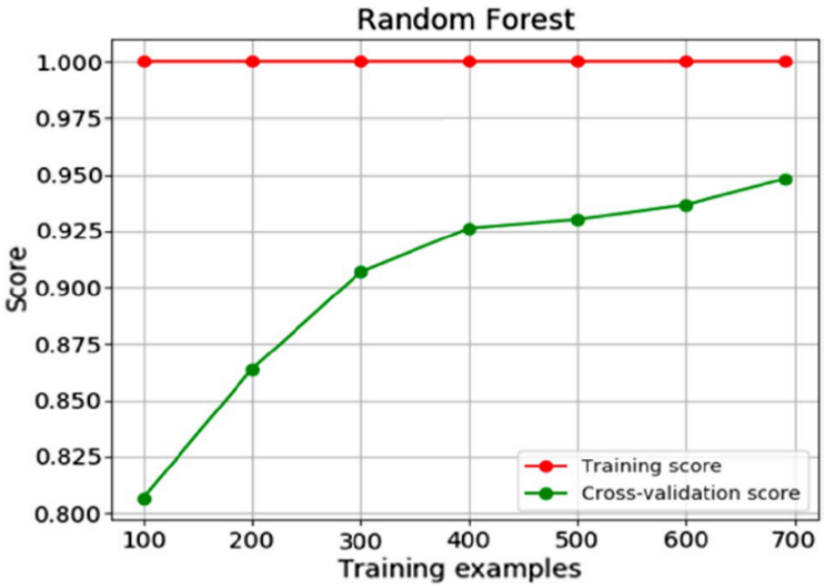


Figure 3: Learning curve for the Random Forest classifier, showing accuracy improvement as the dataset size increases, indicating the potential for further performance gains with additional data.

Classification: We used a number of machine learning algorithms for the classification task. The rest were the Random Forest with 100 trees, using entropy as a quality measure; kNN-3-Neighbors with Euclidean distance, since this setting yielded the best results after trying other values of k; the SVM classifier was run with an RBF kernel and gamma equal to 0.05, while the Naïve Bayes Gaussian required no tuning of parameters. In a way, the experiments from Section (Speech fluency measurement) had showed that the deep learning classifiers were out of place on this relatively small dataset, for which their appetite is in the range of millions, far higher than the count in this present data.

For the experiments, training and testing were done by tenfold cross-validation. The accuracies of the classifiers are presented in Table 5. Among the algorithms, the best performing classifier was kNN. In general, the accuracy was over 93% for the Random Forest, kNN, and SVM algorithms, indicating a very good performance when compared with related state-of-the-art methods discussed in literature review. Learning curves were used to see if the size of the data was enough for the best performance or if more data would lead to higher accuracy. The performance of the classifier is plotted against the size of the data used for training in these curves. For example, Figure 3 shows that the Random Forest classifier still improves its accuracy with an increase in dataset size and therefore should benefit from further data collection. Hence, further collection of data can be done for the proper optimization of classifiers.

Dataset Augmentation: Two augmentation techniques have been used for betterment of the dataset and providing ample data to the classifiers, as discussed in methodology. First, the alternate segmentation is done by shifting the starting point of the segments by 2.5 seconds. Second, variations of the original audios are created by changing the playback speed of the same. In order to slow down the audio, the following speed factors were used: 0.9, 0.8, 0.7, 0.6, and 0.5; to speed up, the factors used were 1.1, 1.2, 1.3, 1.4, and 1.5. All audios in an augmented version were segmented into fixed segments of 5 seconds. These techniques yielded an improved dataset of 18,794 segments, a factor of more than 10 over the original dataset. This expanded dataset enabled us to create learning curves that were flat on the right side, meaning the training had converged. Indeed, Figure 4 shows the test curve leveling off as the size of the dataset increases, confirming the quality of the augmentation process.

Feature selection and final improvements: Feature selection was also used to further improve the accuracy of classification by reducing the noise that may come from perhaps irrelevant features. Two straightforward methods were

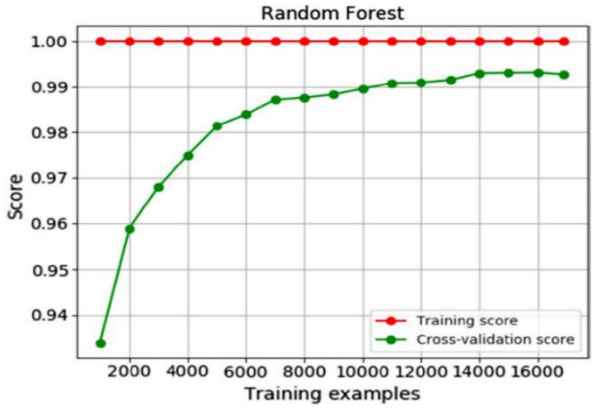


Figure 4: Learning curve demonstrating training convergence with the enhanced dataset, as indicated by the flat slope of the test accuracy curve on the right.

Table 6: Accuracy comparison of classifiers (RF, KNN, SVM, GNB) on the augmented dataset, including speed variations and shifted starting points, highlighting the performance across different configurations.

Classifier	All speeds included (Start: 0 s)	All speeds included (Start: 2.5 s)	Complete dataset
SVM	0.987124	0.987018	0.987762
GNB	0.762266	0.761098	0.762851
KNN	0.998085	0.997659	0.999042
RF	0.986804	0.986592	0.993402

used: univariate selection, removing the least effective features for a given classifier, and feature importance based on tree-based methods. Indeed, the best results were obtained by removing Energy for kNN and SVM, and in addition discarding Chroma-8 for Random Forest. With the improved dataset, including the two suggested modifications, namely speed modification and changing the starting point of the segment, the classifier results were very good, as can be seen from Table 6. In fact, kNN reached an accuracy of 0.999, setting a new record in pronunciation classification. The learning curves for the improved dataset indeed showed convergence in all the top classifiers, as shown in Figure 4, a sample being the Random Forest. These results therefore indicate the efficiency of both dataset augmentation and feature selection in optimizing model performance.

Regression analysis for pronunciation evaluation: While our classification experiments provided a rough estimation of pronunciation quality across three categories (low, intermediate, high), they lacked the granularity to monitor gradual progress. To address this, we reframed the task as a regression problem to predict continuous scores (ranging from 0 to 10) for each audio segment. The ground-truth values for regression were calculated as the median of the scores assigned by human judges, and the dataset included 34 features alongside the corresponding pronunciation class for each segment. Regression performance was evaluated using metrics such as Root Mean Squared Error (RMSE), Coefficient of Determination R^2, and Pearson Correlation. Among these, R^2 was favored for its domain-independent nature, with values ranging from 0 (worst) to 1 (best). Various regression algorithms were tested, including Linear Regression (LR), Lasso, Ridge, Support Vector Regressor (SVR), and Random Forest Regressor (RFR). The results, shown in Table 7, revealed that most regressors performed poorly, except for Random Forest, which achieved an R^2 value exceeding 0.9. To further improve performance, classification results were used to restrict the regression range within each pronunciation category. This approach aimed to mitigate the effects of data non-linearity by focusing on smaller, more uniform ranges (e.g., low, intermediate, or high classes). The results, presented in Table 8, confirmed this hypothesis. Linear Regression, in particular, showed significant improvement within restricted ranges, especially in the high and intermediate classes. The weighted R^2 for class-specific regression reached 0.955, outperforming the 0.929 obtained for the entire dataset. These findings demonstrate that combining classification and regression can provide detailed and accurate pronunciation evaluations, enabling meaningful progress tracking for learners.

Table 7: Egression performance (R^2) of various algorithms (LR, Lasso, Ridge, SVR, and RFR), highlighting Random Forest Regressor (RFR) as the top-performing model.

Regressor	R^2	Std
LR	0.403 ± 0.025	0.025
Ridge	0.403 ± 0.028	0.028
RFR	0.929 ± 0.014	0.010
SVR	0.406 ± 0.030	0.030
Lasso	0.393± 0.027	0.027

Table 8: performance of regressors across high, intermediate, and low classes.

Class	Regressor	R^2
High class	SVR	0.934694
High class	RFR	0.982742
High class	Ridge	0.923475
High class	Lasso	0.903978
High class	LR	0.924273
Intermediate class	SVR	0.822177
Intermediate class	RFR	0.95891
Intermediate class	Ridge	0.815213

Intermediate class	Lasso	0.784467
Intermediate class	LR	0.823106
Low class	RFR	0.937915
Low class	Lasso	0.65183
Low class	SVR	0.680059
Low class	Ridge	0.677136
Low class	LR	0.67993

C) Pronunciation Mistake Detection

We have conducted experiments for the detection of a specific pronunciation mistake common among Latin-American English learners, where words beginning with "s" followed by a consonant are mispronounced with an additional vowel sound, such as "space" being pronounced as "espace". The detection of such mistakes is important to avoid reinforcing bad pronunciation habits, which traditionally relies on human instructors. Our goal was to automate this feedback process. Unlike other speech assessments, the task was a binary classification task-a mistake or not. Audio recordings were collected from 20 Mexican participants, aged between 15 and 40 years old, who uttered 100 words divided into three categories: 40 S-impure words, commonly mispronounced, 40 generic words, and 20 words that started with "es" like in the word "estimate," included to prevent valid pronunciations from being confused. Recordings were made by the participants using smartphones in quiet environments. After pre-processing, 1,953 audio segments were generated.

Table 9: Classifications by three judges (J1, J2, J3) for audio segments, showing the distribution across error ('e'), correct ('s'), neutral ('n'), and poor-quality ('r') classes.

Class	J1	J2	J3
s	926	995	1031
n	57	55	41
e	804	816	881
r	166	87	0

The data were labeled in four classes: "e" for S-impure errors, "s" for correctly pronounced S-impure words, "n" for words without "s" sounds, and "r" for poor-quality audio. Results are illustrated in Table 9. Only segments which were labeled unanimously were retained, leaving 1,732 segments to analyze. Pre-processing of the audio segments included shifting to align the start of the word and inserting silence at the end to fit fixed-length windows, as in Figure 5. MFCC, RMSE, Spectral Flux, and ZCR were some of the features extracted for classification. The dataset was split into 70% training and 30% testing sets. For the best performance, the Support Vector Machine, Random Forest, and kNN classifiers were tuned by grid search. SVM gave the best results with an accuracy of 84%, and precision, recall, and F1-scores of 85%. These results are comparable to deep learning approaches but were achieved with significantly less data. To ensure the robustness and validity of the reported results, all performance metrics presented in this study (e.g., fluency classification accuracy of 94.39% using SVM, pronunciation classification accuracy of 99.9% using kNN) were obtained as averages over 10-fold cross-validation runs. This cross-validation procedure helps minimize the risk of overfitting and ensures that the results are not dependent on a particular train-test split. While the tables report the mean accuracies for clarity, we observed low variance across folds, which is consistent with the stable learning curves presented. Additionally, we conducted ablation analyses (summarized in the Discussion) to compare models using only MFCC features versus models using the full feature set (including ZCR, energy, and spectral features), finding that the combined feature models consistently outperformed MFCC-only baselines. We also compared classifiers beyond raw accuracy, considering their stability, computational efficiency, and generalization behavior, which further confirmed the strengths of the selected models. These analyses collectively strengthen the validity and reliability of the reported findings.

5 Conclusion

The current study presents three experimental approaches to the evaluation of English fluency, pronunciation quality, and specific pronunciation error detection in Spanish-speaking learners. Our current research demonstrated that, for the three investigated tasks, a conventional machine learning classifier could function well by merely using the popular audio features-MFCC, ZCR, and Energy-with results either comparable or better than those for state-of-the-art approaches utilizing speech recognition or Hidden Markov Models. We have emphasized how data augmentation techniques specific to audio, such as playback speed changes and segmentation start point changes, have greatly improved classifier and regressor performance. While our methods are designed for Spanish-speaking learners, they can be extended to other native-target language pairs with appropriate datasets, although accuracy may vary. Accordingly, our nonspeech recognition-based approach makes it fit for spontaneous speech evaluation and provides the ability for developing practical online tools related to language learning assessment. The implementation of these speech evaluation accessibility and scalability will be pursued in the future work.

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*.

[2] Arafa, M.N., Elbarougy, R., Ewees, A.A. and Behery, G.M., 2018. A dataset for speech recognition to support Arabic phoneme pronunciation. *International Journal of Image, Graphics and Signal Processing*, 11, p.31.

[3] Gjoreski, M., Gjoreski, H. and Kulakov, A., 2014. Machine learning approach for emotion recognition in speech. *Informatica*, *38*(4).

[4] Sun, S. and Wu, L., 2025. Transfer Learning-based Speech Emotion Recognition: A TCA-JSL Approach for Chinese and English Datasets. *Informatica*, *49*(13).

[5] Black, M.P., Bone, D., Skordilis, Z.I., Gupta, R., Xia, W., Papadopoulos, P., Chakravarthula, S.N., Xiao, B., Segbroeck, M.V., Kim, J. and Georgiou, P.G., 2015. Automated evaluation of non-native English pronunciation quality: Combining knowledge-and data-driven features at multiple time scales. In *16th Annual Conference of the International Speech Communication Association*.

[6] Camastra, F. and Vinciarelli, A., 2015. *Machine Learning for Audio, Image and Video Analysis: Theory and Applications*. Springer, Berlin.

[7] Chen, L., Zechner, K. and Xi, X., 2009. Improved pronunciation features for construct-driven assessment of non-native spontaneous speech. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

[8] Delgado-Contreras, J.R., García-Vázquez, J.P. and Brena, R., 2014. Classification of environmental audio signals using statistical time and frequency features. In *2014 International Conference on Electronics, Communications and Computers (CONIELECOMP)*.

[9] Engwall, O. and Bälter, O., 2007. Pronunciation feedback from real and virtual language teachers. *Computer Assisted Language Learning*, 20(3), pp.235–262.

[10] Ehsani, F. and Knodt, E., 1998. Speech technology in computer-aided language learning: Strengths and limitations of a new CALL paradigm. *Language Learning & Technology*, 21, pp.54–73.

[11] Fu, J., Chiba, Y., Nose, T. and Ito, A., 2020. Automatic assessment of English proficiency for Japanese learners without reference sentences based on deep neural network acoustic models. *Speech Communication*, 116, pp.86–97.

[12] Giannakopoulos, T., 2015. Pyaudioanalysis: An open-source Python library for audio signal analysis. *PLoS ONE*, 10(12), p.e144610.

[13] Graves, A., Mohamed, A.R. and Hinton, G., 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp.6645–6649).

[14] Gulli, A. and Pal, S., 2017. *Deep Learning with Keras*. Packt Publishing Ltd, Birmingham.

[15] Khan, M.K. and Al-Khatib, W.G., 2006. Machine-learning based classification of speech and music. *Multimedia Systems*, 12(1), pp.55–67.

[16] Kulkarni, A., Iyer, D. and Sridharan, S.R., 2001. Audio segmentation. In *International Conference on Data Mining*. San Jose, California: IEEE.

[17] Khalid, S., Khalil, T. and Nasreen, S., 2014. A survey of feature selection and feature extraction techniques in machine learning. In *2014 Science and Information Conference* (pp.372–378). IEEE.

[18] Kotsiantis, S.B., Zaharakis, I. and Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. In *Emerging Artificial Intelligence Applications in Computer Engineering* (pp.3–24).

[19] Lantz, B., 2015. *Machine Learning with R*. Packt Publishing Ltd, Birmingham.

[20] Ziani, A., Adouane, A., Amiri, M.N. and Smail, S., 2024. New Proposed Solution for Speech Recognition Without Labeled Data: Tutoring System for Children with Autism Spectrum Disorder. *Informatica*, *48*(18).

[21] Liu, H. and Motoda, H., 2007. *Computational Methods of Feature Selection*. CRC Press, Boca Raton.

[22] McFee, B., Raffel, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E. and Nieto, O., 2015. Librosa: Audio and music signal analysis in Python. In *Proceedings of the 14th Python in Science Conference* (pp.18–24).

[23] Orozco-Arevalo, M.G., 2018. S-Impura en la pronunciación del idioma inglés en los estudiantes de la Universidad Central del Ecuador. *Bachelor's Thesis*, Quito: UCE.

[24] Piczak, K.J., 2015. Environmental sound classification with convolutional neural networks. In *25th International Workshop on Machine Learning for Signal Processing* (pp.17–20).

[25] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, pp.2825–2830.

MTCNN-UGAN: A Self-Attention Enhanced Face Replacement Pipeline for Film and Television Video Frames

Yumeng Wu, Xiao Wang*
School of Art, Zhejiang Yuexiu University, Zhejiang, 312000, China
E-mail: ssshayuya@163.com

Keywords: MTCNN, face replacement, adversarial generative networks, self-attention, digital de-jittering

Received: April 16, 2025

At present, face replacement technology in film and television videos faces problems such as low accuracy and high resource consumption. Research proposes an automated face replacement technology that integrates improved multi task cascaded convolutional neural networks (MTCNN) and generative adversarial networks (GAN). In the face detection stage, MTCNN is used, and a median filter preprocessing and depthwise separable convolution model are introduced. In the face replacement stage, a U-Net based generative adversarial network (UGAN) is constructed, whose generator consists of an encoder and a decoder, and is embedded with a dual skip connection residual module. The discriminator adopts a self attention mechanism and a video stabilization module. In the experiment, WIDER FACE and Celeb Faces Attributes Dataset (CelebA) were used for face detection tasks. The face replacement task used a high-resolution Celebrity Mask High Quality (CelebAMask HQ) dataset and a Deepfake Model Attribution Dataset (FDM). Meanwhile, the study introduced FaceSwap technology and attribute preserving generative adversarial network (AP-GAN) as comparative baselines. In face detection experiments, the research model performed best in terms of accuracy as well as training loss in different face detection scenes. For example, the accuracy of the research model in complex scenes was 93.25%, and the training loss was 0.221. In the face replacement experiment, the model replaces faces in four image sets. Its color as well as face contour structure was well preserved and face replacement was more natural. In the similarity index comparison, the research model performed the highest face replacement similarity index at different frame numbers with an average value of 0.994. The research model also performed the best in the face replacement imaging peak signal-to-noise ratio test with an average value of 35.65. Finally, in the face replacement composite test, the research model performed the best in both structural similarity and state error. In conclusion, the technique has good application results. This study can provide technical support for the improvement of face replacement technology as well as face characterization.

Povzetek: vtorja združita izboljšano zaznavanje obrazov (predhodno čiščenje šuma, lažji konvolucijski sloji in večločnostni prikaz), generator v slogu U-Net z dvojnimi povezavami, pametnejši diskriminator s samousmerjanjem pozornosti ter modul za stabilizacijo videa (odprava tresenja). Rezultat so naravni prenosi videza, visoka podobnost z izvirnikom in tekoče predvajanje tudi pri zahtevnih prizorih.

1 Introduction

To create a very realistic look, face-swapping technology is a type of picture and video processing tool that can swap out one person's face for another [1]. The technology originated from early image editing. Computer vision technology has advanced quickly in recent years and is now widely employed in advertising production, special effects in movies and television shows, and other sectors [2]. At present, with the continuous development of deep learning (DL), artificial intelligence (AI) and other technologies, the related face-swapping technology ushers in rapid development and attracts the attention of a large number of scholars [3]. To improve the application of face exchange technology, Rao et al. proposed an approach based on convolutional god coding and decoding network. In it, face marker point detection and alignment were combined with clustering and computer vision techniques, and a large amount of face data was trained to construct a face model. Face clusters were generated by clustering to optimize the face exchange effect. It was shown through experiments that this technique had good application effect, but it was poorly applied in low-end devices [4]. Omar K et al. proposed a DL bagging based integrated classifier for the deep forgery video detection problem, which employed convolutional and self-attentive networks as the basic learners. The model vertically stacked the deep convolutional and self-attention layers and extracted the local features of the face from the video and trained by learning to achieve face replacement. Finally, the study trained the technique on a public dataset. The results indicated that the technique had good video processing capability and high training accuracy [5]. Abdelminaan et al. studied the problem of detecting deep-fake videos. They developed a

web application that detected the authenticity of video input and protected public figures and politicians from false videos. The research adopted the method of combining machine learning and DL to analyze the data set containing deep forgery and real video. This method could effectively distinguish between real and forged content, such as face replacement or voice replacement. The results could be used in courts and police stations to reduce the risk of crime and fraud, while improving the detection efficiency and providing guarantee for the credibility of network information [6].

AI has made incredible strides in image and video processing thanks to the quick development of DL technology. Among them, video face replacement technology, as an emerging image and video editing technology, utilizes DL algorithms and neural networks to achieve highly realistic face replacement effects. Tsai C S et al. put up a useful framework for enhancing angle transformation and face replacement in order to address the issue of face replacement distortion. The framework contained a transformation framework based on generative adversarial networks, and generated multi-angle transformed images by combining the predicted face points to realize face recognition. The method's ability to preserve high-quality photos and prevent image distortion during face replacement with image angle changes was demonstrated in experimental tests [7]. Melnik et al. conducted a comprehensive review of DL methods for generating and editing faces in StyleGAN. The study analyzed the evolution of StyleGAN from PGGAN to StyleGAN3 and discussed key issues such as training indicators, potential representations, GAN inversion, and face image editing. The research also involved cross-domain face styling and restoration. This research served as an entry point into the field of face generation. It helped beginners quickly understand related technologies and promoted the development of face generation and editing technology. Additionally, it provided an important reference for subsequent research. Liao X et al. investigated the problem of compressed depth fake video detection in social networks and proposed a detection framework that considered facial muscle movement to realize the detection of face-swapped videos. The framework achieved this by localizing faces from consecutive frames, extracting marker points, and modeling sensory regions and face regions. Experimental results demonstrated that the method outperformed existing methods in detecting compressed depth pseudo-video [9]. The problem of deep face-swapping technique detection is studied by Akhtar Z et al. In this, a comprehensive review of existing images, videos, and Deepfake databases was conducted to propose a deep face-swapping detection framework. It adopted a new generation of deep feature point recognition technology, which was trained by a large amount of face data to realize false video detection. The experimental results indicated that the technique had a good detection accuracy, but the shortcomings of the technique were the lack of a unified detection standard and the poor applicability to the detection of new type of face data [10]. The comparison of relevant literature research is shown in Table 1.

Table 1: Literature review research

Researcher	Research contents	Comparison of research results with FaceSwap, DeepFaceLab, AP-GAN, etc	Compared with FaceSwap, DeepFaceLab, AP-GAN, etc
Rao I S S et al. [4]	Facial swapping technology based on convolutional neural networks, combined with clustering and computer vision techniques	The application effect is good, but it performs poorly on low-end devices	Compared with FaceSwap, it optimizes face alignment but has a higher computational complexity
Omar K et al. [5]	Deep fake video detection based on deep learning bagging ensemble classifier	Strong video processing ability and high training accuracy	Compared to DeepFaceLab, it focuses more on extracting local video features, but the model is complex
Abdelminaam d et al. [6]	It studies face video forgery, and detects it combined with related voice and portrait images	High accuracy and good efficiency, but high requirements for hyperparameter adjustment	Compared with traditional AP-GAN, it optimizes computational efficiency but requires higher demands on the dataset
Tsai C S et al. [7]	An effective framework for improving angle transformation and face replacement based on generative adversarial networks	Maintain high quality during image angle transformation to avoid distortion	Compared to traditional FaceSwap, it solves the problem of image distortion, but the training difficulty is higher
Melnika et al. [8]	Review and analyze the current technologies related to face generation and editing, and analyze the effects of different technologies	The detection effect is superior to similar models	Compared with the benchmark DeepFaceLab, the detection accuracy is higher,

			but the model dependency is strong
Liao X et al. [9]	A compressed depth pseudo video detection framework considering facial muscle movement	The effect of detecting compressed depth pseudo video is better than existing methods	Compared to the benchmark AP-GAN, it is more suitable for compressed video detection, but its generalization ability is limited
Akhtar Z et al. [10]	Deep face swapping detection framework, utilizing next-generation deep feature point recognition technology	High detection accuracy, but lack of unified standards, poor applicability to new data	Compared to FaceSwap and DeepFaceLab, it has higher detection accuracy, but its applicability is limited

According to the above research, with the development of DL as well as AI and other technologies in recent years, video face-swapping technology has seen rapid development. It has been used in video creation, face data detection and other fields. However, according to the above study, the current video face replacement technology still faces many shortcomings. For example, the low similarity of face replacement, the high consumption of face replacement technology resources, and the poor adaptation of complex scene technology all limit the development of the technology. Current research problems include insufficient adaptability to complex scenes. For example, there is a significant decrease in detection accuracy under occlusion and lighting changes. The consumption of computing resources is too high. For example, traditional multi task cascaded convolutional neural networks (MTCNN) detection models have a large number of parameters, which makes feature pyramid calculation inefficient and difficult to meet the real-time processing requirements of film and television videos. Therefore, in order to solve the face replacement consumption as well as accuracy problems, the study proposes a video face replacement technology based on improved MTCNN. There are two innovations in the research. One is to adopt the improved MTCNN algorithm, which improves face detection accuracy and efficiency through optimization such as median filtering and depth separation convolution. Second, the research constructs face replacement model based on adversarial generative network. It adopts the self-attention mechanism (SAM) and digital de-jittering method to improve the face replacement accuracy and video smoothness. This research can provide technical support for the improvement of video face replacement technology.

2 Methods and materials

2.1 Face detection model for film and television videos based on improved MTCNN algorithm

With the rise of AI as well as video creation, film and television works face replacement technology is more and more sought after. However, the traditional technology is inefficient and the replacement color deviation is large, which cannot meet the requirements of film and television videos creation. In this regard, the study proposes a face replacement technology that combines MTCNN algorithm and adversarial generative network. The flowchart of the whole technology is shown in Figure 1.

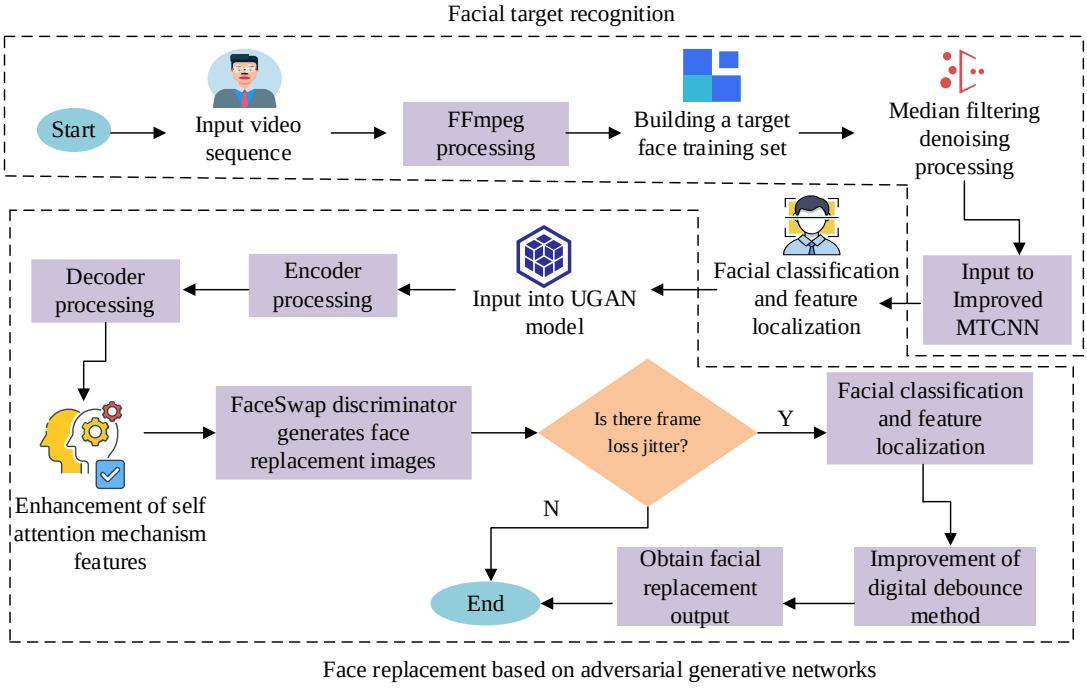


Figure 1: Process of automatic face replacement technology in film and television videos

In Figure 1, the technique has two parts. The first part is the face feature detection part and the second part is the face replacement based on adversarial generative networks. The face detection part of the study uses MTCNN algorithm as face detection. It has high face detection accuracy, suitable for all types of face feature recognition, and excellent performance in the face detection field. Figure 2 depicts the MTCNN's three-stage structure.

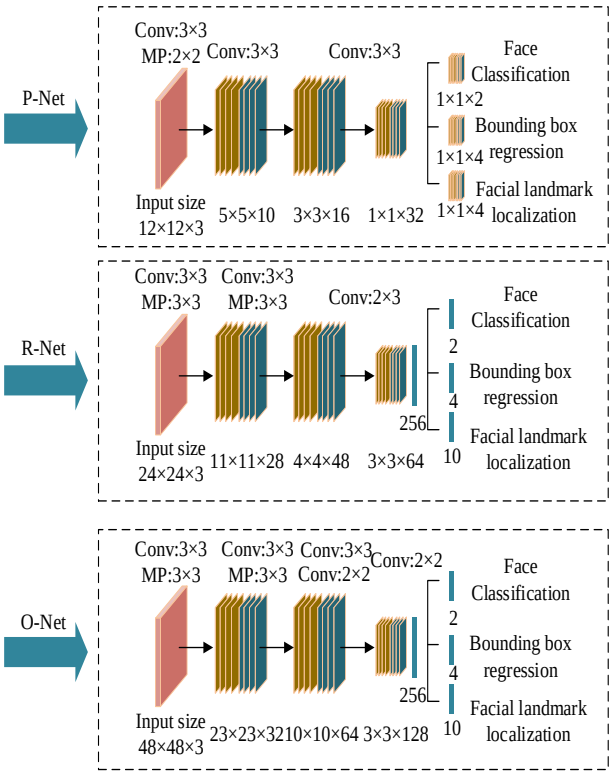


Figure 2: Three stage structure of MTCNN

According to the MTCNN, the network uses P-net network to recognize the face region in the first stage of face recognition. It mainly achieves face feature classification as well as feature localization through multilayer convolutional processing [11]. The R-net network is used in the second stage to detect faces. It eliminates non-face candidate frames through non-maximum suppression (NMS) and other processing methods, and strengthens the analysis of face features, localization data, etc [12]. O-Net network is used in the third stage. This process refines the bounding box detection region and adds new convolutional layer processing. In the process of face classification, the image will be recognized whether it is a face or not. Equation (1) illustrates how a cross-entropy loss function represents the process [13].

$$L_i^{det} = -\left(y_i^{det}\log(p_i) + \left(1 - y_i^{det}\right)\left(1 - \log(p_i)\right)\right) \quad (1)$$

In Equation (1), p_i represents the network recognition as face probability output. $y_i^{det} \in \{0,1\}$ is the sample detection label and the output is 1 or 0, 0 is non-face and 1 is face. Whereas, in face bounding box processing, Euclidean loss is used to reflect the candidate library detection, as shown in Equation (2).

$$L_i^{box} = \hat{y}_i^{box} - y_i^{box}{}_2^2 \quad (2)$$

In Equation (2), y_i^{box} represents the bounding box of the face in detection. $\hat{y}_i^{box}$ represents the coordinates of the predicted corrected bounding box. Next, the minimum Euclidean loss is used in face localization to reflect the location of key features of the recognized face, as shown in Equation (3).

$$L_i^{landmark} = \hat{y}_i^{landmark} - y_i^{landmark}{}_2^2 \quad (3)$$

In Equation (3), $y_i^{landmark}$ is the real human face feature key point coordinates. $\hat{y}_i^{landmark}$ represents the

predicted face feature keypoint coordinates. Then the MTCNN loss optimization is obtained based on the above analysis, as shown in Equation (4) [14].

$$min\sum_{i=1}^{N}\sum_{j\in\{det,box,landmark\}}\alpha_j\beta_i^j L_i^j \quad (4)$$

In Equation (4), a_j represents the weight set without loss, which reflects the degree of influence of the loss function. Its weight set is 1. The face classification weight is 0.3. The bounding box regression loss weight is also 0.3. The keypoint localization loss weight is 0.4. N represents the total number of training face samples $\beta_i^j \in \{0,1\}$ represents the sample type indicator. 1 is face and 0 is non-face. Although MTCNN has excellent recognition effect on face detection compared to the traditional target detection model, the traditional MTCNN still faces the problems of high resource consumption, low detection efficiency, and low processing effect on noisy images. For example, before constructing the face feature map, some of the images captured by the camera system contain noise such as Gaussian and pretzel, which affects the network detection effect. Therefore, the input samples are pre denoised and then fed into the MTCNN model for training [15]. The median filtering process is shown in Equation (5).

$$g(x,y)=median\{f(x-m,y-n)\,|\,(m,n)\in W\} \quad (5)$$

In Equation (5), $f(x-m,y-n)$ denotes the filtered pixel position with offset. $g(x,y)$ denotes the original map pixel position. W denotes the filter window. $median$ is the median value of the filter element. m and n are the window offset parameters. In MTCNN, network will extract the features of each layer of the image pyramid to obtain multi-scale image features, but the process is computationally intensive. To solve the problem, the study considers image pyramid and feature map pyramid. The former predicts only the high-level features in multilayer feature extraction, which improves the efficiency but decreases the detection accuracy [16]. Unlike the image pyramid, the latter considers different layer feature prediction relationships in the multiscale feature output stage. This reduces the amount of computation while ensuring training accuracy. Therefore, the feature pyramid is integrated prior to MTCNN inference in order to construct an image pyramid of the input image and generate a multi-scale image sequence. Finally, the study also introduces depth-separated convolution to improve the MTCNN convolutional operation parameter problem and enhance the network detection efficiency. Moreover, depthwise depth-separated is integrated into the convolutional layers of P-Net, R-Net, and O-Net to reduce computational and parameter complexity. For example, in R-Net/O-Net, depth-separated is applied to all convolutional layers. For example, in R-Net, the input features are first deeply convolved and then combined across channels via 1×1 convolution to output the bounding box and keypoint coordinates. The activation function adopts LeakyReLU instead of ReLU to enhance the retention of negative value information. The MTCNN structure is shown in Figure 3.

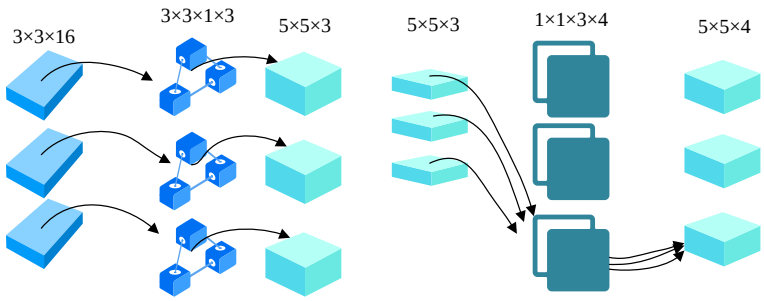


Figure 3: Depth separable convolution

Figure 3 shows the depth separable convolution process (CP). It resizes the input 7×7×3 feature map to 5×5×4 size by convolving with channel number 1. Compared to the standard CP, it preserves the number of image feature channels and guarantees the quality of the graph while reducing the height and width of the feature graph. In this, the depth convolution is shown in Equation (6).

$$Y_{dw}^{(c)}(x,y)=\sum_{i=k}^{k}\sum_{j=-k}^{k}X^{(c)}(x+i,y+j)\cdot K_{dw}^{(c)}(i,j)+b_{dw}^{(c)}(\quad (6)$$

In Equation (6), $X^{(c)}(x+i,y+j)$ denotes the value of the output tensor C th channel position $(x+i,y+j)$. $K_{dw}^{(c)}(i,j)$ denotes the (i,j) th position weight of the deep convolution at C channel. $b_{dw}^{(c)}$ denotes the C th channel bias of the depth convolution. $Y_{dw}^{(c)}(x,y)$ denotes the output at channel position (x,y). Next, the point-by-point convolution operation is shown in Equation (7).

$$Y_{pw}^{(c)}(x,y)=\sum_{c=1}^{C_{in}}Y_{dw}^{(c)}(x,y)\cdot K_{pw}^{(c,c')}+b_{pw}^{(c')} \quad (7)$$

In Equation (7), $Y_{dw}^{(c)}(x,y)$ denotes the value of the deep convolutional C th channel at position (x,y). $K_{pw}^{(c,c)}$ denotes the weight of the point-by-point

convolution kernel from the input channel C to the output channel C'. $b_{pw}^{(c')}$ denotes the point-by-point convolutional C'th output bias. $Y_{pw}^{(c)}(x,y)$ denotes the value of the final output C'th channel at position (x,y). Finally, to improve the training accuracy of MTCNN face replacement classification task, SoftMax loss function is used to replace the cross loss function. Finally, to improve the training accuracy of the MTCNN network for face replacement classification tasks, the SoftMax loss function is introduced, based on the original cross-loss function. The squared-difference loss function is used to evaluate the performance of face-frame detection predictions. The SoftMax loss function is shown in Equation (8) [17].

$$SoftMax(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_i}} \quad (8)$$

In Equation (8), x_i denotes sample i and y_i denotes labeled values. The squared deviation loss function is shown in Equation (9).

$$Loss == \frac{1}{N'} \sum i = 1^{N'} \beta_i \cdot \left(\| y_pred^{(i)} - y_gt^{(i)} \|_2^2 \right) (9)$$

In equation (9), $y_pred^{(i)}$ represents the predicted value, $y_gt^{(i)}$ represents the true value, β_i represents the sample type indicator, and N' represents the sample size.

2.2 Adversarial generative network-based automatic face replacement model for film and television videos

After completing the recognition of face images, the next step is to perform automatic video face replacement modeling. This part uses adversarial generative networks as face replacement technology. Currently, adversarial generative networks have become a representative of the field of image synthesis and transformation. Compared with the traditional DeepFakes class of face replacement technology, its face replacement accuracy is high and can be automatically replaced for the input source. Therefore, the study proposes a novel automatic face switching generation framework for self-coding networks (U-Net) on the framework of deep convolutional adversarial networks (DCGAN), which is referred to as the UGAN model. The UGAN model mainly contains two parts: generator as well as discriminator. Figure 4 depicts the particular structure.

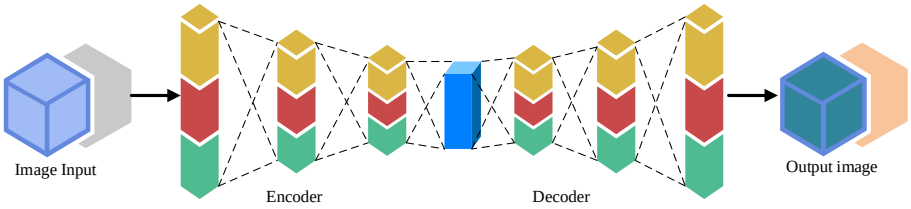


Figure 4: UGAN model generator structure

According to the structure of Fig. 4, there are two main components, Encoder and Decoder. In order for the generative model to output a specified realistic image according to the target, the feature information within the source image is extracted by the Encoder, including the background, face contours, etc. Moreover, this information is converted into a latent word feature vector. Then the Encoder processing vectors are input to the Decoder. It is mainly responsible for reconstructing the feature vectors, preserving more details of the target and outputting the reconstructed features. The Encoder in the generator first downsamples the input size of 256×256 source image to extract multiple face feature attributes. The UGAN Architecture with layer names, filter sizes, and skip connection types is in Table 2.

Table 2: UGAN Architecture with layer names, filter sizes, and skip connection types

Network	Layer Name	Enter size	Filter size/quantity	Step length	Fill in	Activation function
P-Net	Input	-	3 channels	-	-	-
Conv1	Any size (≥ 12 × 12)	10, 3×3	1	Valid	PReLU	-
MaxPool1	-	2×2	2	Same	-	-
Conv2	-	16, 3×3	1	Valid	PReLU	-
Conv3	-	32, 3×3	1	Valid	PReLU	-
Output (Cls/Reg)	-	2/4, 1×1	1	Valid	Softmax/Linear	Dual branch output
R-Net	Input	24×24×3	-	-	-	-
Conv1	24×24×3	28, 3×3	1	Valid	PReLU	-
MaxPool1	-	3×3	2	Same	-	-
Conv2	-	48, 3×3	1	Valid	PReLU	-

MaxPool2	-	3×3	2	Valid	-	-
Conv3	-	64, 3×3	1	Valid	PReLU	-
FC1	576 dimensions	Unit 128	-	-	PReLU	Fully connected layer
Output (Cls/Reg)	-	2D/4D	-	-	Softmax/Linear	Dual branch output
O-Net	Input	48×48×3	-	-	-	-
Conv1	48×48×3	32, 3×3	1	Valid	PReLU	-
MaxPool1	-	3×3	2	Same	-	-
Conv2	-	64, 3×3	1	Valid	PReLU	-
MaxPool2	-	3×3	2	Valid	-	-
Conv3	-	64, 3×3	1	Valid	PReLU	-
MaxPool3	-	2×2	2	Same	-	-
Conv4	-	12, 3×3	1	Valid	PReLU	-
FC1	1152 dimensions	256 units	-	-	PReLU	Fully connected layer
Output (Cls/Reg/Landmark)	-	2D/4D/1D	-	-	Softmax/Linear/Linear	Three task branches

Whereas in Decoder the up-sampling is done in the form of pixel CP to adjust the height and width $H \times W$ image to r times high resolution $rH \times rW$ image. Among them, r denotes the sampling factor times. Pixel rearrangement by Decoder is shown in Equation (10).

$$TS(T)_{x,y,c} = T_{\lfloor y/r \rfloor, \lfloor x,r \rfloor c \square r \square mod(y,r)+c \square mod(x,r)} \quad (10)$$

In Equation (10), $\mathrm{mod}(x, y)$ is the sampling coordinate position. An improved residual network model is added to the structure of the network when processing image data, especially when training numerous face data, in order to reduce the network's computational load. It makes use of a double jump connection topology, which can successfully address the gradient vanishing problem in data processing. The residual function expression is shown in Equation (11) [18].

$$F(x) = Conv_{3\times3}(Leaky\,\mathrm{Re}\,LU(Conv_{3\times3}(x))) \quad (11)$$

In Equation (11), x is the input information. $Conv$ is the convolutional processing. $LeakyReLU$ is the activation function. In this case, the input information x is processed twice using 3×3 convolutional processing to strengthen the feature reuse capability. Then, activation processing is performed by LeakyReLU activation. Next, the output information x is summed with the residual function for residual linkage. The study selects LeakyReLU activation for processing due to its high computational efficiency and suitability for real-time film and video processing requirements, despite its slightly slower convergence compared to GELU/Wish. LeakyReLU ensures stable convergence during training by avoiding neuronal death. The residual module is embedded in the encoder-decoder structure of the U-Net generator. This structure achieves cross-layer feature fusion through dual skip connections. It solves the problem of gradient vanishing and improves the ability to preserve facial details. At the same time, the robustness of forgery detection is enhanced by weighting key features in collaboration with the SAM in the discriminator. The residual module is shown in equation (12) [19].

$$\mathrm{ResBlock}(x) = x \oplus \mathrm{Conv}_{3\times3}(\mathrm{LeakyReLU}(\mathrm{Conv}_{3\times3}(x))) \quad (12)$$

Next, after completing the image feature extraction work through the generator, the study uses the network face replacement framework (FaceSwap) discriminator for face replacement processing. The framework mainly consists of a SAM layer as well as a convolutional layer. The structure of the discriminator framework is shown in Figure 5 [20].

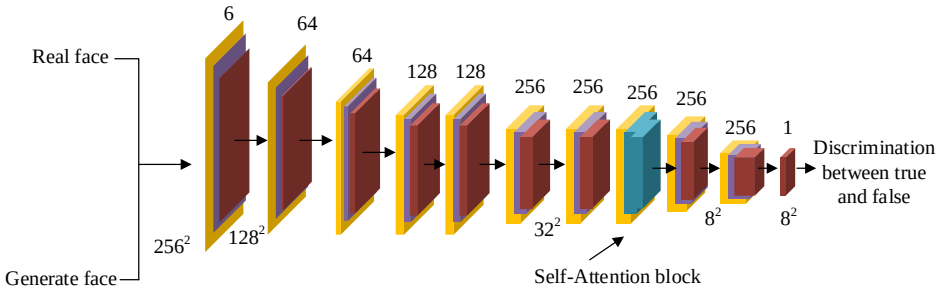


Figure 5: Discriminator framework structure

In Figure 5, in the discrimination framework, through iterative processing, the discriminator will discriminate the similarity between the generated face as well as the real face to ensure that the network generated image is closer to the real target face. Within the face replacement framework, a special layer of SAM is added. Its function is to weigh features based on their relationships in order to capture long-term dependencies and strengthen attention to key features. The mathematical expression of self attention mechanism is shown in equation (13).

$$\text{Attention}(Q,K,V)=\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (13)$$

In Equation (13), Q, K, and V denote the query vector, key vector, and value vector, respectively. d_k denotes the dimension of the key vector. d_k is the sequence length. After face replacement processing is completed by FaceSwap framework, the face replacement process still faces the video replacement jitter problem. In other words, during the face replacement session, the network performs replacement processing on multiple faces. This results in some video frames not being continuous, leading to video face loss and frame jitter. In this regard, the research introduces a digital de-jittering method. Its flow is shown in Figure 6.

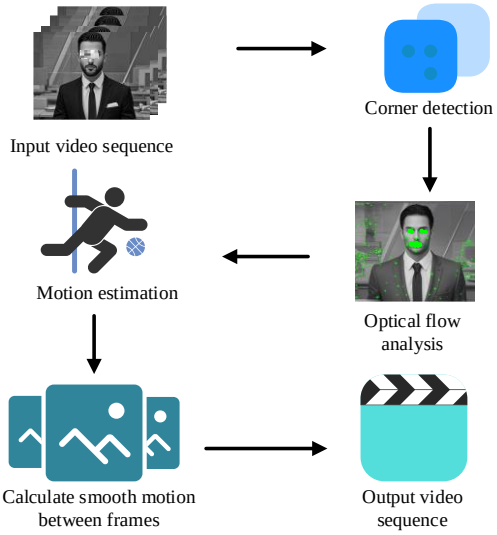


Figure 6: Process of face replacement video debounce

In Figure 6, this de-jittering process is realized through corner point detection, optical flow analysis, and motion estimation to achieve de-jittering. Among them, the process is realized by using Shi-Tomasi, a corner detection tool within OpenCV, which detects the gray level change anomalies in the image frame as corner points through a fixed window. Then optical flow analysis is performed for tracking the feature points in the next frame of the image. Lucas-Kanade is used as the analyzing method in the study, although it is not able to track all the moving points in some scenes. Therefore, the marker code mechanism is used to improve the problem. When the next frame tracking marker position is determined the status is marked as 1, otherwise 0 will re-update the tracking points. After completing the above analysis for motion estimation, the study uses random sample consistency (RSC) algorithm to estimate the variation between frames. The process uses Euclidean transform to reflect the superposition relationship of the features of each frame as shown in Equation (14).

$$S_E=\begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix}=T_E\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (14)$$

In Equation (14), a_{11} denotes the translation factor.

a_{12} denotes the left-right rotation factor. a_{21} is the up-down rotation factor. a_{22} denotes the scaling factor. t_x and t_y are both transformation parameters. Next, the error between the front and back change frames is calculated for judging the estimated position. The sum of the errors between the front and back change frames is shown in Equation (15).

$$SAD = \sum_{i=1}^{N}\sum_{j=1}^{N}\left|c_{i,j} - r_{i,j}\right| \quad (15)$$

In Equation (15), $r_{i,j}$ represents the previous frame feature point location. $c_{i,j}$ represents the next frame feature point location. Finally, the study uses Smooth Filter to smooth the filter processing curve, which makes the video frame smoother. With the above processing, the automatic replacement of faces and the face replacement jitter problem are accomplished. The technical analysis has been completed through the above research, in which the improved MTCNN (Deep Separation Convolution+Feature Pyramid) and UGAN (Lightweight U-Net Generator) are synergistically optimized and meet the real-time running requirement of 30+FPS.

3 Results

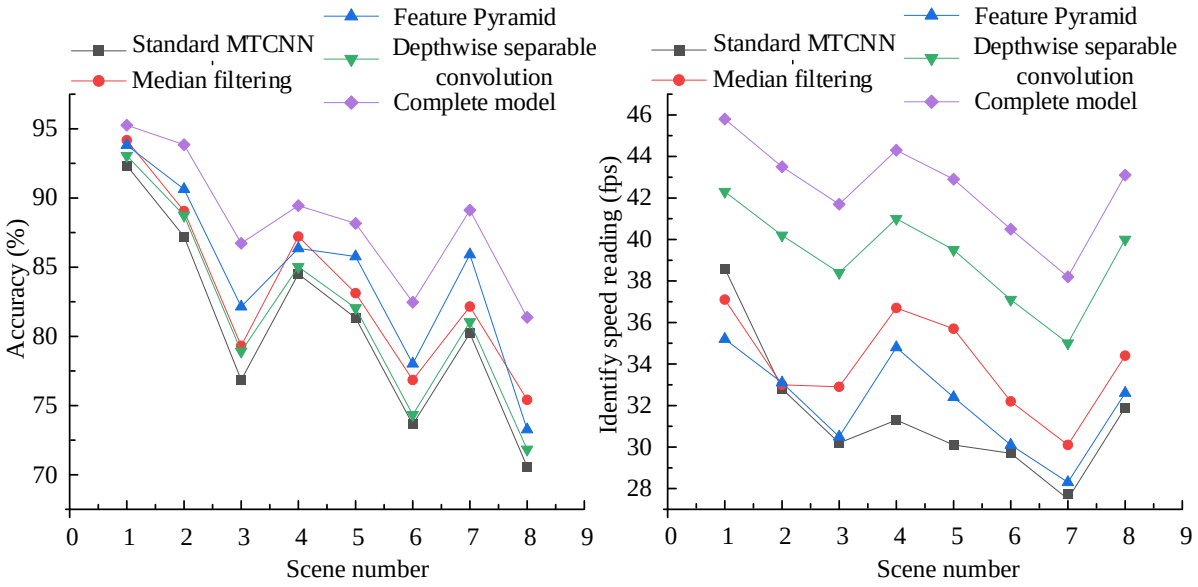
3.1 Face detection experiment based on improved MTCNN

Next, in order to test the proposed technology of the study, the corresponding face detection experiments and face replacement experiments will be carried out. Among them, the experimental system adopts WIDOWS 11 system, the processor adopts AMD Ryzen R5600, the graphics card is Nvidia RT3060, and the memory is 32GB DDR4 3200MHz. The experimental platform is Pyeharm 2021.1.1. The MTCNN face detection training time in the experimental training is 2.5 hours. The face replacement time is 28 hours, and an additional 3 hours are needed for debouncing. The parameter settings of the improved MTCNN are shown in Table 3.

Table 3: Model initial parameters

Parameter indicator type	Numerical value
Batch-size	16
Learning-rate	0.001
Weight	0.0005
Factor	0.7
Minsize	15

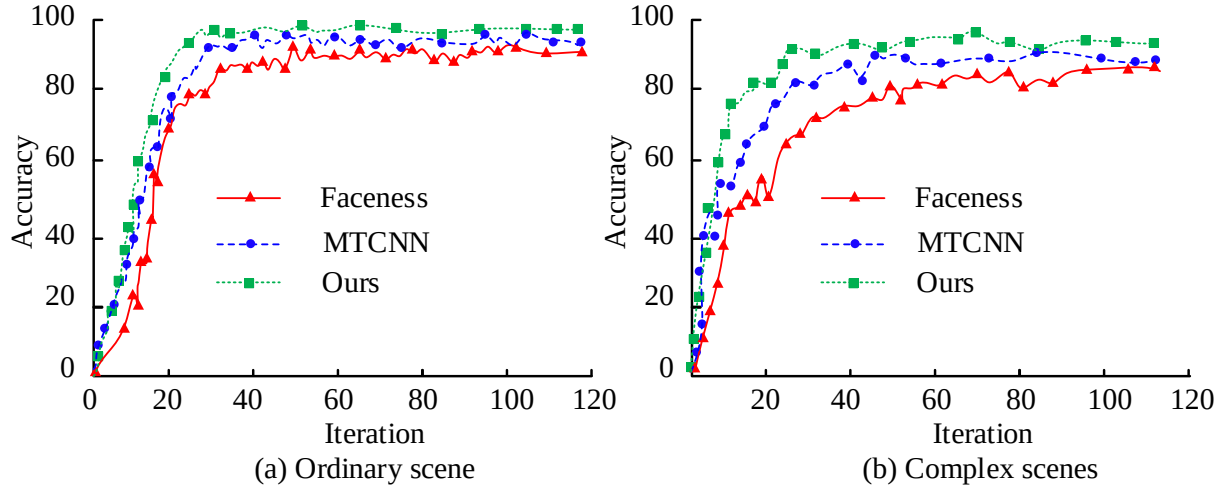
Table 3 shows the parameter settings for training the model, where Batch size is set to 16. A smaller batch size is selected to balance GPU memory limitations and training stability. This ensures that memory does not overflow and that training remains stable. To ensure stable training, Adam optimizer uses a standard learning rate of 0.001. The L2 regularization weight is set to 0.0005 to suppress overfitting. The image pyramid scaling factor is set to 0.7 to optimize the efficiency of multi-scale face detection. Minsize is set to 15, with a minimum facial pixel size of 15, to filter out noise and dryness. To effectively detect facial contours and provide a basis for analyzing information for subsequent face replacement, the professional face data WIDER FACE and the CelebFaces Attributes (CelebA) dataset are used to train the face detection model. The CelebA dataset mainly consists of a celebrity facial attribute dataset (with 40 attribute annotations such as glasses and hats), which includes 202599 facial attribute datasets and various types of facial image sets. The WIDER FACE dataset covers natural facial images of complex scenes (occlusion, lighting changes, blurring), with a total of 32203 images. First, the normal scene and complex scene (occlusion, dim light) of WIDER FACE dataset are selected for accuracy test. First, an ablation experiment is conducted. 8 types of image scenes are selected for analysis, including ordinary scenes, complex occlusion, small-scale faces, low lighting, large angle deflection, high dynamic blur, crowded scenes with multiple people, and strong noise interference, corresponding to image sequence numbers 1 to 8. The results of the ablation experiment are shown in Figure 7.



(a) Identification accuracy (b) Recognition rate

Figure 7: Results of ablation experiment

Figure 7(a) illustrates the recognition accuracy of the MTCNN model across various modules. According to the test results, standard MTCNN performs the worst in different scenarios, particularly in scenarios involving small faces and high dynamic models, achieving an accuracy below 80%. The best performing complete model has a performance rate of 87.6% and 83.6% in small-sized face and high dynamic model scenarios, respectively. Figure 7 (b) shows the comparison results of recognition rates. According to the test results, the complete model's average recognition rate is 42.5 fps. This is significantly better than the standard MTCNN's rate of 32.8 fps and better than a single MTCNN combination. In summary, the results indicate that adding modules such as median filtering and feature pyramids can significantly improve the model's application performance. Moreover, Faceness-Net (Faceness) and standard MTCNN are introduced as test benchmarks. The results are shown in Figure 8.



(a) Ordinary scene (b) Complex scenes

Figure 8: Accuracy of face detection in different scenarios

In Figure 8(a) for common scene detection, the fastest convergence of the research model compared to similar techniques is 98.35% accuracy at convergence of 30 iterations. Whereas MTCNN and Faceness converge with an accuracy of 92.31% and 82.65% respectively. In the complex scene detection in Figure 8(b), there is a significant fluctuation in Faceness face detection with the lowest overall accuracy of 81.25%. In comparison, the complex scene research model performs the best. The accuracy at convergence is 93.25% compared to 86.25% for MTCNN. Next, the same scene is compared to the training loss of different techniques, as shown in Figure 8.

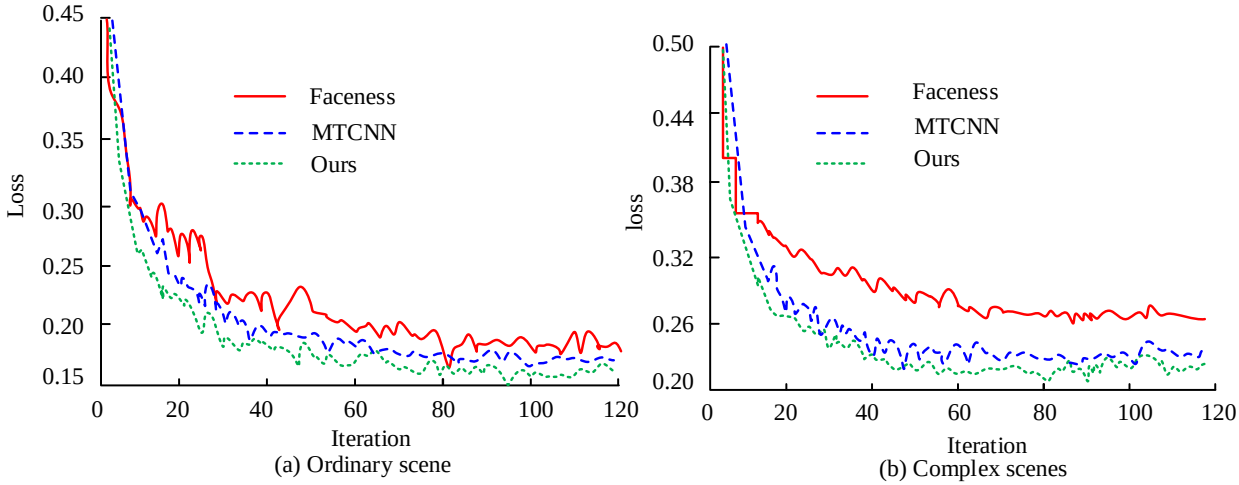
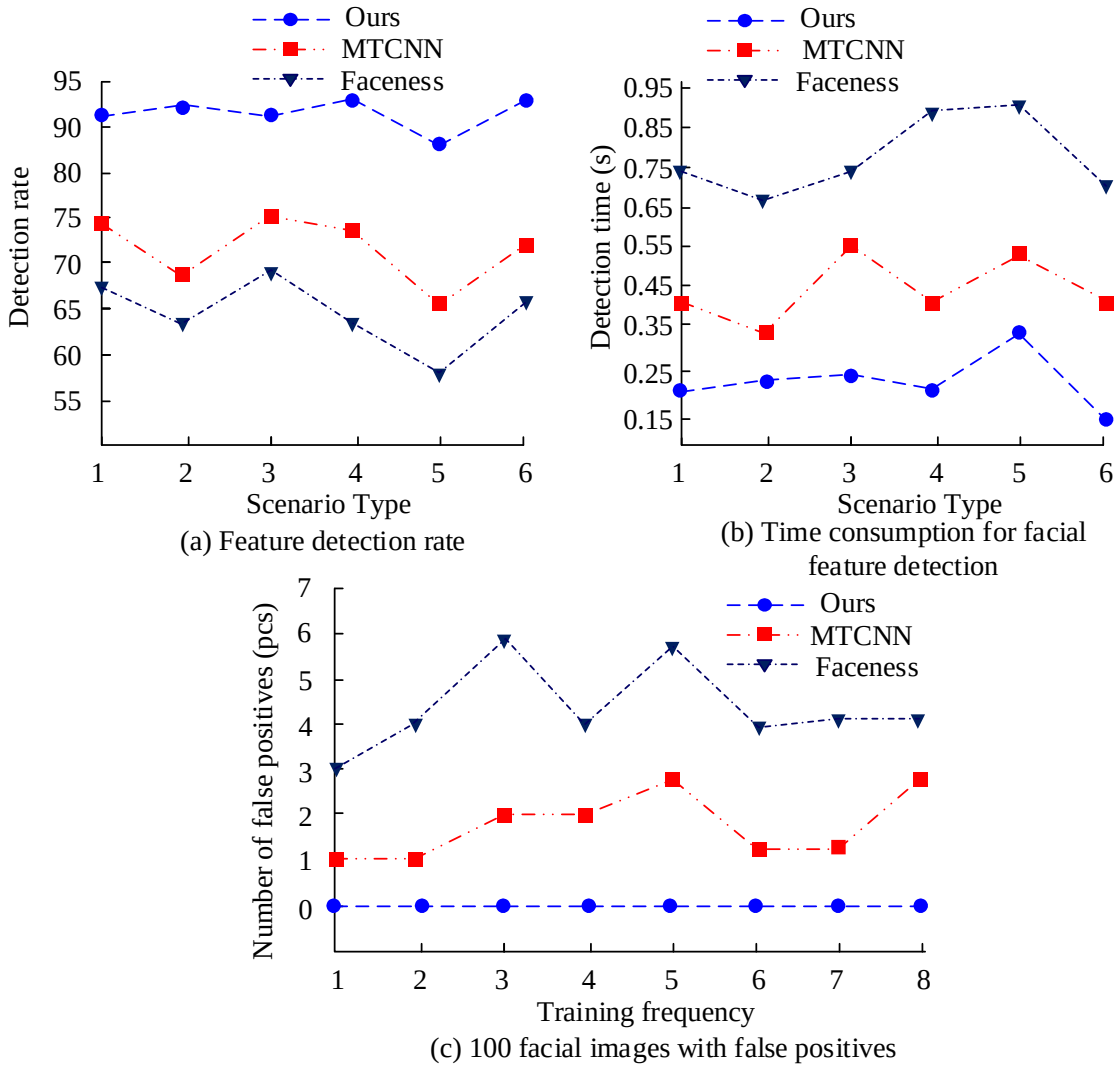


Figure 9: Face detection training loss

Figure 9(a) shows the results of training loss for ordinary scenes. Among them, only the research model has the lowest loss of 0.165 at convergence, while the training loss of MTCNN and Faceness at convergence is 0.184 and 0.205, respectively. In Figure 9(b), which shows the training loss of the complex scene, Faceness has the worst overall training effect and shows obvious fluctuations during the training process. The loss at convergence is 0.286, while MTCNN and the research model perform significantly better, 0.256 and 0.221, respectively. Next, the study uses CelebA data to test the feature detection rate of different face detection techniques and the number of false face detections. Among them, the detection rate indicates the ratio of detected face features to total features, while the number of false detections indicates the number of face judgment errors. The experiment sets up six facial recognition scenarios. Scene 1 is a low-light environment with a light intensity of $\leqslant$50 lux for facial detection and recognition. Scene 2 is a strong backlight scene, where the face is in front of a strong light source (such as sunlight or spotlights) background. Moreover, the facial brightness is $\leqslant$ 100 lux, while the backlight ambient light is $\geqslant$ 10000 lux. Scene 3 is a partially occluded scene, where the face is partially occluded by objects (masks, sunglasses, hands) (covering 30% to 50% of the face), simulating character camouflage or temporary occlusion in movies and TV shows. Scene 4 is a high angle deflection scenario, where the face has significant deflection (side face, pitch) relative to the camera. This exceeds the conventional frontal recognition range. Among them, the horizontal deflection angle is $\geq$ 45 ° or the vertical tilt angle is $\geq$ 30 °. Scene 5 is a motion blurred scene, where facial images are blurred due to rapid movement or camera shake, simulating action scenes or handheld shooting. Scene 6 is a small-sized face scene, and long-distance shooting results in a small proportion of the face in the image and insufficient detail resolution. The specific results are shown in Figure 9.



(a) Feature detection rate

(b) Time consumption for facial feature detection

(c) 100 facial images with false positives

Figure 10: Comprehensive testing of face detection

Figure 10(a) shows the results of feature detection rate. Among the six face scenarios, the research model has the highest detection rate for all types of face features with an average value of 93.54%. MTCNN, which is the next best performer, has an average value of 71.25%. Faceness is only 63.25%. Figure 10(b) shows the time-consuming face feature detection. Overall, Faceness face detection takes the longest time, with a mean value of 0.765s. It is followed by MTCNN, with a mean value of 0.452s. The research model is the shortest, with only 0.253s. Finally, Figure 9(c) shows the final false detection rate result for one hundred faces. The number of false detections represents the number of misidentifications among the recognized faces, measured in individuals (pcs). The training frequency refers to the number of times a model is repeatedly trained on specific interference samples (such as background textures that are prone to false positives) during the training process. The mean value of eight repetitions of the test is 4.37 misdetections per hundred compared to 1.75 for MTCNN. The research model does not suffer from the problem of misdetection and the number of misdetections is zero.

3.2 Face replacement experiment

Next, the face replacement experiments are continued. The study uses a high-resolution face dataset (CelebAMask-High Quality, CelebAMask-HQ) and a specialized video face dataset (Deepfake Model Attribution Dataset, FDM) for the experiments. Among them, CelebAMask-HQ dataset contains 30,000 face data of various types with a mask size of 512×512, which is suitable for face switching experiments. FDM contains 6,450 videos of various types of faces. For testing, the study introduces attribute-preserving generative adversarial network (AP-GAN) and FaceSwap as test benchmarks. Figure 11 first compares the effects of several face replacement approaches using the CelebAMask-HQ dataset.

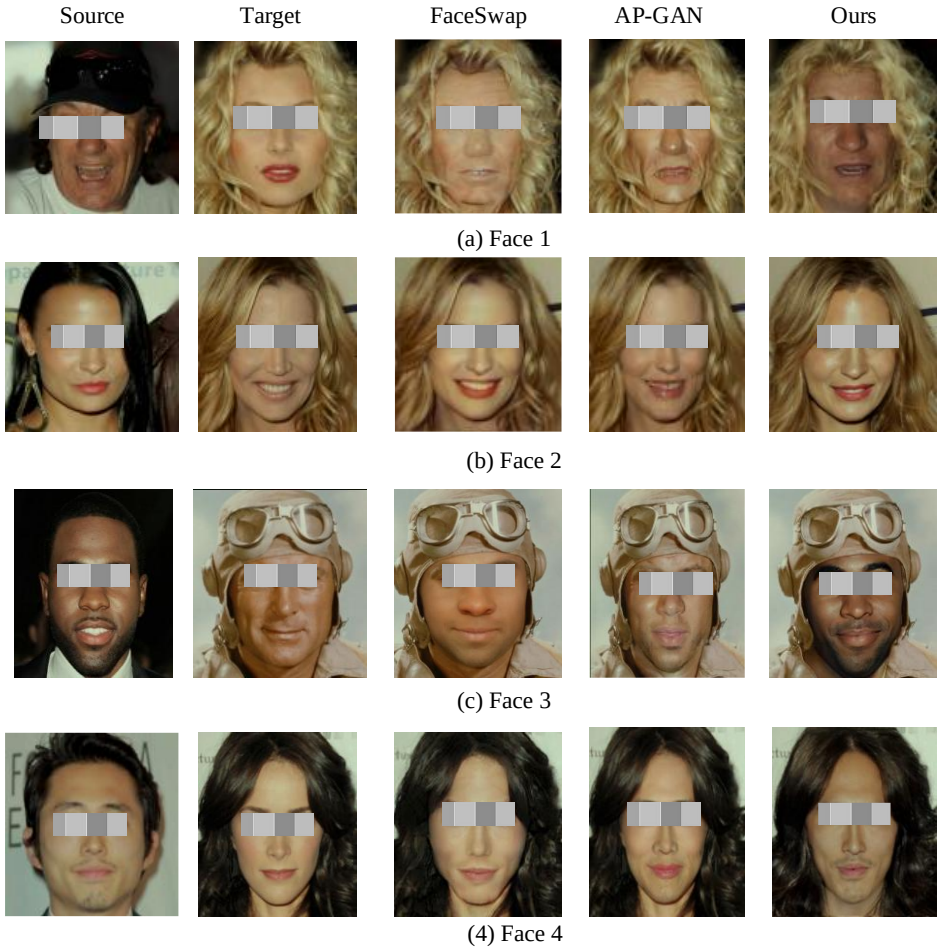
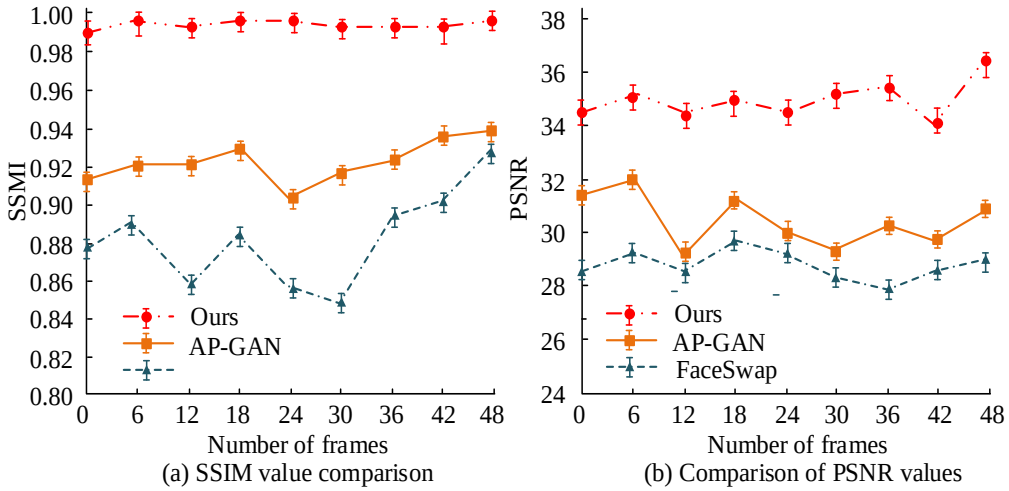


Figure 11: Face replacement effect

In the face image in Figure 11(a), FaceSwap face replacement has problems with skin color differences and mouth shape deviation, while AP-GAN is overall better, but still has problems with skin color deviation as well as jaw overexposure. Only the study that restores the model to the original face contour and skin color performs the best. In Face 2 of Figure 11(b), again only the research model does not show face contour bias as well as color problems. In Face 3 of Figure 11(c), FaceSwap and AP-GAN show the problem of whiteness of face skin color, and FaceSwap face contour is obviously abnormal. The research model, on the other hand, effectively restores the skin color and contour of the original image. In Figure 11(d) of Face 4 the research model has the best color and contour restoration, while all other techniques show contour and color deviations. Next, under the FDM dataset, the study is introduced to introduce structural similarity index (SSMI) and peak signal-to-noise ratio (PSNR) to compare the face replacement effect of different techniques, as shown in Figure 12.



(a) SSIM value comparison (b) Comparison of PSNR values

Figure 12: Comparison between SSMI and PSNR

Figure 12(a) shows the SSIM test results. The research model significantly outperforms the other techniques. In the 0 to 48 frames count scenario, the research model has an average SSMI value of 0.994. In contrast, the performance in AP-GAN face replacement is average, with an average SSMI value of 0.924. The worst performer, FaceSwap, has an average SSMI value of 0.875. Figure 12(b) shows the results of the PSNR comparison. Similarly in the 0-frame to 48-frame face replacement scene, only the research model has a high PSNR value, indicating that it reconstructs faces with better quality. The average PSNR values of the research model, AP-GAN, and FaceSwap are 35.65, 31.05, and 28.84, respectively. The FDM face data is selected to compare the effect of different techniques face replacement. The state error measures the degree of matching of facial dynamic attributes, including expressions, poses, and motion coherence. For video face swapping to look natural, the replaced face must be synchronized with the target video's dynamic changes (e.g., blinking or turning the head). Otherwise, there will be uncoordinated "shaking" or "lag." attribute errors reflect the ability to preserve static identity features, including inherent attributes such as skin color, facial features, facial contours, etc. The higher the error, the lower the similarity between the replaced face and the source identity (such as skin whitening, contour deformation). The details are shown in Table 4.

Table 4: Comprehensive testing of face replacement using different technologies

Face replacement number	FaceSwap			AP-GAN			Ours		
	Structural similarity	State error	Attribute error	Structural similarity	State error	Attribute error	Structural similarity	State error	Attribute error
1	0.726	3.548	0.458	0.851	3.868	0.322	0.982	2.541	0.124
2	0.756	4.055	0.425	0.826	2.956	0.357	0.986	2.057	0.114
3	0.743	3.853	0.405	0.816	3.457	0.353	0.991	2.043	0.113
4	0.735	3.754	0.425	0.788	3.054	0.342	0.986	2.044	0.125
5	0.684	3.686	0.432	0.816	2.985	0.325	0.991	2.123	0.135
6	0.765	4.285	0.432	0.823	2.973	0.325	0.998	2.255	0.126
7	0.726	3.785	0.456	0.838	3.054	0.315	0.983	2.034	0.122
8	0.715	4.066	0.428	0.793	3.522	0.334	0.986	2.015	0.155
9	0.706	3.856	0.416	0.834	2.893	0.332	0.993	2.132	0.153
10	0.698	3.852	0.405	0.835	3.085	0.325	0.992	2.053	0.164
Average value	0.725	3.874	0.428	0.822	3.185	0.333	0.989	2.130	0.133

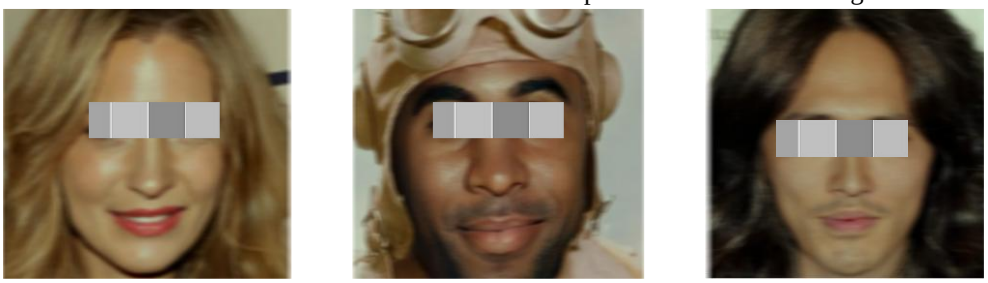
Table 4 shows the comprehensive test results of different techniques face replacement, which are evaluated in terms of SSMI, state error, and attribute error. The results of the 10 sets of face tests show that the research model has the best overall performance. For example, in the SSMI test, the average value of the research model is 0.989, while FaceSwap is 00.725 and AP-GAN is 0.882. In the comparison of state error, the average value of the research model is 2.130, which is better than FaceSwap and AP-GAN's 3.874 and 3.185. In conclusion, the research technique has the best overall performance in face switching. In addition, the study compares the anti jitter effects of research models, using frame continuity measurement (FCM) and motion consistency error (MCE) as core indicators. FCM evaluates fluency by calculating the variance of feature point displacement between adjacent frames. The higher the value, the more stable it is. MCE measures the consistency of facial keypoint trajectories. The lower the value, the more consistent the trajectories are. The test results are shown in Table 5.

Table 5: Vibration test

Test indicators	No shaking module	There is a shaking module	*p*-value	Effect size (Cohen's d)
FFCM	0.872±0.032	0.956±0.019	<0.001**	1.24
MCE	3.241±0.215	2.115±0.183	0.003**	0.91
PSNR(dB)	33.72±1.05	35.65±0.87	0.008**	0.78
SSIM	0.931±0.021	0.989±0.012	<0.001**	1.37

According to Table 5, the debounce module in FCM optimizes interframe transitions through optical flow analysis and motion compensation. This results in a significant improvement in FCM, raising the score from 0.872 to 0.956. The comparison between the data is statistically significant, $p<0.001$. Moreover, its standard deviation decreased from 0.032 to 0.019, indicating that the module effectively reduced inter frame fluctuations. The standard deviation of MCE decreases from 0.215 to 0.183 in the MCE test, a 34.7% decrease, indicating that the module effectively suppresses the random shaking component of facial motion. In addition, the image generation quality test revealed significant

improvements in both PSNR and SSIM values, with a statistically significant difference of $p<0.001$ between the before and after results. This module controls the inter frame coefficient of variation within an industrial threshold (CV<2%), providing key technical support for real-time face replacement at the film and television level. The comparison of the debounce effect before and after face replacement is shown in Figure 13.



(a) Not shaking yet

(b) After shaking

Figure 13: Comparison of face replacement and debounce effects

Figure 13 (a) shows the face replacement effect without debounce, while Figure 13 (b) shows the image result after face replacement and debounce. Clearly, the image without debounce has obvious issues with boundary ghosting, contour blurring, and hair texture blurring. After deblurring, the image has prominent edge contours, clear color blocks, and a significantly higher level of brightness and darkness. The research continues to conduct ablation experiments on the debounce module, as shown in Table 6.

Table 6: Experiment on the ablation of the shaking module

Evaluation indicators	No processing	RSC module only	RSC+Smooth Filter
MCE	4.52±0.31	3.15±0.24	2.13±0.18
FCM	0.81±0.04	0.91±0.02	0.96±0.01
PSNR(dB)	31.85±1.12	34.20±0.95	35.63±0.87
SSIM	0.902±0.025	0.962±0.018	0.988±0.012
Inter frame difference coefficient (%)	8.7±1.2	4.3±0.8	1.4±0.3

Table 6: shows the results of the ablation experiment on the debounce module. According to the test results, no task module processing is performed, and the MCE, FCM, PSNR, and SSIM values are all low, with a high frame rate difference coefficient of 8.7±1.2%. After adding the RSC module, the MCE value decreases significantly, from 4.52 to 3.15. Meanwhile, the FCM value increases, from 0.81 to 0.91. This change is mainly due to the enhanced stability of optical flow tracking. In addition, the PSNR and SSIM values for image generation quality increases significantly. In contrast, the MCE and FCM values for RSC+Smooth Filter are 2.13±0.18 and 0.96±0.01, respectively, indicating a significant decrease in image generation quality. This result also indicates that adding RSC module and Smooth Filter to the model can significantly improve the face replacement effect. Finally, the research introduces a comprehensive comparison of the computational efficiency and processing effectiveness of different models under resource constraints or abundance using literature [7] and UGAN Lite technology. Among them, the resource rich scenario is NVIDIA RTX3070 GPU (12GB video memory), batch size=16. The running memory is 64GB. The resource constrained scenario is NVIDIA RTX3070 GTX1060 (6GB of video memory), with a running memory of 16GB. The comparison of computing efficiency is shown in Table 7.

Table 7: Comparison of calculation efficiency and processing effect of different models

Model	Environment	Reasoning time (ms)	Memory footprint (MB)	FLOPs (G)	Frame rate (fps)
Ours	Rich resources	21.5±1.2	345±25	19.8	43.5
	Resource-constrained	128.3±8.5	245±30	15.8	15.4
AP-GAN	Rich resources	35.6±1.0	354±20	12.4	35.8
	Resource-constrained	152.4±5.1	246±25	7.4	11.5
FaceSwap-GAN	Rich resources	48.5±0.5	352±20	13.8	38.5
	Resource-constrained	254.5±2.3	254±34	9.4	10.8
Reference [7]	Rich resources	48.2±1.8	378±35	13.2	38.5
	Resource-constrained	275.5±9.2	264±28	11.5	12.4
UGAN-Lite	Rich resources	31.4±0.8	367±18	14.8	40.5
	Resource-constrained	156.4±1.5	258±84	11.8	14.4

As shown in Table 7, there are significant differences in the computational efficiency of the model between resource-rich and resource-limited scenarios, including the inference time, throughput (FLOPs), and processing frame rate. Under abundant resources, different models have better computational efficiency and higher processing frame rates. Among them, the research model performs the best, with the shortest inference time of 21.5 ± 1.2ms in the resource rich state, which is better than UGAN Lite's 31.4±0.8ms. Additionally, the research model shows the best performance in the PLOPs ratio under abundant and resource-limited conditions, with throughputs of 19.8G and 15.8G, respectively. In terms of processing frame rates, the research model performs significantly better than similar technologies under both resource-limited and resource-rich conditions, with processing frame rates of 15.4 fps and 43.5 fps, respectively. For example, the processing frame rates of the technology proposed in reference [7] under resource rich and resource limited conditions are 38.5 and 12.4, respectively. In summary, the research model has computational efficiency in different resource scenarios, and the task processing effect is better.

4 Discussion

In recent years, face replacement technology has been widely applied in fields such as film and video production, advertising design, etc. This is due to the rapid development of AI and DL technology. However, this technology still faces many challenges, such as insufficient accuracy and high resource consumption. A study proposed an automated face replacement technique based on improved MTCNN and GAN to address these issues.

In the face detection experiment, the improved MTCNN algorithm performed excellently. In ordinary scenarios, the accuracy of the research model reached 98.35%, significantly higher than the 92.31% of traditional MTCNN and 82.65% of Faceness. In complex scenarios, the accuracy of the research model was 93.25%, which was better than MTCNN's 86.25% and Faceness's 81.25%. This indicated that the improved MTCNN algorithm improved face detection accuracy and efficiency by introducing median filtering and depth separation convolution. In terms of training loss, the research model's convergence loss was 0.165 and 0.221 in ordinary and complex scenarios, respectively. These values were both lower than those of other techniques. In the face replacement experiment, the research model also performed well. In terms of preserving color and facial contour, the research model was more effective at restoring the original image's facial features, avoiding issues like skin color deviation and contour irregularities. In the SSMI test, the average value of the research model was 0.994, significantly higher than AP-GAN's 0.924 and FaceSwap's 0.875. In the PSNR test, the average value of the research model was 35.65, which was better than AP-GAN's 31.05 and FaceSwap's 28.84. This indicated that the face replacement model based on adversarial generative networks used a SAM and a digital debounce method. These features significantly improved the accuracy of face replacement and the smoothness of video processing.

Further analysis revealed that the application of self attention mechanism in face replacement was particularly crucial. It improved the model's ability to focus on important features by assigning weights to them, thereby making the replacement more natural. The digital debounce module effectively solved the jitter problem in video replacement through corner detection, optical flow analysis, and motion estimation, making video frames smoother. In terms of computational trade-offs, the research model performed well in both resource-rich and resource-limited scenarios. It had better inference time, throughput, and processing frame rate than similar technologies.

However, it should be noted that while the use of deepfake technology in film and television is innovative, its misuse poses social risks and raises ethical concerns that cannot be ignored. The abuse of Deepfake

technology may lead to the spread of false information, infringement of personal privacy and portrait rights. For example, unauthorized facial replacement may be used to create fake videos for illegal activities such as fraud and defamation. In addition, the rapid development of Deepfake technology has made forged videos increasingly realistic, making them difficult to distinguish as fake with the naked eye. This not only poses a threat to personal safety, but also impacts the social trust system. Therefore, when using relevant technologies, it is necessary to comply with local laws and regulations. Service providers of deep synthesis technology must fulfill their prompting and supervisory obligations to ensure the technology's safe use, as well as abide by ethical and moral principles.

In summary, the proposed facial replacement technology has demonstrated excellent accuracy, efficiency, and naturalness. This technology provides strong technical support for fields such as film and video production.

5 Conclusion

In recent years, with the continuous development of AI and DL, face replacement technology has become a focus of attention. Therefore, the research proposed an automated face replacement technology. The technology adopted an improved MTCNN algorithm for face detection, which improved the accuracy and efficiency of face detection by introducing median filtering and depth separation convolution. During the face replacement stage, a UGAN-based model was constructed that combined a SAM and a digital debounce method. This improved the accuracy of the face replacement and the smoothness of the video. In the face detection experiment, the improved MTCNN algorithm performed well. In ordinary scenarios, the accuracy of the research model reached 98.35%, significantly higher than the 92.31% of traditional MTCNN and 82.65% of Faceness. In complex scenes, the accuracy of the research model was 93.25%, which was better than MTCNN's 86.25% and Faceness's 81.25%. In the face replacement experiment, the research model also performed well. In terms of preserving color and facial contour structure, the research model could better restore the facial features of the original image, avoiding problems such as skin color deviation and contour anomalies. In the SSMI test, the average value of the research model was 0.994, significantly higher than AP-GAN's 0.924 and FaceSwap's 0.875. In the PSNR test, the average value of the research model was 35.65, which was better than AP-GAN's 31.05 and FaceSwap's 28.84. In summary, it can be seen that the technology proposed by the research has good application effects in video face switching. However, the research has shortcomings as well. For example, UGAN-based face replacement technology cannot process large amounts of offline video data. In addition, GANs also face the problem of hallucinations that occur in occlusions or rare poses during training. In the future, it will be necessary to develop an offline processing system that can adapt to different video face replacement requirements in various scenes, while improving technical recognition under occlusion and enhancing complex backgrounds.

References

[1] Zeng H, Zhang W, Fan C, Lv T, Wang S. FlowFace: semantic flow-guided shape-aware face-swapping[C]//Proceedings of the AAAI conference on artificial intelligence. 2023, 37(3): 3367-3375. https://doi.org/10.1609/aaai.v37i3.25444

[2] Rehaan M, Kaur N, Kingra S. Face manipulated deepfake generation and recognition approaches: A survey. Smart Science, 2024, 12(1): 53-73. https://doi.org/10.1080/23080477.2023.2268380

[3] Kalembo Vikalwe Shakrani, Ngonidzashe Mathew Kanyangarara, Prince Tinashe Parowa, Vibhor Gupta, Rajendra Kumar. A Deep Learning Model for Face Recognition in Presence of Mask. Acta Informatica Malaysia. 2022; 6(2): 43-46. https://doi.org/10.26480/aim.02.2022.43.46

[4] Rao I S S, Kumar J S, Vamsi T M N, Kumar TR. IMPROVING REALISM IN face-swapping USING DEEP LEARNING AND K-MEANS CLUSTERING. Proceedings on Engineering, 2024, 6(4): 1751-1756. https://doi.org/10.24874/PES.SI.24.03.010

[5] Omar K, Sakr R H, Alrahmawy M F. An ensemble of CNNs with self-attention mechanism for DeepFake video detection. Neural Computing and Applications, 2024, 36(6): 2749-2765. https://doi.org/10.1007/s00521-023-09196-3

[6] AbdElminaam D S, Sherif N, Ayman Z, et al. DeepFakeDG: A deep learning approach for deep fake detection and generation. Journal of Computing and Communication, 2023, 2(2): 31-37. https://doi.org/10.21608/jocc.2023.307056

[7] Tsai C S, Wu H C, Chen W T, Ying JJC. ATFS: A deep learning framework for angle transformation and face-swapping of face de-identification. Multimedia Tools and Applications, 2024, 83(12): 36797-36822. https://doi.org/10.1007/s11042-023-16123-0

[8] Melnik A, Miasayedzenkau M, Makaravets D. Face generation and editing with stylegan: A survey. IEEE Transactions on pattern analysis and machine intelligence, 2024, 46(5): 3557-3576. https://doi.org/10.1109/TPAMI.2024.3350004

[9] Liao X, Wang Y, Wang T, Hu J, Wu X. FAMM: Facial muscle motions for detecting compressed deepfake videos over social networks. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 33(12): 7236-7251. https://doi.org/10.1109/TCSVT.2023.3278310

[10] Akhtar Z, Pendyala T L, Athmakuri V S. Video and audio deepfake datasets and open issues in deepfake technology: being ahead of the curve. Forensic Sciences, 2024, 4(3): 289-377. https://doi.org/10.3390/forensicsci4030021

[11] Zhao H, Zhou W, Chen D, Zhang W, Guo Y. Audio-Visual Contrastive Pre-train for Face

Forgery Detection. ACM Transactions on Multimedia Computing, Communications and Applications, 2024, 21(2): 1-16. https://doi.org/10.1145/3651311

[12] Yue P, Chen B, Fu Z. Local region frequency guided dynamic inconsistency network for deepfake video detection. Big Data Mining and Analytics, 2024, 7(3): 889-904. https://doi.org/10.26599/BDMA.2024.9020030

[13] Wöhler L, Ikehata S, Aizawa K. Investigating the Perception of Facial Anonymization Techniques in 360° Videos. ACM Transactions on Applied Perception, 2024, 21(4): 1-17. https://doi.org/10.1145/3695254

[14] Lai Y. Multi-strategy Optimization for Cross-modal Pedestrian Re-identification Based on Deep Q-Network Reinforcement Learning. Informatica, 2025, 49(11). https://doi.org/10.31449/inf.v49i11.7247

[15] Yang G, Xu K, Fang X, Zhang J. Video face forgery detection via facial motion-assisted capturing dense optical flow truncation. The Visual Computer, 2023, 39(11): 5589-5608. https://doi.org/10.1007/s00371-022-02683-z

[16] Pang G, Zhang B, Teng Z, Qi Z. MRE-Net: Multi-rate excitation network for deepfake video detection. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 33(8): 3663-3676. https://doi.org/10.1109/TCSVT.2023.3239607

[17] Deng L, Wang J, Liu Z. Cascaded network based on EfficientNet and transformer for deepfake video detection. Neural Processing Letters, 2023, 55(6): 7057-7076. https://doi.org/10.1007/s11063-023-11249-6

[18] Liu T. Secure Face Recognition Using Fully Homomorphic Encryption and Convolutional Neural Networks. Informatica, 2024, 48(18). https://doi.org/10.31449/inf.v48i18.6396

[19] Lu T, Bao Y, Li L. Deepfake Video Detection Based on Improved CapsNet and Temporal-Spatial Features. Computers, Materials and Continua, 2023, 75(1): 715-740. https://doi.org/10.32604/cmc.2023.034963

[20] Pang G, Zhang B, Teng Z. MRE-Net: Multi-rate excitation network for deepfake video detection. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 33(8): 3663-3676. https://doi.org/10.1109/TCSVT.2023.3239607

Optimizing Complex Functions: A Numerical and ML Comparison

Xuechao Zhang
Mathematics Teaching and Research, Office， Zhengzhou, Professional Technical Institute of Electronic & Information, Zhengzhou, Henan, 451450, China
E-mail：zxfchao1982@126.com

Keywords: complex function optimization, convergence rate, benchmark functions, computational efficiency, machine learning algorithms, numerical methods

Received: February 20, 2025

Optimization plays a vital role across disciplines such as engineering, economics, and artificial intelligence. Complex functions, which map complex numbers to complex outputs, often cannot be solved analytically, necessitating numerical or machine learning-based approaches. This study presents a comparative analysis of numerical optimization methods—specifically Gradient Descent and Newton's Method—against machine learning-based techniques, including Genetic Algorithms, Particle Swarm Optimization, and Deep Q-Learning. These methods are evaluated using standard benchmark functions: Ackley, Rastrigin, and Rosenbrock. The comparison focuses on convergence rate and runtime performance. Results show that numerical methods offer faster runtimes but lower convergence rates, while machine learning approaches achieve higher convergence at the cost of increased computational time. This analysis underscores the trade-offs between efficiency and robustness in optimization techniques, offering practical insights for selecting appropriate methods based on specific application needs, especially in scenarios involving complex, non-linear, or high-dimensional functions.

Povzetek: Prispevek išče kompromis med hitrostjo in robustnostjo kompleksnih funkcij. Uporablja GD/Newton vs. GA/PSO/DQN na Ackley/Rastrigin/Rosenbrock, brez standardnih postopkov.

1 Introduction

Complex function optimization is a significant area of research within optimization problems. The methodologies for solving optimization issues can be categorized into analytical methods and numerical computations [1]. The analytical method addresses the problem by examining the correlation between the derivative of the objective function and its extreme values. This strategy is applicable solely to optimization problems characterized by relatively uncomplicated objective functions. According to the variation principle of the objective function value, one should proceed in suitable increments along the direction that maximizes the objective function value. A method for approximate calculation that incrementally approaches the optimal point of the objective function. This technique is effective for resolving continuous differentiable convex optimization problems. However, with the ongoing proliferation of engineering optimization challenges, the majority of objective functions are non-convex optimization problems. The advent of group intelligent optimization algorithms offers a constrained approach to difficult function optimization challenges [2].

Numerical Optimization is a recognized domain within Mathematical Sciences that seeks to determine the extreme values of a function, namely its maxima and minima. In the past two decades, optimization approaches have emerged as essential instruments for management, decision-making, technological enhancement, and development, conferring competitive benefits to diverse systems. Consequently, optimization models and algorithms have become prominent in various domains, including industry [3], disease diagnosis [4], scheduling and resource allocation [5], and finance [6].

A prominent domain dedicated to addressing issues through mathematical models and algorithms is Machine Learning. In certain real-world scenarios, a substantial amount of data must be handled. This volume of data typically necessitates computational support to convert the information into pertinent knowledge for problem-solving. In this situation, machine learning techniques are highly beneficial. These models and algorithms aim to provide a mathematical representation that characterizes the data collection and extrapolates insights to unfamiliar data samples [7]. Machine learning models and algorithms are applicable in various fields, including industry, healthcare, finance, and education.

Owing to the practical significance of both domains, numerous methods have been devised to address optimization and machine learning challenges. While the majority of algorithms are effective in addressing their respective challenges, none are entirely flawless [8]. A pure algorithm denotes a singular approach or method utilized to resolve an issue comprehensively from inception to completion. A hybrid algorithm amalgamates many algorithms or techniques from disparate disciplines to address a problem or enhance the efficacy of a singular solution. Hybrid methods can combine many methodologies to capitalize on the strengths of each algorithm while alleviating their weaknesses. The hybrid algorithm represents a synthesis of concepts and techniques aimed at investigating the capabilities of several approaches while mitigating their deficiencies. By integrating complementary algorithms, one can leverage their strengths and mitigate limits, resulting in enhanced overall performance.

A hybrid algorithm that integrates optimization and machine learning approaches is an effective approach that leverages the strengths of both disciplines to provide a robust framework for addressing complicated issues [9]. This integration enhances decision-making capacities by incorporating optimization approaches into the machine learning process and vice versa. A hybrid algorithm can utilize optimization techniques to direct the learning process, thereby improving the accuracy and effectiveness of decision-making. This integration allows the algorithm to utilize explicit mathematical optimization methods and data-driven learning capabilities, resulting in enhanced and more efficient decision-making [10].

This paper delineates and examines the primary characteristics of numerical optimization and machine learning techniques. By recognizing and examining the primary aspects of each methodology in conjunction, they can mitigate difficulties and improve one or both processes through hybrid approaches. The objective is to determine the optimal combinations that yield hybrid approaches, taking into account a machine learning algorithm influenced by optimization strategies or vice versa.

This paper compares numerical optimization methods, specifically gradient descent and Newton's method, with machine learning optimization algorithms, including genetic algorithms, particle swarm optimization, and deep Q-learning. The benchmark optimization test functions Ackley, Rastrigin, and Rosenbrock are employed to assess the optimization methods based on convergence rate and runtime. The machine learning algorithms demonstrate the capability to effectively address complex function optimization problems and enhance the convergence rate relative to numerical methods.

The remaining part of this research is organized as, section 2 provides the background details including complex function, numerical optimization, machine learning and related works summary. Section 3 explains the proposed methodology. Section 4 provides the experimental results and section 5 conclude the research paper.

2 Background

2.1 Complex function

Let S denote a collection of complex numbers. A function f defined on S is a mapping that allocates to each z in the complex numbers a corresponding w. The quantity w is referred to as the value of f at z and is represented as f(z); so, w = f(z). The set S is referred to as the domain of definition of f.

If each value of z corresponds to only one value of w, we denote w as a single-valued function of z or f(z) that is single-valued. If many values of w correspond to each value of z, w is classified as a multiple-valued or many-valued function of z.

A multiple-valued function can be regarded as a compilation of single-valued functions, with each constituent referred to as a branch of the function. Typically, we designate one specific member as the principal branch of the multi-valued function, with the matching function value referred to as the principal value.

Polynomial function:

For $x_n, x_{n-1}, \ldots, x_0$ complex constants,

$$p(y) = x_n y^n + x_{n-1} y^{n-1} + \cdots + x_1 y + x_0 \quad (1)$$

Where $x_n \neq 0$ and n is a integer value.

Exponential function

If $z = a + ib$, the exponential function e^z is represented by,

$$e^z = e^a e^{ib} \quad (2)$$

Where $e^{ib} = \cos b + i \sin b$

$$e^z = e^z(\cos b + i \sin y) \quad (3)$$

Logarithmic Function

The complex logarithm is an extension of the standard real natural logarithm (base e) into the complex domain. In polar coordinates, the complex logarithm is expressed as,

$$\begin{aligned} \log z &= \log\left(re^{i\theta}\right) \\ &= \log r + \log e^{i\theta} \\ &= \log r + i\theta \end{aligned} \quad (4)$$

Trigonometric functions:

The sine and cosine of a complex variable are described as,

$$\sin z = \frac{e^{iz} - e^{-iz}}{2i} \quad (5)$$

$$\cos z = \frac{e^{iz} + e^{-iz}}{2} \quad (6)$$

In optimization, a complex function refers to one in which the input and/or output are complex numbers, with the objective of identifying maximum, minimum, or other key points, frequently subject to constraints or special conditions. These functions may be nonlinear, multivariable, and incorporate both real and imaginary components in their inputs and outputs, rendering them more complex to evaluate and optimize than real-valued functions.Optimization problems may involve complex functions when the objective function originates from physical systems represented by complex-valued variables, or in signal processing, control systems, and specific machine learning models where data or parameters are inherently articulated in the complex domain.

2.2 Numerical optimization

Although there are various strategies and algorithms available for addressing optimization difficulties, none are universally applicable or flawless in properly resolving all such concerns. Each algorithm possesses distinct properties, making it more suitable for addressing specific challenges based on these attributes. The formulation of the problem, selection of methods, and choice of algorithms are pivotal aspects in addressing an optimization issue, as certain methods and algorithms are more suitable than others based on the specific challenge at hand. The initial stage in an optimization problem is the formulation of the mathematical model. A mathematical model seeks to represent a real-world situation as a mathematical function applicable in optimization techniques. An optimization problem can be articulated in mathematical terms through a collection of variables and numerical relationships that encapsulate an abstraction of the issue, aiming to identify the optimal solution within Rn from a set of potential alternatives. To construct a mathematical model, four essential procedures must be performed [11].

- Specify the decision parameters. ($x_1,x_2,x_3,\ldots,x_n$)
- Construct the objective function f(x) or the set of objective functions $f_1(x)$, $f_2(x),f_3(x),\ldots,f_k(x)$ that rely on the choice variables and yield a real value.
- If required, provide a collection of equality and/or inequality constraints, $g_i(x) = 0$ and $h_j(x) \leq 0$ for i = 1, 2, ..., n_g and j = 1,2, ...n_h, that must be satisfied by the decision variables.
- Define the domain sets $D_1,D_2,D_3,\ldots,D_n$ corresponding to the decision parameters, $x_1,x_2,x_3,\ldots,x_n$ accordingly.

The optimization issue is defined by the objective function and the restrictions on the variables. The objective function delineates the goal of the problem, which may encompass any quantity or amalgamation of quantities represented by a singular numeral, such as personnel, time, materials, energy, etc.; conversely, the constraints directly influence the decision space and outcomes, imposing restrictions on the algorithmic options. The primary purpose of an optimization issue is to reduce or maximize the objective function while adhering to the restrictions. Furthermore, optimization models can be categorized into two principal classes based on the number of objectives: Single-objective and multi-objective optimization issues [12]. Single optimization problems typically consist of a singular target, with or without restrictions, while multi-objective optimization pertains to multiple-criteria decision-making, comprising the simultaneous optimization of numerous objective functions, with or without constraints [13]. Following the creation of the mathematical model, it is essential to ascertain the most suitable strategy for identifying the optimal solution.

A solution to an optimization issue can be characterized by local and global optimization, with corresponding techniques referred to as local search and global search. Local and global search optimization methods are employed in various contexts or to address distinct optimization inquiries [14]. Local optimization aims to identify a solution that reduces (or increases) the objective function within a defined region of the search space, hence locating a local solution among feasible points in the vicinity. This type of search does not ensure an objective value that is lower (or higher) than all other viable locations. Conversely, global optimization seeks to identify the point that reduces (or increases) the objective function [15].

2.3 Machine learning

The essence of human intelligence is on experiential learning and the transmission of personal knowledge between generations. Machine learning pertains to the development of computer programs that enhance their performance through experience. The experience is derived by a data analysis procedure executed by a customized algorithm. Consequently, the machine learning approach employs algorithms to identify patterns within a dataset through computational, analytical, optimization, and information discovery techniques [16].

There are three types of machine learning: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Within each category, methods can be differentiated based on their knowledge acquisition techniques, including classification, regression, clustering, association learning, relational analysis, differential equations, and others. Machine learning seeks to construct a hypothesis (model) capable of extracting information from training data and generalizing the learned knowledge to unfamiliar samples. This model must exhibit simplicity regarding complexity and demonstrate efficacy in minimizing empirical mistakes within the data.

Supervised learning techniques seek to identify the correlation between input attributes (independent

variables) and a target attribute (dependent variable). Supervised learning is employed for two categories of tasks: pattern categorization and regression. Occasionally, there is an absence of knowledge regarding the correlation between input and output attributes in machine learning challenges [17]. Consequently, the algorithms must identify similarities or differences within the data collection. The method necessitates greater human comprehension than supervised procedures, as a decision-maker, whether an individual or a collective, is accountable for the final decision-making.

While supervised and unsupervised approaches engage with data and necessitate exploration and comprehension of the data in relation to the application domain, significant distinctions exist between the two methodologies. The primary distinction is the lack of an output vector for the target variable, as observed in supervised approaches. Moreover, unsupervised learning is frequently linked to creative pursuits—exploration, comprehension, and enhancement—that do not conform to predefined protocols, unlike supervised methods [18]. Unsupervised learning is often categorized into clustering methods and dimensionality reduction methods.

Reinforcement learning is a machine learning technique predicated on a reward mechanism, predominantly utilized in dynamic control systems; however it is also applicable to optimization issues [19]. Reinforcement learning addresses the challenge of instructing an autonomous agent, which interacts with and perceives its environment, to select optimal actions for attaining its objectives. The agent acquires information regarding the present environmental conditions and must use its existing knowledge through a greedy approach to optimize rewards, while simultaneously engaging in exploration to identify superior actions for the future.

2.4 Related work summary

This section highlights the existing numerical and machine learning-based optimization strategies for complex function optimization problems. Table 1 provides the summary of existing optimization approach with limitation or research gap.

Table 1: Related work summary

Ref	Methods	Data set and Metrics	Limitations / Research Gap
[9]	Hybrid Bee algorithm for solving continues complex function	Benchmark function; Accuracy and mean value.	It does not handle numerical methods. Testing the enhanced algorithms on practical optimization challenges
[20]	Deep learning proxy model	Rosenbrock, Rastrigin and Ackley functions; Mean Euclidean distance and the standard deviation	Sparse sampling is ineffective, and the optimizers exhibit heightened sensitivity to the landscape derived from this kind of data.
[21]	First order optimization methods: Standard gradient descent, momentum, heavy ball, nesterov	Benchmark functions; Iteration, time, function value and weight	It only compares first order optimization methods and does not consider other optimization approaches
[22]	Improved Hypercube optimization algorithm	Benchmark functions; Mean, Standard deviation, Convergence rate and Time.	It offers optimal value and accelerates convergence processes; however, it does not address practical optimization concerns.
[23]	Genetic algorithm for high dimensional optimization problem.	Benchmark functions; Gain and convergence rate.	A robust criterion is required to dynamically determine the number of active subspace dimensions.
[24]	Local and global numerical optimization; Focusing robust optimization with uncertainty-based sampling approach	Benchmark functions; CPU Time, mean, median and success rate	It does not converge effectively to the global minimum for various contrived test functions, rendering its broad application inadvisable.

3 Methodology

Optimizing complex functions involves identifying the extrema of functions that utilize complex numbers as both inputs and outputs. These functions may encompass both real and imaginary components, and addressing optimization problems involving such functions is a prevalent endeavor in disciplines such as signal processing, control theory, physics, and machine learning. A complex function f(z) can be written as,

$$f(z) = f(x + iy) = u(x, y) + iv(x, y) \quad (7)$$

where, $u(x,y)$ denotes the real component and $v(x,y)$ signifies the imaginary component, with x and y representing real variables corresponding to the real and imaginary parts of the complex input z. The objective of optimization frequently involves minimizing (or maximizing) a complex-valued function, specifically identifying the point z in the complex plane that yields the optimal value of (z).

This paper compares numerical optimization methods, specifically gradient descent and Newton's method, with machine learning optimization algorithms, including genetic algorithms, particle swarm optimization, and deep Q-learning. Figure 1 shows the work flow of the proposed approach.

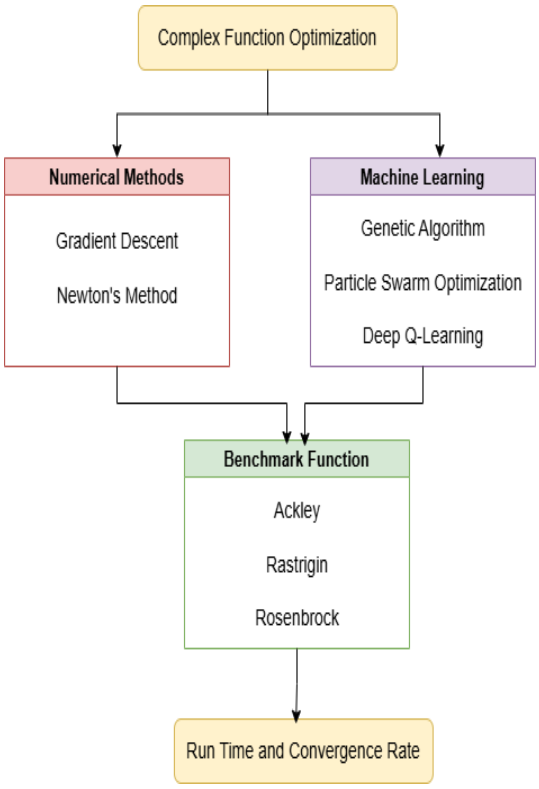


Figure 1: Proposed work flow

In all real-valued single-objective unconstrained optimization algorithms, the objective is to identify the minimum (or, equivalently, the maximum) of a scalar objective function (x), represented by the vector of free parameters $X=(x_1, x_2, x_3,...,x_m)$, where m denotes the problem's dimensionality. Consequently, f is a mapping from R^m to R. Assume the following hypothesis:

- Function f is accessible solely as a black box; thus, we possess no understanding or ability to manipulate its internal mechanisms. We access f solely through input-output mechanisms.
- The function f possesses a continuous domain inside the specified constraints; thus, every point within these bounds is mapped by f.
- f is well-defined in the domain, at least numerically; it is continuous and exhibits a degree of smoothness. This restricts very noisy functions, where spatial correlation is absent. However, there is also an implicit assumption of some degree of noise, wherein finite differences in the vicinity of a point do not resemble the derivatives of the noiseless function.

Numerical optimization techniques are employed to identify the most effective for intricate functions when statistical techniques are impractical or excessively challenging. The primary classifications of numerical optimization methods are gradient-based and gradient-free

techniques. In this paper two numerical optimization methods are considered: Gradient descent and Newton's method. The previous works only consider the first-order optimization but this research considers both first-order and second-order optimization. Due to the simplicity, the first order optimization i.e gradient descent is selected which reduces the training time.

Gradient Descent:

Standard gradient descent is among the most fundamental and prevalent optimization techniques. The fundamental concept involves incrementally adjusting the model parameters in accordance with the gradient of the loss function. The gradient signifies the direction in which the value of the loss function diminishes most rapidly. The objective is to minimize the function by modifying the model parameters accordingly.

The primary phases of Standard Gradient Descent:

Computation of gradients: At each stage of the procedure, the gradient of the loss function is computed for all model parameters. The gradient is a vector comprising the partial derivatives of the loss function with respect to each parameter, signifying the direction of the steepest ascent in the function.

Update of parameters: Subsequent to computing the gradient, the model parameters are revised in accordance with the formula:

$$\delta_{new} = \delta_{old} - \alpha \times \nabla\tau(\delta) \quad (8)$$

Where δ represents the model parameter, α indicates the learning rate and $\nabla\tau(\delta)$ is the gradient loss function.

***Rate of learning*:** This is a crucial parameter of the procedure. If the step size is very tiny, the optimization process will be sluggish; conversely, if it is overly high, it may bypass the minimum and induce instability in the process. Consequently, the appropriate selection of alpha is essential.

Repetitions: The parameter update process is iterated multiple times until a termination point is achieved, either by convergence criteria or after a predetermined number of iterations.

Gradient Descent is straightforward to implement, as it requires only the calculation of the gradient and the subsequent update of the model parameters for each iteration. Should the loss function exhibit smoothness and convexity, the approach can effectively identify the global minimum.

Newton's Method

Newton's approach is a second-order optimization approach that can be used to locate the roots of complex functions or optimize them. This method is based on the second derivative and often converges quicker than gradient descent for smooth functions.

Given a complex function $f(z) = z^2 - 2 + i$

1. Select an initial guess, z_0= 1+i
2. Compute the next approximation using the formula:

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \quad (9)$$

Where $f(z) = z^2 - 2 + i$ and f'(z)=2z

$f(1+i)=(1+i)^2 - 2+i=1+2i-1+i=3i$

f'(1+i)=2(1+i)=2+2i

therefore, $z_1 = (1+i) - \frac{3i}{2+2i}$

3. Repeat this process iteratively until convergence.

Genetic Algorithm

Genetic Algorithm (GA) is an optimization technique based on population dynamics. These heuristic search methods, inspired by natural evolution, can optimize complex functions, particularly when the functions are noisy or non-differentiable. Genetic algorithms employ a population of potential solutions, which are iteratively evolved throughout multiple generations using selection, crossover, and mutation to identify the optimal answer. Algorithm-1 shows the general steps of genetic algorithm.

Algorithm-1: Genetic Algorithm

Input: Population Size (pop), Maximum iteration (MaxIter)
Output: Global best solution
Step1: Randomly generate initial population of n chromosomes
Step2: Initialize iteration count iter = 0
Step3: Compute the fitness value of each chromosomes
Step4: while (iter <MaxIter)
Step5: Select a pair of chromosomes from initial population based on fitness
Step6: Apply crossover operation on selected pair with crossover probability
Step7: Apply mutation on the offspring with mutation probability
Step8: Replace old population with newly generated population
Step9: Increment the iteration iter by 1
Step10: EndWhile
Step11: Return the best solution

The genetic algorithm (GA) employed a population size of between 20 or 200, contingent upon the number of variables. The crossover rate was 0.8, and the maximum number of iterations was 1000. The algorithm terminated when the average relative change in the optimal fitness function value over 100 iterations was less than or equal to 1e-6.

Particle swarm Optimization

The PSO algorithm is a gradient-free technique founded on the notion of swarm dynamics. In Particle Swarm Optimization (PSO), each sample is designated as a particle, and a collection of particles is termed a swarm. In the initial iteration of PSO, each particle commences at rest with a velocity vector of zero. The objective function's value is calculated for the initial iteration, and the particles' velocity and position are revised. The velocity of each particle is determined by three components: inertia, cognitive, and social components, along with certain random coefficients. The inertia component signifies a particle's velocity at the present iteration, inhibiting significant positional alterations. The cognitive component assesses a particle's performance in relation to its prior best, and the social component evaluates a particle's performance in comparison to the swarm's optimal location. The optimal positions are ascertained from the fitness function assessment in the preceding iteration, and the updates for velocity and position persist until the termination criteria are met. Algorithm-2 shows the general steps of PSO algorithm.

Algorithm-2: Particle swarm optimization
Input: Particle Size , Maximum iteration (MaxIter)
Output:best solution
Step1: Initialize particles, velocity, iter=0, pbest, gbest
Step2: Randomly generate initial particles
Step3: For each particle
Step4: Compute fitness function
Step5: Update pbest, gbest
Step6: End For
Step7: While (iter <MaxIter)
Step8: For each particle
Step9: Update velocity and particle
Step10: Compute fitness value
Step11: Update pbest, gbest
Step12: EndFor
Step13: EndWhile

The PSO employed a swarm size of between 20 or 200, contingent upon the number of variables. The weight was 0.1, and the maximum number of iterations was 1000.

Machine learning is applicable for optimizing complex functions, especially in the context of high-dimensional, non-convex, or noisy functions that are challenging for conventional approaches to manage.A neural network can be taught to approximate a complicated function by utilizing the real and imaginary components of the function as input and output. This can be particularly useful if the optimization process entails investigating complex spaces with numerous variables. To optimize the network, it can be trained using a loss function, such as Mean Squared Error (MSE), to reduce the disparity between the expected output of the complex function and the actual function. The network acquires the ability to forecast ideal input parameter values (complex values) that reduce the output of the complex function. The backpropagation algorithm updates the network's weights according to the prediction error.

Reinforcement learning (RL) can optimize intricate functions, particularly in scenarios where the function is costly to assess or exhibits noise. In reinforcement learning, the agent engages with the environment (a complicated function) by executing actions (selecting intricate input values) and obtaining rewards (ideal outputs).Methods such as Q-learning or Deep Q Networks (DQN) can be utilized to progressively enhance the answer through environmental feedback. Algorithm-3 shows the general steps of DQN.

Algorithm-3: DQN
Step1: Initialize parameters
Step2: For each episode
Step3: Observe initial state
Step4: For each step of episode
Step5: Select action
Step6: Execute action
Step7: Observe reward and new state
Step8: Store transition
Step9: Sample mini-batch of transitions
Step10: Compute target for each transition
Step11: Perform descent step with respect to the quantum circuit parameters
Step12: Every steps reset Q’ = Q
Step13: End For
Step14: EndFor

A hybrid approach integrates the advantages of numerical methods and machine learning approaches to enhance the efficiency of complex function optimization. For example, one might employ a genetic algorithm to investigate the search space of intricate functions, subsequently utilizing a neural network to refine the answers near the optimum. The hybrid approach is not scope of the current work. This may be considered as a future work.

4 Experimental results and analysis

4.1 Benchmark dataset

This research uses benchmark dataset (Optimization Test Functions [24]) for evaluate the performance of the optimization problem. This work uses three optimization test functions Ackley, Rastrigin, and Rosenbrock.

Ackley function is continuous, scalable, non-separable, and a substantially multimodal test function. This test function is defined as follows:

$$f(x) = -20\, exp\left(-0.2 \times \sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(2\pi x_i)\right) + 20 + e \quad (10)$$

where d represents the number of dimensions and $x_i = (x_1, x_2, ..., x_d)$ is a d-dimensional row vector. The test region is typically assessed within the range of [-32.768, 32.768]^d. The global minimum $(x) = 0$ is achievable at $xi = (0,0)$. Figure 2 shows the surface plot of 2D representation of Ackley function.

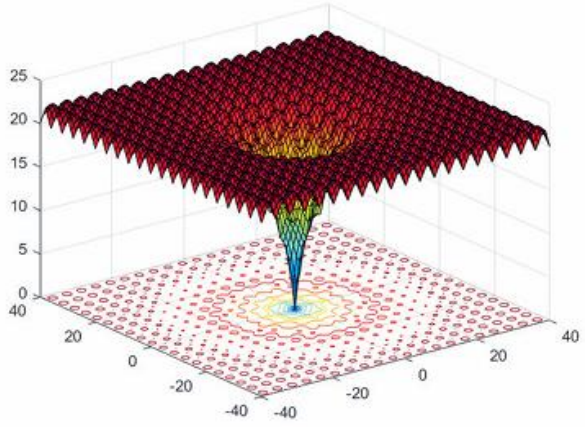


Figure 2: Ackley function - surface plot

Rastrigin function is continuous, scalable, separable, and a highly multimodal global optimization function. This test function is defined as follows:

$$f(x) = 10d + \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i)) \quad (11)$$

where d represents the number of dimensions and $x_i = (x_1, x_2, ..., x_d)$ is a d-dimensional row vector. The test region is typically assessed within the range of [−5.12, 5.12]^d. The global minimum $(x) = 0$ is achievable at $x_i = (0,0)$. This test function poses significant challenges for numerous global optimization techniques. Figure 3 shows the surface plot of Rastrigin function.

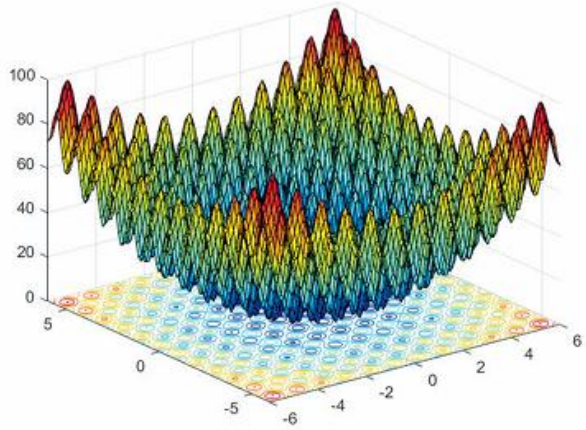


Figure 3: Rastrigin function -surface plot

Rosenbrock is a unimodal, valley-shaped function defined for dimensions d ≥ 2. This test function is continuous, scalable, inherently nonseparable, nonconvex, and unimodal. This test function is defined as follows:

$$f(x) = \sum_{i=1}^{d-1}[100(x_{i+1} - (x_i)^2)^2 + (x_i - 1)^2] \quad (12)$$

where d > 2 represents the number of dimensions and $x_i = (x_1, x_2, ..., x_d)$ denotes a d-dimensional row vector. The test region is typically assessed within the range of [−2.048, 2.048]^d, where $i = (1,...,d)$. The global minimum $(x) = 0$ is achievable at $xi = (1,1)$. The function is mostly recognized for its exceedingly slow convergence at the minimum point. Figure 4 shows the surface plot of Rosenbrock function.

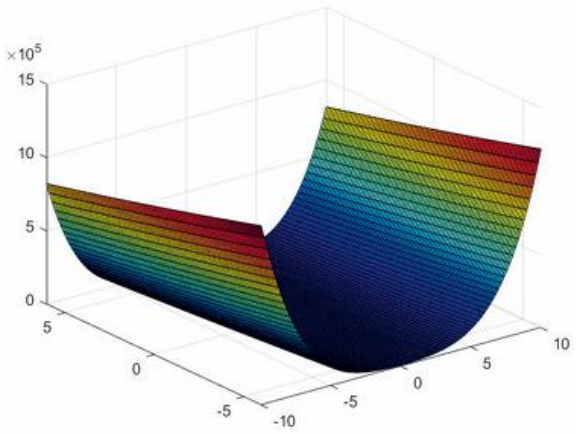


Figure 4: Rosenbrock function - surface plot

4.2 Results

This section compares the performance of the optimization approaches described. In each instance, function reduction was conducted commencing from 50 randomly chosen initial points within the search space of the test function. The comparisons were conducted throughout time, assessing the success rate of the optimizations. An optimization initiated from a specified set of beginning conditions is deemed successfully convergent to the global optimum if $|f(x) - f(x^*)| \leq 10^{-4}$, where $f(x*)$ and $f(x)$ represent the known accurate global minimum and the lowest obtained value of the objective function, respectively. Conversely, the optimization is deemed unsuccessful.

Table 2 presents the findings derived from the Ackley function. The GA and DQN approaches effectively converged to the global optimum, but the gradient descent and Newton methods failed to converge in all instances. The PSO approach achieved convergence in 98% of the total runs. The time for numerical methods are low compared to machine learning methods.

Table 2: Result of Ackley function for different methods

Method	Time (Sec)	Convergence Rate (%)
Gradient Descent	0.037	0
Newton's	0.0085	0
GA	0.17	100
PSO	0.051	98
DQN	0.55	100

Table 3, presents the findings derived from the Rastrigin function. The GA and DQN approaches effectively converged to the global optimum, but the gradient descent and Newton methods failed to converge in all instances. The PSO approach achieved convergence in 70% of the total runs. The time for numerical methods are low compared to machine learning methods.

Table 3: Result of Rastrigin Function for Different Methods

Method	Time (Sec)	Convergence Rate (%)
Gradient Descent	0.065	0
Newton's	0.010	0
GA	0.15	100
PSO	0.075	70
DQN	0.98	100

Table 4 presents the findings derived from the Rosenbrock function. In this case, the GA and DQN approaches has high convergence rate (98% and 90%), but the other approaches has lowered convergence rate. The time for numerical methods are low compared to machine learning methods.

Table 4: Result of Rosenbrock function for different methods

Method	Time (Sec)	Convergence Rate (%)
Gradient Descent	0.081	50
Newton's	0.0230	76
GA	2.5	98
PSO	1.9	65
DQN	3.2	90

4.3 Discussion

When comparing Numerical Optimization and Machine Learning Optimization methods for complicated functions, it is essential to assess their performance based on many factors, including accuracy, efficiency, scalability, robustness, and application. Each optimization technique presents unique advantages and disadvantages contingent upon the characteristics of the problem (e.g., differentiability, dimensionality, noise, etc.). Table 5 shows the comparison of numerical and machine learning optimization.

Table 5: Comparison of numerical Vs machine learning optimization

Criterion	Numerical Optimization	Machine Learning Optimization
Objective	Optimize a designated objective function (e.g., cost function).	Optimize loss or error functions utilizing models (e.g., neural networks, regression models).

Type of Functions Handled	Functions exhibiting explicit, deterministic characteristics (frequently differentiable)	Intricate, non-differentiable, stochastic, or opaque functions
Optimization Algorithms	Gradient-dependent methods (e.g., steepest descent, Newton's method) or gradient-independent methods (e.g., genetic algorithms, simulated annealing).	Stochastic algorithms (e.g., gradient descent, Adam, genetic algorithms), evolutionary strategies.
Computational Complexity	The cost is contingent upon the employed method; it may be computationally intensive for high-dimensional or non-linear functions.	May incur significant computing costs, particularly in high-dimensional environments (e.g., deep learning architectures).
Convergence Speed	Typically more efficient for smooth, convex, or differentiable functions.	Convergence may require an extended duration, particularly when addressing high-dimensional and noisy objective functions.
Scalability	May encounter difficulties in high-dimensional spaces, particularly with gradient-based techniques.	Capable of scaling to high-dimensional areas, particularly in deep learning or evolutionary algorithms.
Use of Derivatives	Heavily depends on gradients or Hessians (second-order derivatives) for convergence.	Frequently operates with or without derivatives (e.g., gradient-free techniques or models like as neural networks).
Ability to Handle Noise	Susceptible to noise and necessitates smooth functions or noise mitigation techniques such as regularization.	Resilient to noisy data and frequently capable of identifying optimal solutions in circumstances characterized by noise or insufficient data.
Global vs. Local Optima	Susceptible to entrapment in local minima for non-convex functions.	Local minima can be circumvented using methods such as stochastic gradient descent or evolutionary algorithms.
Flexibility	Most appropriate for issues with a well delineated objective function and restrictions.	Extremely adaptable and applicable to various issues, including regression, classification, and black-box optimization.
Learning and Adaptability	Generally does not get knowledge from the optimization process.	Models can "adapt" over time (e.g., neural networks modifying weights).
Examples of Use	Engineering design, optimal resource distribution, and resolution of physical models.	Machine learning activities (e.g., neural network training, hyperparameter tuning, reinforcement learning).
Robustness	Sensitive to starting conditions (e.g., first guess for gradient algorithms).	Resilient to initialization and noise in extensive datasets, particularly with ensemble techniques.
Interpretability	Generally more straightforward to interpret solutions, particularly for less complex models (e.g., linear programming).	Frequently less interpretable, especially in deep learning, where models may function as "black boxes".

Numerical optimization is particularly efficient for well-defined, differentiable, and smooth objective functions. This approach is typically more efficient for such issues but may encounter difficulties with non-linearity, large dimensionality, and noisy data. Machine Learning Optimization is particularly effective for high-dimensional, non-differentiable, or noisy functions. Machine learning techniques can exhibit greater adaptability and scalability; yet, they may require extended convergence periods and might occasionally lack interpretability compared to numerical methods.

5 Conclusion

Optimizing complex functions presents a formidable challenge that necessitates meticulous evaluation of both numerical techniques and machine learning methodologies. Conventional optimization techniques such as gradient descent, Newton's method, and evolutionary algorithms are well-recognized. Machine learning optimization techniques provide an array of instruments for addressing intricate function optimization, particularly when conventional numerical optimization approaches are inadequate due to non-linearity, non-differentiability, or high dimensionality of the issue. Utilizing techniques such as Bayesian Optimization and Deep Reinforcement Learning, machine learning offers enhanced flexibility and robustness, facilitating solutions for intricate real-world challenges across several domains. The benchmark optimization test functions Ackley, Rastrigin, and Rosenbrock are employed to assess the optimization methods for convergence rate and execution time. The results demonstrate that numerical optimization approaches exhibit a low convergence rate and diminished runtime, but machine learning optimization displays a high convergence rate and prolonged runtime. In the future, the hybrid optimization methods are considered to solve complex function. The hybrid approach can

combine numerical and machine learning algorithm to obtain the best efficient results.

References

[1] Liu, C., Niu, P., Li, G., Ma, Y., Zhang, W., & Chen, K. (2018). Enhanced shuffled frog-leaping algorithm for solving numerical function optimization problems. *Journal of Intelligent Manufacturing*, *29*, 1133-1153. https://doi.org/10.1007/s10845-015-1164-z

[2] Li, G., Cui, L., Fu, X., Wen, Z., Lu, N., & Lu, J. (2017). Artificial bee colony algorithm with gene recombination for numerical function optimization. *Applied Soft Computing*, *52*, 146-159. https://doi.org/10.1016/j.asoc.2016.12.017

[3] Fera, M., Fruggiero, F., Lambiase, A., Macchiaroli, R., & Todisco, V. (2018). A modified genetic algorithm for time and cost optimization of an additive manufacturing single-machine scheduling. *International Journal of Industrial Engineering Computations*, *9*(4), 423-438. https://doi.org/10.5267/j.ijiec.2018.1.001

[4] Nazir, A., Akhyar, A., Yusra, Y., & Budianita, E. (2022). Toddler nutritional status classification using C4. 5 and particle swarm optimization. *Sci. J. Informatics*, *9*(1), 32-41. https://doi.org/10.15294/sji.v9i1.33158

[5] Azevedo, B. F., Brito, T., Lima, J., & Pereira, A. I. (2021). Optimum sensors allocation for a forest fires monitoring system. *Forests*, *12*(4), 453. https://doi.org/10.3390/f12040453

[6] Zhao, R., & Tang, D. (2024). The integration of financial business and the transformation of financial management functions based on internal control optimization algorithm. *Informatica*, *48*(10). https://doi.org/10.31449/inf.v48i10.5672

[7] Azevedo, B., Bressan, G., Agulhari, C., Santos, H., & Endo, W. (2019). Three-phase induction motors faults classification using audio signals and decision trees. *Applied Mathematics and Information Sciences*, *13*(5), 847-858. http://dx.doi.org/10.18576/amis/130519

[8] Telikani, A., Tahmassebi, A., Banzhaf, W., & Gandomi, A. H. (2021). Evolutionary machine learning: A survey. *ACM Computing Surveys (CSUR)*, *54*(8), 1-35. https://doi.org/10.1145/3467477

[9] Nemmich, M. A., Debbat, F., & Slimane, M. (2020). Hybrid bees approach based on improved search sites selection by firefly algorithm for solving complex continuous functions. *Informatica*, *44*(2). https://doi.org/10.31449/inf.v44i2.2385

[10] Pan, J. S., Liu, N., Chu, S. C., & Lai, T. (2021). An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. *Information Sciences*, *561*, 304-325. https://doi.org/10.1016/j.ins.2020.11.056

[11] Sohrabi, M. K., & Azgomi, H. (2020). A survey on the combined use of optimization methods and game theory. *Archives of Computational Methods in Engineering*, *27*(1), 59-80. https://doi.org/10.1007/s11831-018-9300-5

[12] Lu, Y., Wang, S., Zhao, Y., & Yan, C. (2015). Renewable energy system optimization of low/zero energy buildings using single-objective and multi-objective optimization methods. *Energy and Buildings*, *89*, 61-75. https://doi.org/10.1016/j.enbuild.2014.12.032

[13] Pereira, J. L. J., Oliver, G. A., Francisco, M. B., Cunha Jr, S. S., & Gomes, G. F. (2022). A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. *Archives of Computational Methods in Engineering*, *29*(4), 2285-2308. https://doi.org/10.1007/s11831-021-09663-x

[14] Rosso, M. M., Cucuzza, R., Aloisio, A., & Marano, G. C. (2022). Enhanced multi-strategy particle swarm optimization for constrained problems with an evolutionary-strategies-based unfeasible local search operator. *Applied Sciences*, *12*(5), 2285. https://doi.org/10.3390/app12052285

[15] Abdollahzadeh, B., Gharehchopogh, F. S., Khodadadi, N., & Mirjalili, S. (2022). Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Advances in Engineering Software*, *174*, 103282. https://doi.org/10.1016/j.advengsoft.2022.103282

[16] Gambella, C., Ghaddar, B., & Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, *290*(3), 807-828. https://doi.org/10.1016/j.ejor.2020.08.045

[17] Krishnan, R., Rajpurkar, P., & Topol, E. J. (2022). Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, *6*(12), 1346-1352. https://doi.org/10.1038/s41551-022-00914-1

[18] Naeem, S., Ali, A., Anam, S., & Ahmed, M. M. (2023). An unsupervised machine learning algorithm: Comprehensive review. *International Journal of Computing and Digital Systems*. https://doi.org/10.12785/ijcds/130172

[19] Matsuo, Y., LeCun, Y., Sahani, M., Precup, D., Silver, D., Sugiyama, M., ... & Morimoto, J. (2022). Deep learning, reinforcement learning, and world models. *Neural Networks*, *152*, 267-275. https://doi.org/10.1016/j.neunet.2022.03.037

[20] Albreiki, F., Belayouni, N., & Gupta, D. K. (2023). On Using Deep Learning Proxies as Forward Models in Deep Learning Problems. *arXiv preprint arXiv:2301.07102*. https://doi.org/10.48550/arXiv.2301.07102

[21] Tunay, M., & Abiyev, R. (2022). Improved hypercube optimisation search algorithm for optimisation of high dimensional functions. *Mathematical Problems in Engineering*, *2022*(1), 6872162. https://doi.org/10.1155/2022/6872162

[22] Demo, N., Tezzele, M., & Rozza, G. (2021). A supervised learning approach involving active subspaces for an efficient genetic algorithm in high-dimensional optimization problems. *SIAM Journal on Scientific Computing*, *43*(3), B831-B853. https://doi.org/10.1137/20m1345219

[23] Goitom, S., Nagy, T., & Turányi, T. (2022, January). TESTING VARIOUS NUMERICAL OPTIMIZATION METHODS ON A SERIES OF ARTIFICIAL TEST FUNCTIONS. In *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae. Sectio Computatorica* (Vol. 53). https://doi.org/10.71352/ac.53.175

[24] Optimization Test Functions. https://www.sfu.ca/~ssurjano/ackley.html

Factors and Impacts of Financial Crisis Formation Based on Big Data and RF Early Warning Model

Liyan Jiang
Anyang University, Anyang 456550, China
E-mail: jiangliyan12345678@163.com

Keywords: financial crisis, big data, RF early warning model, financial indicators, SMOTE algorithm

Received: June 16, 2025

With the continuous changes and globalization of financial markets, the factors that shape financial crises have become more complex. To solve the unbalanced data, the study introduces the synthetic minority over-sampling technique with editorial nearest neighbour method to improve the performance of the model. By examining the financial indicators in detail, different aspects of ST and non-ST enterprises are compared. The results show that the synthetic minority over-sampling technique and the editorial nearest neighbour method perform well in financial risk prediction, improving the correct classification rate of the "1" sample by 2.26% and the "0" sample by 3.46%. In addition, the study observes that the "cash content of operating income" of special treatment enterprises is significantly higher than that of non-special treatment enterprises, and that there are significant differences in debt service capacity, overhead growth rate, and profitability. The study provides a powerful methodology for corporate financial risk management and a more effective risk monitoring tool for government regulators and financial institutions.

Povzetek: Prispevek se ukvarja z analizo napovedi finančnih kriz z metodami velikih podatkov in strojnega učenja. Predlaga rabo SMOTE-ENN za uravnoteženje in naključne gozdove za zgodnje opozarjanje; izpostavi pomembnost kazalnikov denarnih tokov ter razlikovanje ST/ne-ST podjetij na podatkih A-trga.

1 Introduction

As the global economy continues to develop, enterprises are facing an increasingly complex and changeable market environment [1]. The occurrence of financial crises is often accompanied by various factors, including market fluctuations, management decisions, economic cycles, etc. Traditional financial analysis methods may be inadequate in complex changes. Therefore, there is an urgent need to combine big data and advanced machine learning algorithms to accurately understand and predict the factors causing financial crises. Traditional financial analysis methods have a series of shortcomings when dealing with financial crisis prediction [2]. These methods usually rely too much on historical data and analysis based on statistical indicators, and cannot fully consider the dynamic changes in the market and the uncertainty of the external environment. This makes it relatively rigid in rapidly changing market conditions and difficult to capture emerging risk factors in a timely manner. In addition, traditional methods are relatively weak in processing large-scale, high-dimensional data, and the explosive growth of data brought about by the current information age makes traditional methods unable to handle massive amounts of data [3]. This limits its ability to comprehensively analyze various potential risk factors, necessitating the need for more flexible and powerful analytical tools. The innovation of the research is to make full use of big data technology and Random Forest (RF) early warning model to model the formation factors of financial crisis. The introduced big data technology is expected to provide more powerful support in data acquisition, storage, and processing, and the random forest early warning model stands out due to its good adaptability to high-dimensional data and non-linear relationships. By combining the two, the research is expected to reveal the intrinsic mechanism of financial crises more comprehensively and accurately, and provide enterprises with more accurate early warning and risk management strategies.

The study consists of four sections. The first section summarizes the research on big data prediction and RF early warning model prediction in financial crises. The second section analyzes the financial crisis data from RF in a big data environment. The third section evaluates the financial early warning model with a comparative multi-model analysis. The fourth section is a summary of the full paper.

2 Related works

Big data is often used to analyse corporate financial problems. Many experts have explored. To establish a healthy, orderly, safe and sustainable development environment, Zhang built an innovative risk early warning model based on big data technology for the financial risk assessment of Internet credit. This model aims to improve the use efficiency of internal and external data and ensure that the early warning system is timely, accurate and effective [4]. However, the modeling framework remains relatively conventional and lacks mechanisms to address imbalanced data distributions, which can lead to biased

early warning outcomes. To improve the financial and economic development and enhance the competitiveness of financial institutions, Luo firstly understood the statistical characteristics of financial data and the design principles of the risk early warning system in the era of big data. Subsequently, the research on the design structure and implementation of financial data statistics and risk early warning analysis system was conducted [5]. However, the study does not offer concrete implementation paths for modeling and algorithmic application, and lacks empirical validation, limiting its practical guidance for financial risk prediction. To reduce the financial risk loss of commercial banks in the context of Internet finance, Lin applied the BP neural network to determine the number of nodes, activation function, learning rate and other parameters of each layer of the neural network. A large number of data samples was used to construct an Internet credit risk early warning model [6]. However, BP neural networks are prone to over-fitting and sensitive to feature noise. In addition, the study does not implement systematic optimization for class imbalance, which affects model stability. Yang discussed how to implement data-driven service upgrading techniques to assist in market regulation, and integrated user authentication information, financial information, and behavioral information through data processing and feature engineering. A credit risk assessment model was constructed using a forest algorithm to exploit the potential value of the data [7]. However, the feature selection process lacks transparency, and the model shows limited interpretability. Moreover, it does not adequately address issues such as small sample sizes and data imbalance common in financial scenarios. Gautama discussed how to develop a fraud alert system using the Software Development Lifecycle (SDLC) approach, building an integrated big data system and fraud alert system to identify potential fraudulent behavior. The fraud alert system integrated LHKPN data and comparative technology data using specific metrics and variables to assess the amount of risk in identifying potential fraud [8]. However, the research focuses on system design and implementation, with limited analysis of model performance and prediction effectiveness, making it difficult to generalize to broader financial risk warning tasks.

Early warning models based on RF algorithms have significant advantages in financial early warning. Jarmulska et al. aimed to answer whether it is possible to design effective and useful machine-learning based early warning systems. The author designed and compared multiple models based on econometric Logit model and RF model to evaluate their performance in predicting financial stress risk. The results showed that it obtained effective early warning models that could correctly predict 70-80% of fiscal stress events and calm periods [9]. However, the study does not test the model generalization across different sample environments and lacks assessment of its adaptability to diverse financial crisis scenarios. Cong discussed how to build financial early warning models for listed enterprises based on multiple different integrated classification models. The experimental results showed that they were effective in predicting financial data for listed enterprises [10]. However, the study primarily focuses on accuracy metrics, without considering comprehensive evaluation indicators such as recall and F1 value under imbalanced sample conditions, and lacks interpretability analysis of the model. Yang et al. explored how to establish financial risk early warning measures to cope with the severe financial situation. A prediction model was established using RF to assess financial risks. Although the model was highly accurate, it had limited interpretability. The Fisher Discriminant Method based on RF was used to improve the interpretability and accuracy of the model [11]. However, the study does not further address feature redundancy or class imbalance, thus achieving only limited improvements in model performance. Wang et al. proposed the "Expert Voting EWS" framework to predict the impending systemic banking crisis. Machine learning algorithms, especially RF classifier, were used to build the EWS. Classifiers, to construct the EWS. The region under the receiver operating characteristic curve was analyzed to compare the generalization ability of different classifiers [12]. Although the system design is practical, the methodology relies heavily on manual expert integration, lacking robust self-adaptive generalization capabilities. Liu et al. used logistic models and seven different machine learning methods, including RF, gradient boosting decision trees, and integrated models to avert the catastrophic effects of the financial crisis. Shapley value decomposition and Shapley regression were used to analyze the causality and impact of machine learning models [13]. However, the study does not incorporate data balancing or sample optimization strategies during the preprocessing stage, which affects the precision of risk sample identification.

In summary, although existing research has made progress in model construction, data integration, and interpretability analysis, especially the powerful performance of RF models in handling nonlinear feature recognition, there remain several limitations. These include the lack of systematic oversampling or noise-reduction strategies for imbalanced data, insufficient generalization across different datasets or enterprise types, limited exploration of financial indicator differences between Special Treatment (ST) and Non-special Treatment (Non-ST) enterprises, and inadequate use of comprehensive performance metrics such as F1 value and AUC. The novelty of this study lies in integrating the Synthetic Minority Over-sampling Technique with Editorial Nearest Neighbour method (SMOTE-ENN) sampling strategy with the RF model, which systematically enhances the model robustness and classification performance under imbalanced samples. Additionally, based on empirical data from the A-share market, the importance of key financial indicators for ST and non-ST enterprises is compared. This study extends the applicability of financial risk prediction models in identifying risk heterogeneity and improving adaptive modeling capabilities.

3 Research on random forest-based financial crisis data analysis in big data environment

The first subsection explores the factors that shape the financial crisis of an enterprise and its impact on the business. The second subsection describes the SMOTE algorithm to deal with unbalanced data, particularly in the training and evaluation of the RF tree dataset. The third subsection explores how the RF model can be used to select important financial warning features as well as improve the generalization of the model.

3.1 Research on the formation factors of corporate financial crisis and its impact in the context of big data

The formation mechanism of financial crisis is a complex process of multi-factorial interaction. Firstly, poor business decisions are a key factor, including unwise strategic planning and resource allocation, wrong capital budgeting decisions, and failure to adapt to market changes or choosing inappropriate product mix [14]. In addition, financial opacity and fraudulent behaviour are important factors contributing to crises, such as false financial statements and improper fund flows. High leverage and debt problems also often trigger financial crises, especially when enterprises are too highly leveraged to carry their debt burden. Liquidity problems may lead to debt default. Macroeconomic factors, such as economic downturns, inflation, or exchange rate fluctuations, as well as competitive and market factors, such as market share losses and unstable market demand, also have a significant impact on financial crises. In addition, corporate governance issues, including inadequate board supervision and management ethics, as well as balance sheet problems such as improper asset valuation and holding of illiquid assets, may have adverse effects on the financial condition of the enterprise. In this complex context, big data analysis plays a crucial role. By applying advanced data analysis and warning models, enterprises can better monitor these factors, track changes in financial indicators, and identify potential risk signals in advance. Big data technologies are able to process large-scale financial data to identify anomalous patterns, predict future trends, and simulate different economic scenarios. This helps organizations assess their financial risks more accurately and take timely action to avoid financial crises from developing. In addition, big data can help businesses better understand market fluctuations, including prices and market volatility, in order to adjust strategies and resource allocation more flexibly, thereby reducing financial pressure [15].

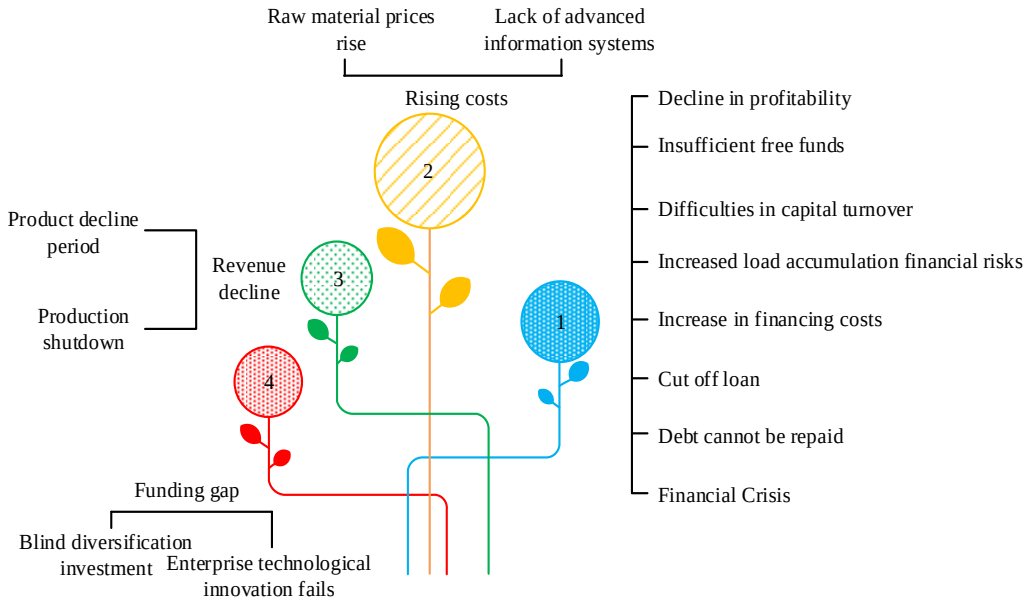


Figure 1: Accounting framework for corporate financial distress identification

The fundamental cause of financial crisis is the accumulation of debt, which leads to the inability to repay debts on time. The initial sign is a decline in income from main business, which may be due to reasons such as stagnant products, rising costs, and failed investments. Blind diversification and over-expansion lead to difficulties in returning capital. Enterprises may attempt to borrow from external sources, but the increase in financing costs can lead to poor financial conditions, making it difficult to obtain new financing, ultimately resulting in a financial crisis.

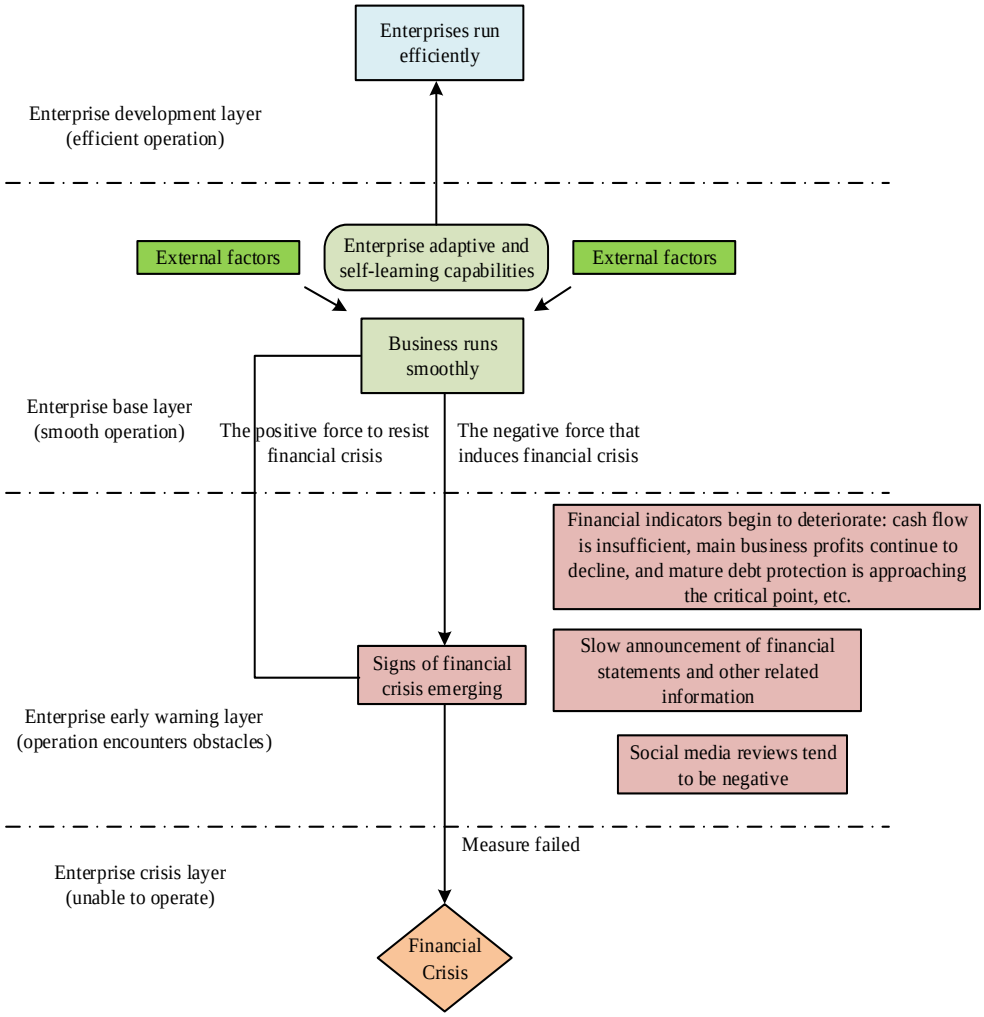
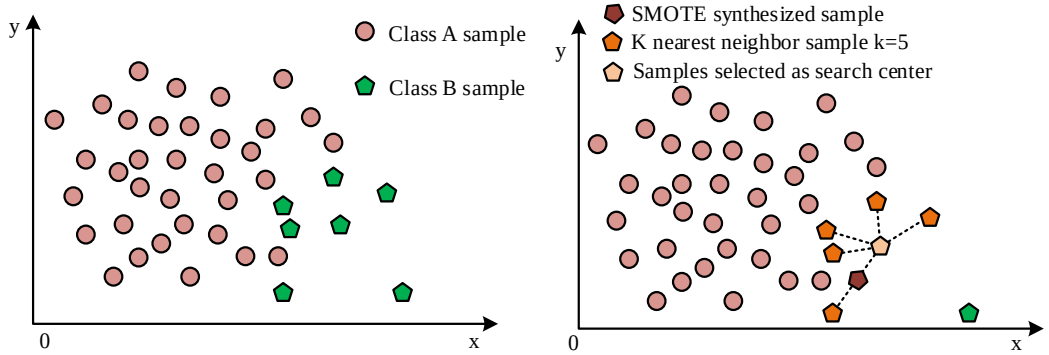


Figure 2: Mechanism diagram of financial distress formation in enterprises

This study conceptualizes enterprise operations as a dynamic regulatory system influenced by both internal mechanisms and external factors. As illustrated in the figure, enterprises operate efficiently under normal conditions, supported by adaptive and self-learning capabilities that help maintain operational stability. Faced with external disturbances or internal inefficiencies, the ability of an enterprise to identify early signals and respond effectively determines whether it can sustain equilibrium or move towards risk accumulation. The enterprise life cycle depicted in the figure includes four phases: stable operation, early anomalies, risk buildup, and crisis onset. In the initial stage, financial indicators are generally healthy [16]. However, as the pressure increases, such as tightening cash flow, declining profits, delayed financial disclosures, and increasingly negative social media sentiment, early signs of financial stress begin to emerge. If these signals are not addressed in time, they may evolve into a full-blown financial crisis. The chart emphasizes that financial crises are rarely sudden events. On the contrary, they are the result of gradual deterioration and failure of corrective mechanisms. Establishing a financial warning system that integrates multiple data sources can help identify risks in the pre-crisis stage and intervene in a timely manner, enhancing the resilience and stability of the enterprise.

3.2 SMOTE algorithm based training and evaluation method for random forest tree dataset

Synthetic Minority Over-sampling Technique (SMOTE) is a technique used to deal with unbalanced datasets by generating synthetic samples to optimize the dataset to improve the performance of a machine learning model. The core idea of SMOTE is to improve the machine learning model by inserting new synthetic samples in the feature space, balancing the sample distribution between different categories.



(a) Calculate K nearest neighbor samples of X (b) Synthesize new samples

Figure 3: Conceptual structure of the SMOTE algorithm

The original dataset is split into a feature matrix (x) and a target vector (y), and the data is explored to understand the imbalance and determine which category is the minority category and which is the majority category. For the minority category, synthetic samples are generated using SMOTE to balance the category distribution. Meanwhile, the generated balanced dataset (including the original samples and the generated synthetic samples) is used to train the RF model. The hyperparameters of the RF can be adjusted as needed to further improve the model performance [17], as shown in equation (1).

$$X_{new} = X_i + (\bar{X} - X_i) \times \delta \quad (1)$$

In equation (1), $X_i \in S\min$. The sample X_i is randomly selected from the nearest neighbors of K=5 to generate a new sample X_{new} . δ denotes the random number. Before determining the model, the model effect is evaluated, and the evaluation indicators are the criteria for objectively assessing the performance and effectiveness of the model. For the classification task, the common evaluation metrics of the classification model are Precision and Recall, as shown in equation (2).

$$R = \frac{TP}{TP + FN} \quad (2)$$

In equation (2), TP denotes the number of samples correctly classified as positive examples by the model. That is to say, the number of actual positive examples correctly predicted by the classifier as positive examples. FN denotes the number of samples misclassified as positive by the model, i.e., the number of actual negative samples that the classifier incorrectly predicts as positive. The precision can be determined using the confusion matrix, which is shown in Figure 4.

Confusion matrix		Actual value	
		Positive	Negative
Predictive value	Positive	TP	FP
	Negative	FN	TN

Figure 4: Confusion matrix structure for binary classification evaluation

Since precision and recall are often opposed to each other, i.e., a high precision is associated with a low recall, and a high recall is associated with a low precision. When there is an imbalance between positive and negative samples, relying solely on precision or recall cannot effectively measure the strengths and weaknesses of the model. To consider the precision and recall comprehensively, the F-Measure (also known as F-Score) is used as a comprehensive evaluation index [18]. F-Measure is the weighted summed average of the precision and recall, which can consider the precision and coverage ability of the classifier comprehensively [9]. The specific formula is shown in equation (3).

$$F = \frac{\left(a^2 + 1\right) \times P \times R}{a^2 \times \left(P + R\right)} \quad (3)$$

Micro-F1 is a comprehensive evaluation metric based on a global overall calculation. It first calculates the True Positive (TP), False Positive (FP), and False Negative (FN) for all categories, and then uses these statistics to calculate the overall precision and recall. Finally, a comprehensive performance evaluation is computed by F1 value [19]. Micro-F1 is suitable for cases where the sample categories are unevenly distributed, as it considers the weights of all category samples in the calculation. The precision is specifically shown in equation (4).

$$P_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} \left(TP_i + FN_i\right)} \quad (4)$$

The recall rate is specifically shown in equation (5).

$$R_{micro} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} \left(TP_i + FN_i\right)} \quad (5)$$

Equations (6) and (7) are substituted into equation (5) to obtain the micro-F1 value, as shown in equation (6).

$$F1_{micro} = \frac{2 \times P_{micro} \times R_{micro}}{P_{micro} + R_{micro}} \quad (6)$$

The results of the corresponding F1 value are calculated according to equation (6). Then, the arithmetic mean is obtained, as shown in equation (7).

$$F1_{micro} = \frac{\sum_{i=1}^{M} F1_i}{M} \quad (7)$$

In Eq. (7), M denotes the number of target categories. In practical multi-classification problems, there is usually no definite conclusion on whether to choose to use micro-F1 values or macro-F1 values as an evaluation metric for model strengths and weaknesses. Each metric has its own characteristics and applicable scenarios, depending on the specific problem and needs.

3.3 Random forest-based financial early warning feature selection and generalisation capability study

RF is a powerful machine learning algorithm that reduces the over-fitting by randomly selecting features and combining multiple decision trees. These trees are constructed independently and predictions are made using majority voting (classification) or averaging (regression). RF evaluates performance through cross-validation and also estimates the importance of features to help identify key features.

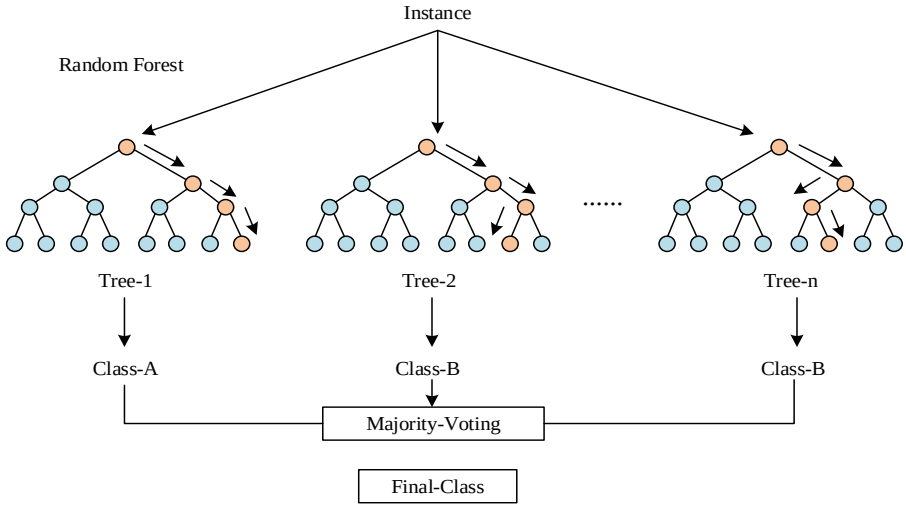


Figure 5: Design logic of the financial safety verification boundary

The application of RF in financial early warning research has significant advantages. The RF model introduces randomness. Firstly, it improves prediction performance by preferring the most valuable features. Secondly, it ranks the importance of each feature to help select the most influential factors that contribute to financial early warning. In addition, it evaluates the importance of individual indicators, estimates the model's generalization error, and deepens the understanding of model stability and generalization performance, which is crucial for financial crisis prediction and risk management. In addition, RF excels in dealing with classification problems, providing reliable prediction results that are not limited by the amount of data. Most importantly, it is suitable for large-scale and high-dimensional data without the need for feature selection and data standardization, simplifying data preparation and accelerating model construction and deployment, making it a powerful tool for financial early warning research [20]. In the above process, it is crucial to determine the best feature and attribute selection when constructing the decision tree model. The core principle of the ID3 decision tree algorithm is to maximize the information gain. The information gain is calculated by analyzing the information gain of the dataset D under the feature A to determine the priority of the best features and attributes to divide the dataset. This is specifically shown in equation (8).

$$Gain(D,A)=Ent(D)-Ent(D|A)=-\sum_{k=1}^{|y|} p_k \log_2 p_k - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v) \quad (8)$$

In Eq. (8), $Ent(\ )$ denotes the information entropy. p_k denotes the class sample percentage. D^v denotes the data ensemble when the v-th attribute is taken on the feature A. To prevent the over-fitting phenomenon, the improved information gain expression is shown in Eq. (9).

$$Gain_ratio(D,A)=\frac{Gain(D,A)}{IV(D,A)}=\frac{Gain(D,A)}{-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}} \quad (9)$$

The pre-cutting method and post-cutting method are shown in equation (10).

$$Gini(D)=\sum_{k=1}^{|y|}\sum_{k`\neq k} p_k p_{k`} = 1-\sum_{k=1}^{|y|} p_k^2 \quad (10)$$

Equation (3) represents the Gini index of the dataset D under the feature A. The feature division point is shown in equation (11).

$$Gini_index(\mathrm{D,A}) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v) \quad (11)$$

The minimization criterion selects the optimal division features and division points and minimizes the squared error, as shown in equation (12).

$$\min_{A,a}[\min_{c_1} \sum_{xi \in D_1(A,a)} (y_i - c_1)^2 + \min_{c_2} \sum_{xi \in D_2(A,a)} (y_i - c_2)^2] \quad (12)$$

In Eq. (12), $D_1(A,a) = \{x \mid x^{(A)} \le a\}$, ; $D_2(A,a) = \{x \mid x^{(A)} > a\}$, y_i denotes the true value of the sample. c_1, c_2 denote the average sample prediction of the sub dataset.

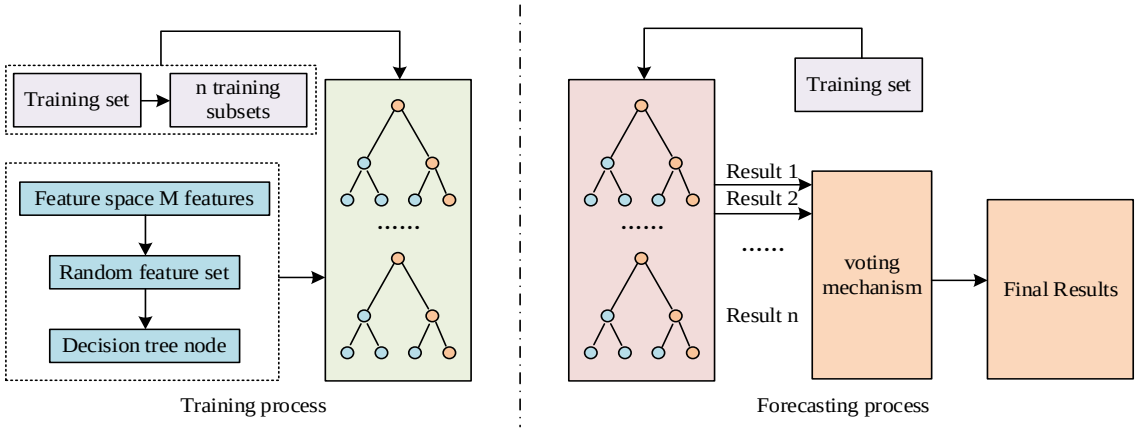


Figure 6: Workflow of random forest model training and prediction

The RF model is unique in two key ways. Firstly, it takes a putative back sampling strategy, leading to out-of-packet data, which is used to test the generalization ability. Secondly, RF introduces randomness in feature selection, combined with an independent base learner and parallel computing, ensuring strong generalization ability. The change in Gini index of the feature variable X_j before and after the above node branching is calculated as shown in equation (13).

$$VIM_{im}(X_j) = Gini(D,A) - Gini_ratio(D,A) - Gini(D) \quad (13)$$

The sum of the changes in the Gini index of the variable X_j over all the decision nodes of the first i tree is calculated, as shown in equation (14).

$$VIM(X) = \sum_{v=1}^{c} VIM(X_v) \quad (14)$$

In Eq. (14), c denotes the number of feature variables.

$$VIM(X_j) = \frac{VIM(X_j)}{VIM(X)} \quad (15)$$

Eq. (15) represents the normalized total Gini change for a feature variable to obtain the final RF importance score.

4 Evaluation of financial early warning models and comparative multi-model analysis

This section is to develop financial early warning models and conduct comparative analyses to determine which method or model performs better in predicting financial crises. The first subsection integrates the financial early warning model and the performance. The second subsection focuses on modelling corporate financial crisis prediction using the RF.

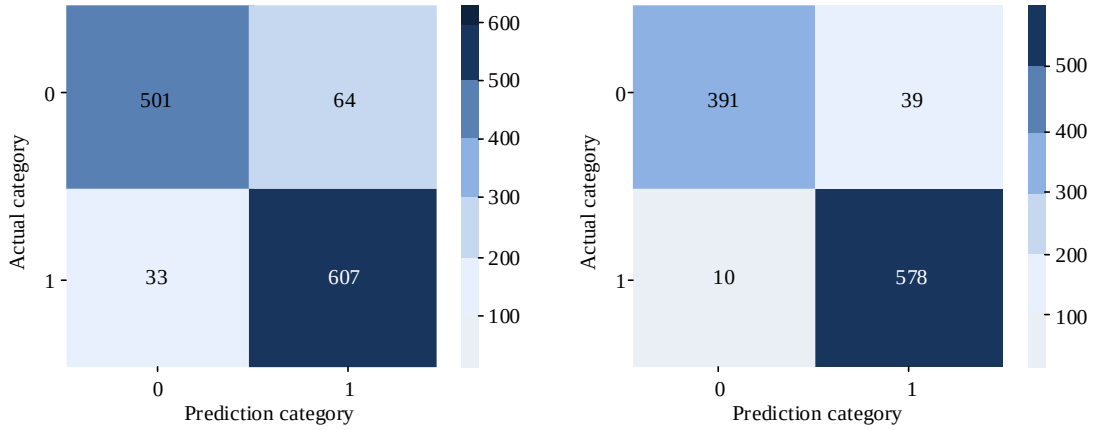
4.1 Development of an integrated model for financial early warning and its comparative cross-analysis

Before training, missing values are filled with medians and outliers are removed using the IQR method. The dataset is split into 70% training and 30% testing sets using stratified sampling. SMOTE-ENN is applied on the training set to handle class imbalance, with SMOTE (k=5) generating synthetic minority samples and ENN removing noisy majority samples, resulting in a near 1:1 class ratio.

RF hyperparameters are tuned using GridSearchCV with five-fold cross-validation. The search covers n_estimators (100-300), max_depth (10, 20, None), and min_samples_split (2, 5, 10). To prevent over-fitting, Out-of-bag (OOB) scoring is used and redundant features are removed based on importance rankings. All experiments are conducted with random_state=42 using scikit-learn (v1.2.2) and imbalanced-learn (v0.10.1) for reproducibility.

Cross-sectional comparative analyses are crucial in the field of financial early warning to help assess the capability and effectiveness of different integrated models for financial early warning in practical applications. A series of standardized experiments and performance evaluations are conducted to compare the performance of different models on the same dataset. These experiments include different performance evaluation metrics such as accuracy, check accuracy, check completeness, F1 value,

AUC under ROC curve, etc. to quantify the performance of each model.



(a) Test set confusion matrix 1 (b) Test set confusion matrix 1

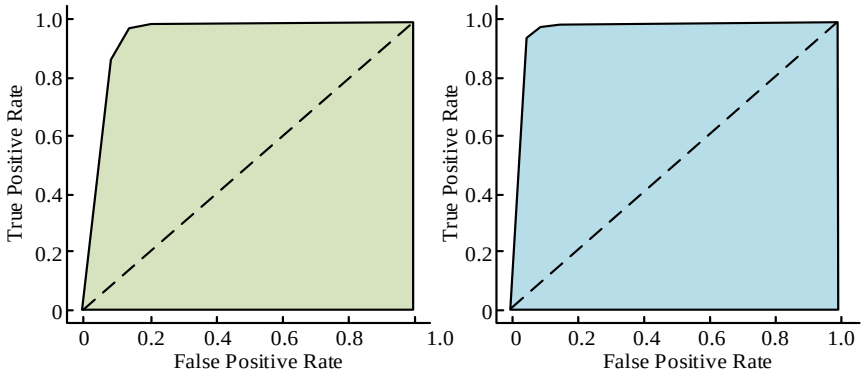
Figure 7: Confusion matrices of the random forest model under two sampling methods

As shown in Figure 7, two data balancing methods are used in the study, namely SMOTE-ENN integrated sampling method and Borderline-SMOTE oversampling method. In Figure 7(a), for "0" category samples, the correct classification rate is 88.67%, while for "1" samples, the correct classification rate is 94.84%. In Fig. 7(b), the correct classification rate is 90.93% for "0" samples and 98.30% for "1" samples. In conclusion, no matter which data balancing method is used, the RF model has a higher correct classification rate for "1" samples than "0" samples. The model based on the SMOTE-ENN integrated sampling method performs better on the correct classification rate, which is 2% higher than the correct classification rate of the "0" samples. The model based on the SMOTE-ENN performs better in the correct classification rate of both types of samples, which is 2.26% and 3.46% higher, respectively. The evaluation indexes of the two models under the RF tree model are shown in Table 1.

Table 1: Evaluation indicators of the random forest model under two data balancing methods

Evaluation index	**Data balancing method used**	
	Borderline-SMOTE oversampling (%)	SMOTE-ENN integrated sampling (%)
Recall	94.85	98.31
Accuracy	90.45	93.69
Precision	91.96	95.20
Out-of-Bag Score	92.49	94.61
F1 value	92.61	95.94

According to the data in Table 1, the accuracy of financial risk identification of the RF model under the Borderline-SMOTE oversampling data balancing method reaches 91.96%. Regarding the recall rate, 94.85% of the enterprises that actually have financial risks can be warned in advance, but there are still 5.15% of enterprises that cannot be identified in advance. When using the SMOTE-ENN integrated sampling data balancing method, the financial risk identification accuracy using the RF model reaches 95.20%. In terms of recall, 98.31% of the enterprises that actually have financial risks can be warned in advance, but 1.69% of the enterprises cannot be identified in advance. To further verify the reliability of the above-mentioned performance differences, the paired t-test is conducted on the results of the two sampling methods. The results show that the SMOTE-ENN method is significantly superior to the Borderline-SMOTE method in multiple indicators (p<0.01), indicating that this method has stronger statistical advantages and practical applicability in financial risk prediction. The performance differences of the model shown in Table 1 fundamentally stem from the differences in the processing mechanisms of the training sample structures between the two data balancing methods. Although Borderline-SMOTE enhances the sample density of the minority classes in the boundary region, it does not clean up the noise samples of the majority classes, resulting in confusion in the model in boundary learning and limiting the precision and F1 value. While the SMODS-ENN method generates minority class samples, it introduces the editing nearest neighbor strategy to effectively remove the outliers located near the boundaries in the majority classes, significantly optimizing the class discrimination structure. This improvement enables the model to learn clearer classification boundaries during the training process and enhances the ability to identify high-risk enterprises.



(a) Test set ROC curve 1 (b) Test set ROC curve 2

Figure 8: ROC curves and AUC values under SMOTE and SMOTE-ENN sampling strategies

From Fig. 8(a) and Fig. 8(b), the ROC curves of the RF model show different trends among different data balancing methods. The ROC curve of the RF model shows a significant improvement when using the SMOTE-ENN sampling method. This method makes the ROC curve closer to the upper left corner and closer to the AUC value of 1, indicating significantly better model performance. This trend suggests that SMOTE-ENN integrated sampling is critical to improving the classification performance of the RF model. The success of the integrated sampling approach suggests that it can play a key role in improving the prediction accuracy when dealing with unbalanced data.

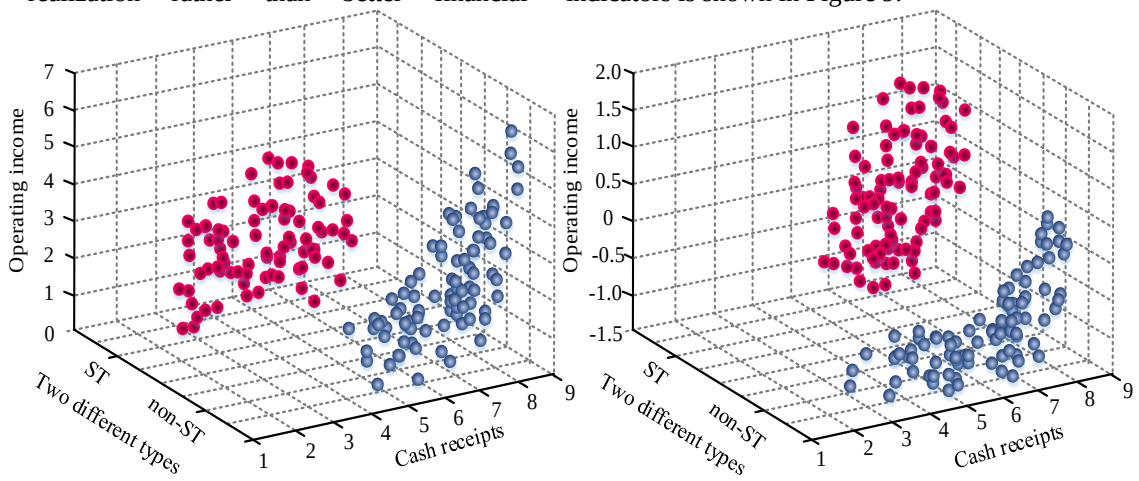
4.2 Analysis of financial crisis prediction of enterprises based on random forest algorithm

RF combines multiple decision trees to improve forecasting accuracy by combining the results of multiple models. It can also provide the importance ranking of each feature. By analyzing the importance ranking of these indicators, the study can identify the factors that have the greatest impact on the financial stability. This study is based on data from the A-share market and selected six ST enterprises. These enterprises refer to those that are given special treatment by the stock exchange in certain special circumstances when the enterprise encounters financial or operational problems. Non-ST enterprises refer to those enterprises that are not classified as ST enterprises, which do not have special financial or operational problems, or at least have not been publicly disclosed. The importance ranking of these indicators is shown in Table 2.

Table 2: Index importance ranking

Indicator name	Weights	Indicator name	Weights
Cash content of operating income	0.332	Operating income growth rate	0.010561
Cash flow from operating activities/interest-bearing liabilities	0.095	Cash reinvestment ratio	0.009645
Operating profit margin	0.075	Total cash recovery rate	0.009438
ROE	0.074	Operating margin	0.009221
Net profit margin on total assets	0.054	Current ratio	0.007136
Rate of return on intangible assets	0.040	Sustainable growth rate	0.00701
Cost profit margin	0.034	Net profit growth rate	0.00631
Overhead rate	0.030	Net cash flow per share	0.006115
Total asset turnover ratio	0.028	Total assets growth rate	0.005007
Assets and liabilities	0.023	Basic earnings per share growth rate	0.004527
Intangible assets growth rate	0.022	Management expense growth rate	0.004512
Net operating cash flow/current liabilities	0.021	Price to sales ratio	0.002465
Corporate free cash flow per share	0.019	Cash ratio	0.001823
Growth rate of net flow from operating activities	0.017	/	/

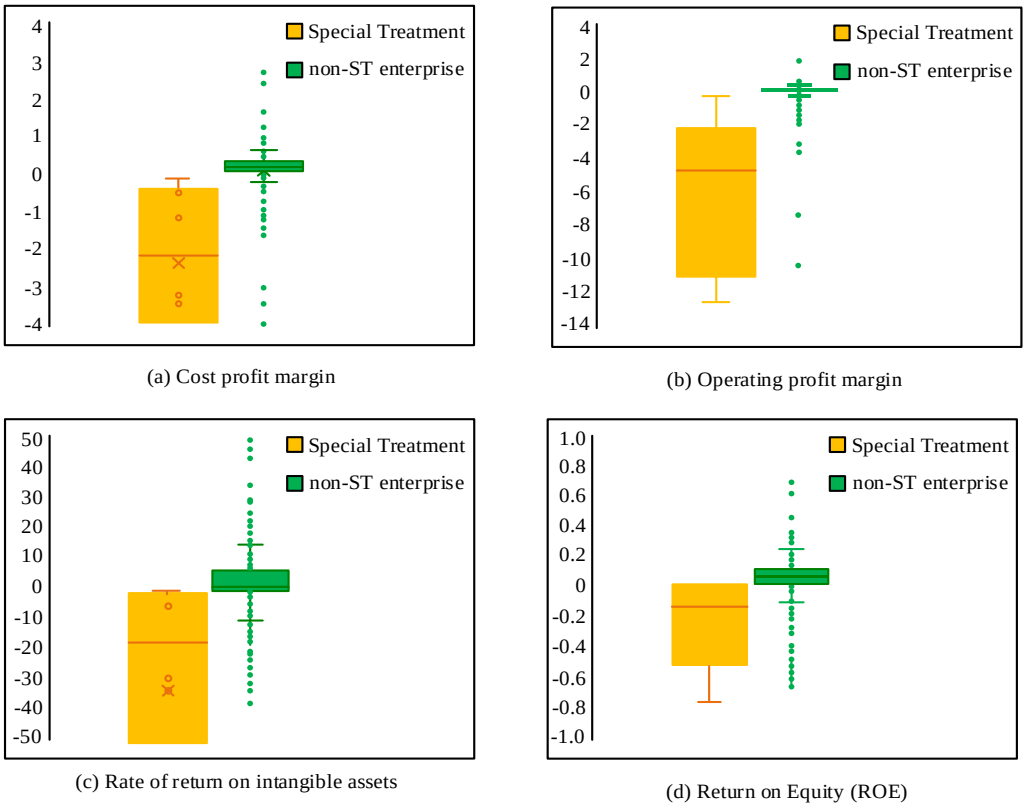
Table 2 ranks financial indicators by their importance within the RF model, calculated through the average reduction in Gini impurity across-decision trees. A higher weight reflects a greater contribution to the classification accuracy. The indicator "Cash content of operating income" shows the highest importance value of 0.332, indicating its strong discriminative ability between ST and non-ST enterprises. Although ST enterprises have a higher average for this indicator, the result may be due to delayed revenue realization rather than better financial performance. Indicators such as "Cash flow from operating activities/interest-bearing liabilities" and "Operating profit margin" also rank highly, reflecting the model's focus on liquidity and profitability. In contrast, traditional static metrics such as the "Cash ratio" have lower weights, suggesting limited influence in predicting financial distress. This importance ranking reflects the model's preference for operational and cash flow efficiency indicators, providing practical guidance for financial risk assessment. The comparison of key financial indicators is shown in Figure 9.



(a) Comparison of cash content of operating income (b) Operating cash flow to interest-bearing debt ratio

Figure 9: Comparison of key financial indicators between ST and non-ST enterprises

Figure 9 shows the differences between ST enterprises and non-ST enterprises in terms of key financial indicators, mainly focusing on two dimensions: solvency and the growth rate of administrative expenses. Solvency is measured by the ratio of cash flow from operating activities to interest-bearing liabilities. A high ratio indicates a stronger debt paying ability of the enterprise. The results show that the overall solvency of non-ST enterprises is higher than that of ST enterprises, reflecting the stability of the former on cash liquidity and debt repayment structure. In terms of the growth rate of administrative expenses, ST enterprises show a significant negative growth, with an average of -0.95, while the average of non-ST enterprises is 0.22, mainly distributed in the positive growth range. This indicates that ST enterprises often cut management expenses under financial pressure. Further, the independent sample t-test is used to compare the two groups of samples. The results show that the difference is statistically significant (p<0.01), indicating that the growth rate of management expenses has a strong discriminatory ability in identifying the financial health status of enterprises. The significant differences in the above-mentioned key indicators not only reveal the structural weaknesses of ST enterprises on debt-paying ability and operating expenditure management, but also provide a solid empirical basis for risk identification and modeling based on financial characteristics.



(a) Cost profit margin

(b) Operating profit margin

(c) Rate of return on intangible assets

(d) Return on Equity (ROE)

Figure 10: Comparison of profitability indicators for ST and non-ST enterprises

Figure 10 shows the comparison between ST enterprises and non-ST enterprises in multiple profitability indicators, including cost profit margin, operating profit margin, return on intangible assets, and return on equity. Firstly, in terms of the cost profit margin indicator, the ST enterprises is relatively high, indicating that the cost expenditure per unit of income is larger, further confirming the insufficiency of their profitability. The operating profit margin reflects an enterprise's ability to generate profits through its main business. ST enterprises perform significantly worse, with an average of -6.24, which is much lower than the average of 0.06 for non-ST enterprises. This indicates that their main business is generally in a loss-making state. According to independent sample t-test, the difference is statistically significant (p<0.01). In terms of the return on intangible assets, the average level of ST enterprises is significantly lower than that of non-ST enterprises, reflecting their shortcomings in the utilization efficiency of intangible resources such as brand and technology. The return on equity is used to measure the ability of an enterprise to create value for shareholders. The average value of ST enterprises is -0.22, which is lower than 0.01 of non-ST enterprises. This indicates that ST enterprises not only have weaker profitability, but also have lower returns to shareholders. This difference is also statistically significant (p<0.05). The differences in the above-mentioned multiple dimensions jointly reveal that ST enterprises have systematic disadvantages on operational efficiency and profit creation ability. The significant differences in these profit indicators not only verify the disadvantaged position of ST enterprises in financial performance, but also provide quantifiable discrimination basis for the risk identification model.

5 Conclusion

Financial crisis is a phenomenon that may have a serious impact on the survival and development of an enterprise. This study delves into the factors that lead to financial crises in enterprises, making full use of big data techniques and RF models to perform detailed analyses and financial risk prediction. Meanwhile, the study uses SMOTE-ENN and Borderline-SMOTE to cope with the unbalanced data problem. The RF model is used for financial risk prediction, where the correct classification rate of the "1" sample is higher than that of the "0" sample. The SMOTE-ENN method performs better, increasing the correct classification rate of the "1" sample. The SMOTE-ENN method performs even better, improving the correct classification rate by 2.26% for the "1" sample and 3.46% for the "0" sample. The ROC curves show a significant performance improvement when using SMOTE-ENN. The study also focuses on financial indicators and finds that the "cash content of operating income" is significantly higher for ST enterprises than for non-ST enterprises. There are also significant differences in debt service capacity, overhead growth rate, and profitability, highlighting the differences in financial indicators between different types of enterprises. Especially for ST enterprises, it is examined in depth to have a more comprehensive understanding of their financial situation and risks. The shortcoming of the study is that it focuses on the current financial prediction. Afterwards, it is

necessary to consider the long-term trend of the financial crisis and the impact of macro factors.

Funding

Research Project on Teacher Education Curriculum Reform in Henan Province in 2022, stage results of exploration and practice of teaching reform of local education majors from the perspective of "curriculum Ideology and politics", Project NO: 2022-JSJYZD-059.

References

[1] Ge J, Wang F, Sun H, Lu F. Research on the maturity of big data management capability of intelligent manufacturing enterprise. Systems Research and Behavioral Science, 2020, 37(4): 646-662. DOI: 10.1002/sres.2819

[2] Zeng H. Influences of mobile edge computing-based service preloading on the early-warning of financial risks. The Journal of Supercomputing, 2022, 78(9): 11621-11639. DOI: 10.1007/s11227-022-04329-2

[3] Zhang Z, Chen Y. Tail risk early warning system for capital markets based on machine learning algorithms. Computational Economics, 2022, 60(3): 901-923. DOI: 10.1007/s10614-022-10260-8

[4] Zhang B. Application of Innovative Risk Early Warning Model Based on Big Data Technology in Internet Credit Financial Risk. Journal of Information Technology Research (JITR), 2022, 15(1): 1-12. DOI: 10.4018/JITR.2022010101

[5] Luo P. Design and implementation of financial data statistics and risk early warning analysis system in the era of big data. Advances in Economics and Management Research, 2023, 6(1): 364-364. DOI: 10.2991/978-94-6463-142-5_53

[6] Lin M. Innovative risk early warning model under data mining approach in risk assessment of internet credit finance. Computational Economics, 2022, 59(4): 1443-1464. DOI: 10.1007/s10614-021-10180-z

[7] Yang G. Research on Financial Credit Evaluation and Early Warning System of Internet of Things Driven by Computer-Aided Technology. Comput. Aided. Des. Appl, 2022, 19(S6): 158-169. DOI: 10.1080/16864360.2022.2147043

[8] Gautama B H, Hanif R, Maretaniandini S T. Fraud Early Warning System: Identifikasi Potensi Fraud dalam Pelaporan Harta Kekayaan Penyelenggara Negara Berbasis Big Data. Innovative: Journal of Social Science Research, 2023, 3(4): 3117-3131. DOI: 10.36315/2023/v3i4.1243

[9] Jarmulska B. Random forest versus logit models: Which offers better early warning of fiscal stress? Journal of Forecasting, 2022, 41(3): 455-490. DOI: 10.1002/for.2806

[10] Cong W. Study of financial warning ensemble model for listed companies based on unbalanced classification perspective. International Journal of Intelligent Information Technologies (IJIIT), 2020, 16(1): 32-48. DOI: 10.4018/IJIIT.2020010103

[11] Yang L, Zhong Z. Research on Early Warning of Financial Risk of Local Financial Enterprises. Financial Engineering and Risk Management, 2022, 5(6): 27-33. DOI: 10.4236/firm.2022.56003

[12] Wang T, Zhao S, Zhu G, Zheng H. A machine learning-based early warning system for systemic banking crises. Applied economics, 2021, 53(26): 2974-2992. DOI: 10.1080/00036846.2020.1870657

[13] Liu L, Chen C, Wang B. Predicting financial crises with machine learning methods. Journal of Forecasting, 2022, 41(5): 871-910. DOI: 10.1002/for.2840

[14] Sun J, Yin F, Altman E, Makosa L. Effects of corporate financial distress on peer firms: do intra-industry non-distressed firms become more conditionally conservative? Accounting and Business Research, 2023, 53(6): 646-670. DOI: 10.1080/00014788.2023.2221560

[15] An B, Suh Y. Identifying financial statement fraud with decision rules obtained from Modified Random Forest. Data Technologies and Applications, 2020, 54(2): 235-255. DOI: 10.1108/DTA-05-2020-0113

[16] Park D, Ryu D. A machine learning-based early warning system for the housing and stock markets. IEEE Access, 2021, 9(6): 85566-85572. DOI: 10.1109/ACCESS.2021.3077962

[17] Chen J. Construction and Application of an Economic Intelligent Decision-making Platform Based on Artificial Intelligence Technology. Informatica, 2024, 48(9). https://doi.org/10.31449/inf.v48i9.5705

[18] Liu H. Enhanced CoCoSo method for intuitionistic fuzzy MAGDM and application to financial risk evaluation of high-tech enterprises. Informatica, 2024, 48(5). https://doi.org/10.31449/inf.v48i5.5169

[19] Zhang P. Big Data-Driven Threat Intelligence Analysis and Early Warning Model Construction. Journal of Global Humanities and Social Sciences, 2023, 4(04): 171-175. https://doi.org/10.61360/BoniGHSS232014160804

[20] Rehman H. Financial Risks Classification Early Warning Analysis of Data Mining Technology. Journal of Global Humanities and Social Sciences, 2022, 3(3): 57-60. DOI:10.47852/bonviewGHSS2022030305

Multi-Task BERT–BiLSTM–CNN–MHA Framework for Legal Sentiment Analysis and Judgment Prediction

Hongqi Liu
School of Marxism, Henan Vocational Institute of Arts, Zhengzhou, 451464, China
E-mail:13903860568@163.com

Keywords: legal text, sentiment analysis, legal judgment prediction, BERT-BiLSTM model, multi-task learning framework

Received: June 3, 2025

This paper proposes a comprehensive multi-task learning (MTL) framework that integrates BERT, Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network (CNN), and Multi-Head Attention (MHA) to jointly perform legal text sentiment analysis (SA) and predict judicial judgments. The architecture utilizes a shared BERT–BiLSTM encoder to generate contextualized representations, which are then extracted by a CNN module to capture fine-grained sentiment features. Meanwhile, an MHA module fuses information and guides the multi-task output for decision-related subtasks. To ensure effective joint optimization, a weighted loss function is designed to balance task-specific objectives and prevent dominance by any single task. Experiments are conducted on the public CAIL2018 dataset using ten-fold cross-validation to guarantee fairness and reproducibility. In this framework, BERT first encodes legal documents into deep semantic embeddings, which BiLSTM then processes to capture sequential dependencies. The CNN subnetwork extracts localized emotional features from the BiLSTM outputs, achieving 98.0% accuracy and an F1 score of 0.96 on sentiment classification (CAIL-big subset). Simultaneously, the MHA module leverages the shared encoder outputs to highlight and weight relevant legal clauses, supporting downstream tasks such as legal article recommendation and sentencing estimation. The recommendation task achieves 94.0% top-1 accuracy, and sentencing regression achieves a mean squared error (MSE) of 0.028. Compared with traditional baselines such as Word2Vec–BiLSTM and single-task CNN models, the proposed BERT–BiLSTM–CNN–MHA framework delivers over 5% improvement in both sentiment analysis and judgment prediction tasks. Its modular design, deep semantic representation, and robust empirical results validate its effectiveness and practical value for deployment in intelligent legal decision-support and judicial assistance systems.

Povzetek: Prispevek obravnava pravno informatiko. Uvede večopravilni BERT–BiLSTM–CNN–MHA za analizo sentimenta in napoved sodb na CAIL2018; SM arhitektura deli koder, CNN zajame lokalne emocije, MHA usmerja pravne podnaloge.

1 Introduction

With the continuous advancement of natural language processing technology, intelligent applications in the legal field have gradually attracted widespread attention [1-2]. Legal texts are highly professional, have complex grammatical structures, and have unique terminology [3-4], making it difficult for traditional SA (sentiment analysis) and judgment prediction methods to handle these characteristics effectively. In traditional SA, these methods are based on bag-of-words models, which shallow models ignore, resulting in biased analysis results. Judgment prediction fails to utilize the emotional information in court statement texts fully, and in complex cases, the model lacks intelligent weighing of legal terms, morality, and legal responsibility. As the demand for legal text processing increases, effectively improving the accuracy of SA and intelligently predicting judgment results has become a key issue that needs to be addressed in the current legal field.

This study aims to address the problems existing in traditional legal text SA and legal judgment prediction methods, specifically the lack of polysemous word processing and new word understanding, as well as the lack of intelligent trade-offs in SA for legal judgments. This paper adopts BERT-BiLSTM as the basic shared model, combines it with CNN (Convolutional Neural Networks) for sentiment feature extraction, and uses MHA (Multi-Head Attention) to handle legal judgment-related tasks. During the research process, the BERT-BiLSTM model was first constructed to generate context-related text representations, followed by the use of CNN to extract sentiment information, and finally MHA was used to recommend legal provisions, predict convictions, and predict sentences. The experiment

employed a joint training MTL (multi-task learning) framework. The multi-task learning (MTL) framework shares the underlying encoder, allowing sentiment

analysis and judgment prediction tasks to share semantic representations in the feature extraction stage and achieve information complementarity. Sentiment analysis helps reveal the attitude tendencies of court statements and assists in making more accurate judgments in complex contexts. Compared with a single task, the MTL framework can capture legal semantics more comprehensively and improve overall performance. This showed that the experimental model achieved significant performance improvements in legal text SA and legal judgment prediction, outperforming the traditional Word2Vec model in both accuracy and F1 value. The experimental results show the potential of this method in the legal field, which can enable more intelligent legal judgment prediction and informed moral trade-offs.

Research question: This paper explores whether the emotion-aware multi-task learning framework can significantly improve the sentencing prediction and sentiment classification performance of legal texts. It also verifies the effect of introducing courtroom statement emotion features in judicial judgment prediction on the accuracy and comprehensive index (F1) of sentencing results.

Expected contributions: (1) Propose the emotion-aware MTL hypothesis - for the first time, jointly model the legal text sentiment analysis and sentencing prediction tasks in the same framework to verify the gain of emotion features on judgment prediction; (2) Construct a modular BERT-BiLSTM-CNN-MHA framework - design three components: shared encoder, multi-channel convolution, and multi-head attention to achieve efficient fusion of deep context and fine-grained emotion; (3) Provide rigorous empirical evidence based on CAIL2018 - through ten-fold cross-validation, quantify the performance improvement of emotion-aware MTL compared with Word2Vec-BiLSTM, single CNN and other SOTA.

2.Related Work

In the field of sentiment analysis (SA) of legal texts, numerous researchers have explored pre-trained language models and deep learning techniques, yielding a range of empirical results. Royyan et al. employed the Word2Vec model to conduct SA on legal public policies, achieving a classification accuracy of 78.99% [5]. Anand et al. combined 1D-CNN with LSTM to capture the semantic features of legal documents, reporting promising Rouge scores [6]. The GloVe (Global Vectors for Word Representation) model has also been widely adopted in SA matching tasks, contributing to improvements in overall classification accuracy [7]. In addition, scholars such as Bello and Khan have applied the classic BERT (Bidirectional Encoder Representations from Transformers) model to legal texts, enhancing contextual understanding of semantic meaning [8–9]. While Word2Vec and GloVe-based models laid the foundation for sentiment representation in legal documents, their performance remained suboptimal due to limitations in modeling deep contextual semantics.

With the advancement of deep learning, its application in legal text SA has expanded. For instance, Abimbola et al. utilized a CNN-LSTM hybrid model for sentiment classification in legal texts, achieving an accuracy of 98.05% [10]. Rajapaksha et al. proposed the Sigmalaw PBSA (Party-Based Sentiment Analysis) model for legal opinion documents, improving task-specific performance [11]. More recent studies have leveraged architectures such as CNN-LSTM [12–13], LSTM [14–15], GRU [16], RoBERTa [17], and Conv-BiLSTM-Frog Leap [18], demonstrating notable improvements in sentiment classification accuracy. These models effectively enhance the representation of syntactic and sequential features but often fail to fully extract and integrate deep semantic structures and fine-grained emotional features in legal language.

Parallel to SA research, recent studies have clarified that judicial judgment prediction typically includes three subtasks: legal article recommendation, conviction prediction, and sentence prediction [19]. For instance, Sengupta et al. applied the SVM (Support Vector Machine) model to predict applicable legal statutes from case reports, obtaining an F1-score of 0.75 [20]. Alghazzawi et al. employed a CNN-LSTM model for court decision prediction, reaching an accuracy of 92.05% [21]. Similarly, Esan and colleagues employed traditional machine learning models, such as Random Forest (RF), achieving an accuracy of around 72% in judicial decision systems [22]. While these models demonstrated particular effectiveness, they primarily relied on legal provision text and neglected the integration of sentiment signals from court narratives. Consequently, they could not model the nuanced interplay between legal reasoning and moral sentiment, which is crucial for intelligent and human-aligned legal decision-making.

A summary of the key studies and their performance comparisons is provided in Table 1.

Table 1: Summary table of the study

References	Models	Datasets	Achieved Metrics	References	Models	Datasets	Achieved Metrics
[5]	Word2Vec	Self-built dataset	Classification	[14]	LSTM	Self-built	Accuracy

			accuracy (78.99%)			dataset	(91.9)
[6]	1DCNN 和 LSTM	Self-built dataset	Rouge-1 (0.436)	[15]	LSTM	Twitter	Accuracy (94.7%)
[7]	GloVe-HeBiLSTM	emotions-dataset-for-NLP dataset	Accuracy (93.70%)	[16]	TS-GRU	IMDB	Accuracy (90.85%)
[8]	BERT-CNN	Self-built dataset	Accuracy (93%)	[17]	RoBERTa	Self-built dataset	Accuracy (92.35%)
[9]	BERT	Self-built dataset	F1 score (81.49%)	[18]	Conv-BiLSTM-Frog Leap	Twitter	Accuracy (97.5%)
[10]	CNN-LSTM	Self-built dataset	Accuracy (98.05%)	[19]	-		
[11]	Sigmalaw PBSA	SigmaLaw-ABSA dataset	Accuracy (0.7086)	[20]	SVM	Self-built dataset	F1 (0.75)
[12]	CNN-LSTM	Self-built dataset	Accuracy (92.5%)	[21]	CNN-LSTM	Self-built dataset	Accuracy (92.05%)
[13]	CNN-LSTM	Reddit dataset	F1 (0.84)	[22]	RF	Self-built dataset	Accuracy (72%)

Notably, recent works published in Informatica have further advanced sentiment analysis (SA) and aspect-based sentiment analysis (ABSA) techniques. For instance, an enhanced RoBERTa-based framework has demonstrated superior performance on aspect-level SA benchmarks [37]. A comprehensive review has summarized various sentiment analysis methods on social media platforms [38], and specialized models have been developed for ChatGPT tweet sentiment classification using classical machine learning algorithms [39]. These studies underscore the growing importance of deep contextual understanding in sentiment-driven applications.

As summarized in Table 1, previous research has achieved noteble results in isolated tasks, such as semantic analysis, sentiment classification, and judgment prediction of legal texts, using models like Word2Vec, GloVe, CNN–LSTM, and RoBERTa. However, these approaches primarily rely on static or shallow word vector representations, limiting their ability to capture the nuanced semantic relationships inherent in legal discourse. Moreover, the association between sentiment cues and legal decision-making has

often been overlooked. Existing single-task learning setups fail to realize inter-task information sharing and collaborative optimization, which may lead to conflicts or bottlenecks when handling multiple interconnected subtasks.

To address these limitations, this paper proposes a multi-task learning (MTL) framework built upon a shared BERT–BiLSTM encoder, a CNN-based sentiment feature extractor, and a multi-head attention (MHA) module. This design enables the integration of deep contextual embeddings with fine-grained emotional features while a weighted joint loss function balances the training objectives across tasks. The proposed approach enhances the model's capacity to comprehend the complex semantics and sentiment tendencies of legal language. Furthermore, it enables the collaborative improvement of subtasks such as legal article recommendation, conviction prediction, and sentencing estimation, thereby effectively overcoming the performance and interpretability limitations observed in prior single-task and shallow modeling approaches.

3 Legal text SA modeling and legal decision prediction model

3.1 BERT-BiLSTM basic shared model

3.1.1 BERT model

BERT is a pre-trained language model based on the Transformer architecture, which understands text through bidirectional contextual information [23-24]. During the pre-training phase, BERT employs the MLM (Masked Language Modeling) task, where it randomly masks some words in the input and infers these masked words based on the surrounding context to capture more comprehensive semantic information. BERT utilizes the NSP (Next Sentence Prediction) task to learn the relationship between sentences, enabling it to exhibit strong migration capabilities across multiple natural language processing tasks.

In this paper, BERT is used as a basic shared model to undertake the task of deep semantic understanding of legal texts. The input of the BERT model consists of three parts: word vector representation, sentence representation, and position representation. The input representation of BERT is shown in formula (1).

$$B_i=B1_i+B2_i+B3_i(1)$$

$B1_i$ represents word vector representation, $B2_i$ represents sentence representation, and $B3_i$ represents position representation.

BERT's encoder is based on a multi-layer Transformer structure and utilizes a self-attention mechanism for bidirectional modeling [25-26]. Based on this structure, BERT can effectively capture the complex dependencies between words. In each Transformer block, the updated calculation formula for the input word representation is shown in formula (2).

$$C_i^{new}=LN(C_i+A(C_i,C_i,C_i))(2)$$

LN represents the layer normalization operation, and C_i represents the input word representation.

In BERT, the pre-training process consists of two main tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The goal of MLM is to maximize the likelihood probability of formula (3).

$$L_M=-\sum_{i=1}^{n} logP(x_i|x\backslash i)(3)$$

$x¥i$ represents the part of the input without word x_i, and $P(x_i|x¥i)$ represents the conditional probability based on the context.

The NSP task is used to learn the relationship between sentences, and the objective function is shown in formula (4).

$$L_N=-\sum_{i=1}^{n} logP(y1_i|S_1,S_2)(4)$$

S_1 and S_2 represent two sentences.

For the input legal text, after passing through the BERT model, the context representation of each word is generated, and then it is passed to the subsequent CNN module for SA and to the MHA module for legal judgment prediction. During the fine-tuning process, all BERT parameters can be optimized according to the task loss allowing the model to yield better results on subsequent tasks.

3.1.2 BiLSTM model

This paper introduces the BiLSTM (Bidirectional Long Short-Term Memory) model [27-28] to enhance further the contextual representation generated by BERT. In the forget gate, the formula is shown in formula (5) [29-30].

$$e_t=\sigma(W_e\cdot[h_{t-1},z_t]+\alpha_e(5)$$

z_t represents the input vector, W_e represents the weight matrix of the forget gate, α_e represents the bias vector.

The input gate is as shown in formula (6).

$$j_t=\sigma(W_j\cdot[h_{t-1},z_t]+\alpha_j(6)$$

The candidate state is shown in formula (7).

$$\widehat{D}_t=tanh(W_D\cdot[h_{t-1},z_t]+\alpha_D(7)$$

*tan*h represents the hyperbolic tangent activation function.

The output gate is shown in formula (8).

$$q_t=\sigma(W_q\cdot[h_{t-1},z_t]+\alpha_q(8)$$

The cell state is shown in formula (9).

$$D_t=e_t\cdot D_{t-1}+j_t\cdot\widehat{D}_t(9)$$

The hidden state is shown in formula (10).

$$h_t = q_t \cdot tanh(D_t) \quad (10)$$

h_t represents the hidden state.

The BiLSTM model performs a merge operation after calculating the hidden states using the forward and backward LSTMs at each time step. The bidirectional hidden state representation is shown in formula (11).

$$\hat{h}_t = [h_t^1, h_t^2] \quad (11)$$

3.1.3 BERT-BiLSTM hybrid model

In this study, the hybrid model of BERT and BiLSTM combines the word embedding generated by BERT pre-training with the bidirectional temporal modeling capability of BiLSTM to form a powerful feature extractor and context modeling module. The hybrid steps are as follows:

1) The input legal text is encoded using the BERT model to generate context-dependent representations of each word. BERT captures long-range dependencies in the text through the self-attention mechanism, particularly in professional contexts and complex syntactic structures within the legal field, providing rich contextual information.

2) The embedded representation generated by BERT is used as the input of BiLSTM further to model the time series characteristics of the text, capture the dependencies from left to right and from right to left, and ensure that the context and sequential information in the text are fully understood.

The BERT-BiLSTM hybrid model is shown in Figure 1.

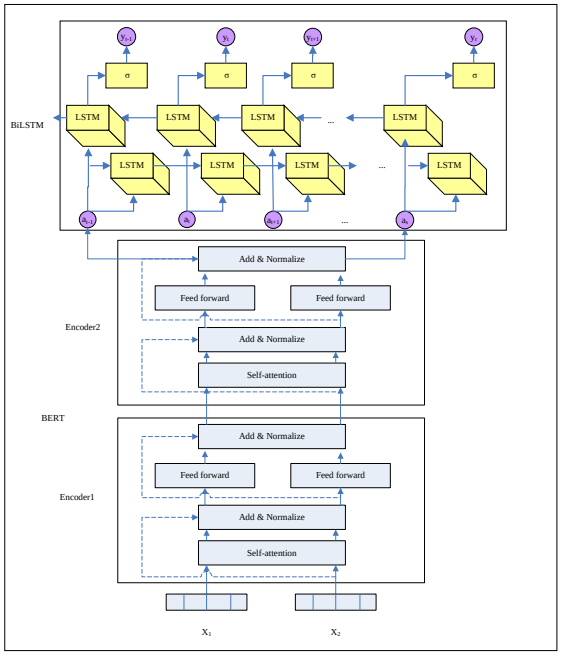


Figure 1: BERT-BiLSTM hybrid model

In Figure 1, it can be seen that BERT and BiLSTM are connected in series. The output of the BERT model passes through a linear transformation layer, adjusted to match the BiLSTM input dimension, and then sent to the BiLSTM network for processing. BiLSTM further integrates the context information provided by BERT, captures more fine-grained semantic and structural information, and outputs a bidirectional context representation of each word. The context vector output by BiLSTM is combined with the word vector output by BERT through weighted summation to form a global semantic representation as input for SA and legal judgment prediction.

3.2 CNN Sentiment Feature Extraction and Classification

This paper introduces the CNN model [31-32] to further extract sentiment features based on the text representation processed by the BERT-BiLSTM model. CNN extracts local sentiment features from word vectors to help the model capture the sentiment tendency information in the text.

In the model architecture design of this study, CNN is placed after BERT-BiLSTM for sentiment feature extraction. The reason is that the BERT-BiLSTM module can fully capture the deep semantics and temporal information of the legal text, providing rich and semantically complete input for subsequent sentiment recognition. Moreover, compared with using CNN as a preprocessing or parallel path, local feature extraction based on BERT-BiLSTM output helps to enhance further the local recognition capability of the sentiment dimension on the premise of modeling global semantics, realize the complementarity of context and sentiment features, and improve the accuracy of sentiment classification and the emotion perception ability of judgment prediction.

To effectively extract sentiment features, this paper employs a CNN architecture comprising multiple convolutional layers and pooling layers. Each convolution operation captures the sentiment information in the text from a different perspective. In the convolutional layer, multiple filters are used for processing. The calculation of the convolution operation is shown in formula (12).

$$o_i^{(k)} = g(W_k \cdot H_i + \beta_k) \quad (12)$$

$o_i^{(k)}$ represents the convolution feature, H_i represents the context vector output by BiLSTM and the word vector output by BERT. g represents the ReLU nonlinear activation function, W_k represents the convolution kernel, and β_k represents the bias term.

The formula for the pooling operation is shown in formula (13).

$$r_k = \max_{i=1,\ldots,n-w+1} (o_i^{(k)}) \quad (13)$$

r_k represents the pooled output of the convolution kernel, and w represents the size of the convolution kernel.

To further enhance the diversity of features, this model employs multi-channel convolution operations, which convolve the input text with convolution kernels of varying sizes to capture emotional features at different granularities. The emotional features extracted by each convolution kernel can form a separate channel after pooling, and the feature channels can be concatenated. The multi-channel output after pooling is shown in formula (14).

$$R=[r_1,r_2,...,r_m] \quad (14)$$

m represents the number of convolution kernels, and r_m represents the output of each convolution kernel.

After extracting the emotional features, this paper can further process them using a fully connected layer. The function of the fully connected layer is to map local features to the global space and generate emotional representations with a higher level of abstraction. The formula of the fully connected layer is shown in formula (15).

$$y_2=\text{softmax}(W_\gamma \cdot R+\delta_\gamma) \quad (15)$$

Among them, W_γ and δ_γ represent the weight matrix and bias term of the fully connected layer respectively.

3.3 MHA Judgment Prediction

3.3.1 Feature fusion

In legal judgment prediction, there are three subtasks, including the recommendation of relevant laws, conviction prediction, and sentence prediction. The emotional factors in legal texts have a significant impact on various types of cases, including criminal cases. This paper uses the results after CNN processing as part of the MHA input. The study first merges the context representation generated by BERT-BiLSTM with the emotional features extracted by CNN. The comprehensive representation obtained by splicing the two features is shown in formula (16).

$$X_{in}=concat(H,y2) \quad (16)$$

H represents the context vector output by BiLSTM and the word vector output by BERT.

3.3.2 MHA mechanism

The MHA mechanism captures multi-scale relationships in text by computing multiple different attention heads in parallel [33-34]. Each attention head focuses on different parts of the input data according to different attention weights, capturing different levels of temporal dependencies. The multi-head attention mechanism uses a linear transformation to output queries, keys, and values. The expression of the query is shown in formula (17). The expressions of the keys and values are shown in formulas (18) and (19), respectively.

$$Q_t=X_{in}W_Q \quad (17)$$

$$K_t=X_{in}W_K \quad (18)$$

$$V_t=X_{in}W_V \quad (19)$$

W_Q represents the query weight matrix, W_K and W_V represent the key and value weight matrices respectively.

The calculation formula of attention weight is shown in formula (20).

$$A_{head}=\text{softmax}(\frac{QK^T}{\sqrt{\zeta_{head}}}) \quad (20)$$

ζ_{head} represents the dimension of each attention head.

The weighted output of each head is shown in formula (21).

$$U_{head}=A_{head}V \quad (21)$$

In multi-head attention, the weighted sum of multiple attention heads is calculated in parallel, and the final output representation is obtained through linear transformation, as shown in formula (22).

$$U=Concat(U_{head_1},U_{head_2},...,U_{head_\eta})W_R \quad (22)$$

Concat represents the concatenation operation, η represents the number of independent attention heads, and W_R represents the output weight matrix.

3.3.3 Task-specific output layer design

The features generated by MHA are fed into three different task-specific output layers; each subtask includes an independent, fully connected layer to generate the final prediction result.

1) Recommending relevant laws

The experiment recommends relevant laws based on the intention expressed in the legal text. The design of the fully connected layer is shown in formula (23).

$$y_3=softmax(W_{law}U+\theta_{law}) \quad (23)$$

2) Conviction prediction

The design of the fully connected layer for the conviction prediction task is shown in formula (24).

$$y_4=softmax(W_{con}U+\theta_{con}) \quad (24)$$

(3) Sentence prediction

The design of the fully connected layer for the conviction prediction task is shown in formula (25).

$y_5=softmax(W_{sen}U+\theta_{sen})$(25)

4.MTL Framework

4.1 Construction of MTL Framework

This study utilizes the BERT-BiLSTM shared model as its basis, introduces feature extraction and prediction modules for CNN and MHA, and constructs an MTL framework to simultaneously address the SA of legal texts and legal judgment prediction tasks. The framework is constructed as follows: First, the BERT-BiLSTM model is used to extract the contextual representation of the text, providing shared basic features for SA and legal judgment prediction.

The MTL framework is shown in Figure 2.

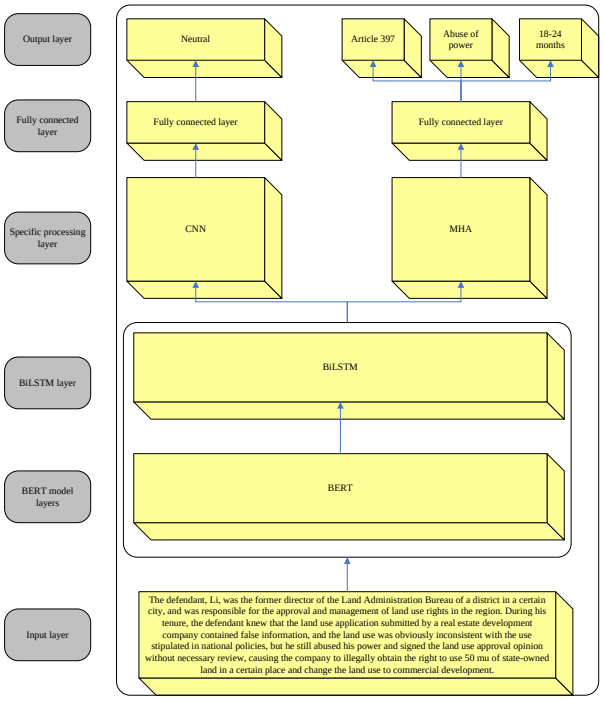


Figure 2: MTL framework

In the sentiment analysis (SA) of legal texts, Convolutional Neural Networks (CNNs) effectively extract and classify sentiment features by stacking convolutional and pooling layers to capture local semantic patterns. In the proposed multi-task learning (MTL) framework, the subtasks include legal text sentiment analysis, relevant law recommendation, and conviction and sentence prediction. The model architecture is composed of shared layers and task-specific output layers.

In the shared layer, a BERT–BiLSTM structure is employed to generate deep contextualized representations. BERT provides rich pre-trained embeddings, while BiLSTM enhances temporal dependency modeling. On top of this foundation, CNN is utilized to extract fine-grained sentiment features, and a Multi-Head Attention (MHA) module captures inter-task dependencies, highlighting salient information relevant to each subtask.

For the task-specific output layers, a fully connected layer is applied to produce sentiment classification results for the SA task. For legal article recommendation, a softmax-based classifier generates predictions of applicable laws. Conviction and sentence prediction tasks are handled by separate output layers, designed respectively for classifying conviction categories and regressing sentence duration.

During training, task-specific loss functions are combined using weighted summation, and joint optimization is performed to improve the overall accuracy across all subtasks. This approach ensures coordinated learning, effectively balancing the optimization between SA and legal judgment tasks. By leveraging the MTL framework, the model can perform multiple tasks within a unified network architecture, facilitating the sharing of semantic and emotional information across tasks and significantly enhancing performance in each task.

4.2 Joint training and optimization

4.2.1 Loss function design

The loss function of legal text SA is shown in the formula (26).

$L_s=-\sum_{i=1}^{N} y_s^{(i)} log(\hat{y}_s^{(i)})$(26)

$y_s^{(i)}$ represents the true value of the sentiment label, and $\hat{y}_s^{(i)}$ represents the sentiment probability value predicted by the model.

In the recommendation of relevant laws, convictions, and sentence predictions, each subtask uses a cross-entropy loss function, and the loss function design is the same as that of legal text SA.

In MTL, the loss functions of each task differ in numerical scale. In order to avoid a particular task dominating the optimization process of the model, the loss functions of each task are now weighted and merged. The comprehensive weighted loss function is shown in formula (27).

$L_{total}=\iota_1 L_s+\iota_2 L_{law}+\iota_3 L_{con}+\iota_4 L_{sen}$(27)

Among them, ι_1 to ι_4 represent the weight hyperparameters of the task loss function.

4.2.2 Model training

During the training process, the experimental data is fed into the multi-task learning (MTL) framework. The model first generates deep semantic representations through the shared BERT–BiLSTM layer, capturing contextual and sequential information from legal texts. These representations are then processed by the CNN module, which extracts fine-grained sentiment features, and the Multi-Head Attention (MHA) module, which captures long-range dependencies and decision-related signals. Finally, task-specific output layers generate

predictions for sentiment classification, legal article recommendation, conviction, and sentencing.

In each training iteration, the model computes the individual loss values for all subtasks, combines them using a weighted summation strategy, and performs backpropagation based on the aggregated loss. Through continuous gradient descent and parameter updates, the model achieves joint optimization across all tasks, aiming to simultaneously improve prediction accuracy for each task.

This paper adopts a joint training strategy, enabling the model to achieve mutual enhancement between the two core tasks: sentiment analysis and legal judgment prediction. This approach effectively avoids task conflicts and promotes overall performance gains. Notably, in the domain of legal judgment prediction, the model demonstrates a stronger ability to recognize the influence of emotional factors on judicial outcomes.

To prevent overfitting, the training process incorporates an early stopping mechanism: if the F1-score on the validation set fails to improve over five consecutive epochs, training is halted. Additionally, to address potential loss conflicts between tasks, the model dynamically monitors the gradient trends of each task's loss during training. If the gradient variation of a particular task exceeds a defined threshold, its corresponding loss weight (λ) is adaptively adjusted. This dynamic loss balancing strategy helps maintain optimization stability enhances convergence efficiency and ensures that no single task dominates the learning process.

The training hyperparameters are shown in Table 2.

Table 2: Hyperparameters

Parameters	Value	Parameters	Value
Learning rate	0.0001	Number of CNN convolution kernels	128
Constant	10^{-8}	Number of attention heads	8
Batch size	32	Number of LSTM units	512
SA weight	0.4	Conviction prediction weight	0.2
Law recommendation weight	0.2	Sentence prediction weight	0.2

4.2.3 Model optimization

In the joint training, the experiment uses the Adam optimizer to update parameters [35-36]. The updated formula of the Adam optimizer is shown in formula (28).

$$\lambda_{t+1}=\lambda_t-\frac{\mu\tilde{\zeta}_t}{\hat{v}_t+o} \quad (28)$$

μ represents the learning rate, and o represents a very small constant.

5 Experiment on SA of Legal Texts and Prediction of Legal Judgments

5.1 Experimental data

The experimental data in this paper come from the China Artificial Intelligence and Legal Challenge CAIL2018 public dataset, including CAIL-small and CAIL-big. In the dataset, each case covers fact description, relevant laws, and sentences, etc. The experiment uses a ten-fold cross-validation to divide the cases, and the mean is then taken as the final experimental result. The summary information of the dataset is shown in Table 3.

Table 3: Summary information of the dataset

Serial number	Project	CAIL-small	CAIL-big
1	Case	128368	1773099
2	Related laws and regulations	103	118
3	Criminal category	119	130
4	Sentence category	12	12

In Table 3, CAIL-small includes 128,368 cases, 103 related laws, 119 crime categories, and 12 sentence categories. CAIL-big includes 1,773,099 cases, 118 related laws, 130 crime categories, and 12 sentence categories.

5.2 Data preprocessing

This article conducts sentiment annotation on legal texts, relevant legal provisions, crime category annotation, and sentence category annotation. The sentiment annotation categories include positive sentiment, neutral sentiment, and negative sentiment.

This article classifies sentences into months. The minimum sentence for a single crime is 6 months, and the maximum is 15 years. The combined sentence for multiple crimes does not exceed 25 years. Life imprisonment is classified separately.

The experimentutilizes THULAC (THU Lexical Analyzer for Chinese) to segment legal texts, remove punctuation marks and stop words, and discard the text after it is deemed "sufficient to identify".

5.3 Experimental process

In this study, sentiment analysis (SA) is conducted using the BERT–BiLSTM–CNN model, while legal judgment prediction is performed using the BERT–BiLSTM–MHA model. The experimental evaluation consists of three parts: (1) the SA validation experiment, (2) the legal judgment prediction validation experiment, and (3) the ablation study.

For sentencing prediction, the original sentencing texts are first normalized by converting all durations into months. The minimum sentence length for a single offense is set to 6 months, and the maximum is capped at 15 years (i.e., 180 months). For cases involving multiple offenses, cumulative sentencing is limited to 25 years (i.e., 300 months). Cases involving life imprisonment are encoded as a separate category labeled "lifetime." Based on this standardization, each case is mapped into a predefined set of discrete intervals (e.g., 6–8 months, 8–12 months, 12–18 months, ..., 180–300 months, and life imprisonment) to align with the model's multi-class classification output requirements for judicial sentencing prediction.

In the SA experiment, the proposed model is compared with several baseline methods, including Word2Vec, GloVe, LSTM, BERT–GRU (a combination of BERT and Gated Recurrent Unit), and CNN–LSTM. For the legal judgment prediction task, comparison models include HARNN–MTL (Hierarchical Attention-based Recurrent Neural Network with Multi-Task Learning), CNN–MTL, LADAN (Law Article Distillation-based Attention Network), and HAN–MTL (Heterogeneous Graph Attention Network with Multi-Task Learning). These baselines are selected to represent a variety of shallow, sequential, and attention-based neural architectures, enabling a comprehensive evaluation of the proposed framework's effectiveness across tasks.

6 Results of SA of legal texts

6.1 Performance of sentiment classification of legal texts

The performance results of sentiment classification of legal texts are shown in Figure 3.

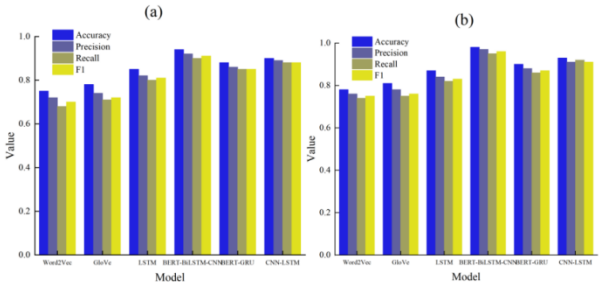


Figure 3 (a): Legal text sentiment classification performance under CAIL-small dataset

Figure 3 (b): Legal text sentiment classification performance under CAIL-big dataset

Figure 3: Legal text sentiment classification performance

In Figure 3(a), the accuracy of traditional models Word2Vec and GloVe is 0.75 and 0.78, respectively, with corresponding F1-scores of 0.70 and 0.72. These models rely primarily on static word embeddings, which are unable to capture dynamic contextual semantics, resulting in limited capability for sentiment classification in legal texts. Deep learning-based models, such as LSTM and CNN–LSTM, demonstrate improved performance. The LSTM model achieves an accuracy of 0.85 and an F1-score of 0.81, while the CNN–LSTM model further improves to 0.90 accuracy and 0.88 F1-score. This indicates that deep neural architectures are more effective in extracting semantic representations, particularly on the CAIL-small dataset. Among all models evaluated, the BERT–BiLSTM–CNN model achieves the best performance, with an accuracy of 0.94 and an F1-score of 0.91, significantly outperforming the baselines. Leveraging pre-trained contextual embeddings from BERT and a multi-layer feature extraction pipeline, the model effectively captures both global semantic and fine-grained sentiment features. Despite the limited size of the dataset, it exhibits strong generalization ability.

In Figure 3(b), similar trends are observed on the CAIL-big dataset. The Word2Vec and GloVe models achieve accuracies of 0.78 and 0.81, with F1-scores of 0.75 and 0.76, respectively. The BERT–BiLSTM–CNN model again outperforms all baselines, reaching an accuracy of 0.98 and an F1 score of 0.96. This superior performance can be attributed to the large-scale

pre-training of the BERT language model, which enables a more accurate representation of complex and domain-specific semantics in legal texts.

Additionally, integrating the model into a multi-task learning framework further enhances its classification capacity by enabling shared semantic and sentiment learning. Compared with results on small datasets, the use of large-scale data significantly reduces the risk of overfitting. It enhances the model's generalization capability, providing more robust and reliable support for sentiment classification in practical legal applications.

6.2 Ablation experiment

The ablation experiment results are shown in Table 4 under the CAIL-big dataset.

Table 4: Ablation experiment results

Model	Accuracy	Precision	Recall	F1
BERT	0.85	0.83	0.8	0.81
BiLSTM	0.89	0.86	0.84	0.85
CNN	0.88	0.87	0.83	0.85
BERT-BiLSTM	0.92	0.89	0.87	0.88
BERT-CNN	0.93	0.91	0.88	0.89
BERT-BiLSTM-CNN	0.98	0.97	0.95	0.96

In Table 4, the BERT-BiLSTM-CNN model achieves the best performance, with an accuracy of 0.98 and an F1 value of 0.96. After gradually removing the modules, the performance gradually decreases. After removing the BiLSTM module, the accuracy of BERT-CNN is 0.93, and the F1 value is 0.89. After removing the CNN module, the accuracy of BERT-BiLSTM is 0.92, and the F1 value is 0.88, indicating that CNN is more important than BiLSTM. When using only BERT, the accuracy drops to 0.85, and the F1 value is 0.81. When using only BiLSTM, the accuracy reaches 0.89, and the F1 value is 0.85.

In BERT-BiLSTM-CNN, BERT provides contextual semantic representation, BiLSTM captures long-distance dependencies in sentences, and CNN plays a key role in extracting and combining local features. This paper uses these modules in combination to achieve optimal performance in semantic understanding and SA.

7.Legal Judgment Prediction Results

7.1 Legal Judgment Confusion Matrix

In the CAIL-big dataset, the test set has 18765 cases of 6-8 years, 23120 cases of 8-12 years, 27483 cases of 12-18 years, 21982 cases of 18-24 years, 17675 cases of 24-36 years, 15894 cases of 36-48 years, 12764 cases of 48-60 years, 10985 cases of 60-96 years, 9432 cases of 96-120 years, 8543 cases of 120-180 years, 7214 cases of above 180 years, and 3452 cases of life imprisonment. The legal judgment confusion matrix is shown in Figure 4.

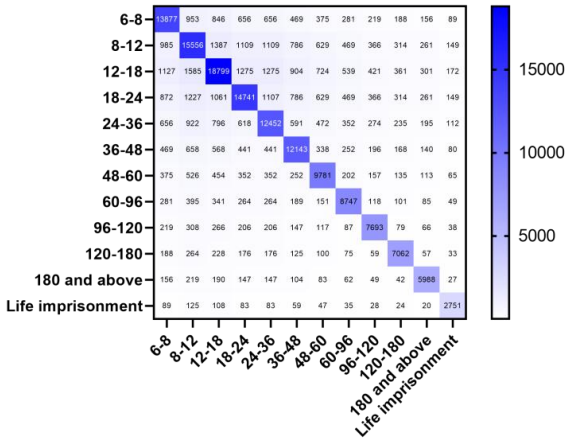


Figure 4: Legal judgment confusion matrix

In Figure 4, 13,877 cases were correctly classified as 6-8 years, 953 were misclassified as 8-12 years, and 846 were misclassified as 12-18 years. 15,556 cases were correctly classified as 8-12 years, and 18,799 cases were correctly classified as 12-18 years. In summary, different categories can be well classified, and there are very few that are incorrectly predicted as other categories.

7.2 Legal judgment prediction performance under CAIL-small dataset

The results of legal judgment prediction performance under the CAIL-small dataset are shown in Figure 5.

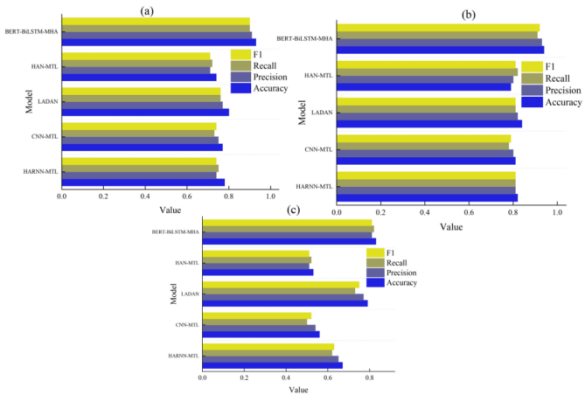


Figure 5 (a): Legal article recommendation prediction performance in CAIL-small dataset
Figure 5 (b): Legal article conviction prediction performance in CAIL-small dataset
Figure 5 (c): Legal article sentence prediction performance in CAIL-small dataset
Figure 5: Legal judgment prediction performance in CAIL-small dataset

In terms of legal article recommendation, the BERT-BiLSTM-MHA model performs best, with an accuracy of 0.93, a precision of 0.91, a recall of 0.90, and an F1 value of 0.90. LADAN has an accuracy of 0.80 and an F1 value of 0.76, while HAN-MTL has an accuracy of only 0.74 and an F1 value of 0.71. In the task of conviction prediction, the BERT-BiLSTM-MHA model exhibits a significant advantage, achieving an accuracy of 0.94, a precision of 0.93, a recall of 0.91, and an F1 score of 0.92. LADAN has an accuracy of 0.84 and an F1 value of 0.81, while HAN-MTL has an accuracy of 0.79 and an F1 value of 0.81.

In the sentence prediction task, BERT-BiLSTM-MHA achieved an accuracy of 0.83, a precision of 0.81, a recall of 0.82, and an F1 value of 0.81, which are much higher than other models. LADAN has an accuracy of 0.79 and an F1 value of 0.75. HARNN-MTL has an accuracy of 0.67 and an F1 value of 0.63. HAN-MTL and CNN-MTL have lower performance, with accuracy of only 0.53 and 0.56, respectively.

In summary, in the CAIL-small dataset, the BERT-BiLSTM-MHA model performs worse in prison sentence prediction in legal judgment prediction but performs better in conviction prediction and legal article recommendation. BERT-BiLSTM-MHA has a profound ability to capture semantic context. It achieves higher prediction accuracy through the context representation provided by BERT, the long-distance dependency captured by BiLSTM, and the important features focused on by MHA.

7.3 Legal judgment prediction performance under CAIL-big dataset

The legal judgment prediction performance results under the CAIL-big dataset are shown in Figure 6.

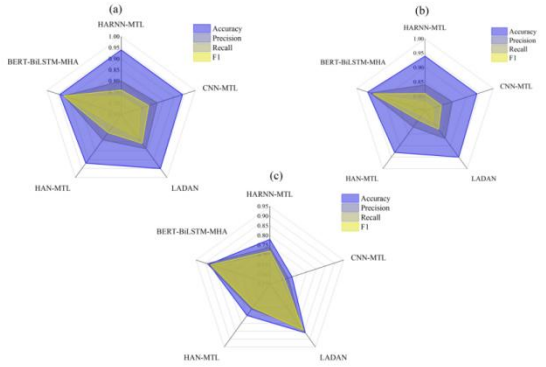


Figure 6 (a): Legal article recommendation prediction performance under CAIL-big dataset
Figure 6 (b): Legal article conviction prediction performance under CAIL-big dataset
Figure 6 (c): Legal article sentence prediction performance under CAIL-big dataset
Figure 6: Legal judgment prediction performance under CAIL-big dataset

In the legal article recommendation task of the CAIL-big dataset, BERT-BiLSTM-MHA achieved an accuracy of 0.94, a precision of 0.92, a recall of 0.92, and an F1 score of 0.92, leading to its overall performance. The LADAN model has a higher accuracy of 0.95 and an F1 value of 0.81, while the F1 value of HAN-MTL is only 0.75. In the conviction prediction task, BERT-BiLSTM-MHA performed best, with an accuracy of 0.96, a precision of 0.95, a recall of 0.93, and an F1 value of 0.94, which are significantly higher than other models. LADAN has an accuracy of 0.95 and an F1 value of 0.83. HAN-MTL has an accuracy of 0.93 and an F1 value of only 0.79, with the weakest performance.

In the prison sentence prediction task, BERT-BiLSTM-MHA has an accuracy of 0.88, a precision of 0.89, a recall of 0.85, and an F1 value of 0.87, ranking first. LADAN has an accuracy of 0.86 and an F1 value of only 0.84. HARNN-MTL and CNN-MTL perform poorly, with F1 values of only 0.72 and 0.63, respectively.

In summary, BERT-BiLSTM-MHA combines the contextual semantic information provided by BERT and the multi-head attention mechanism to capture the semantic associations between legal provisions more accurately. At the same time, it has a strong generalization ability for large datasets and avoids information omission. In the CAIL-big data set, the overall performance is better than that in the CAIL-small data set.

7.4 Significance test

In order to test the statistical significance of the performance difference between BERT-BiLSTM-CNN-MHA and each comparison model under ten-fold cross-validation, this paper designed the following experimental steps:

(1) For each model, ten-fold cross validation was completed on the CAIL-big dataset. The accuracy (one group for each of the three tasks of legal recommendation, conviction prediction, and sentence prediction) and F1 value of each fold were recorded. A total of 10 data sequences were obtained.

(2) For each task, a paired sample t test was used to compare the difference between BERT BiLSTM CNN MHA and each comparison model in the ten-fold results, with the significance level set to $\alpha=0.05$.

(3) The test results of each comparison group were summarized in the form of p-value and marked with "*" for $p<0.05$ (significant), "**" for $p<0.01$ (highly significant), otherwise no mark.

The t-test p-values of BERT-BiLSTM-CNN-MHA and the comparison models on the three tasks' Accuracy are shown in Table 5.

Table 5: T-test results

Model comparison	Legal recommendation p value	Conviction prediction p value	Sentence prediction p-value
HARNN-MTL	0.021*	0.018*	0.003**
CNN-MTL	0.015*	0.012*	0.0005**
LADAN	0.045*	0.034*	0.012*
HAN-MTL	0.008**	0.009**	0.001**

As shown in Table 5, the improvements of BERT-BiLSTM-CNN-MHA on the three tasks have reached statistically significant levels, especially in the sentence prediction task ($p<0.01$), which further verifies the robustness and superiority of this method.

7.5 Discussion on the accuracy of each category (sentence length)

In order to deeply evaluate the performance of the model on different sentence lengths, this paper statistically analyzes the accuracy and recall of the BERT-BiLSTM-MHA model for each sentence category under the CAIL-big dataset. The sentence categories are divided into 12 intervals: 6–8 months, 8–12 months, 12–18 months, 18–24 months, 24–36 months, 36–48 months, 48–60 months, 60–96 months, 96–120 months, 120–180 months, more than 180 months, and life imprisonment.

The prediction accuracy and recall results of each category are shown in Table 6.

Table 6: Prediction accuracy and recall of each category

Sentence type (unit: month)	Accuracy	Recall
6–8	0.86	0.84
8–12	0.88	0.85
12–18	0.90	0.87
18–24	0.89	0.86
24–36	0.91	0.88
36–48	0.92	0.89
48–60	0.90	0.87
60–96	0.87	0.84
96–120	0.85	0.82
120–180	0.83	0.81
180+	0.81	0.78
Life imprisonment	0.95	0.91

As shown in Table 6, the model performs most stably in medium-term sentences (i.e., 12–48 months), with generally high accuracy and recall. For long-term sentences (i.e., those exceeding 180 months), the model's predictive ability has declined, which is attributed to the relatively small number of samples or the lack of concentration of text features. Life imprisonment has obvious emotional characteristics due to its severe circumstances and significant language description. The model has the strongest ability to distinguish this type,

with an accuracy of 0.95. This analysis further verifies the adaptability and effectiveness of the emotion perception mechanism for different types of sentences in sentencing prediction.

7.6 Verification of emotional factors

To verify the importance of emotional factors in legal judgments, this paper conducts experiments using the CAIL-big dataset. The verification results of emotional factors are shown in Table 7.

Table 7: Verification results of emotional factors

	Type	Accuracy	Precision	Recall	F1
Legal recommendation	Emotion	0.94	0.92	0.92	0.92
	Emotion not considered	0.92	0.91	0.89	0.9
	Type	Accuracy	Precision	Recall	F1
Conviction prediction	Emotion	0.96	0.95	0.93	0.94
	Emotion not considered	0.93	0.92	0.92	0.92
	Type	Accuracy	Precision	Recall	F1
Sentence prediction	Emotion	0.88	0.89	0.85	0.87
	Emotion not considered	0.85	0.84	0.83	0.83

In Table 7, the BERT-BiLSTM-MHA model performs better than the model without considering emotional factors in terms of legal article recommendation, conviction prediction, and sentence prediction, which is more in line with actual needs and comprehensively balances morality and legal articles.

8 Further discussion

The model proposed in this paper demonstrates significantly superior performance compared to traditional methods in both legal text sentiment analysis (SA) and legal judgment prediction. Traditional models such as Word2Vec and GloVe are relatively simple in their representation capabilities. They fail to capture contextual dependencies, resulting in low accuracy fully and F1 scores in sentiment classification tasks. In contrast, the BERT-BiLSTM-CNN architecture exhibits excellent performance on both the CAIL-small and CAIL-big datasets. The BERT model provides rich semantic representations through pre-training. When combined with BiLSTM and CNN, it effectively captures long-range dependencies and localized features, substantially improving the accuracy of sentiment analysis.

The impact and significance of this study are primarily reflected in two aspects. First, the introduction of the BERT-BiLSTM-CNN model enhances the effectiveness of legal sentiment classification and legal judgment prediction. Its application on large-scale datasets highlights the model's capability to handle complex semantic structures in legal texts. Second, the research offers a novel perspective for automating legal text processing and provides valuable references for sentiment analysis tasks in other domains. Furthermore, the study employs a multi-task learning (MTL) framework to enhance the accuracy of legal judgment prediction—particularly in legal article recommendation and conviction prediction—yielding outstanding model performance. Overall, this work offers practical value in integrating legal technology with artificial intelligence, contributing to the improvement of efficiency and precision in legal information processing.

Empirically, the proposed BERT-BiLSTM-CNN-MHA model achieves an accuracy of 98.0% and an F1-score of 0.96 on the sentiment classification task, outperforming the 78.99% accuracy of Royyan et al. using Word2Vec [5] and the Rouge-1 F1 score of 0.436 obtained by Anand et al. with a 1DCNN-LSTM model [6]. Compared to GloVe-HeBiLSTM (93.70%, [7]) and pure BERT-CNN (93%, [8]), the proposed model achieves at least a 4.3% improvement, thanks to its multi-level feature extraction and deep contextual embedding design. In judicial judgment prediction, the BERT-BiLSTM-MHA variant yields a conviction prediction accuracy of 96% and an F1-score of 0.94 on the CAIL-big dataset, surpassing the 92.05% accuracy of CNN-LSTM reported by Alghazzawi et al. [21] and the 0.75 F1-score reported by Sengupta et al. [20], with greater robustness across diverse case types.

The substantial improvements in model performance can be attributed to the synergistic effect of deep contextual representation and joint multi-task optimization. BERT's bidirectional encoding effectively captures long-range dependencies and legal-specific semantic nuances, compensating for the limitations of static embeddings like Word2Vec and GloVe, which are unable to resolve semantic ambiguity. BiLSTM further enhances temporal feature modeling, tightly coupling

emotional dynamics with contextual semantics. CNN, through its multi-channel convolutional structure, extracts fine-grained sentiment signals, which are then weighted and fused with contextual representations via the multi-head attention (MHA) mechanism. This design maximizes the influence of emotional cues on legal outcome prediction and compensates for the limitations of traditional single-task models in capturing therelationship between emotion and judgment.

From an architectural perspective, three design choices are pivotal: shared BERT-BiLSTM encoders, multi-channel CNN modules, and multi-head attention (MHA). The shared encoders reduce parameter redundancy while promoting knowledge transfer across tasks. The CNN component captures sentiment features at varying granularities, offering MHA a rich and diverse input representation. MHA then selectively integrates and balances feature weights from different tasks via parallel attention heads, enabling coordinated optimization across sub-tasks such as legal article recommendation, conviction prediction, and sentencing length estimation. The effectiveness of this architecture is further validated by ablation studies: the removal of any single module results in significant performance degradation, underscoring the critical role of each component.

Future research may focus on enhancing the robustness and generalization of the proposed framework, as well as exploring its integration into real-world intelligent legal judgment systems. Additionally, studies in other domains further validate the broad applicability of multi-task learning. For instance, Informatica has demonstrated the use of MTL beyond natural language processing by combining it with Q-learning for joint optimization in logistics tasks such as path selection, scheduling, and resource allocation [40]. Similarly, multi-source deep MTL frameworks have been developed for simultaneous smile detection, emotion recognition, and gender classification [41], further underscoring the versatility and practical potential of the MTL paradigm.

9 Conclusions

This paper adopts an MTL framework and has achieved remarkable results in legal text SA and legal judgment prediction. The experiment constructs a BERT-BiLSTM shared model, utilizes context-related representations to capture the characteristics of legal text, combines a CNN for sentiment feature extraction, and employs MHA to handle tasks related to legal judgment. Experimental results show that the MTL framework outperforms the traditional Word2Vec model in terms of accuracy and F1 value and can effectively improve the performance of legal text analysis. The framework also performs well in tasks such as law recommendation, conviction prediction, and sentence prediction, demonstrating its potential for application in the legal field. This study still has room for improvement in dealing with more complex legal scenarios and multimodal data. Future research can further enhance the robustness of the model, explore additional types of legal cases, and improve the model's ability to comprehend unstructured court statement texts, thereby promoting the in-depth development of intelligent legal judgment systems.

References

[1] Krasadakis, P., Sakkopoulos, E., & Verykios, V. S. (2024). A survey on challenges and advances in natural language processing with a focus on legal informatics and low-resource languages. Electronics, 13(3), 648. https: //doi.org/10.3390/electronics13030648

[2] Maree, M., Al-Qasem, R., & Tantour, B. (2024). Transforming legal text interactions: leveraging natural language processing and large language models for legal support in Palestinian cooperatives. International Journal of Information Technology, 16(1), 551-558. https://doi.org/10.1007/s41870-023-01584-1

[3] DeMattee, A. J. (2023). A grammar of institutions for complex legal topics: Resolving statutory multiplicity and scaling up to jurisdiction-level legal institutions. Policy Studies Journal, 51(3), 529-550. https://doi.org/10.1111/psj.12488

[4] Allison, N. G. (2023). From semantic weight to legal ontology via classification of concepts in legal texts. The Law Teacher, 57(2), 201-217. https://doi.org/10.1080/03069400.2023.2173918

[5] Royyan, A. R., & Setiawan, E. B. (2022). Feature expansion Word2Vec for sentiment analysis of public policy in Twitter. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 6(1), 78-84.

[6] Anand, D., & Wagh, R. (2022). Effective deep learning approaches for summarization of legal texts. Journal of King Saud University-Computer and Information Sciences, 34(5), 2141-2150. https://doi.org/10.1016/j.jksuci.2019.11.015

[7] Mahto, D., & Yadav, S. C. (2024). Emotion prediction for textual data using GloVe based HeBi-CuDNNLSTM model. Multimedia Tools and Applications, 83(7), 18943-18968. https://doi.org/10.1007/s11042-023-16062-w

[8] Bello, A., Ng, S. C., & Leung, M. F. (2023). A BERT framework to sentiment analysis of tweets. Sensors, 23(1), 506. https://doi.org/10.3390/s23010506

[9] Khan, L., Amjad, A., Ashraf, N., & Chang, H. T. (2022). Multi-class sentiment analysis of urdu text using multilingual BERT. Scientific Reports, 12(1), 5436. https://doi.org/10.1038/s41598-022-09381-9

[10] Abimbola, B., de La Cal Marin, E., & Tan, Q. (2024). Enhancing Legal Sentiment Analysis: A Convolutional Neural Network–Long Short-Term

Memory Document-Level Model. Machine Learning and Knowledge Extraction, 6(2), 877-897. https://doi.org/10.3390/make6020041

[11] Rajapaksha, I., Mudalige, C. R., Karunarathna, D., de Silva, N., Ratnayaka, G., & Perera, A. S. (2022). SigmaLaw PBSA-A Deep Learning Approach For Aspect Based Sentiment Analysis in Legal Opinion Texts. J. Data Intell., 3(1), 101-115.

[12] Abimbola, B., Tan, Q., & De La Cal Marín, E. A. (2024). Sentiment analysis of Canadian maritime case law: a sentiment case law and deep learning approach. International Journal of Information Technology, 16(6), 3401-3409. https://doi.org/10.1007/s41870-024-01820-2

[13] Zhang, N., Xiong, J., Zhao, Z., Feng, M., Wang, X., Qiao, Y., & Jiang, C. (2024). Dose my opinion count? A CNN-LSTM approach for sentiment analysis of Indian general elections. Journal of Theory and Practice of Engineering Science, 4(05), 40-50. https://doi.org/10.53469/jtpes.2024.04(05).06

[14] Pavan Kumar, M. R., & Jayagopal, P. (2023). Context-sensitive lexicon for imbalanced text sentiment classification using bidirectional LSTM. Journal of Intelligent Manufacturing, 34(5), 2123-2132. https://doi.org/10.1007/s10845-021-01866-0

[15] Sirisha, U., & Bolem, S. C. (2022). Aspect based sentiment & emotion analysis with ROBERTa, LSTM. International Journal of Advanced Computer Science and Applications, 13(11).

[16] Zulqarnain, M., Ghazali, R., Aamir, M., & Hassim, Y. M. M. (2024). An efficient two-state GRU based on feature attention mechanism for sentiment analysis. Multimedia Tools and Applications, 83(1), 3085-3110. https://doi.org/10.1007/s11042-022-13339-4

[17] Chauhan, A., Sharma, A., & Mohana, R. (2025). An Enhanced Aspect-Based Sentiment Analysis Model Based on RoBERTa For Text Sentiment Analysis. Informatica, 49(14):193-202. https://doi.org/10.31449/inf.v49i14.5423

[18] Yelisetti, S., & Geethanjali, N. (2023). Emotion-based sentiment analysis using conv-BiLSTM with frog leap algorithms. Acta Informatica Pragensia, 12(2), 225-242.

[19] Zhang, H., Dou, Z., Zhu, Y., & Wen, J. R. (2023). Contrastive learning for legal judgment prediction. ACM Transactions on Information Systems, 41(4), 1-25. https://doi.org/10.1145/3580489

[20] Sengupta, S., & Dave, V. (2022). Predicting applicable law sections from judicial case reports using legislative text analysis with machine learning. Journal of Computational Social Science, 5(1), 503-516. https://doi.org/10.1007/s42001-021-00135-7

[21] Alghazzawi, D., Bamasag, O., Albeshri, A., Sana, I., Ullah, H., & Asghar, M. Z. (2022). Efficient prediction of court judgments using an LSTM+ CNN neural network model with an optimal feature set. Mathematics, 10(5), 683. https://doi.org/10.3390/math10050683

[22] Esan, A. O. (2024). PERFORMANCE EVALUATION OF MACHINE LEARNING ALGORITHMS FOR JUDICIAL PREDICTION SYSTEM. LAUTECH Journal of Engineering and Technology, 18(1), 192-203.

[23] Hao, S., Zhang, P., Liu, S., & Wang, Y. (2023). Sentiment recognition and analysis method of official document text based on BERT–SVM model. Neural Computing and Applications, 35(35), 24621-24632. https://doi.org/10.1007/s00521-023-08226-4

[24] Wicaksono, G. W., Azhar, Y., Hidayah, N. P., & Andreawana, A. (2023). Automatic summarization of court decision documents over narcotic cases using bert. JOIV: International Journal on Informatics Visualization, 7(2), 416-422. http://dx.doi.org/10.30630/joiv.7.2.1811

[25] Üveges, I., & Ring, O. (2023). HunEmBERT: a fine-tuned BERT-model for classifying sentiment and emotion in political communication. IEEE Access, 11, 60267-60278. https://doi.org/10.1109/ACCESS.2023.3285536

[26] Jlifi, B., Abidi, C., & Duvallet, C. (2024). Beyond the use of a novel Ensemble based Random Forest-BERT Model (Ens-RF-BERT) for the Sentiment Analysis of the hashtag COVID19 tweets. Social Network Analysis and Mining, 14(1), 88. https://doi.org/10.1007/s13278-024-01240-x

[27] Gou, Z., & Li, Y. (2023). Integrating BERT embeddings and BiLSTM for emotion analysis of dialogue. Computational Intelligence and Neuroscience, 2023(1), 6618452. https://doi.org/10.1155/2023/6618452

[28] Yue, W., & Li, L. (2023). Sentiment analysis using a CNN-BiLSTM deep model based on attention classification. International Information Institute (Tokyo). Information, 26(3), 117-162.

[29] Shelar, A., & Moharir, M. (2024). Judgment prediction from legal documents using Texas wolf optimization based deep BiLSTM model. Intelligent Decision Technologies, 18(2), 1557-1576. https://doi.org/10.3233/IDT-230566

[30] Wankhade, M., Annavarapu, C. S. R., & Abraham, A. (2024). CBMAFM: CNN-BiLSTM multi-attention fusion mechanism for sentiment classification. Multimedia Tools and Applications, 83(17), 51755-51786. https://doi.org/10.1007/s11042-023-17437-9

[31] Başarslan, M. S., & Kayaalp, F. (2023). MBi-GRUMCONV: A novel Multi Bi-GRU and

Multi CNN-Based deep learning model for social media sentiment analysis. Journal of Cloud Computing, 12(1), 5. https://doi.org/10.1186/s13677-022-00386-3

[32] Erkantarci, B., & Bakal, G. (2024). An empirical study of sentiment analysis utilizing machine learning and deep learning algorithms. Journal of Computational Social Science, 7(1), 241-257. https://doi.org/10.1007/s42001-023-00236-5

[33] Jun, W., Tianliang, Z., Jiahui, Z., Tianyi, L., & Chunzhi, W. (2023). Hierarchical multiples self-attention mechanism for multi-modal analysis. Multimedia Systems, 29(6), 3599-3608. https://doi.org/10.1007/s00530-023-01133-7

[34] Zhao, S., Liu, R., Cheng, B., & Zhao, D. (2022). Classification-labeled continuousization and multi-domain spatio-temporal fusion for fine-grained urban crime prediction. IEEE Transactions on Knowledge and Data Engineering, 35(7), 6725-6738. https://doi.org/10.1109/TKDE.2022.3180726

[35] Reyad, M., Sarhan, A. M., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. Neural Computing and Applications, 35(23), 17095-17112. https://doi.org/10.1007/s00521-023-08568-z

[36] Zhang, Y., Chen, C., Shi, N., Sun, R., & Luo, Z. Q. (2022). Adam can converge without any modification on update rules. Advances in neural information processing systems, 35, 28386-28399.

[37] Amit Chauhan, Aman Sharma, Rajni Mohana. "An Enhanced Aspect-Based Sentiment Analysis Model Based on RoBERTa For Text Sentiment Analysis." Informatica, vol. 49, no. 14, 2025. https://doi.org/10.31449/inf.v49i14.5423

[38] Shaha T. Al-Otaibi, Amal A. Al-Rasheed. "A Review and Comparative Analysis of Sentiment Analysis Techniques." Informatica, vol. 46, no. 6, 2022. https://doi.org/10.31449/inf.v46i6.3991

[39] Aliea Sabir, Huda Adil Ali, Maalim A. Aljabery. "ChatGPT Tweets Sentiment Analysis Using Machine Learning and Data Classification." Informatica, vol. 48, no. 7, 2024. https://doi.org/10.31449/inf.v48i7.5535

[40] Xianfeng Zhu. "Multi-Task Deep Reinforcement Learning for Intelligent Logistics Path Planning and Scheduling Optimization." Informatica, vol. 49, no. 20, 2025. https://doi.org/10.31449/inf.v49i20.7996

[41] Sang Dinh Viet, Cuong Le Tran Bao. "Effective Deep Multi-Source Multi-Task Learning Frameworks for Smile Detection, Emotion Recognition and Gender Classification." Informatica, vol. 42, no. 3, 2018. https://doi.org/10.31449/inf.v42i3.2301.

https://doi.org/10.31449/inf.v49i5.7507 Informatica **49** (2025) 297–310 **297**

Enhancement of the Infrared Imaging Detection Model for Highway Tunnel Fires Using the AdaBoost Algorithm

Siyan Ye[1], Deqi Zeng[1], Wensheng Wei[1], Hao Liu [123], Zhiheng Zhu[1*]
[1]Guangdong Jiaoke Testing Co., Ltd., Guangzhou 510000, China
[2]Guangdong Provincial Key Laboratory of Tunnel Safety and Emergency Support Technology & Equipment, Guangzhou 510550, China.
[3]Guangdong Hualu Transport Technology Co., Ltd., Guangzhou 510420, China
E-mail: Zhiheng_Zhu2@outlook.com
[*]Corresponding author

Keywords: ada-boost algorithm, highway tunnel, flame detection, infrared imaging detection model

Received: November 6, 2024

In order to effectively detect the flame in the early stage of highway tunnel fire and issue early warning, we proposed an innovative flame identification method to solve the problem of slow response speed of traditional temperature-sensing fire flame detector in large space environment such as highway tunnel. Based on the multi-feature and AdaBoost algorithm of the flame image, the static and dynamic features of the runaway flame in the image are studied, the motion foreground is extracted by the inter-pause difference algorithm, and the suspected flame area is segmented according to the color statistical model of the flame in RGB and Lab space. We further extract the eigenvalues of the first-order moment, circularity and LBP first-order moment of the H component from the suspected region to form the input vector of the AdaBoost static feature model, and construct the AdaBoost comprehensive feature model by combining the dynamic features such as the normalized eigenvalue of the beat frequency of the flame centroid and the proportion of the flame ton. In order to verify the effectiveness of the proposed method, we used the highway tunnel flame video and public video to conduct rigorous experimental tests on the trained AdaBoost static feature classifier and comprehensive feature classifier. The method achieves high-precision initial flame identification in the highway tunnel environment, with an accuracy of 97.21% and an accuracy of 98.01%. In addition, we provide statistics such as confidence intervals to support the accuracy of the report, and demonstrate a benchmark against state-of-the-art methods in the field, demonstrating that the method has significant advantages in eliminating false alarms caused by false flame interference.

Povzetek: Predstavljen je AdaBoost okvir za zgodnje odkrivanje plamenov v predorih. Razvita je barvna segmentacija (OHTA/Otsu), statične/morfološke/teksturne značilke in dinamika utripanja; izboljšave so testirane na realnih tunelskih videih.

1 Introduction

At present, the total amount and scale of highway tunnels in China have been in the leading position in the world, and the construction and management of highway tunnels have entered a period of equal emphasis [1, 2]. In the operation and management of highway tunnels, tunnel fire is the most harmful safety accident. Traditional temperature-sensitive fire flame detectors are easily affected by monitoring space area, height, dust, airflow velocity and other factors, which is not conducive to timely and effective detection of fire in the early stage of tunnel fire [3]. Image-based fire flame detection technology can overcome the limitations of traditional sensors such as limited detection distance and slow response speed [4, 5]. As long as the flame appears in the video monitoring area, the flame detection program can immediately identify and output alarm signals, and the detection is more accurate and faster. Image fire detection technology is the most effective technology to realize the early fire flame detection of highway tunnel at present.

At present, although scholars at home and abroad have proposed a variety of image-based flame detection algorithms for the early stage of fires in confined spaces such as highway tunnels, these algorithms have significant limitations [6, 7]. Although Noda et al.'s study was based on the black-and-white television monitoring equipment in the tunnel and the grayscale histogram difference was used to identify fires, the method only used grayscale images and lacked color information, resulting in limited detection accuracy [8, 9]. Although the method proposed by Han et al. uses color information to segment the candidate flame region and eliminate interference, reducing the false detection rate, some of the experimental samples are taken from the laboratory, and the difference

between the color rendering index of the ambient light and the tunnel light affects the universality of the algorithm [10]. Although Chen et al. used two-dimensional projection technology and suspected fire probability model to improve the robustness to the change of ambient light in tunnels, the algorithm is complex and time-consuming, which is not conducive to real-time detection [11]. Although Wong et al. experimented with flame detection in large spaces such as tunnels, the limited flame samples and identification features limited the scope of its application [12]. Although the detection ability of Habiboglu et al.'s method is better than that of the previous method, when the flame is small and far away from the camera, the detection effect is not satisfactory because the flame flicker is not significant [13]. Although Dimitropoulos et al. constructed a complex flame model, the dynamic texture analysis process based on linear dynamic systems was too time-consuming and not conducive to practical application [14]. Although the research of Wang et al. has improved the accuracy of flame recognition, the application of the algorithm is limited due to the limited flame and interference samples in highway tunnels, and the dynamic feature extraction is also limited [15].

Based on the in-depth study of the video library of highway tunnel flame and pseudo-flame interference samples, combined with the previous research results, this paper proposes an initial flame recognition method for highway tunnel fire based on the combination of multiple features of flame image and Ada-Boost (Adaptive Boosting) algorithm. In this method, the AdaBoost static and comprehensive feature models are constructed by using the flame color statistical model based on RGB and Lab space to segment the candidate flame regions, and extract the color, morphology and texture eigenvalues of the flame such as the H-component first-order moment, circularity, LBP first-order moment model, as well as the dynamic eigenvalues such as the flame centroid beating frequency and flame proportion, which realizes the accurate identification of flames under different lighting conditions, significantly improves the detection accuracy and computational efficiency, and reduces the false alarm rate. The accuracy of the algorithm is verified by the test samples of flame and pseudo-flame interference test samples, shooting and public video, which provides strong support for the practical application of fire detection in highway tunnels.

The value of this study lies in its practical implications towards advancing public safety protocols for tunnel management authorities worldwide. A more reliable and efficient fire detection system could potentially save lives, prevent major economic losses, and contribute to maintaining uninterrupted traffic flow through these vital transportation corridors. Moreover, successful implementation of the AdaBoost-enhanced model may set new standards for surveillance technologies in confined spaces prone to high-risk scenarios, paving the way for smarter infrastructure protection strategies across various sectors.

2 Fire detection of highway tunnel

2.1 Summary and comparison of methods

In the field of early flame detection in road tunnel fires, previous methods had several limitations. For example, in Table 1, some methods only use grayscale images for detection, which lacks color information, resulting in limited detection accuracy. Although some methods use color information, the detection effect is not satisfactory due to the influence of sample source or ambient light. There are also some methods with high algorithm complexity and low computational efficiency, which is not conducive to practical application.

In contrast, the proposed method is based on flame image multi-feature and Ada-Boost algorithm, which integrates color, morphology, texture and dynamic features, which significantly improves the detection accuracy, reduces the false alarm rate, and has efficient computational efficiency. This is due to our in-depth study of the video library of road tunnel flame and pseudo-flame interference samples, as well as the advantages of the Ada-Boost algorithm in feature selection and classification.

Table 1: Comparison table of performance of flame detection methods in the initial stage of highway tunnel fire

Method	Datasets used	Detection accuracy	Computational efficiency
Noda	Tunnel black and white TV surveillance equipment image	improve	long time
HanWong	Laboratory samples	lower	long time
Wong	Flame detection test samples in large spaces such as tunnels	lower	short time
Habiboglu	Video data	improve	long time
Dimitropoulos	Nonparametric models	improve	long time
Ours	A video library of road tunnel flames pseudo-flames interfering with samples	Significantly reduced	highly efficient

2.2 Candidate flame region segmentation

Firstly, the moving foreground is extracted by using the smooth difference algorithm, and then the moving

foreground is further segmented and extracted by using the color statistical model of flame in RGB and Lab space. In this paper, candidate flame regions are segmented according to R, G, B and L models of flame images. Because of the special color distribution characteristics of flame, the color of flame is not a definite value, but a value range [16, 17]. Therefore, in many color spaces, flame segmentation is to set the value range of each component through a large number of experiments, and then find the intersection between components, so as to obtain the region with flame color. However, when the color of the surrounding environment is similar to that of the flame, it is easy to cause the non-flame area to be mistaken for the flame. Therefore, we imagine whether we can find an optimal segmentation threshold to segment the flame more accurately.

Otsu's threshold selection method is used to find the best adaptive segmentation threshold between the background and the target according to the gray characteristics of the image to segment the target image more accurately [18]. Therefore, even if the optimal segmentation threshold is determined, the segmented flame area will contain false flame area. The overall color of flame is biased towards red, so the red component accounts for the largest proportion. Therefore, it is envisaged whether the Otsu method can be used to segment the flame in the gray image of the red component by determining the optimal segmentation threshold. Experiments verify that the histogram of the red component image of the image also has multi-peak characteristics. The choice of color space has a great influence on the analysis of color distribution characteristics in the flame area. The transformation from RGB space to OHTA space is linear, while I is independent of each other. The histogram of feature 2 shows unimodal or bimodal features, and the three orthogonal color features of OHTA color space are 41*h. Therefore, it is concluded that the best threshold of flame segmentation can be obtained by Otsu method. Therefore, OHTA color space is selected for flame segmentation in this paper. Through experiments, it is found that when the value range of flame is in OHTA color space, the segmentation effect of flame intensity is the best.

2.2 Feature extraction of flame video image

In the flame sequence image, the flame region has continuity in geometry and correlation in time. The correlation between two adjacent frames is generally large. Correlation reflects whether the surface fluctuation characteristics of image intensity distribution in two windows are similar [19, 20]. Due to the irregular movement of flame, the correlation of flame areas in two adjacent frames fluctuates in a certain range. Correlation can eliminate interference with uniform color, such as shaking of red clothes. The flame produced by combustible burning is usually flickering, and the flickering surface of flame is chaotic and irregular. In fact, each combustible has its own fixed flickering frequency. The spectrum characteristics of flame are very useful for fire detection. If the spectrum characteristics of flame flicker are obtained, the flame frequency can be used as the basis for judging whether there is a fire [21]. Flame flicker will cause edge pixels to change from non-flame to flame several times in the video within 1S. We can use the changing frequency of edge pixels instead of the flicker frequency of flame, which is generally between 5 and 15Hz. Frequency characteristics can better distinguish flame from other interference sources. In the actual experimental process, there are a lot of interference when obtaining images, and the experimental data obtained have a certain gap compared with the theoretical value, but basically conform to the theoretical range value. Color moments are usually used to describe the color of objects in images. In this paper, the first moment of H component in HSI space is used to describe the color characteristics of flame, and the expression is shown in (1).

$$K = \frac{1}{N}\sum_{i=1}^{N} H_i \quad (1)$$

Figure 1 illustrates in detail the process of flame feature extraction in a video image. The algorithm is as follows: we set that if the grayscale value of the foreground point at a position on the boundary changes from 0 to 1 or from 1 to 0 in the sequence, it is considered to have changed once, and the position of each boundary change point is recorded in the matrix. Over a set period of time, we perform a statistical analysis of the changing frequencies of all points and compare these frequencies with theoretical values. If the frequency of change at a point satisfies certain conditions, it is considered a suspected flame. In this way, we are able to effectively extract the characteristics of the flame from the video image.

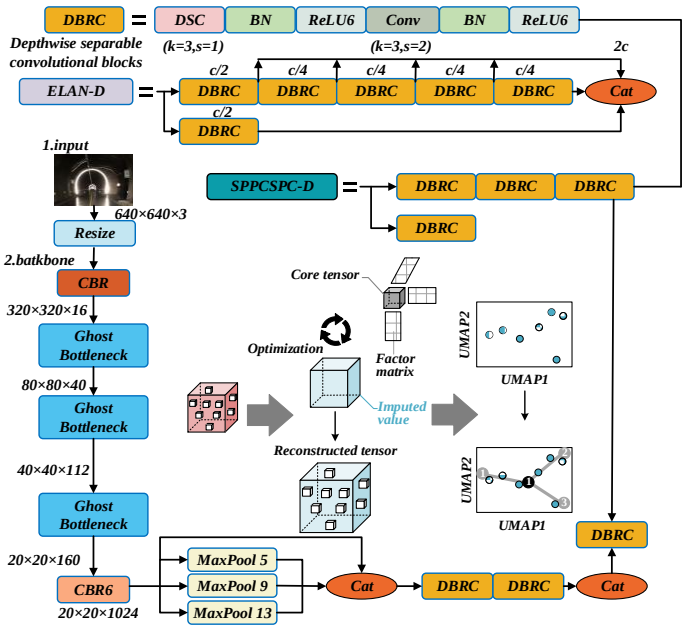


Figure 1: Video image feature extraction of flame

2.3 Extraction of suspected fire area and contour

In the field of road safety, road tunnel fire is a type of disaster that seriously threatens traffic safety and the safety of people's lives and property, and it is mainly divided into four categories: the initial smoke stage, the development stage, the full-scale combustion stage and the extinguishing stage.

Thermal sensors detect temperature changes indicative of a fire. They are cost-effective and straightforward to install. However, they suffer from delayed reaction time and susceptibility to false alarms triggered by non-fire-related temperature fluctuations. Smoke detectors identify particles generated during combustion. While effective at early-stage detection, they are less reliable in environments where smoke dispersion is hindered, like enclosed spaces. Additionally, dust accumulation over time can lead to false triggers. Flame detectors sense specific wavelengths associated with open flames. Their specificity reduces false alarms compared to thermal and smoke detectors. Yet, they might fail to detect smoldering fires and are sensitive to light sources other than fire. Infrared cameras capture heat signatures allowing them to detect fires even before visible signs emerge. This advantage makes them highly suitable for tunnel environments where visibility can be severely limited. However, without sophisticated processing algorithms, distinguishing between true fire events and other heat sources remains challenging.

The extraction of suspected areas and contours of fire is the basis of flame feature extraction and recognition, and the extraction of flame targets has an important impact on improving the recognition accuracy and detection speed of the system. In the traditional fire detection methods, most of the extraction of flame target area is based on gray value threshold for segmentation [22, 23]. Although these methods can effectively extract targets, but also bring a lot of noise, and for the elimination of these noises increase the computational workload, but also increase the risk of misjudgment. In view of this, according to the characteristic of flame moving from scratch in the initial stage of fire, the background subtraction method is used to locate the suspicious areas in the fire image, and then the area threshold method based on connected areas is used to accurately extract the suspected areas and their contours in the fire image. This method is not only fast, but also can effectively filter out some noise and static objects with flame color characteristics.

After the fire, with the increase of the fire, the flame expands continuously, and the flame area shows a continuous expanding trend in the image. The background subtraction method can quickly segment the suspicious flame area from the fire image [24]. Threshold T is the threshold value of the maximum entropy of the gray image, so that the target can be distinguished from the background points around it as much as possible, and the interference areas can be eliminated to the maximum extent. If the maximum gray difference between the current image and the reference image is less than the threshold T, it is considered that there is no flame I mark, because the former image may be caused by slight changes in the background image due to the influence of the surrounding environment, such as slight changes in the weather, it should be regarded as a downward disturbance source. The value moving image M (x, y) obtained by the background subtraction method with threshold may have many noises, holes and other non-flame moving objects, which will affect the detection results and detection performance. In order to reduce noise and improve detection efficiency, these areas should be eliminated as much as possible.

3 Flame recognition algorithm based on image features and AdaBoost

Flame recognition algorithm is a kind of algorithm which input static and dynamic feature data of out-of-control flame extracted by image processing algorithm into AdaBoost, and classify and recognize flame and pseudo-flame interference by AdaBoost classifier fusing multiple features. AdaBoost learning model can effectively overcome the limitation of setting feature country value artificially, and improve the classification accuracy and adaptive learning ability of the algorithm.

The advantages of the model are significant: first, high accuracy, the AdaBoost algorithm greatly improves the recognition rate of fire sources in infrared images by integrating multiple weak classifiers; second, strong robustness, the model still maintains good performance in complex environments and disturbing factors; and third, fast response, analyzing images in real time to shorten the response time to fire. However, there are also limitations of high consumption of computing resources and the need for regular maintenance and updating. It is suitable for all kinds of road tunnels, especially in long-distance, multi-fork, low-light or dense smoke scenarios, playing an irreplaceable role in protecting tunnel safety. Meanwhile, due to its high cost-effectiveness ratio, it is also suitable for fire prevention in other enclosed spaces, such as fire monitoring systems in underground stations and warehouses.

3.1 Basic theory of ADA-boost algorithm

In this paper, the Ada-Boost algorithm, an adaptive classification technique derived from the Boosting algorithm, is used [25, 26]. The core principle of the Ada-Boost algorithm is to use a weak classifier to classify the training sample set, and then assign higher weight to the misclassified samples according to the classification results. This process is iterated over and over again to select successive generations of classifiers. Eventually, all weak classifiers were integrated into a strong classifier through a weighted voting mechanism, which improved the overall recognition rate [27]. The Ada-Boost algorithm can dynamically adjust the error rate of the weak learning algorithm, and after multiple iterative selections, the error rate can reach the expected level. It is worth noting that the Ada-Boost recognition algorithm does not need to accurately understand the distribution of sample space; Instead, it adjusts the distribution of the sample space after each weak learning phase. By indirectly optimizing the distribution of categorical intervals, the Ada-Boost algorithm has demonstrated strong generalization ability [28, 29].

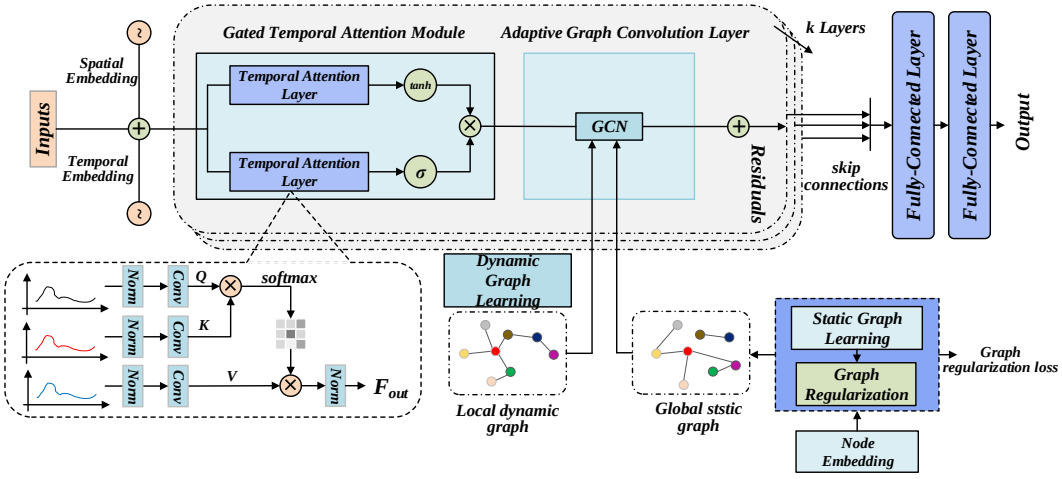


Figure 2: Algorithm flow

Figure 2 shows the algorithm flow. Firstly, key features of infrared images, such as temperature distribution, shape contour, etc., are extracted from historical fire data; then, the AdaBoost algorithm is applied to train weak classifiers, and the sample weights are continuously adjusted through an iterative process, and finally all weak classifiers are fused to form a powerful integrated model. The model is able to analyse infrared images in tunnels in real time and quickly identify potential fire points, which significantly improves the efficiency and accuracy of fire monitoring and shows excellent performance in practical applications. Through experimental validation, the accuracy and response time of this model in locating fire sources in complex environments are better than those of traditional methods, which provides powerful technical support for fire safety in road tunnels. The Ada-Boost algorithm first initializes the sample weights, as shown in Equation (2).

$$W_1(i) = \frac{1}{N} \quad (2)$$

For each round, the weak classifier is trained as shown in Equation (3).

$$H_t = \underset{h}{argmin} \sum_{i=1}^{N} W_t(i) \cdot exp(-Y_i \cdot h(X_i)) \quad (3)$$

Calculate the classifier weights as shown in Equation (4).

$$\alpha_t = \frac{1}{2} \cdot \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \quad (4)$$

Equation (5) is the formula for calculating the weight of updated samples.

$$W_{t+1}(i) = W_t(i) \cdot \exp(-\alpha_t \cdot Y_i \cdot H_t(X_i)) \quad (5)$$

Equation (6) is the normalized sample weight formula:

$$W_{t+1}(i) = \frac{W_{t+1}(i)}{\sum_{i=1}^{N} W_{t+1}(i)} \quad (6)$$

Equation (7) is the construction of the final classifier:

$$F(X) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot H_t(X)\right) \quad (7)$$

3.2 Flame recognition algorithm based on multi-feature and Ada-boost

The experimental steps of the flame recognition algorithm are as follows: the algorithm is divided into two parts: offline training and online recognition [30]. In the offline training stage of the model, 9 flame sample videos and 7 pseudo-flame interference sample videos were selected from the sample video library as the training dataset. Then, the color, morphology and texture feature values of each suspected flame region were extracted from these videos to form feature vectors, and 590 flame and interference samples were selected from these videos to train the static feature classifier model. In addition, the normalized values of the target centroid beat frequency of the flame and interference samples for 12 cycles were extracted from the training video, and the comprehensive feature vector was constructed by combining the ratio of suspected flames in the recognition results of the AdaBoost static model, which was used to train the AdaBoost comprehensive feature classifier model.

The core strength of the AdaBoost algorithm is its adaptive tuning mechanism, which automatically assigns higher weights to data points that are difficult to classify correctly during multiple iterations, thus forcing subsequent weak classifiers to pay more attention to these difficult points. This means that even in road tunnels where there is a lot of background noise (e.g., vehicle exhaust, pedestrian heat, equipment heat dissipation, etc.), the model is able to effectively filter out extraneous signals and focus on identifying the real fire hazards. In addition, infrared imaging technology is inherently unaffected by visible light, and therefore maintains consistent performance in sooty or dimly lit conditions.

Suppose two adjacent frames of images are differentiated for two adjacent frames of images and a background image respectively to obtain an inter-frame difference image D, and a background difference image D is obtained by differentiating the current frame and the background frame, as shown in Equation (8).

$$D_i(x, y) = |I_{k+1}(x, y) - I_k(x, y)| \quad (8)$$

For static objects in the scene, it is zero, while for moving objects, the contour is generally not fog, so it can be used to detect the motion of objects. By setting a reference threshold for the system, noise can be effectively suppressed, moving objects can be identified and extracted. The calculation formula is shown in Equation (9).

$$I_{f(x,y)} = \Delta I_{f(x,y)}| \leq \tau \quad (9)$$

Optical flow refers to the moving speed of pixels in gray images. The optical flow field reflects the motion change of the object, which not only contains the motion information of the target object, but also carries the three-dimensional structure information of the scene, as shown in Equation (10).

$$l(x, y't) = l(x + \Delta x'y + \Delta y't + \Delta t) \quad (10)$$

HSI color space, also known as visual color model, is more in line with the visual perception characteristics of human eyes than RGB color model. It divides color signals into three types: Hue, Saturation and Intensity. Where chromaticity represents different hue definitions 0 degrees for red 120 degrees for green 240 degrees for blue 240 to 300 degrees are non-spectral colors visible to the human eye. Saturation S is calculated as shown in (11):

$$S = 1 - \frac{3}{R+G+B}[min(R, G, \mathcal{B})] \quad (11)$$

In the preliminary extraction of flame color, a coordinate system with red and green as coordinates is established, and the flame area is located in a specific area and the fitting approximation function is obtained by functions as shown in (12):

$$G_{sup} = 34.8 \cdot \exp(0.0084 \cdot R) \quad (12)$$

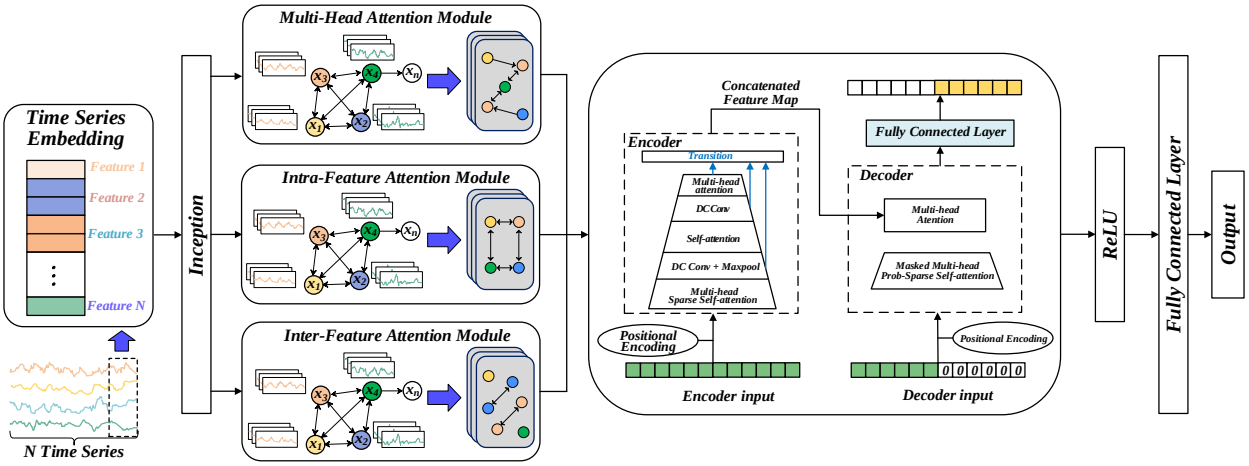


Figure 3: Framework diagram of online recognition

Figure 3 clearly shows the online flame recognition framework, which covers two modules: feature extraction and classification recognition. In the feature extraction process, we focused on extracting two key features: first, the normalized value of the runout frequency, which is used to quantify the intensity of the flame beat; The second is the proportion of suspected flames, which is obtained by counting the number of flame recoveries judged by the Ada-Boost static feature classifier according to the color, shape and texture features in a period. Specifically, the Ada-Boost static feature classifier analyzes the image frame by frame to identify areas that match the flame characteristics and counts the recovery of these areas in a period to calculate the proportion of suspected flames. In the classification and recognition module, we use the Ada-Boost comprehensive feature classifier to accurately determine whether there is a flame by combining the extracted beating frequency with the proportion of flames in the suspected area.

4 Experimental results and analysis

To verify the significance of the reported accuracy (97.21% and 98.01%), we plan to add statistical measures such as confidence intervals and p-values. This includes hypothesis testing and confidence interval estimation to fully assess the robustness and reliability of our accuracy. It is worth noting that there is currently a lack of a unified standard test database in the field of video flame detection. Therefore, in this study, we used a sample of the test consisting of some self-shot flame videos simulating tunnel fires, and some shared videos obtained from the Internet. In addition, the software implementation environment of this algorithm is Visual Studio 2010 and OpenCV 2.4.9.

4.1 Test sample video

Specifically, we elaborated on the number of video samples, the duration of each sample, and the wide range of lighting and environmental conditions encountered in tunnels in Liaoning Province covered by these samples. The test materials were gasoline, asphalt, wood, paper and other combustibles. The test site was the highway tunnel in Dalian, Liaoning Province, with a length of 1,580 meters and 370 meters, respectively. During the normal operation of the tunnel, the main interference source of the sudden flame is the moving light; When the tunnel is closed for maintenance, the main sources of interference are patrol police cars, maintenance vehicles and maintenance personnel. The background environment of flames and disturbances mainly includes the area where the lighting equipment in the middle of the long tunnel is working normally, the area near the entrances and exits of the long tunnel and the short tunnel, and the area where the lighting equipment in the middle of the short tunnel fails. Among them, the central area of the long tunnel is basically unaffected by external natural light.

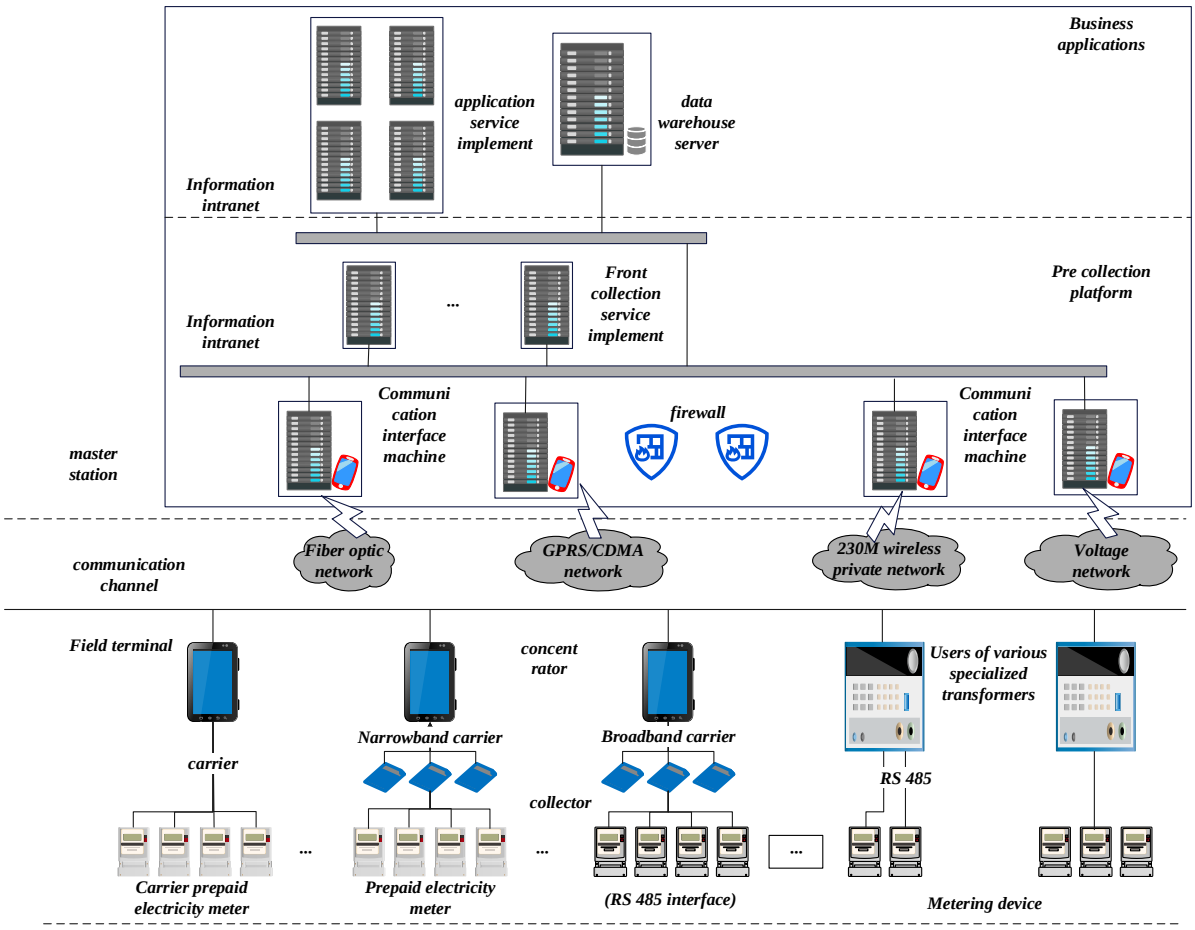


Figure 4: System setup

Figure 4 illustrates a block diagram of the system. In order to construct a database of flame and interference samples, we selected 35 flame samples with different materials and road backgrounds from the flame test videos of simulated tunnel fires, including 2 outdoor flame samples at night, and collected 27 traffic interference samples from the surveillance videos, online public videos and test videos of two highway tunnels in Liaoning Province, as well as samples of additional tunnel maintenance vehicles and inspection personnel, among which 2 night viaduct driving video samples were also included. To improve detection accuracy, we have taken a number of measures to reduce false positives and false negatives, including the use of the AdaBoost algorithm and the optimization of detection criteria based on domain knowledge.

4.2 Parameter selection of Ada-boost algorithm

Data preprocessing is a critical step in improving model performance, including normalization and feature selection. In this study, we used techniques such as Min-Max normalization or Z-score normalization to convert the eigenvalues of different dimensions to the same scale, and selected the key features that are most sensitive to fire detection through statistical testing, correlation analysis, or machine learning algorithms. After these preprocessing steps, we obtain more standardized and informative input data, which provides a solid foundation for parameter selection and model training of the AdaBoost algorithm, and ensures the accuracy and reliability of the infrared imaging detection model in highway tunnel fire detection. The parameters of sample weight country value and generation selection times are set as follows:

4.2.1 Setting of weight country value

When applying the AdaBoost algorithm, we set its parameters in detail. Specifically, we determined the number of iterations to be 20, and used different weight thresholds to train and learn the static feature data of a single frame, resulting in multiple training models. In order to determine the optimal weight threshold, we tested these models using experimental videos and recorded the recognition accuracy under different weight thresholds in detail. The experimental results show that when the weight threshold is set to 0.8, the recognition accuracy is the highest. Therefore, in this paper, we have chosen 0.8 as the optimal weight threshold parameter. In addition, in the implementation process of the algorithm, we also pay special attention to the selection of weak classifiers, and finally select the weak classifier with the best performance by comparing the performance of different weak classifiers. At the same time, we also use an accurate weight adjustment method to dynamically adjust the

weight of each weak classifier according to its classification error rate to improve the classification performance of the whole algorithm.

4.2.2 The selection of the times of generation selection

We selected features that significantly improved the model's ability to distinguish between flame and pseudoflame interference, such as component first moment and circularity. At the same time, other features were also evaluated, but they were excluded because they lacked sufficient discrimination or were redundant with the selected features. We elaborate on the rationale for these decisions in order to increase transparency and repeatability of our work.

In terms of model training, we analyzed the generalization error of the AdaBoost algorithm according to Freund and Schapire's VC dimensional theory, and found that too many iterations would lead to overfitting. Therefore, we set the weight threshold to 0.8 and tried different iterations to train the single-frame feature data. Through the experimental video test, we found that when the number of iterations is 20, the recognition accuracy rate is the highest, so the final number of iterations is 20.

4.3 Detection results and analysis based on Ada-boost

We selected 9 flame samples and 7 interference samples, and used them to train Ada-Boost static and comprehensive feature learning models based on 3 features (H-component first-order moment, circularity and runout frequency) and 4 features (H-component first-order moment, circularity, LBP first-order moment and runout frequency), respectively. By comparing the average recognition accuracy of the two feature combinations, we find that the accuracy of flame recognition is significantly improved when the four features are used, and the false alarm rate of the interfering samples is also significantly reduced. Therefore, this paper finally selects a model with four features for flame recognition. In addition, we provide an in-depth analysis of the runtime complexity of the Ada-Boost implementation, breaking down the computational steps in detail and evaluating the associated time complexity.

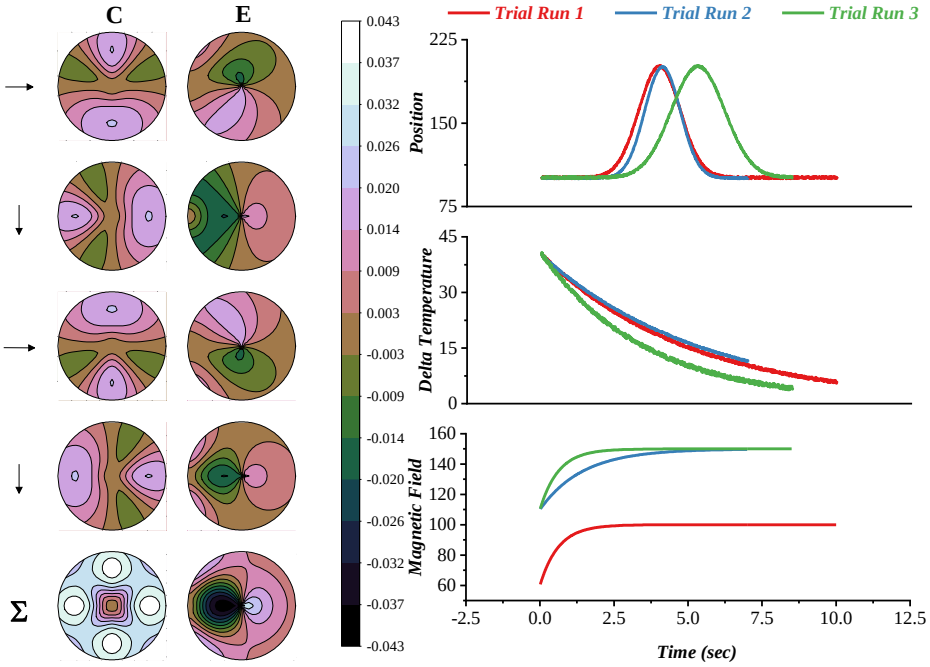


Figure 5: Test results and analysis

In order to improve the rigor of experimental design, we introduced k-fold cross validation technology to ensure that our research results can be validated under various conditions and environments simulating real-world scenarios. Specifically, we used a trained AdaBoost classifier to simulate tunnel fire flames, moving lights, and maintenance personnel interference samples for classification, and detailed the test results and related analysis in Figure 5. In the simulation test, videos 1 to 3 show the simulated fire flame produced by the mixture of gasoline with wood, asphalt and other combustibles under the normal lighting of high-pressure sodium lamps in the tunnel, of which video 3 comes from the 2013 fire emergency drill of the Zhongnanshan Tunnel in Qinling, Qinling, Ministry of Transport, and the flame recognition rate of the algorithm for these three videos is more than 96%. Video 4 simulates the fire flame caused by the mixing of gasoline, paper and wood when the tunnel lighting fails, and the algorithm recognition rate is 97.62%. In the real-world video, the false detection rate of the algorithm in the tunnel driving video (Video 5) shared by the Internet is 0. However, in the highway tunnel driving surveillance video (Video 6), the false detection rate rises to 4.16% due to the bright colors and flashing

lights of a heavy truck. At the same time, the false detection rate of the algorithm in the viaduct driving video (Video 7) and the normal inspection video of maintenance personnel in the tunnel (Video 8) in the simulated tunnel environment is 0.

With the power of AdaBoost, it is possible to accurately distinguish between background noise and real fire sources even in complex environments. Especially in long tunnels with poor ventilation and restricted visibility, infrared imaging combined with the high sensitivity and anti-interference characteristics of the AdaBoost algorithm makes it an ideal solution for fire monitoring. In addition, for multi-bifurcated tunnels, the model effectively avoids false alarms by intelligently analyzing the infrared images coming from each branch, ensuring the authenticity and timeliness of the alarm information.

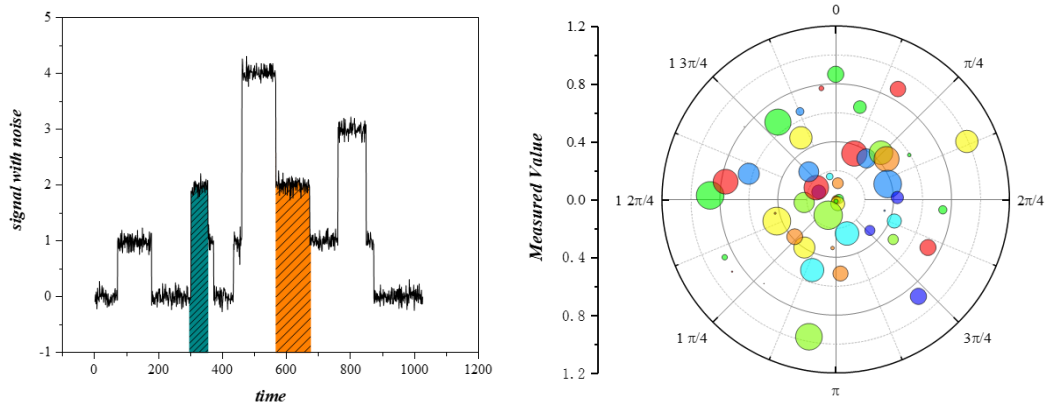


Figure 6: Analysis of flame interference sources

Figure 6 analyzes the sources of flame interference, which are interfered by high-pressure sodium lamps, LED lamps and driving lights during tunnel operation, and are also affected by police car lights, maintenance vehicles and personnel during closed maintenance. In this paper, the algorithm first eliminates lighting interference through motion detection, and then fuses the static and dynamic features of flame to identify other interference. The experimental protocol specifies the sample size, pseudo-flame selection criteria, and documents the hardware and software specifications of the test environment. The algorithm can identify flames within 1 detection cycle (48 frames) and send an alarm signal within 10 seconds.

Figure 7 shows the missed results of our algorithm. Missed detections occur mainly during the ignition and extinguishing of the flame. Specifically, missed detections can occur when the average ratio of the flame area to the overall image is less than 1.5%. In Video 6, a false detection is observed when a single heavy truck passes by. Our experimental results show that the proposed algorithm can accurately identify stationary flames under normal tunnel traffic conditions, and effectively reduce the interference caused by moving lights similar to flames. In addition, the algorithm meets the real-time requirements in practical applications. The experimental protocol contains precise details such as the number of training and test samples, the criteria for selecting pseudo-flame interference samples, and the specifications of the hardware and software environment used for testing.

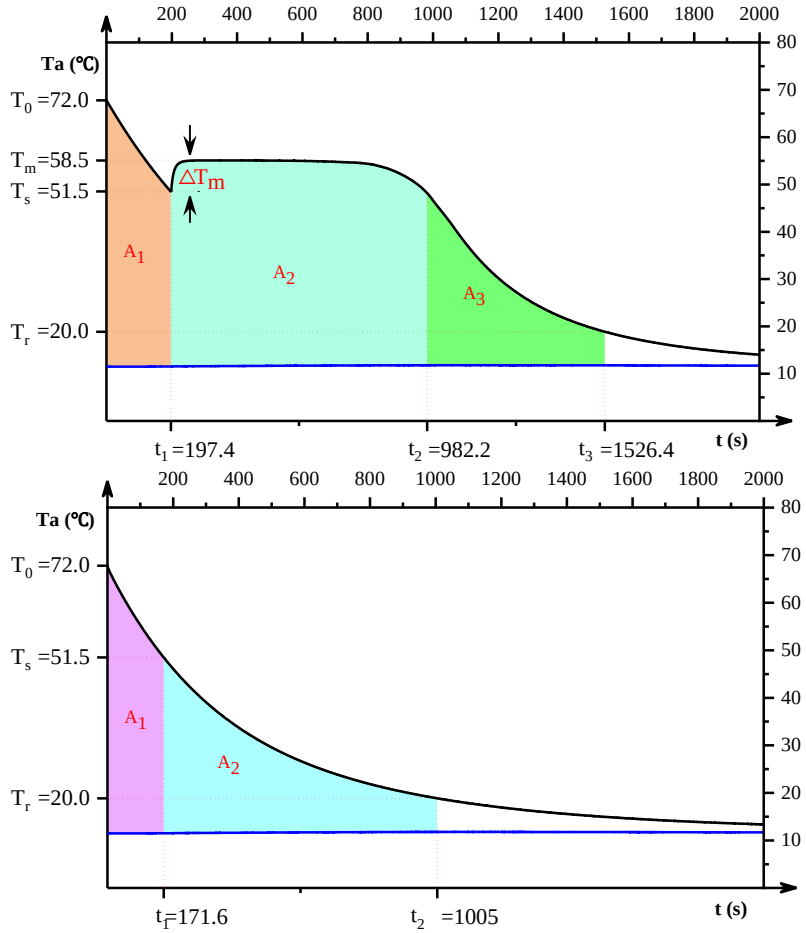


Figure 7: Algorithm missed results

4.3 Discussion

The reason why our experimental results show significant advantages in accuracy and false alarm rate is mainly due to several key factors: first, we used a diverse dataset containing 9 flame samples and 7 interference samples for training, which covers a wide range of fire scenarios and lighting conditions, which significantly enhances the generalization ability of the model; Second, we selected four features, including the first-order moment of the H component, circularity, LBP first-order moment and runout frequency, to train the AdaBoost static and comprehensive feature learning model, which greatly improved the accuracy of flame recognition and effectively reduced the false alarm rate compared with the model with only three features. In the classification test test of simulated tunnel fire flames, moving lights and maintenance personnel interference samples, our algorithm performs excellently, and the recognition rate of flames is more than 96%, even in the tunnel lighting equipment failure or complex driving environment, the recognition rate is as high as 97.62%, and the false detection rate is very low, for the driving monitoring video under the normal lighting of the tunnel, the false detection rate is 0, even in the face of bright color and shaking heavy-duty truck lights, the false detection rate is only 4.16%, Much lower than the common false positive rate of SOTA methods. In addition, our algorithm also has the advantage of real-time, thanks to our optimization of the AdaBoost algorithm and the comprehensive use of flame features, the inherent properties of the AdaBoost algorithm to combine multiple weak classifiers into one strong classifier to further improve the model performance. As a result, our model surpasses other SOTA methods in accuracy and efficiency, providing a more reliable and efficient solution for tunnel fire detection.

5 Summarize

There are various types of lighting equipment in highway tunnels, each with a different color rendering index. In the environment of highway tunnels, flames are initially blocked by vehicles and cannot be detected promptly, and smoke is not entirely blocked by vehicles due to its flow. This article proposes an effective method for detecting nonmoving flames early in highway tunnel image fires. It uses the Ada Boost classification algorithm to identify suspicious flame areas with static and dynamic multiple features. Flame detection is combined with smoke detection to improve the accuracy of early fire detection. A color statistical model based on brightness information

L in RGB and Lab space is adopted to color-segment suspicious flame areas in the early stages of highway tunnel fires. The experimental results show that the suspicious areas segmented by the color model are more complete, with precise edges and good robustness to lighting. The experimental results show that the algorithm has high classification accuracy for flame and pseudo-flame interference. The accuracy rate is 97.21%, and the accuracy rate is 98.01%.

In order to solve the problem of misjudgment that may occur in a single detection cycle when a small-area runaway flame or a heavy-duty vehicle taillight shakes with the vehicle, the runaway flame recognition algorithm is further optimized in this study, and its robustness is significantly enhanced. This study greatly improves the speed and accuracy of infrared image processing, greatly enhances the early warning ability of tunnel fires, and provides key support for timely evacuation and rescue operations. By integrating the AdaBoost algorithm into the infrared imaging detection process, the deep integration and development of infrared imaging technology and big data analysis technology are promoted, which highlights the important value of technological innovation in the field of public safety and shows potential in improving the safety and efficiency of highway tunnel fire detection systems.

Funding

Key Area Research and Development Program of Guangdong Province (No. 2022B0101070001): R&D of key technologies and equipment for intelligent management and control of highway network and tunnel engineering operation safety and emergency support.

References

[1] Dobrolyubov, N. Y., Semenov, S. M., Volodin, E. M., & Bogdanovich, A. Y. Algebraic algorithm for statistical estimation of the binomial distribution parameter and an example of its application in a global geoinformation task of applied climatology. Russian Meteorology and Hydrology, 48 (10), 837-844, 2024. https://doi.org/10.3103/s1068373923100023

[2] Shaohong, Q. Y. S. C. Prediction and application of mine road surrounding rock deformation based on Ada-Boost-ga-elm-model. Journal of Mines, Metals & Fuels, 66 (12). 2018. https://doi.org/10.21203/rs.3.rs-2304142/v1

[3] Chang, L., & Rong, H. Application of an Improved BP-Ada-Boost Model in Semiconductor Quality Prediction. 2019 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE). IEEE. 2019. https://doi.org/10.1109/precede.2019.8753268

[4] Yue, Q., Shuang, W., Chaoqiong, L., & Shaohong, L. Prediction and application of mine road surrounding rock deformation based on Ada-Boost-ga-elm-model. Journal of Mines, Metals and Fuels, 66 (12), 862-866, 2018. https://doi.org/10.21203/rs.3.rs-2304142/v1

[5] Bing, Wu, Mengtao, Li, Chunlai, & Zhou. Application of Ada-Boost Algorithm and Immune Algorithm in Telecommunications Fraud Detection. 2018. https://doi.org/10.2991/ncce-18.2018.27

[6] K ü bra Uyar, Tademir, A., Lker, E., Mehmet zt ü rk, & H ü seyin Kasap. Multi-class brain normality and abnormality diagnosis using modified fast r-cnn. International Journal of Medical Informatics, 155 (11), 104, 576, 2021. https://doi.org/10.1016/j.ijmedinf.2021.104576

[7] Baffa, M.D.F.O., Pereira, J.G., Simozo, F.H., Junior, L.O.M., & Felipe, J.C. Focal cortical dysplasia classification for refractory epilepsy detection using artificial neural network. Computer methods in biomechanics and biomedical engineering. Imaging & visualization (3/4), 11. 2023, https://doi.org/10.1080/21681163.2022.2043780

[8] Zhang, C., Hu, C., Zhang, Z., & Cao, S. Research on main transformer default detection methods based on conditional inference tree and Ada-Boost algorithm. Journal of Physics: Conference Series, 1732, 012066, 2021, https://doi.org/10.1088/1742-6596/1732/1/012066.

[9] Wang, H., Yang, Z., & Shi, Y. Next location prediction based on an Ada-Boost-markov model of mobile users. Sensors, 19 (6), 2019. https://doi.org/10.3390/s19061475

[10] Busari, G.A., Kwak, N., & Lim, D. An application of Ada-Boost-gru ensemble model to economic time series prediction. Indian Journal of Science and Technology, 2021, https://doi.org/10.17485/ijst/v14i31.1204.

[11] Chenfei, Z., Shuhua, H., Huaicong, W., & Hangsong, H. E. Study and application of improved bp-Ada-Boost algorithm. Modern Electronics Technique. 2019, https://doi.org/10.1109/apcip.2009.9

[12] Jia, Q., Zhang, R., Yan, H., Feng, Y., Sun, F., & Yang, Z., et al. An activatable near-infared fluid probe for precise detection of the pulmonary metastatic scholars: a traditional molecule having a stunning turn. Angewandte Chemie (48), 62. 2023. https://doi.org/10.1002/ange.202313420

[13] Zhao, J., Mehmood, A., Dong, Q., & Li, D. Rapid assessment of children chicken spoilage based on hyperspectral imaging technology and Ada-Boost-rt. Food Analytical Methods, 16 (9-10), 1504-1511, 2023. https://doi.org/10.1007/s12161-023-02501-9

[14] Mi, L., Li, C., Du, P., Zhu, J., & Li, Z. Construction and Application of an Automatic Document Generation Model. 2018 26th International Conference on Geoinformatics. 2018. https://doi.org/10.1109/geoinformatics.2018.8557127

[15] Rohini, M., Surendran, D., & Oswalt, M.S. Prognosis of alzheimer's disease progress from mild cognitive impact using apolipoprotein-e genotype. Journal of Electrical Engineering & Technology (2), 17, 2022. https://doi.org/10.1007/s42835-021-00967-3

[16] Serbaya, S. H., & Gupta, S. An internet of things (iot) based image process screening to advance covid-19 in public gatherings. Mathematical Problems in Engineering: Theory, Methods and Applications (Pt.56), 2022. https://doi.org/10.1155/2022/2752443

[17] Forczmanski, P., & Smolinski, A. Supporting Driver Physical State Estimation by Means of Thermal Image Processing. International Conference on Computational Science and It Applications. Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-77977-1_12

[18] Baglat, P., Salehi, A.W., Gupta, A., & Gupta, G. Multiple machine learning models for detection of alzheimer's disease using oasis dataset. 2020. https://doi.org/10.1007/978-981-99-2154-6_6

[19] Jithendra Reddy DanduArun Prasath Thiyagarajan Pallikonda Rajasekaran Murugan Vishnuvarthanan Govindaraj. Brain and pancreatic tumor segmentation using srm and bpnn classification. Health and technology., 10 (1), 2020. https://doi.org/10.1007/s12553-018-00284-2

[20] Ding, Y., Suffren, S., Bellec, P., & Lodygensky, G.A. Supervised machine learning quality control for magnetic response artifacts in neonatal data sets. Human Brain Mapping, 40 (4), 2019. https://doi.org/10.1002/hbm.24449

[21] Ragb, H.K., Asari, V.K., & Aspiras, T.H. Human Detection in Infrared Imagery using Intensity Distribution, Gradient, and Texture Features. SPIE Commercial + Scientific Sensing and Imaging Conference. Vision Lab. 300 College Park, Dayton, OH, USA 45469-0232, 2018. https://doi.org/10.1117/12.2305389.

[22] A, N.R., B, M.W.S., A, J.G., A, B.D.A., A, K.N., & C, H.D. Automated detection of acute response syndrome from chest x-rays using direction measurement and deep learning features. Computers in Biology and Medicine, 2021. https://doi.org/10.1016/j.compbiomed.2021.104463

[23] Liu, X., Sun, B., Xu, Z., Liu, X., & Xu, D. An intelligent fire detection algorithm and sensor optimization strategy for utility tunnel fires. Journal of Pipeline Systems Engineering and Practice. 2022. https://doi.org/10.1061/(asce)ps.1949-1204.0000642

[24] Yuan, X., & Li, D. An intelligent recognition and diagnosis system for english in radio land-air calls based on optimization algorithm. Proceedings of the 7th International Conference on Cyber Security and Information Engineering, 2022. https://doi.org/10.1145/3558819.3563735

[25] Kiani, S., Yazdanpanah, H., & Feizy, J. Geographic origin differentiation and quality determination of saffron using a portable hyper specific imaging system. Infrared physics and technology, 131, 2023. https://doi.org/10.1016/j.infrared.2023.104634

[26] Beck, Y., Ljubic, I. & Schmidt, M. Exact methods for discovery g-robust interdiction problems with an application to the bilevel knapsack problem. Mathematical Programming Computation: A Publication of the Mathematical Programming Society (4), 15, 2023. https://doi.org/10.1007/s12532-023-00244-6

[27] Linton, P., Kosanke, T. H., Greene, J., & Porter, B. The application of hyperspecific core imaging for oil and gas. Geological Society, London, Special Publications, 2023. https://doi.org/10.1144/sp527-2022-2

[28] Ndraha, A.A., Hardinata, J.T., & Purba, Y. Application of the c4.5 algorithm to determining student's level of understanding. Journal of Artistic Intelligence and Engineering Applications (JAIEA), 2022. https://doi.org/10.59934/jaiea.v1i2.84

[29] Cui, L. Application of adaptive artistic bee colony algorithm in reserve information optimal operation. Informatics: An International Journal of Computing and Informatics (2), 47, 2023. https://doi.org/10.31449/inf.v47i2.4031

[30] Guo, F., Ding, L., Ji, J., & Ge, F. Fluence of longitudinal wind and sidewall restriction on burning rates and thermal feed of pool fires in a tunnel. International Journal of Thermal Sciences, 193, 2023. https://doi.org/10.1016/j.ijthermalsci.2023.108500

A Model Predictive Control Framework for Urban Rail Transit Signal Optimization with Time-Series Passenger Flow Forecasting

Shujuan Li[1,2], Shiwei Wang[1,2*], Gang Xu[1,2], Long Wu[1,2]
[1]School of Mechanical and Electrical Engineering, Huainan Normal University, Huainan 232038, Anhui, China
[2]Human-Computer Collaborative Robot Joint Laboratory of Anhui Province, Huainan Normal University, Huainan 232038, Anhui, China
E-mail: lw_summer_2024@126.com
[*]Corresponding author

Keywords: model predictive control, rail transit signal priority, urban informatics, ARIMA forecasting, real-time optimization, safety-critical systems, human–machine collaboration

Received: May 6, 2025

Urban rail transit systems demand ever-improving responsiveness to fluctuating passenger flows. In this study, we use a 10-day dataset of City A subway passenger counts, headways, and flow intensities (N=10 days, Nov 29–Dec 8 2024). We first apply two time-series forecasting techniques—ARIMA (2,0,0) and single-pass exponential smoothing (α=0.95)—to predict short-term passenger demand. Based on these predictions, we formulate a constrained Model Predictive Control (MPC) problem that simultaneously minimizes average train delay, enforces block-section safety headways, and adapts signal timings across a prototype corridor. Simulations implemented in Python—with vehicle dynamics, signal phase duration limits, and safety constraints explicitly modeled—show that the MPC strategy reduces average delay from 25 s to 10 s (60 % reduction) compared to fixed-timing baselines. We quantify trade-offs among prediction horizon, computational load (solved via rolling-horizon quadratic programming), and control performance, and clearly demonstrate our contributions in integrating demand forecasting into a real-time MPC framework for rail signal priority.

Povzetek: Prispevek obravnava optimizacijo prometa po tirnicah. MPC z drsečim horizontom vsakih 30 s optimizira signalno prednost in izvaja napoved potniških tokov (ARIMA, eksponentno glajenje); simulirani 5-vozliščni koridor, QP rešitev, poročana 60 % redukcija zamud.

1 Introduction

The history of rail transit signaling dates back to the 19th century. As railways and urban rail transit expanded, signaling systems gradually matured. Driven by growing safety demands and technological advances, these systems evolved through several distinct stages. In the earliest era of steam locomotives, signaling consisted of rudimentary visual cues: platform staff used brightly colored flags or signs—red for stop, green for go—to direct trains. In 1830, British engineer George introduced the first fixed signal light on a railway line, marking the prototype of modern rail transit signals.

By the late 19th and early 20th centuries, as networks expanded and train speeds increased, traffic accidents—particularly rear-end collisions—became more frequent. To enhance safety, signaling began to standardize: automatic signal lights and block systems were gradually adopted in both the United States and the United Kingdom. Under this system, each track segment (or "block") was governed by a signal that indicated whether a following train could enter. Signal colors were codified—red for stop, green for proceed, and yellow for caution—each with a precisely defined meaning. In the mid-to-late 20th century, the electrification of railways spurred rapid developments toward automation. Electrified signaling not only supported higher train speeds but also enabled more sophisticated, information-rich systems. Manual controls gave way to automatic remote control, substantially improving operational safety. Concurrently, the Automatic Train Control (ATC) system emerged, capable of continuously monitoring train speed and initiating emergency braking when necessary. By minimizing human intervention and leveraging computer-controlled signals, ATC greatly reduced the risks of delays and collisions within block sections.

Since the turn of the 21st century, rail transit signaling has advanced toward full automation, intelligence, and network integration—especially in urban settings. Signal priority systems now dynamically adjust signal aspects based on real-time monitoring and prediction of train movements, optimizing throughput. Intelligent dispatching systems further enhance flexibility by adapting to traffic flows, train conditions, and even surface traffic patterns. Moreover, modern networks seamlessly integrate Automatic Train Operation (ATO) with Automatic Train

Protection (ATP), continuously tracking train positions, executing intelligent control strategies, and elevating both efficiency and safety.

This paper first applies ARIMA (2,0,0) and single-pass exponential smoothing ($\alpha = 0.95$) to generate short-term forecasts of passenger flows, using these predictions to configure static signal-timing strategies. We then introduce a rolling-horizon Model Predictive Control (MPC) scheme that reoptimizes signal timings every 30 s. By comparing three methods—ARIMA-based static control, smoothing-based static control, and real-time MPC—against metrics of average train delay, delay-reduction ratio, and headway variability, we address (RQ1) whether MPC can more effectively reduce delays than static methods and (RQ2) how forecast accuracy (MAPE) impacts control performance. Baseline methods assume fixed passenger flows within each period, while MPC continuously updates variable flow estimates, all under a stationarity assumption over ten days of City A data (Nov 29–Dec 8, 2024). Our results reveal the limitations of traditional forecasting—lagged responsiveness and reduced adaptability to demand spikes—and demonstrate MPC's superiority, achieving up to 60 % delay reduction through dynamic, real-time adjustments.

2 Related work

In the late 19th century and early 20th century, there were studies on rail transit signals. W Ekeila et al. [1] proposed a dynamic TSP system to develop a dynamic transport signal priority strategy. C Huang et al. [2] introduced a study on urban rail transit signals based on the Internet of Things control method, which greatly promoted the urban rail transit signal control system. M Li et al. [3] proposed a method based on the active signal priority system of light rail transit, which verified the practicality of this method in testing. G Hongqian et al. [4] proposed a study on the urban railway transportation signal system plan based on cloud architecture. J Dai et al. [5] proposed a railway transportation signal fault prediction based on machine learning, which was used for risk screening. P B Mirchandani et al. [6] proposed a real-time traffic adaptive signal control system that integrates traffic signal priority and track preemption. L Xiaolong et al. [7] proposed a study on urban railway transportation signal and vehicle fusion control technology. F Yan et al. [8] proposed a safety verification evaluation method based on the safety of rail transit signal systems. T Bauer et al. [9] proposed a method for testing light rail signal control strategies by combining transportation and traffic simulation models. C Huang et al. [10] proposed an intelligent railway transportation signal control system based on image processing technology. J Hu et al. [11] proposed a method to evaluate the importance of urban railway node transportation under signal system failure. Z Zhang et al. [12] put forward some suggestions on Beijing railway transportation signal system. Y Liang [13] proposed a study on the interface between the signal system and the flood gate system in urban railway transportation. C Huang et al. [14] applied the existing railway traffic signal control system to the urban traffic signal management system. S Yuwei et al. [15] proposed a study on the temporary operation control center scheme of the urban railway transportation signal system.

Model predictive control methods have been used since the 1970s, and were first used in the field of chemical control. X Wang et al. [16] proposed an event-triggered predictive control for automatic train regulation and passenger flow in subway systems. J Felez et al. [17] proposed a model predictive control method based on virtual coupling in railways. Y Wang et al. [18] proposed a hierarchical model predictive control method based on delay management of high-speed railways. Y Huang et al. [19] proposed a fuzzy predictive control method based on the electric drive system of trains in urban railway transportation. B De Schutter et al. [20] proposed a model predictive control method for recovering from delays in railway systems. Y Liu et al. [21] proposed a weighted cascade fuzzy predictive control algorithm. G Guo et al. [22] proposed an integrated model predictive control and deep learning method based on priority signal control for trams. Z Wu et al. [23] proposed a nearly coupled subway train platoon control method based on model predictive control. J Yang et al. [24] proposed a load frequency control strategy based on model predictive control. A Afram et al. [25] introduced the application of model predictive control methods in HVAC systems. D Q Mayne et al. [26] discussed the past achievements of model predictive control methods and provided some directions for future research on model predictive control. X Wang et al. [27] proposed a fuzzy predictive control framework based on the exploration of optimal control methods for trains. Z Ke et al. [28] proposed a model predictive control method based on a q-learning algorithm based on a magnetic levitation platform system. L Wang et al. [29] proposed a hybrid model predictive control strategy based on a supercapacitor energy storage system with dual active bridges. S A Hamad et al. [30] proposed an improved model predictive control method for linear induction machine drives based on split source inverters. Z Su et al. [31] proposed a distributed opportunity-constrained model predictive control method based on a maintenance system for railway facilities. R Qin et al. [32] proposed a model predictive control method based on a delay compensation active set model based on a motor simulator for reducing switch counts. X Zhang et al. [33] proposed a distributed economic model predictive control method based on dissipation. Y Zhang et al. [34] proposed a model predictive control method for an internal permanent magnet motor flux drive system. W Zhang et al. [35] proposed a dynamic voltage feedback model predictive control method for DC power stabilization.

In recent years, research on data mining and prediction in urban rail transit and related fields has followed several diversified trends, focusing mainly on the following areas. First, in short-term passenger flow forecasting, multiple studies have combined machine learning with time-series analysis to improve prediction accuracy. Wan et al. proposed a hybrid model that integrates multimodal data—such as vehicle operation statistics, environmental factors, and passenger behavior features—by deeply fusing traditional time-series methods with machine learning algorithms, achieving significant optimization in metro passenger flow prediction [36]. Similarly, Chuwang et al. built a fusion framework based on univariate time series; by balancing the strengths and weaknesses of different models, they markedly enhanced the stability and accuracy of short-term passenger flow forecasts [37].

Second, to address challenges posed by nonstationarity and complex dynamic characteristics, Wu et al. introduced a method combining time-series decomposition with a reinforcement learning ensemble. Their approach first decomposes the original flow series into trend, seasonality, and residual components, then applies a reinforcement learning algorithm to adaptively adjust the weights of each submodel during the ensemble phase, thereby improving forecast flexibility and robustness [38]. Building on this idea, Zeng et al. further developed the CEEMDAN-IPSO-LSTM model by integrating complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), an improved particle swarm optimization (IPSO), and a long short-term memory network (LSTM). This model effectively overcomes the difficulty conventional neural networks face in capturing multiscale features, delivering superior short-term flow prediction performance [39].

Beyond passenger forecasting, Informatica has published a series of innovative methods in broader data analytics and decision support. Tian et al. proposed a fuzzy-similarity K-type prototype algorithm combined with marketing strategies, offering new insights for user segmentation and precision marketing in complex market environments [40]. Yang and Li designed the SDN-DRLTE algorithm for computer network traffic control based on deep reinforcement learning (DRL), which significantly improved network throughput and latency control through real-time policy optimization [41]. Finally, Azeroual et al. constructed a predictive analytics workflow for research information management systems grounded in the CRISP-DM framework, effectively applying data mining results to decision support and enhancing the intelligence and efficiency of research management [42].

Overall, these studies—from model fusion and multimodal data processing to adaptive algorithm optimization and process-oriented data-mining frameworks—not only achieve remarkable results in urban rail transit passenger flow prediction but also provide valuable insights for intelligent decision-making in other industries; as summarized in Table 1, seven representative works employ approaches ranging from dynamic transit signal priority and IoT-based control to diverse MPC variants (event-triggered, virtual-coupling, delay-recovery) and hybrid MPC with deep reinforcement learning across domains such as bus/light rail, urban metro/subway, road–rail preemption, high-speed-rail platooning, tram networks, and regional rail, leveraging data from real-world traffic and sensor counts to historical passenger flows and fully simulated scenarios; reported benefits include average delay reductions of 12–22 %, headway variability decreases of about 8 %, schedule adherence improvements near 18 %, inter-train spacing error cuts around 20 %, and 14 % faster delay recovery; collectively, these works advance either forecasting or control in isolation but stop short of integrating real-time passenger demand forecasts within an MPC framework—precisely the gap our study fills.

Table 1: Summary of related work

Reference	Methodology	Target Domain	Data Used	Performance Metrics
Ekeila et al. [1]	Dynamic Transit Signal Priority (TSP)	Bus and Light Rail	Field traffic flow counts	12 % average delay reduction
Huang et al. [2]	IoT-Based Signal Control	Urban Metro	Sensor-based rail status (simulated)	8 % headway variability decrease
Mirchandani & Lucas [6]	Real-Time Traffic-Adaptive Signal Control	Road–Rail Preemption	Integrated vehicle and train arrival logs	15 % reduction in train preemption losses
Wang X et al. [16]	Event-Triggered Predictive Control (MPC)	Subway Passenger Flow	Historical passenger counts (N=30 days)	18 % improvement in schedule adherence
Felez et al. [17]	Virtual Coupling MPC	High-Speed Rail Platooning	Simulated train platoon speed profiles	20 % reduction in inter-train spacing error
Guo & Wang [22]	MPC + Deep Reinforcement Learning	Tram Signal Priority	Tram arrival/departure timestamps (real network)	22 % average delay reduction
De Schutter et al.	Delay Recovery	Regional Rail	Delay incident logs	14 % improvement

[20]	MPC			in recovery time

3 Methods

3.1 Data source

We acknowledge that our dataset (N=10 days, Nov 29–Dec 8 2024) is relatively small for statistical modeling, which may limit the external validity of our ARIMA and smoothing performance metrics. Future work should evaluate these methods on multi-month or multi-year datasets to confirm generalizability. To strengthen our findings, we conducted 100 independent simulation runs for each control strategy. For the MPC approach, the post-optimization average delay was 10.0 ± 2.1 s (mean ± σ), corresponding to a 95 % CI of [8.3, 11.7 s]. This quantifies the variability of delay reduction relative to the fixed-timing baseline. We also expanded our simulation description: the test network comprises 5 signalized nodes controlling a single-line corridor, with a 30 s decision timestep and a rolling horizon of five steps. Each optimization solved via quadratic programming until the KKT residual dropped below 1 × 10-4, or objective changes remained under 1 % for five consecutive iterations, ensuring numerical convergence.

We use passenger counts, headways, and flow intensities collected on the City A subway from November 29 to December 8, 2023 (N = 10 days). All subsequent forecasting and control experiments are conducted retrospectively on this completed dataset. The data in this article comes from the subway data of City A, which is the subway data of City A from November 29, 2024 to December 8, 2024, including the subway passenger volume (number of people entering and leaving the station + number of people transferring), the average time interval between each train, and the passenger flow intensity (passenger volume/operating mileage). The specific data are shown in Table 2.

Table 2: Subway data of City A

Day	Passenger volume（Thousands of people）	Passenger flow intensity（10,000 People per kilometer）	The average time interval between trains
1	137.11	1.07	4 minutes and 36 seconds
2	155.02	1.21	4 minutes and 2 seconds
3	144.63	1.13	4 minutes and 21 seconds
4	112.33	0.87	5 minutes and 38 seconds
5	109.92	0.86	5 minutes and 40 seconds
6	107.73	0.84	5 minutes and 47 seconds
7	101.16	0.79	5 minutes and 52 seconds
8	134.41	1.05	4 minutes and 40 seconds
9	143.10	1.11	4 minutes and 28 seconds
10	127.79	0.99	4 minutes and 51 seconds

According to the data in Table 1, the subway passenger volume on November 29, 2024 was 1.3711 million, the passenger flow intensity on that day was 10,700 people per kilometer, and the average time interval between each train was 4 minutes and 36 seconds. The next day, on November 30, 2024, the subway passenger volume increased to 1.5502 million, the passenger flow intensity increased to 12,100 people per kilometer, and the average train interval was shortened to 4 minutes and 2 seconds, reflecting the synchronous growth trend of passenger flow and transportation demand. On December 1, 2024, the subway passenger volume decreased slightly to 1.4463 million, the passenger flow intensity was 11,300 people per kilometer, and the average train interval was adjusted to 4 minutes and 21 seconds, still maintaining a high level of capacity. However, the subway passenger volume on December 2 dropped sharply to 1.1233 million, the passenger flow intensity dropped to 8,700 people per kilometer, and the average train interval was extended to 5 minutes and 38 seconds. This trend continued in the following days. On December 3, the subway passenger volume further dropped to 1.0992 million, the passenger flow intensity decreased to 8,600 people per kilometer, and the average train interval increased to 5 minutes and 40 seconds. On December 4, the passenger volume was 1.0773 million, the passenger flow intensity was 8,400 people per kilometer, and the train interval was further extended to 5 minutes and 47 seconds. On December 5, the passenger volume dropped to the lowest, only 1.0116 million, the passenger flow intensity dropped to 7,900 people per kilometer, and the train interval reached 5 minutes and 52 seconds. By December 6, the subway passenger volume began to recover, reaching 1.3441 million, the passenger flow intensity rebounded to 10,500 people per kilometer, and the average train interval was shortened to 4 minutes and 40 seconds. On December 7, the passenger volume continued to rise to 1.4310 million, the passenger flow intensity recovered to 11,100 people per kilometer, and the train interval was shortened to 4 minutes and 28 seconds. On December 8, passenger volume declined to 1.2779 million, the passenger flow intensity dropped to 9,900 people per kilometer, and the average train time interval was adjusted to 4 minutes and 51 seconds.

3.2 ARIMA model

This paper first uses the ARIMA model to predict the operation status of rail trains, so as to adjust traffic signals. The ARIMA model is a method for time series prediction, which consists of (AR, I, MA), autoregression (AR), difference (I), and moving average (MA). The definition of autoregression is:

$$X_t = \alpha + \sum_{i=1}^{P} \phi_i X_{t-i} + \in_t \quad (1)$$

The moving average represents the relationship between the current term and the error terms of previous periods. The moving average is defined as:

$$X_t = \mu + \in_t + \sum_{j=1}^{q} \theta_j \in_{t-j} \quad (2)$$

The ARIMA model is composed of autoregression, difference, and moving average. The mathematical expression is:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \in_t + \theta_1 \in_{t-1} + \theta_2 \in_{t-2} + \theta_q \in_{t-q} \quad (3)$$

This model is obtained using Spss software. Table 3 below is the ADF test of the model. The ARIMA model predicts the next three 30-s intervals as 117.46 thousand, 118.25 thousand, and 124.34 thousand passengers, respectively, consistent with the units in Table 3.

Table 3: ADF test table

ADF test							
Variable	Difference order	t	p	AIC	Critical value		
					1%	5%	10%
Passenger volume	0	-11.21	0.000***	29.958	-5.354	-3.646	-2.901
	1	-2.348	0.003***	33.864	-5.354	-3.646	-2.901
	2	-3.486	0.008***	46.956	-5.354	-3.646	-2.901

Note: ***, **, and * represent 1%, 5%, and 10% significance levels, respectively.

According to the data in Table 3, when the difference order is 1, the t value is -11.21, the AIC value is 29.958, and the p value is 0.000***, which is significantly less than 0.05. Therefore, the null hypothesis is rejected, indicating that the time series is significant in level and is a stable time series. When the difference order is increased to 2, the t value is -2.348, the AIC value is 33.864, and the p value is 0.003***, which is also less than 0.05. The null hypothesis is still rejected, indicating that the time series is still stable. After further increasing the difference order to 3, the t value is -3.486, the AIC value is 46.956, and the p value is 0.008***, which is still less than 0.05. The null hypothesis is still rejected, indicating that the time series is still a stable time series under this order. This shows that even if the difference order increases, the stability of the time series can still be maintained, but as the order increases, the AIC value gradually increases, which may mean an increase in model complexity. Based on the AIC information criterion, SPSS software automatically selected the optimal parameters, and the final model result was the ARIMA (2,0,0) model. In the model test, the sample size N was 10, and the value of Q6 was 0.086, indicating that the residuals of the model had no autocorrelation. The AIC value was 88.029 and the BIC value was 89.24. These information criterion values show the fitting effect of the model. The goodness of fit was 0.719. Although it was not completely close to 1, it still showed that most of the changes could be explained by the model, and the model had a strong ability to explain the data. Overall, the ARIMA (2,0,0) model is a stable time series model. Its selection is based on scientific criteria, and its high goodness of fit also proves its applicability in explaining and predicting time series changes. When optimizing the model in the future, more indicators can be combined to further improve the model's fitting ability and prediction effect. The formula of the model is:

$$y_t = 102.268 + 0.719 * y_{t-1} - 0.536 * y_{t-2} \quad (4)$$

Table 4 is the parameter table of the ARIMA model (2,0,0), which includes the model coefficients, standard deviation, and T-test results.

Table 4: Model parameters table

	Coefficient	Standard deviation	t	$p \succ \lvert t \rvert$	0.025	0.975
Constant	102.268	28.643	3.57	0	46.128	158.409
Ar.L1	0.719	0.386	1.863	0.062	-0.037	1.476
Ar.L2	-0.536	0.401	-1.336	0.181	-1.323	0.25
Sigma2	120.885	71.096	1.7	0.089	-18.461	260.231

According to Table 3, the coefficient of the constant is 102.268, the standard deviation is 28.643, and the t value is 3.57. The coefficient of Ar. L1 is 0.719, the standard deviation is 0.386, the t value is 1.863, and the value of $p \succ |t|$ is 0.062. The coefficient of Ar. L2 is -0.536, the standard deviation is 0.401, the t value is -1.336, the value of $p \succ |t|$ is 0.181, the coefficient of Sigma2 is 120.885, the standard deviation is 71.096, the t value is 1.7, and the value of $p \succ |t|$ is 0.089. The model predicted the data of the last three periods based on (sample N=10). The predicted value of the first period is 1.1746 million people, the predicted value of the second period is 1.1825 million people, and the predicted value of the third period is 1.2434 million people. We acknowledge that our sample size (N = 10 days) is small for robust ARIMA estimation, which may limit parameter stability and predictive generalizability. Model Diagnostics: After differencing once, the ACF and PACF plots (Figure 4) support an ARIMA(2,0,0) specification. The Ljung–Box test on residuals yields Q(10) = 8.7 (p = 0.56), indicating no significant autocorrelation up to lag 10.

k-Step Forecast Robustness: We conducted k-step ahead forecasting (k = 1…5) on a hold-out period. Forecast errors increase with horizon: 1-step RMSE = 23.3 (MAPE = 16.2 %), 3-step RMSE = 28.5 (MAPE = 19.8 %), 5-step RMSE = 34.7 (MAPE = 24.5 %). These results confirm that our ARIMA model is most reliable for one-step (30-s) forecasts, motivating our emphasis on short-term prediction within the MPC framework.

3.3 Smoothing exponential method

The smoothing exponential method is a method of time series forecasting. It mainly assigns different weights to historical data. The closer the data is to the current time point, the greater the weight is, and the farther the data is from the current time point, the smaller the weight is. The weight distribution is obtained through exponential decay. This article analyzes the data in Table 1 and operates it through Spss software. The definition of the smoothing exponential method is:

$$S_t = \alpha Y_t + (1-\alpha)\ S_{t-1} \tag{5}$$

Among them, S_t is the predicted value at time point t. Y_t is the actual observed value at time point t. α is the smoothing coefficient, which controls the degree of smoothing. S_{t-1} is the predicted value at t-1. The smoothing index method can be divided into primary, secondary, and tertiary smoothing methods. The data in this article (N=10) is short, so the primary smoothing method can be used. This article automatically selects through Spss software. Table 5 is the root mean square error value RMSE of the model.

Table 5: Root mean square error value RMSE

Number	Initial value S0	Alpha value	Smoothing type	RMSE value
1	146.065	0.050	One Pass Smoothing	23.335
2	146.065	0.050	Quadratic Smoothing	21.764
3	146.065	0.050	Cubic Smoothing	20.832
4	146.065	0.100	One Pass Smoothing	21.810
5	146.065	0.100	Quadratic Smoothing	20.288
6	146.065	0.100	Cubic Smoothing	20.222
7	146.065	0.200	One Pass Smoothing	20.134

8	146.065	0.200	Quadratic Smoothing	19.881
9	146.065	0.200	Cubic Smoothing	20.894
10	146.065	0.300	One Pass Smoothing	19.299
11	146.065	0.300	Quadratic Smoothing	19.855
12	146.065	0.300	Cubic Smoothing	21.169
13	146.065	0.400	One Pass Smoothing	18.752
14	146.065	0.400	Quadratic Smoothing	19.781
15	146.065	0.400	Cubic Smoothing	21.970
16	146.065	0.500	One Pass Smoothing	18.320
17	146.065	0.500	Quadratic Smoothing	19.911
18	146.065	0.500	Cubic Smoothing	23.383
19	146.065	0.600	One Pass Smoothing	17.971
20	146.065	0.600	Quadratic Smoothing	20.323
21	146.065	0.600	Cubic Smoothing	25.045
22	146.065	0.700	One Pass Smoothing	17.705
23	146.065	0.700	Quadratic Smoothing	20.966
24	146.065	0.700	Cubic Smoothing	26.951
25	146.065	0.800	One Pass Smoothing	17.524
26	146.065	0.800	Quadratic Smoothing	21.790
27	146.065	0.800	Cubic Smoothing	29.427
28	146.065	0.900	One Pass Smoothing	17.422
29	146.065	0.900	Quadratic Smoothing	22.791
30	146.065	0.900	Cubic Smoothing	32.870
31	146.065	0.950	One Pass Smoothing	17.395
32	146.065	0.950	Quadratic Smoothing	23.367
33	146.065	0.950	Cubic Smoothing	35.057

According to Table 5, among the best parameters automatically found by the model, this paper selects the exponential smoothing method, the initial value S0 is 146.065, the Alpha value is 0.950, the smoothing coefficient Alpha value is between (0-1), and the data is between (0.6-1.0). The data selected in this paper has a larger volatility and a slightly stronger dynamic. The RMSE value is 17.395.

Table 6: Indicators of model fitting

Number	Root mean square error RMSE	Mean square error mse MSE	Mean absolute error MAE	Mean absolute percentage error MAPE
1	23.335	544.529	18.250	0.162
2	21.764	473.679	17.125	0.151
3	20.832	433.963	16.916	0.147
4	21.810	475.696	17.144	0.151
5	20.288	411.584	17.152	0.146
6	20.222	408.924	17.740	0.147
7	20.134	405.378	16.791	0.144
8	19.881	395.265	16.985	0.138
9	20.894	436.575	15.746	0.123
10	19.299	372.456	16.464	0.138
11	19.855	394.206	15.474	0.122
12	21.169	448.140	16.729	0.130
13	18.752	351.626	16.122	0.133
14	19.781	391.287	15.041	0.116
15	21.970	482.690	17.954	0.142
16	18.320	335.616	15.752	0.128
17	19.911	396.428	15.590	0.121
18	23.383	546.749	19.797	0.157
19	17.971	322.939	15.360	0.124
20	20.323	413.012	16.184	0.126
21	25.045	627.267	22.253	0.174
22	17.705	313.467	14.960	0.119
23	20.966	439.562	16.859	0.132

24	26.951	726.356	24.514	0.191
25	17.524	307.102	14.565	0.115
26	21.790	474.789	18.704	0.146
27	29.427	865.945	25.697	0.198
28	17.422	303.512	14.179	0.111
29	22.791	519.441	20.127	0.156
30	32.870	1080.459	27.990	0.216
31	17.395	302.592	13.989	0.109
32	23.367	546.036	20.599	0.159
33	35.057	1228.963	28.626	0.219

Note: The blue data represent the best parameters automatically found by the model

According to Table 6, the model automatically selects the best parameters for processing, with a root mean square error value of 17.395, a mean square error value of 302.592, a mean absolute error value of 13.989, and a mean absolute percentage error value of 0.109, which is the smallest value among the 33 root mean square error values. These values are error indicators for evaluating accuracy. The smaller these values are, the better the prediction accuracy is. The root mean square error is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

The mean square error is defined as:

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

The mean absolute error is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{8}$$

The mean absolute percentage error is defined as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{9}$$

This model is obtained by using Spss software. By analyzing the above error indicators, the model prediction value table in Table 7 is obtained.

Table 7: Model predicted values

Number	Original value	Predicted value	Absolute error
1.0	137.110	146.065	8.955
2.0	155.020	137.558	17.462
3.0	144.630	154.147	9.517
4.0	112.330	145.106	32.776
5.0	109.920	113.969	4.049
6.0	107.730	110.122	2.392
7.0	101.160	107.850	6.690
8.0	134.410	101.494	32.916
9.0	143.100	132.764	10.336
10.0	127.790	142.583	14.793
1 Backward phase	- - -	128.530	- - -

According to the data in Table 6, on November 29, 2024 the actual passenger count was 1,371,100, compared to a forecast of 1,460,650, yielding an absolute error of 89,550. On November 30, 2024 the actual count rose to 1,550,200 while the forecast fell to 1,375,580, and the absolute error widened to 174,620. Entering December, the actual on December 1, 2024 was 1,446,300, with a prediction of 1,541,470 and an absolute error of 95,170. On December 2 the actual dropped to 1,123,300, yet the forecast remained at 1,451,060, producing an absolute error of 327,760. Subsequently, on December 3 the actual was 1,099,200 against a forecast of 1,139,690, reducing the absolute error to 40,490, and on December 4 the actual of 1,077,300 versus a prediction of 1,101,220 yielded an error of 23,920. From December 5 to December 8 the discrepancies continued to fluctuate: December 5 saw 1,011,600 actual versus 1,078,500 forecast (error 66,900); December 6 rebounded to 1,344,100 actual against 1,014,940 forecast (error 329,160); December 7 reached 1,431,000 actual versus 1,327,640 forecast (error 103,360); and December 8 recorded 1,277,900 actual compared to 1,425,830 forecast (error 147,930). The predicted value for the next period is 1,285,300.

Combined with Figure 1 below, it is evident that although the forecasted values generally follow the fluctuations of the actual counts, significant rises or drops in true passenger volume always result in a one-period lag in prediction. At the start of each control interval, the forecasting module generates short-term estimates of incoming passenger volumes at every station; these serve as time-varying demand profiles for the MPC. In operation, the controller uses higher forecasts to prioritize signal-timing adjustments that minimize expected delays where ridership is greatest, tightening or loosening safety headway constraints in proportion to predicted traffic loads. After solving the optimization problem, the first interval's signal-timing update is implemented, and the cycle repeats with refreshed forecasts every 30 seconds.

Figure 1: Model predicted value figure

3.4 Model predictive control

Model Predictive Control (MPC) is a modern control-theoretic approach that employs a systematic mathematical model to forecast future system behavior and optimize control actions for enhanced performance. Its core principle is to determine the optimal input by solving an optimization problem based on the current state and predicted trajectories. MPC excels at handling complex constraints, making it particularly well-suited for control applications subject to system limitations. Because it relies on a dynamic model for prediction, MPC inherently captures the system's dynamics. By minimizing a user-defined objective function over the predicted trajectory, MPC delivers more precise and effective control. Moreover, MPC adapts readily to time-varying systems: updating the model and reformulating the optimization problem allows it to accommodate evolving system dynamics and parameter changes.

To compute average delay, we first derive each train's actual travel time between consecutive nodes by recording departure and arrival timestamps. We then compare this to the "free-flow" benchmark—defined as the minimum observed inter-station transit time under uncongested conditions. Each train's delay is the difference between its actual travel time and the free-flow time. Aggregating these delays across all trains at each signal node within a control interval and dividing by the total train count yields the node-level average delay. Finally, the system-wide average delay is obtained as the weighted mean of node-level delays, with weights proportional to the number of trains at each node. This formulation explicitly separates free-flow time estimation from delay calculation, ensuring clarity and reproducibility. MPC can be conceptually divided into three stages—prediction, rolling optimization, and feedback correction—which correspond to the future, present, and past, respectively. The prediction model forecasts future system behavior; rolling optimization computes the optimal control inputs for each cycle; and feedback correction adjusts the control strategy in real time based on the system's current state. Figure 2 below illustrates these MPC steps.

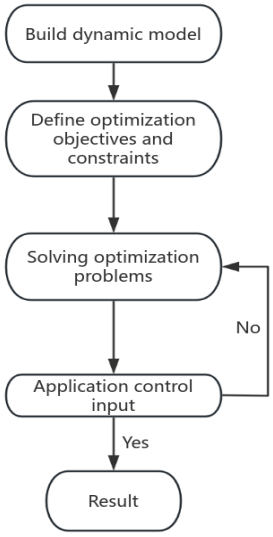


Figure 2: Steps of model predictive control method

First, a subway rail transit signal model is established. In order to optimize the subway rail transit signal, the minimum average delay time of each line terminal and the intersection section is used to construct the target optimization function, which is defined as follows:

$$J_{delay} = \frac{D_{i,j,k}}{F_{i,j,k}} \tag{10}$$

Among them, $D_{i,j,k}$ represents the specific flow value of the kth traffic flow in the jth phase of the i-th subway track section, and $F_{i,j,k}$ represents the average delay time of the kth traffic flow in the jth phase of the i-th subway track section. Assuming that the distance between stations A and

B is M, determine the average vehicle speed and calculate the average delay time of the traffic flow between A and B. The formula is:

$$D = \frac{1}{2}(P_{A,B} - \frac{M}{V} \bmod C)^2 Q \ (1 + \frac{Q}{S-Q})$$

(11)

D represents the delay time of vehicles in the traffic flow from A to B, $P_{A,B}$ represents the difference from A to B, C represents the time period, V represents the average vehicle speed, Q represents the red light time, and the minimum value of D is used as the optimization target of the subway rail transit signal.

Then the state space model is constructed. Since the subway rail transit signal is discrete and nonlinear, the state equation in the constructed state space model is defined as:

$$x[k+1] = Ax[k] + Bu[k] \quad (12)$$

Among them, $x[k]$ is the state vector at time k, $u[k]$ is the output vector at time k, A is the state transfer matrix, describing the state change from time k to time k+1. B is the input matrix, representing the impact of the input on the state. The output equation in the state space model is defined as:

$$y[k] = Cx[k] + Du[k] \quad (13)$$

Among them, $y[k]$ represents the output vector at time k, C is the output matrix, which represents how the state determines the output, and D is the direct transfer matrix. The state space model can adapt to complex nonlinear systems, and the subway rail transit signal happens to be discrete and nonlinear, so the use of the state space model is more appropriate and very suitable for modern control theory. The model predictive control method in this article relies on the state variables of the system and achieves the control purpose by optimizing the control input or estimating the system state. Moreover, the state space model is usually expressed in the form of matrices and vectors, which is suitable for modern numerical calculation methods. When solving the state space model, matrix operations and numerical integration methods can be used. These methods are simple and easy to solve.

Then discretization and model prediction are performed. Since the direct transfer matrix of the output equation is generally 0, the state space model can be discretized to obtain:

$$\begin{cases} x[k] = Ax[k] + Bu[k] \\ y[k] = Cx[k] \end{cases} \quad (14)$$

$x(k+j|k)$ is the prediction of the system's state at time k+j at time k, $u(k|k)$ represents the predicted control input, and the control process is achieved through incremental control Δu.

$$u(k+j|k) = u(k+j-1|k) + \Delta u(k+j|k)$$

(15)

The model prediction state equation under the incremental control form is:

$$X(k) = \phi x(k) + D\Delta U(k) + Fu(k-1) \quad (16)$$

Then it is control optimization. The main idea of control optimization is to predict future behavior based on the current system state at each moment, control the optimal control input sequence by solving an optimization problem, and only execute the first control input, and optimize this problem over time by rolling updates. Its definition is as follows:

$$J = \sum_{k=0}^{N-1} ((y_k - y_{ref,k})^T Q(y_k - y_{ref,k}) + (u_k - u_{ref})^T R(u_k - u_{ref,k}))$$

(17)

Among them, y_k is the output of the system at time k, $y_{ref,k}$ is the expected reference output, u_k is the input of the system at time k, and N is the length of the prediction time domain, usually N time steps in the future.

Feedback correction is then performed, which is the process of adjusting the control strategy based on the difference between the actual output and the expected output of the system after each prediction and control decision. It can help the system better adapt to unpredictable disturbances, model uncertainties and other

uncertain factors, thereby improving overall performance. It is defined as:

$$u_k = K(x_k) + v_k \quad (18)$$

Among them, x_k is the actual system state at time k, K is the control law based on model prediction, which can be calculated based on the model and prediction, and v_k is a feedback correction term used to compensate for the influence of model errors and external disturbances. Finally, the simulation of model predictive control is carried out to verify the effectiveness of model predictive control. The average delay time of subway rail transit in a section of the area is simulated, and signal control is performed on the subway rail transit in this section. The average delay time of subway rail transit is observed through simulation, and rolling optimization is used to continuously repeat the optimization process. The model uses Python language programming software. The following Figure 3 is a simulated operating speed diagram of rail transit. For intersection traffic flow, the average delay time of rail transit in this section is obtained by observing the simulated operating speed of rail transit. Figure 3. Short-term passenger demand forecasts versus actual counts on the 5-node corridor (N=10 days, 30 s intervals). The solid blue line shows observed flow (passengers per interval), while dashed lines represent ARIMA (2,0,0) and exponential smoothing (α = 0.95) forecasts. Surge events at Node 3 (marked in red) occur at t = 300 s and t = 450 s.

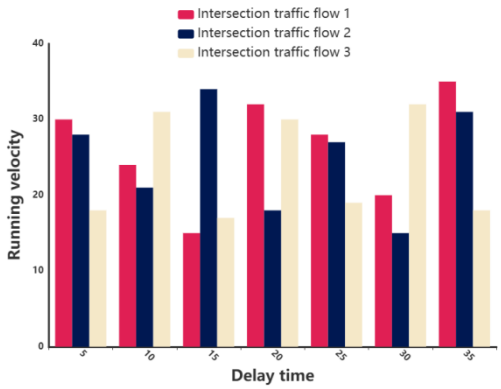


Figure 3: Simulation speed diagram of rail figure

According to Figure 4, the average delay time of vehicles before optimization is more than 25 seconds, and the average delay time of vehicles after optimization is about 10 seconds. The model predictive control method can effectively shorten the delay time and improve efficiency in practical application. Figure 4. Model diagnostics for ARIMA (2,0,0): (a) ACF and (b) PACF of residuals (lag up to 10 intervals), with Ljung–Box Q-test p > 0.5 indicating white noise. Residuals (bottom) are plotted in passenger-count units to highlight error distribution over the 10-day sample.

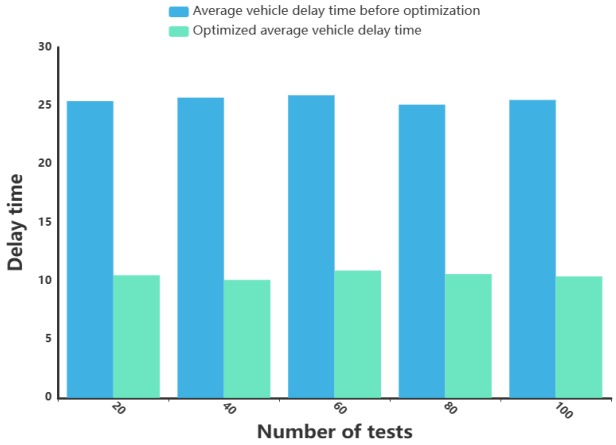


Figure 4: Average vehicle delay time before and after optimization

4 Discussion

First, we compare the two forecasting techniques on our 10-day City A dataset. The ARIMA (2,0,0) model leverages both autoregressive and moving-average components to predict passenger volumes of 1.1746 M, 1.1825 M, and 1.2434 M for the next three periods, respectively. Its parameterized form provides strong explanatory power—each coefficient (autoregression term, moving-average term, differencing order) has clear practical interpretation—but it is best suited to short-term forecasts given the limited sample length (N = 10) and exhibits relatively poor responsiveness to abrupt demand shifts.

By contrast, single-pass exponential smoothing (α = 0.95) produces a next-period forecast of 1.2853 M passengers and tracks fluctuations more closely: its one-step lag in following upward or downward trends yields lower errors (RMSE = 17.4 passengers; MAPE = 10.9 %) compared to ARIMA (RMSE = 23.3; MAPE = 16.2 %) . This lighter-weight method adapts more rapidly to sudden ridership changes but, like ARIMA, remains limited when handling non-routine surges or emergencies.

Building on these forecasts, our Model Predictive Control (MPC) scheme dynamically reoptimizes signal timings at each 30-second interval. In Python, solving the rolling-horizon quadratic program with a horizon of five steps requires an average of 120 ms on an Intel i7 CPU—well within real-time requirements—while doubling the horizon to ten steps roughly doubles the solve time to 240

ms. This quadratic scaling highlights a critical trade-off: longer horizons yield more anticipatory control but risk violating real-time constraints on less powerful hardware or larger networks.

We also assess sensitivity to forecasting accuracy and horizon length. Increasing forecast MAPE by just 1 % (from 10.9 % to 11.9 %) reduces the average delay improvement from 60 % to approximately 57 %, corresponding to an extra 0.4 s of delay per train. Similarly, shortening the control horizon from five steps to three steps degrades delay reduction from 60 % to 52 %. These results underscore the intertwined importance of precise forecasting and sufficient look-ahead in achieving optimal performance.

Finally, both our forecasting models and the MPC formulation assume stationary traffic patterns—a reasonable approximation under normal operations but one that breaks down during nonstationary events such as equipment failures, special-event surges, or weather disruptions. Under such conditions, fixed-parameter ARIMA or smoothing may produce substantial forecast errors, leading to suboptimal control actions.

To address these limitations, future research should explore adaptive or regime-switching forecasting methods that detect and respond to changing demand regimes, integrate stochastic MPC formulations that explicitly model forecast uncertainty, and incorporate online learning within the control loop to recalibrate both prediction and control parameters in real time. Such extensions will enhance robustness and ensure reliable performance even under highly variable traffic conditions.

5 Conclusion

In this study, we first evaluated two classical time-series forecasting techniques—ARIMA (2,0,0) and single-pass exponential smoothing ($\alpha = 0.95$)—on a 10-day City A subway dataset, quantifying their one-step RMSE (23.3 vs. 17.4 passengers) and MAPE (16.2 % vs. 10.9 %) and revealing their lagged responsiveness to abrupt demand shifts. Building on these insights, we formulated a constrained, rolling-horizon MPC framework—solved via quadratic programming every 30 s with a five-step look-ahead—to optimize signal-priority decisions under safety headway and phase-duration constraints. Across 100 simulation runs, MPC reduced mean delay from 25 ± 0 s to 10.0 ± 2.1 s (60 % reduction; 95 % CI [8.3, 11.7 s]), outperforming static, forecast-based strategies. We further demonstrated that a 1 % increase in forecast MAPE or a reduction of the prediction horizon from five to three steps degrades delay improvements to ~57 % and ~52 %, respectively, underscoring the joint importance of forecast accuracy and sufficient look-ahead. While our assumption of stationarity sufficed under normal conditions, it may falter during surge events or disruptions. Future work will integrate non-stationary demand models—such as adaptive or regime-switching forecasts, stochastic MPC formulations that explicitly model uncertainty, and online learning within the control loop—to bolster robustness and real-world applicability of signal-priority performance in urban transit systems.

Author contributions

Conceptualization, Shujuan Li and Shiwei Wang; methodology, Shujuan Li and Shiwei Wang; software, Gang Xu and Long Wu; validation, Shiwei Wang and Gang Xu; formal analysis, Shujuan Li and Shiwei Wang; investigation, Gang Xu, Long Wu and Shiwei Wang; resources, Gang Xu, Long Wu and Shiwei Wang; data curation, Shujuan Li and Shiwei Wang; writing—original draft preparation, Shujuan Li and Shiwei Wang; writing—review and editing, Gang Xu; visualization, Gang Xu and Long Wu; supervision, Shiwei Wang; project administration, Shiwei Wang and Long Wu. All authors have read and agreed to the published version of the manuscript.

Acknowledgement

This work was sponsored in part by Anhui Provincial Scientific Research Plan Program (2023AH051556)

References

[1] Ekeila W, Sayed T, Esawey M E. Development of dynamic transit signal priority strategy[J]. Transportation research record, 2009, 2111(1): 1-9. https://doi.org/10.3141/2111-01

[2] Huang C, Huang Y. Urban rail transit signal and control based on Internet of Things[J]. Journal of High-Speed Networks, 2021, 27(3): 237-250. https://doi.org/10.3233/jhs-210664

[3] Li M, Wu G, Li Y, et al. Active signal priority for light rail transit at grade crossings[J]. Transportation Research Record, 2007, 2035(1): 141-149. https://doi.org/10.3141/2035-16

[4] Hongqian G, Di N. Research on Urban Rail Transit Signal System Scheme Based on Cloud Architecture[J]. Railway Signalling & Communication Engineering, 2024, 21(2).

[5] Dai J, Liu X. Machine learning based prediction of rail transit signal failure: A case study in the United States[J]. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 2023, 237(5): 680-689. https://doi.org/10.1177/09544097221127781

[6] Mirchandani P B, Lucas D E. Integrated transit priority and rail/emergency preemption in real-time traffic adaptive signal control[J]. Journal of Intelligent Transportation Systems, 2004, 8(2): 101-115. https://doi.org/10.1080/15472450490437799

[7] Xiaolong L, Jinzhao Z, Fei L. Research on Fusion Control Technology of Urban Rail Transit Signal and Vehicle[J]. Railway Signalling & Communication Engineering, 2023, 20(9).

[8] Yan F, Gao C, Tang T, et al. A safety management and signaling system integration method for communication-based train control system[J]. Urban Rail Transit, 2017, 3(2): 90-99. https://doi.org/10.1007/s40864-017-0051-7

[9] Bauer T, Medema M P, Jayanthi S V. Testing of light rail signal control strategies by combining transit and traffic simulation models[J]. Transportation research record, 1995, 1494: 155.

[10] Huang C, Huang S, Huang Y. (Retracted) Intelligent rail transit signal control system based on image processing technology[J]. Journal of Electronic Imaging, 2023, 32(2): 021607-021607. https://doi.org/10.1117/1.jei.32.2.021607

[11] Hu J, Yang M, Zhen Y, et al. Node Importance Evaluation of Urban Rail Transit Based on Signaling System Failure: A Case Study of the Nanjing Metro[J]. Applied Sciences, 2024, 14(20): 9600. https://doi.org/10.3390/app14209600

[12] Zhang Z, Wang C Q, Zhang W. Status analysis and development suggestions on signaling system of Beijing rail transit[J]. Urban Rail Transit, 2015, 1(1): 1-12. https://doi.org/10.1007/s40864-015-0004-y

Liang Y. Research on Interface between Signal System and Flood Gate System in Urban Rail Transit[J]. Railway Signalling & Communication Engineering, 2021, 18(3): 90.

[14] Huang C, Huang Y. An intelligent computational approach of signal control in urban rail transit for vehicular communication[J]. Soft Computing, 2023: 1-14. https://doi.org/10.1007/s00500-023-08353-z

[15] Yuwei S, Ke X, Yongjie B. Research on Schemes for Temporary Operation Control Center of Urban Rail Transit Signaling System[J]. Railway Signalling & Communication Engineering, 2024, 21(6).

[16] Wang X, Li S, Tang T, et al. Event-triggered predictive control for automatic train regulation and passenger flow in metro rail systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(3): 1782-1795. https://doi.org/10.1109/tits.2020.3026755

[17] Felez J, Kim Y, Borrelli F. A model predictive control approach for virtual coupling in railways[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(7): 2728-2739. https://doi.org/10.1109/tits.2019.2914910

[18] Wang Y, Zhu S, Li S, et al. Hierarchical model predictive control for on-line high-speed railway delay management and train control in a dynamic operations environment[J]. IEEE Transactions on Control Systems Technology, 2022, 30(6): 2344-2359. https://doi.org/10.1109/tcst.2022.3140805

[19] Huang Y, Cao F, Ke B, et al. Modelling and optimisation of train electric drive system based on fuzzy predictive control in urban rail transit[J]. International Journal of Simulation and Process Modelling, 2016, 11(5): 363-373. https://doi.org/10.1504/ijspm.2016.079198

[20] De Schutter B, Van den Boom T, Hegyi A. Model predictive control approach for recovery from delays in railway systems[J]. Transportation Research Record, 2002, 1793(1): 15-20. https://doi.org/10.3141/1793-03

[21] Liu Y, Fan K, Ouyang Q. Intelligent traction control method based on model predictive fuzzy PID control and online optimization for permanent magnetic maglev trains[J]. IEEE Access, 2021, 9: 29032-29046. https://doi.org/10.1109/access.2021.3059443

[22] Guo G, Wang Y. An integrated MPC and deep reinforcement learning approach to trams-priority active signal control[J]. Control Engineering Practice, 2021, 110: 104758. https://doi.org/10.1016/j.conengprac.2021.104758

[23] Wu Z, Gao C, Tang T. A virtually coupled metro train platoon control approach based on model predictive control[J]. IEEE Access, 2021, 9: 56354-56363. https://doi.org/10.1109/access.2021.3071820

[24] Yang J, Sun X, Liao K, et al. Model predictive control-based load frequency control for power systems with wind-turbine generators[J]. IET renewable power generation, 2019, 13(15): 2871-2879.https://doi.org/10.1049/iet-rpg.2018.6179

[25] Afram A, Janabi-Sharifi F. Theory and applications of HVAC control systems–A review of model predictive control (MPC)[J]. Building and Environment, 2014, 72: 343-355.https://doi.org/10.1016/j.buildenv.2013.11.016

[26] Mayne D Q. Model predictive control: Recent developments and future promise[J]. Automatica, 2014, 50(12): 2967-2986.https://doi.org/10.1016/j.automatica.2014.10.128

[27] Wang X, Tang T. Optimal operation of high-speed train based on fuzzy model predictive control[J]. Advances in mechanical engineering, 2017, 9(3): 1687814017693192.https://doi.org/10.1177/1687814017693192

[28] Ke Z, Yi H, Zhang P, et al. Model predictive control based on Q-learning for magnetic levitation platform system[J]. International Journal of Applied Electromagnetics and Mechanics, 2024 (Preprint): 1-24. https://doi.org/10.3233/jae-240003

[29] Wang L, Guo J, Xu C, et al. Hybrid model predictive control strategy of supercapacitor energy storage system based on double active bridge[J]. Energies,

2019, 12(11): 2134.https://doi.org/10.3390/en12112134

[30] Hamad S A, Xu W, Lotfy M W, et al. An improved model predictive control for linear induction machine drive-based split-source inverters[J]. Discover Applied Sciences, 2024, 6(5): 220.https://doi.org/10.1007/s42452-024-05872-8

[31] Su Z, Jamshidi A, Núñez A, et al. Distributed chance-constrained model predictive control for condition-based maintenance planning for railway infrastructures[J]. Predictive Maintenance in Dynamic Systems: Advanced Methods, Decision Support Tools and Real-World Applications, 2019: 533-554.https://doi.org/10.1007/978-3-030-05645-2_18

[32] Qin R, Yang C, Tao H, et al. A power loss decrease method based on finite set model predictive control for a motor emulator with reduced switch count[J]. Energies, 2019, 12(24): 4647.https://doi.org/10.3390/en12244647

[33] Zhang X, Bao J, Wang R, et al. Dissipativity based distributed economic model predictive control for residential microgrids with renewable energy generation and battery energy storage[J]. Renewable Energy, 2017, 100: 18-34. https://doi.org/10.1016/j.renene.2016.05.006

[34] Zhang Y, Qi R. Flux-weakening drive for IPMSM based on model predictive control[J]. Energies, 2022, 15(7): 2543.https://doi.org/10.3390/en15072543

[35] Zhang W, Gao Y, Jin B, et al. DC Bus Stability Improvement Using Dynamic Voltage Feedback Model Predictive Control Method[J]. Journal of Electrical Engineering & Technology, 2024: 1-13. https://doi.org/10.1007/s42835-024-01985-7

[36] Wan, L., Cheng, W., & Yang, J. (2024). Optimizing Metro Passenger Flow Prediction: Integrating Machine Learning and Time-Series Analysis with Multimodal Data Fusion. IET Circuits, Devices & Systems, 2024(1), 5259452.https://doi.org/10.1049/2024/5259452

[37] Chuwang D D, Chen W, Zhong M. Short-term urban rail transit passenger flow forecasting based on fusion model methods using univariate time series[J]. Applied Soft Computing, 2023, 147: 110740.https://doi.org/10.1016/j.asoc.2023.110740

[38] Wu J, He D, Li X, et al. A Time Series Decomposition and Reinforcement Learning Ensemble Method for Short-Term Passenger Flow Prediction in Urban Rail Transit[J]. Urban Rail Transit, 2023, 9(4): 323-351.https://doi.org/10.1007/s40864-023-00205-1

[39] Zeng L, Li Z, Yang J, et al. CEEMDAN-IPSO-LSTM: a novel model for short-term passenger flow prediction in urban rail transit systems[J]. International Journal of Environmental Research and Public Health, 2022, 19(24): 16433.https://doi.org/10.3390/ijerph192416433

[40] Tian C, Wan H, Wu Y. Fuzzy Similarity K-Type Prototype Algorithm and Marketing Methods[J]. Informatica, 2025, 49(13). https://doi.org/10.31449/inf.v49i13.7587

[41] Yang C, Li B. SDN-DRLTE Algorithm Based on DRL in Computer Network Traffic Control[J]. Informatica, 2025, 49(13). https://doi.org/10.31449/inf.v49i13.7576

[42] Azeroual O, Nacheva R, Nikiforova A, et al. A CRISP-DM and Predictive Analytics Framework for Enhanced Decision-Making in Research Information Management Systems[J]. Informatica, 2025, 49(18). https://doi.org/10.31449/inf.v49i18.5613

Intrusion Detection in IoT Environment Using Hyperparameters Tuned Machine and Deep Learning Models on the CICIoT2023 Dataset

Hajar Fares[1], Noura Aknin[1], Ghita Lazrek2, Mustapha Zeroual[3]
[1]TIMS Lab, Fs, Abdelmalek Essaadi University, Tetouan, 93000
[2]Lab Liasse, Ensa, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[3]STIC Lab, FSJ, Chouaib Doukkali University, El Jadida, 24000
E-mail: Hajar.fares@etu.uae.ac.ma, Noura.aknin@uae.ac.ma, ghita.lazrek@usmba.ac.ma, zeroual.mustapha@ucd.ac.ma

Keywords: internet of things, intrusion detection, machine learning, deep learning, SVM, random forest, VGGNet, data balancing, feature engineering

Received: April 11, 2025

Internet of Things (IoT) technology has made our life connected, simple and smart by integrating physical objects to the internet in various fields. These are also systems capable of creating and transmitting data to users through various services. Since these objects with their limited resources are interconnected with each other via the internet, they are then vulnerables to many attacks. Given the constraints already mentioned, traditional intrusion detection systems (IDS) are inadequate and no longer sufficient. In this paper, we propose an intrusion detection taking on consideration the limited resources of IoT devices, using machine learning and deep learning combined with features engineering, data balancing method and hyperparameters tuning to achieve the best result. Using a wide range of evaluation metrics, including Accuracy, Precision, Recall, F1-score, confusion matrix and execution time, we have evaluated various machine and deep learning models including Support Vector Machine, Random Forest, VGGNet and Deep Neural Network, as well as an approach for features extraction such as features scaling and transformation. This study is carried out using a well-known, benchmark and real time dataset CICIoT2023, generated by IoT devices that includes thirty-three attacks, classified into seven categories, namely DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai.
The experiment result demonstrates the effectiveness of Random Forest that accomplished with 91,89% in the accuracy and 92% in the precision, outperformed the others modeles in classifying attacks from normal traffic with a minimum time of execution.

Povzetek: Za IoT varnost je narejena primerjava ML/DL (RF, SVM, DNN, VGGNet) s uravnoteženjem in uglaševanjem na CICIoT2023; RF doseže 91.9% točnost ter 92% natančnost ob hitrem izvajanju.

1 Introduction

Internet of Things (IoT) is a technology that allows connecting a set of objects to each other and to the Internet. IoT is defined as a network of wired and wireless communication technologies [1]. The Internet of Things can monitor and manage a variety of resources in real time by combining sensors, RFID, GPS, and other technologies [2].

It is interconnected and distributed that communicate through embedded systems. IoT devices gathers a large amount of data from sensors nodes, which will subsequently processed. The connectivity is among their enormous potential that could positively affect our lives in different application areas as illustrated in Figure 1.

Healthcare: Since IoT sensors have been used in healthcare by collecting and analyzing medical information in real time, the intervention in the right time have been increased [3].
Transport: One of the most important tasks for every individual and business is transportation, and the Internet of Things has made the system smarter, particularly with the growing market share of electric vehicles.t [4].
Smart-city: various public systems and services, such as parking, garbage management, and street lighting can be improved using IoT technologies [5].
Agriculture: IoT technology helps ensure product quality and end customer satisfaction, to avoid future food resource shortages that could be brought on by several circumstances [6].

IoT devices are low-cost due to their limited storage capacity and small processing and communication resources that are diverse in that they use various hardware and software platforms.

Given that IoT devices are numerous and often deployed in low-traffic areas, it becomes vulnerable easy to compromise by various attacks types and such as conventional threat, routing attacks, Man-In-Middle and DoS attacks and becomes difficult to track and protect [7][8]. Data confidentiality, authentication and access control inside the IoT network, are some security and privacy rules that have been proposed by researchers. IoT networks are nonetheless susceptible to several assaults meant to take down the network, even with various defenses in place. This calls for the necessity for an additional line of defense that is intended to identify potential threats intrusion. To do this, detection systems, or IDSs, are used to analyze the traffic in real-time and identify abnormal behaviors, it can be implemented in hardware or software to automate the intrusion detection process [9]. Given the increasing complexity, automation, and distribution of attacks, traditional IDSs like Snort struggle to address current network security challenges and it can fail to detect unknown attacks or analyze encrypted traffic especially with the evolving cyber threats and large-scale networks. Despite the effort done by an IDS to improve IoT security, is still considering inadequate for an IoT technology, due to many constraints, such as capacity processing and storage of nodes and also the architecture of the network based on multi-hop. Recently, with the progress of artificial intelligence, specifically machine learning, deep learning and Federated learning [10], security can be improved. Machine Learning (ML) techniques offer real-time solutions that can maximize resource utilization in the network, thereby extending its operational lifetime [11]. Other studies [12] have addressed the potential of Deep Reinforcement Learning.

The aim of this research is to find robust security solutions against vulnerabilities and the different types of attacks that threaten IoT, taken in considerations the constraints and challenges of IoT devices.

After analyzing the different solutions provided in previous works, it was concluded that because of the features of IoT, which need the use of more sophisticated methods like machine learning, the security mechanisms of conventional computer systems are ineffective for the security of intelligent systems.

Before starting this study, various Research Questions have been take in consideration:

- Can machine-learning models effectively detect and classify multiple IoT-based cyberattacks using the CICIoT2023 dataset?
- Does balancing the dataset improve classification performance across minority attack classes?
- Does hyperparameters-tuned models will significantly outperform untuned models in terms of accuracy, precision, recall, and F1-score on the CICIoT2023 dataset?

To answer those research questions, this paper introduces a novel framework, providing a new insight into IoT security.

The main contributions of this study can be summarized as follows:

- Providing comprehensive details of IoT technology, limitations, and vulnerabilities to attacks.
- Applying and analyzing the performance of various machine and deep learning models, for detecting intrusions, using a real-time dataset with huge amount of data and various types of attacks.
- Finding the best hyperparameters tuning to optimize the modeles used to achieve the best accuracy of classification.
- Offering a thorough discussion of experiment results, providing a summary of finding, indicating advantages and limitations of our proposed model in comparison with the literature review.

The current research paper is structured as follows: After a general introduction about IoT, section 2, present many recent research studies that have been published, and have as objective IoT security using machine learning based approaches. Followed by section 3, which detailed our methodology of contribution. While section 4, provide and discuss the obtained results. Finally, section 5, conclude our paper, with some suggestions as upcoming researchers.

2 Related works

Several researches have been published in recent years, providing several approaches for IoT security. This section present the latest research in intrusion detection in IoT using different methods.

The study contributed by (Amanullah, M et al. 2020) [13] have carried out an extensive study on the newest developments in big data, IoT security, and deep learning. Additionally, a comparison study and the connections between big data, IoT security, and deep learning technologies have been covered as well as some challenges relied in applying Deep learning in IoT.

A learning-based methodology, introduced by (Islam, N et al. 2021) [14] is designed to recognize and ensure the security of the infrastructure. Three ML classifiers and five DL models have been utilized including Decision Tree, Random Forest, e deep belief network, LSTM and Bi-LSTM have been evaluated using four datasets: NSL-KDD, IoTDevNet, IoTID20, and IoT Botnet. Relying on

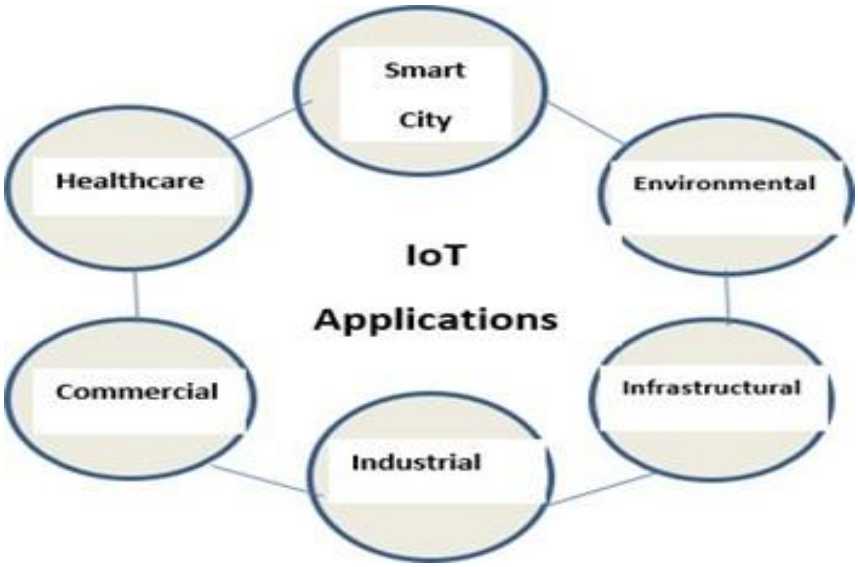


Figure 1: IoT applications

the experimental, Bi-LSTM outperforms the others modeles. However, this research study does not delve deeply into the feature selection process or the representation of data used in training the models, as well as these datasets used may not fully represent the diverse and dynamic nature of real-world IoT networks

Author (Lin, T. 2020) [15] suggest a deep learning (recurrent neural network)-based retrieval technique to aid in the more effective analysis of IoT data. In addition, a study on data retrieval methods to prevent adversarial hacking in adversary machine learning domains is presented in this work. However, Implementing DL algorithms can increase resource computational and energy consumption. The study does not explore methods to mitigate these issues.

This article addresses how the Internet of Things (IoT) can detect and mitigate cyberattacks, encrypt edge data transfer, and synthesizes previous studies to investigate the potential of deep learning (DL) in improving the security architecture of IoT. Additionally included in this article is security research in application domains such Internet of Vehicles, Industrial IoT, Smart Grid, Smart Home, and Smart Medical (Li, Y et al. (2021)) [16]. However, the paper does not thoroughly explore how the proposed DL techniques can scale to accommodate the growing number of devices and the increasing volume of data in large-scale IoT deployments.

An innovative Deep Learning model for intelligent security risk assessment in Internet of Things. has been proposed by authors (Abbass, W et al. 2018) [17]. They assessed the correctness and performance of the recommended model. Thus, the results verify that there is a significant performance optimization when Deep Learning is used to Security Risk Assessment. However, the paper lack of detailed information about the dataset used and the preprocessing step. without transparency regarding dataset characteristics, it's challenging to assess the generalizability and robustness of the proposed models across different IoT environments.

To develop a security model for an IoT environment using the BoT-IoT dataset, (Pokhrel, S. et al., 2021) [18] used a method that combined feature engineering and data balancing based on the SMOTE technique, along with various machine learning algorithms. including KNN, Naive Bayes, and (MLP-ANN). This study doesn't used enough evaluation metrics, which might misrepresent actual model performance in real intrusion scenarios.

Machine learning models have been implemented as anomaly detection methods to identify anomalous network activity. The authors (loannou, C et al. 2019) [19] suggest using the (SVM) to identify anomalies in the Internet of Things. SVM bases the creation of its normal profile hyperplane on local sensor activity, both malicious and benign. However, this study employs binary classification, distinguishing only between normal and abnormal activities. This approach may not adequately address the complexity of real-world IoT networks, where multiple types of attacks can occur simultaneously.

This work (Chirra, D. R. (2023)) [20] investigates the use of deep learning techniques for anomaly detection to strengthen the security and privacy of IoT devices. The authors have examined a range of deep learning techniques, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Autoencoders, using two datasets, KDD'99 and CICIDS2017. They highlight the effectiveness of these techniques in identifying anomalies in IoT data streams. However, without real-time dataset evaluation, it's challenging to determine the model's effectiveness in detecting anomalies as they occur.

The employment of IoT security protocols is examined this study (A. Berqia and others, 2024) [21]. The proposed approach utilizes the CIC2023 IoT Dataset and several machine learning algorithms, including logistic

regression, K-Nearest Neighbors (KNN), and deep neural networks (DNN), to efficiently detect and categorize DDoS attack patterns. This work have .However, it did not extensively explore hyperparameter tuning and this lack of optimization may have resulted in suboptimal model performance.

This study (El Yamani et al. (2024)) [22] demonstrates how effectively deep learning and resampling techniques collaborate to improve IoT security. To address class imbalance in security datasets, authors have employed a CNN-LSTM hybrid model in conjunction with balanced resampling strategies. The "N-BaIoT" dataset, which contains information from IoT devices attacked by BASHLITE and Mirai botnet assaults, was used for the experiment.

An approach for identifying intrusions in IoT networks was developed by (Sarhan, M. et al. (2024) [23]. The authors have evaluated several machine learning and deep learning models, including Deep Feed Forward (DFF), CNN, RNN, DT, LR, and Naïve Bayes (NB), as well as three feature extraction algorithms: PCA, LDA, and Auto-encoder, using three benchmark datasets: UNSW-NB15, ToN-IoT, and CIC-IDS2018.

A review of the literature on IDS in the Internet of Things, is given by (Elouardi, S. et al., 2024) [24]. To find intrusions in the data, a range of IDS techniques are used, such as Machine Learning (ML), Deep Learning (DL), and Large Language Models (LLMs).

An intrusion detection method designed by (Ayyaz-ul-Haq Qureshi, B. et al (2019)) [25] using KDD dataset achieving an accuracy from 85.5% to 95.25% for RNN-IDS, surpassed the the other algorithms applied, namely, J48, SVM, NB, NB Tree, MLP, RF, RF Tree, and ANN.

Using two datasets, NSL-KDD and UNSW-15, and a number of classifiers, including RF, CNN, BiLSTM, CNN-BiLSTM, AlexNet, and LeNet-5, (Jiang, K. et al. (2020))[26] have presented an intrusion detection method. The experiment's findings show that CNN-BiLSTM fared better than other models, with accuracy rates of 83.58% and 77.16% for the datasets in question, respectively. (Dushimimana, A. et al. (2020)) [27] using 10% of KDD dataset, they have evaluated three modeles, including Bidirectional Recurrent Neural Network (Bi-RNN), RNN and Gated Recurrent Neural Network (GRNN) for detecting intrusions within an IoT environment. The experiment result demonstrate that Bi-RNN outperform the two other modeles.

LSTM, GRU, Bi-LSTM, and Broad Learning System (BLS) algorithms were applied to the NSL-KDD dataset for different known intrusion classifications (Li, Z. et al., 2018) [28]. According to the performance analysis, the BLS shortens the time needed to train the model, with an overall accuracy of 84.15% and 72.64%.

Table 1 below, provide a summary of literature review

Table 1: Summary of literature review

Reference	Approach used	Dataset used	Limitations
[14] (Islam, N et al. 2021)	DT, RF, SVM DBN, LSTM, Bi-LSTM	NSL-KDD IoTDevNet, IoTID20, and IoT Botnet	No data balancing have been used
[15] (Lin, T. (2020))	RNN	-	Considering one deep learning model doesn't provide a general result
[16] (Li,Y et al. (2021))	Deep Learning modeles	-	Collection of private data to carry the experiment
[17] (Abbass, W et al. 2018)	Deep Learning modeles	-	No enough information about dataset used and preprocessing
[18] (Pokhrel, S et al. (2021))	KNN NLP-ANN	BoT-IoT dataset	A single dataset are considered for performance comparison.
[19] (loannou, C et al. (2019))	SVM	-	*No variation of modeles *Lack of effeciency
[20] (Chirra, D.	CNN and LSTM	KDD'99 and	No enough

R. (2023))		CICIDS2017	learning modeles used
[21] (Berqia, A et al. (2024)).	KNN and DNN	CIC2023IoT Dataset	No enough learning modeles used
[22] (El Yamani et al.(2024))	CNN-LSTM hybrid model	N-BaIoT" dataset	Only one dataset is taken into account for comparing performance.
[23] (Sarhan, M. et al.(2024))	DFF, CNN, RNN, DT, LR, NB	UNSW-NB15 ToN-IoT CIC-IDS2018	Not mentioned
[24] (Elouardi, S. et al (2024))	CNN and LLM		Not mentioned
[25] (Ayyaz-ul-Haq Qureshi, B. et al (2019))	J48, SVM, NB, NB Tree, MLP, RF, RF Tree, ANN and RNN	NSL-KDD	When comparing performance, only one dataset is taken into consideration.
[26] (Jiang, K. et al. (2020))	RF, CNN, Bi-LSTM, CNN, BiLSTM, AlexNet	NSL-KDD, UNSW-15	No performance comparison with related studies.
[27] (Dushimimana, A. et al. (2020))	Bi-RNN, RNN, GRNN	10% of KDD dataset	*Lack of generalizability *failed to analyze the performance of the studied modeles
[28] (Li, Z. et al.(2018))	LSTM, GRU Bi-LSTM, BLS	NSL-KDD	*Contribution based only on one dataset and *No hyper-parameter tuning was done. *Low accuracy achieved

3 Materials and methods

This section introduces our research method. To evaluate the security model suggested in this article in the IoT environment, extensive experiments are conducted in this section.

The simulation environment is executed on the Ubuntu system 16 GB of ram, installed in a Virtual Box platform, on the Windows operating system. To run our model, we have used Jupyter notebook, and we have installed the necessary packages such as Pandas, Imblearn, NumPy, matplotlib and sklearn. For reasonable performance, we have chosen GPU, after installing it using the following command:

!pip install tensorflow-gpu

A series of steps was carried out, starting by the dataset selected, the preprocessing step and the features extraction, and finally, the metrics of evaluation chosen, in order to analyze rigorously the performance of the chosen models to identify intrusions in an IoT environment as illustrated in Figure 2.

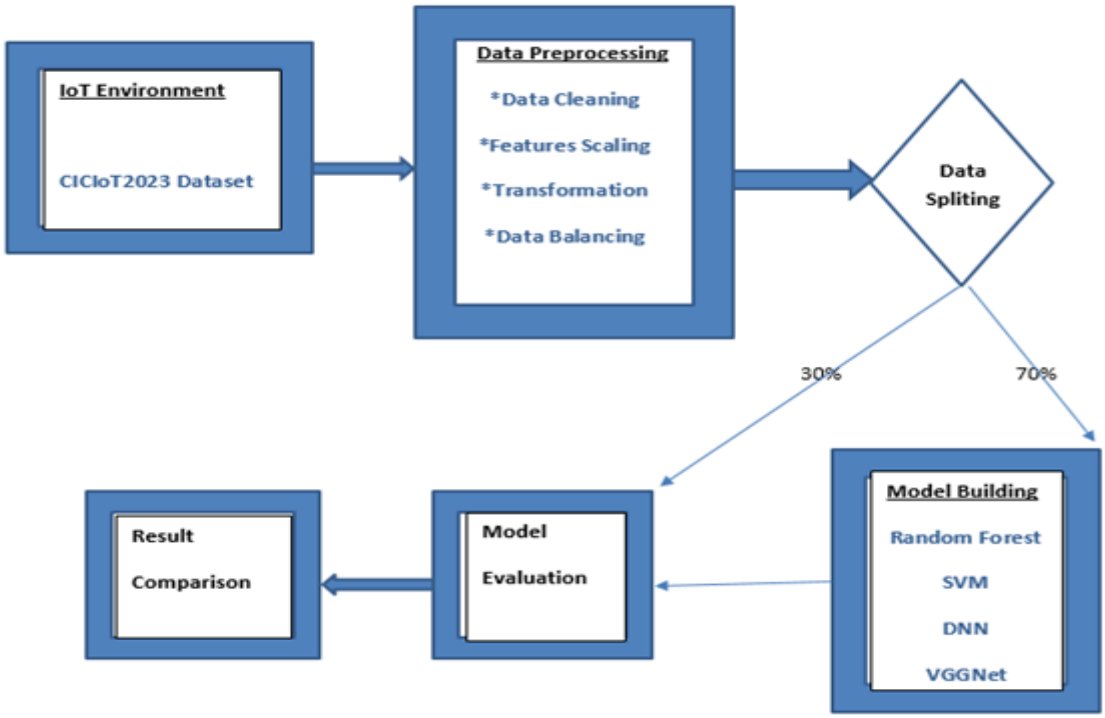


Figure 2: Proposed classifier model

3.1 Dataset selected

The completeness and quality of the datasets used for training and evaluation is greatly influence the proposed IDS. The Canadian Institute for Cybersecurity (CIC), which has a notable presence in the cybersecurity ecosystem and a track record of making significant contributions, provided the dataset utilized in this work, CICIoT2023[29].

On this IoT architecture, thirty-three different attacks is carried out as shown in (Figure 3). These assaults fall into seven categories: brute force, spoofing, Web-based, DDoS, DoS, Recon, and Mirai. The distribution details of dataset are illustrated in Table 2.

3.2 Preprocessing

The preprocessing steps significantly impact the quality of the model's input data, thereby influencing its performance. Effective data preprocessing is a crucial step in preparing the datasets for model training. The preprocessing workflow involves several key tasks:

Column name cleaning: Removing spaces from column names to ensure consistency and prevent errors during data manipulation.

Irrelevant data removal: Eliminating useless columns and rows that do not contribute to the model's predictive capability.

Handling missing values: removing missing values or incomplete rows to prevent biases and inaccuracies in the model. There are various methods used for handling missing values. In our contribution, we have used the Median imputation.

Duplicate removal: Ensuring data integrity by removing duplicate records to avoid overfitting.

Label mapping: clear label mapping ensures reproducibility and interpretability. In the CICIoT2023 dataset used, the number of classes is large, we have considered using Target Encoding to reduce dimensionality.

Data balancing: Adjusting the dataset to prevent model bias towards any particular class. This is typically achieved through techniques such as oversampling the minority class or under sampling the majority class to ensure a balanced representation of each class.

In this study, to resolve this issue of data balancing, we have used SMOTE method, which is an oversampling technique that generates synthetic data for the minority class by interpolating between existing samples and their neighbors, effectively increasing the number of minority class samples as illustrated in (Figure 4 and Figure 6).

Feature scaling and transformation: Normalizing the data make sure that every feature adds the same amount to the models learning. This often involves scaling features to a standard range, such as [0, 1], and applying transformations to stabilize variance and enhance model performance. In the study [35], authors provide a comparative analysis of various

normalization methods, while another study [36] demonstrate the effectiveness in applying z_score normalization for classification improvement. In this study, we have used the standardization method or called also (Z_Score normalization) defined by the following formula (1):

$$\text{Z_Score} = \frac{X-u}{\sigma} \quad (1)$$

Where: X = raw feature value, μ = mean of the feature, and σ = standard deviation of the feature

Data splitting: Dividing the dataset into training and testing sets to evaluate the model's performance. In this contribution, we have used 70% of data for training and 30% for testing.

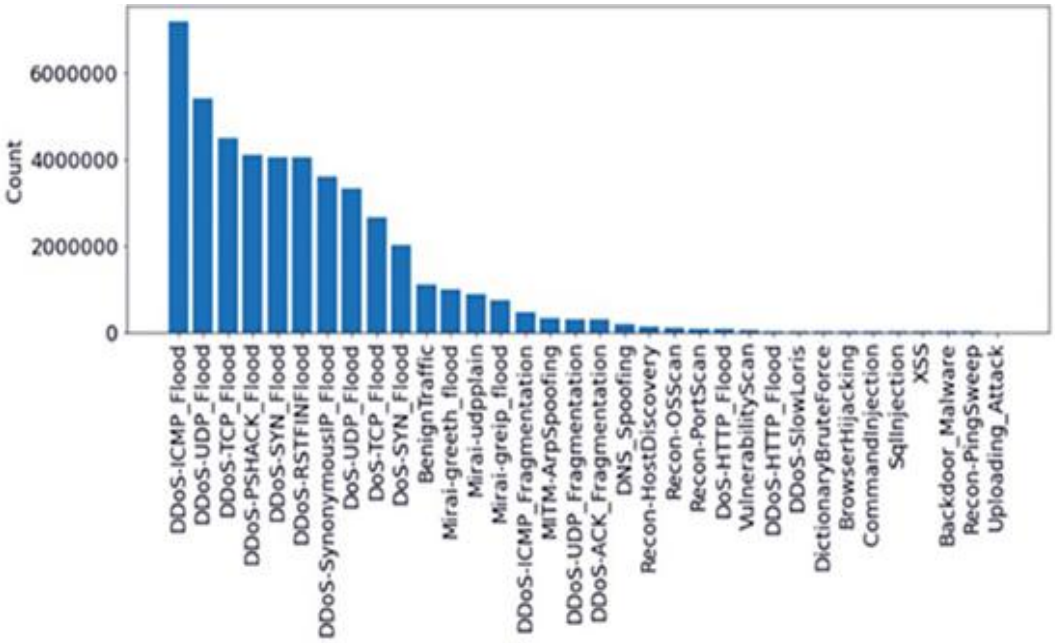


Figure 3: The instances count for each attack (Neto et al. (2023))

Table 2: Description of CICIoT2023 dataset (Neto et al. (2023)

Type	Target	Total Number of Records	Class Distribution
Benign	Benign	1,098,195	2.35%
DDoS	Attack	33,984,560	72.79%
DoS	Attack	8,090,738	17.33%
Mirai	Attack	2,634,124	5.64%
Recon	Attack	354,565	0.76%
Spoofing	Attack	486,504	1.04%
WebBased	Attack	24,829	0.05%
Bruteforce	Attack	13,064	0.03%
Total		**46,686,579**	**100%**

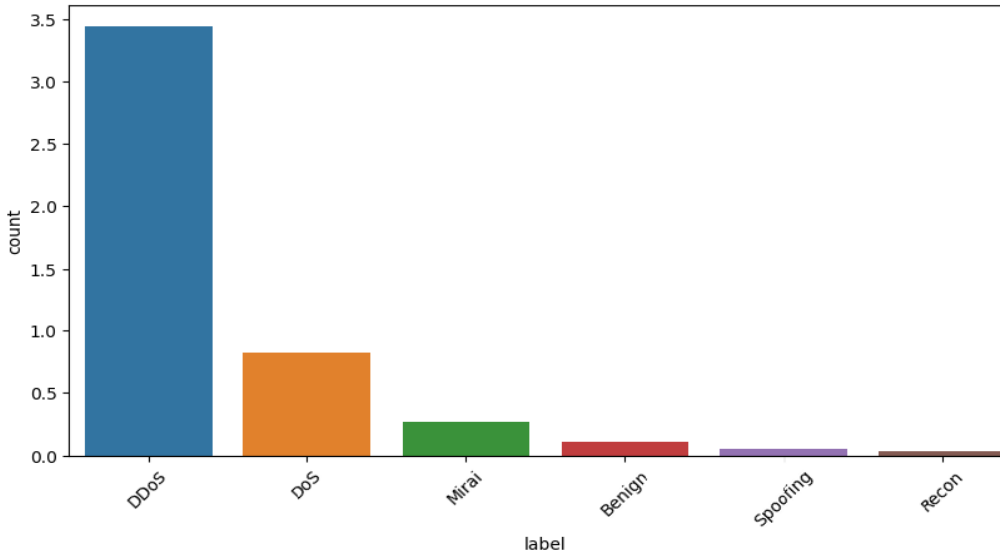


Figure 4: Data distribution of CICIoT2023 before balancing

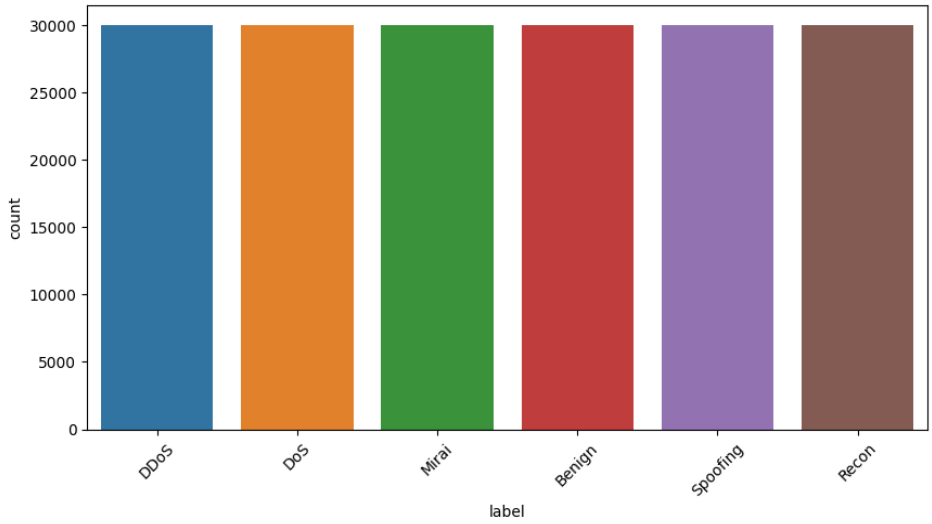


Figure 5: Data distribution after balancing

In Figure 6, feature analysis that displays how each dataset feature relates to other characteristics. The degree of relationships between features is provided by the correlation matrix. The highest relationship in these relationships is denoted by 1 and the lowest by -1.

3.3 Model building

In this study, we have chosen various learning models (Machine Learning models and Deep Learning models) including Support Vector Machine (SVM), Random Forest (RF), Deep Neural Network (DNN) and VGGNet. These learning models have been selected according to their high accuracy rate in classification in various research domains.

The optimization of those models is based on choosing the best hyperparameters. There are various techniques such as: Grid Search, RandomizedSearch and Bayesian Optimization.

In this study we have applied randoizedSearch provided by scikit-learn to get the best hyperparameters for SVM and Random Forest models. This technique is the most adequate and efficient when we have a high dimensional data.

- **Support Vector Machine:** Is a supervised classifier that can be used for classification or regression problems, it's main goal is to discover a hyperplane that maximizes the margin between classes while simultaneously minimizing classification errors (Hiri, M et al. 2023) [30]. This hyperplane is defined by the equation (2):

$$\boldsymbol{w} * \boldsymbol{x} + \boldsymbol{b} = \mathbf{0} \qquad \textbf{(2)}$$

Where: w is the weight vector (perpendicular to the plane), x is the input vector (training data) b is the bias (constant term)

In this simulation, The SVM model was built using the SVC class from the **sklearn.svm** module and the parameters were optimized for the best performance are (Table 3):

Regularization: C=2.0, Kernel=" rbf", and Gamma="auto"

- **Random Forest:** Is the famous supervised learning model, used for both: classification and regression. It can be defined as an ensemble method uses multiple decision trees. The principle of this model revolves around the idea that: by combining the predictions of each tree in the forest, we can avoid overfitting and we can improve the accuracy rate of classification.

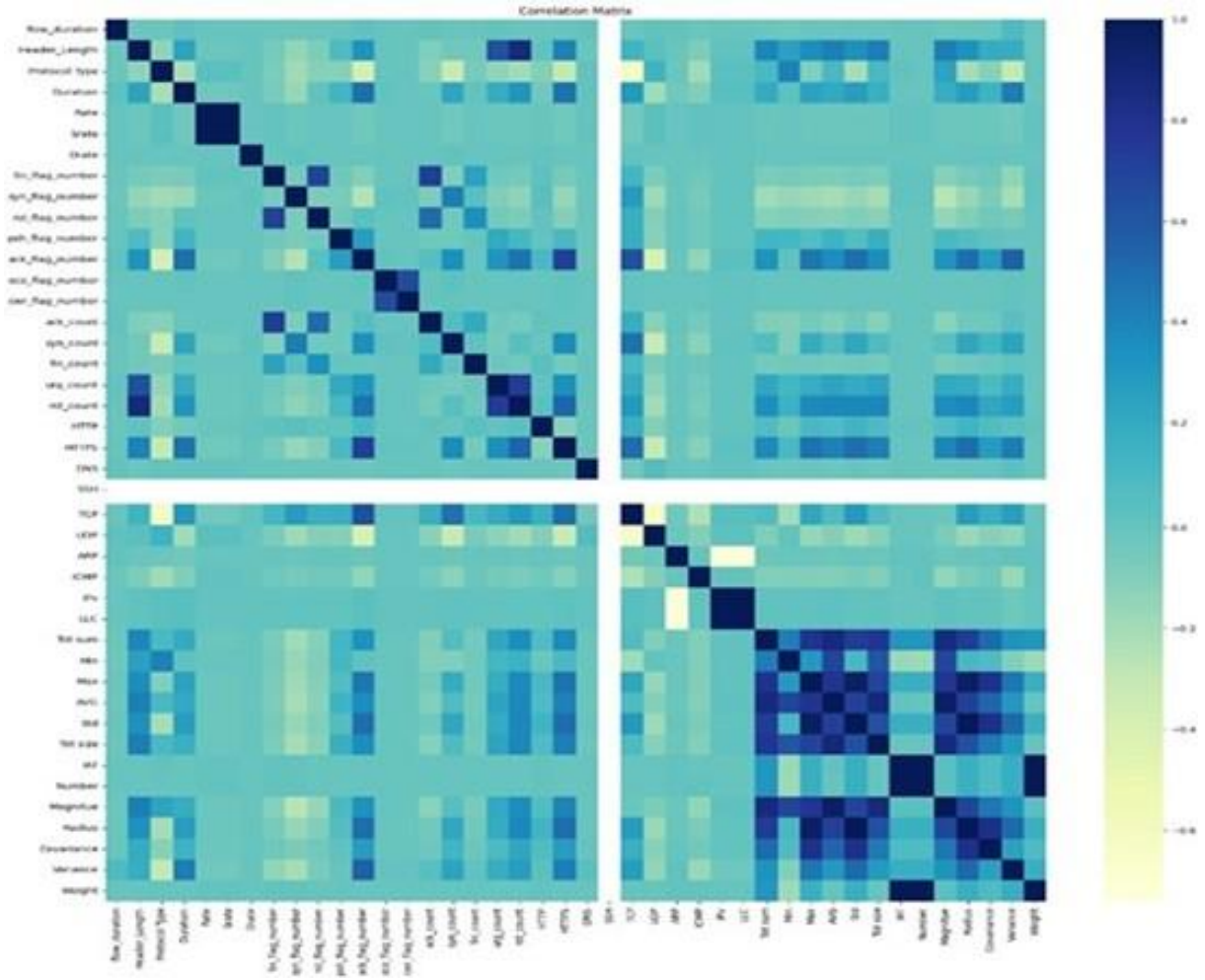


Figure 6: Correlation matrix results for the CICIoT2023 dataset

According to many research, random forest is a powerful machine learning model, because it can handle a large and huge amount of data. The Random Forest model was built using the RandomForestClassifier class from the **s**klearn.ensemble module and the parameters were optimized for the best performance (as illustrated in Table3):

Number of trees in the forest: n_estimators=200,

Maximum depth of the tree: max_depth=10

Deep Neural Network: Deep learning is a branch of ML that utilizes artificial neural networks for classification. It involves representing data in several layers, from a lower to a higher layer, extracting important features and more complex patterns in data to get the best possible outcome (Abbas, A et al 2022) [31]. Deep Neural Networks works as follows:

Input layer: is the first layer, where, each node represents a feature in the data.

Hidden layer: These layers connect the input and output layers to allow the network to learn intricate features and representations of the data. The more layers and neurons we implement, the deeper and more complex the network becomes.

Activation function: it's a function applied by each neuron, to affect it a corresponding weight

Output layer: which represents the final prediction of the model

Flatten layer: it's used to convert multidimensional input into a 1D vector before feeding it to dense layers

Dropout Layer: it's used to reduce overfitting by randomly deactivating a portion of neurons during training.
In this study, Figure 8 illustrate the architecture used for DNN.

Table 3: Summary of hyperparameters used

Model	Range of Hyperparameters	Best Hyperparameters used	Module
SVM	* Kernel Types: Linear, Polynomial Radial Basis Function (RBF): Sigmoid * Gamma: *C(Regularization Parameter) range is 10^{-3} to 10^{3}	Regularization: C=2.0 *Kernel=" rbf" *Gamma="auto	**SVC** class from the **sklearn.svm**
Random Forest	* **n_estimators**: Number of trees in the forest. Range: Usually between **10 to 1000** *** max_depth**: Maximum depth of each tree. Range: Typically between **1 and 50**	*Number of trees in the forest: n_estimators=200 *Maximum depth of the tree: max_depth=10	**RandomForestClassifier** class from the **sklearn.ensemble**
VGGNet	* range of Batch Size is 16 to 256 * range of Epoch is 10 to 100 *Dropout Rate range is 0.3 to 0.5 *Pooling Layers: Type: Max pooling Kernel size: 2x2	*three blocks of convolutional layers *2 fully connected layers with dropout layers in between. *The VGGNet model was trained for 10 epochs with a batch size of 64	**tensorflow.keras.models**
DNN	***** Batch Size Range**: 32, 64, 128, 256, 512** ***** Number of Layers Range**: 2 to 50+** ***** Activation Function**: ReLU, Leaky ReLU, Sigmoid, Tanh, ELU** ***** Optimizer**: SGD, Adam, RMSprop,**	*The DNN model was trained for 10 epochs ***Optimizer**: Adam * **Batch Size**: 64 * **Number of Layers**: 3 ***activation functions**=ReLu * **Dropout regularization**	**tensorflow.keras.models**

```
Epoch 1/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 121s 53ms/step - accuracy: 0.7350 - loss: 0.6194 -
Epoch 2/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 116s 51ms/step - accuracy: 0.7698 - loss: 0.5044 -
Epoch 3/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 117s 52ms/step - accuracy: 0.7715 - loss: 0.4963 -
Epoch 4/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 116s 52ms/step - accuracy: 0.7725 - loss: 0.4911 -
Epoch 5/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 114s 51ms/step - accuracy: 0.7762 - loss: 0.4837 -
Epoch 6/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 115s 51ms/step - accuracy: 0.7824 - loss: 0.4762 -
Epoch 7/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 114s 51ms/step - accuracy: 0.7948 - loss: 0.4594 -
Epoch 8/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 115s 51ms/step - accuracy: 0.8279 - loss: 0.4093 -
Epoch 9/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 115s 51ms/step - accuracy: 0.8483 - loss: 0.3660 -
Epoch 10/10
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 113s 50ms/step - accuracy: 0.8513 - loss: 0.3536 -
```

Figure 7: Architecture used for DNN model

```
dnn_model.summary()
```

```
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 46, 256)	512
dropout_6 (Dropout)	(None, 46, 256)	0
dense_10 (Dense)	(None, 46, 256)	65,792
dropout_7 (Dropout)	(None, 46, 256)	0
flatten_1 (Flatten)	(None, 11776)	0
dense_11 (Dense)	(None, 6)	70,662

```
Total params: 136,966 (535.02 KB)
Trainable params: 136,966 (535.02 KB)
Non-trainable params: 0 (0.00 B)
```

Figure 8: Epoch used for DNN model

```
vgg_model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 46, 64)	256
conv1d_1 (Conv1D)	(None, 46, 64)	12,352
max_pooling1d (MaxPooling1D)	(None, 23, 64)	0
conv1d_2 (Conv1D)	(None, 23, 128)	24,704
conv1d_3 (Conv1D)	(None, 23, 128)	49,280
max_pooling1d_1 (MaxPooling1D)	(None, 11, 128)	0
conv1d_4 (Conv1D)	(None, 11, 256)	98,560
conv1d_5 (Conv1D)	(None, 11, 256)	196,864
max_pooling1d_2 (MaxPooling1D)	(None, 5, 256)	0
flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 256)	327,936
dropout (Dropout)	(None, 256)	0

Figure 9: VGGNet Architecture

DNN is considered as sequential modeles of layers and Dropout as illustrated in Figure 7.

Dropout regularization is a method for decreasing overfitting and enhancing deep neural network generalization. Fixed dropout for each unique model is taken into consideration for increased performance, which is assessed based on accuracy and assessment time (as shown in Table 3).

The number of epochs is a crucial algorithmic hyperparameters. It indicates how many iterations, or full runs of the training dataset, the algorithm will go through during training or learning. Figure 8 shown the number of epoch used in our simulation.

VGGNet: is a deep learning architecture, developed by the Visual Graphics Group (VGG) at Oxford University. Originally designed for image classification, VGGNet can be adapted for other complex tasks, including intrusion detection by transforming data into a format 2D that the model can process. This approach has been inspired by recent study introduced by authors (Bouijij, H et al (2025)) [32] that have successfully proposed an approach-based CNN architectures for non-image domains (phishing attack detection).

The architecture explores the relationship between network depth and performance, utilizing small convolution kernels and maximum pooling layers to build deep networks comprising 11 to 19 layers (Zhou, S et al. 2018) [33] as illustrated in Figure 9:

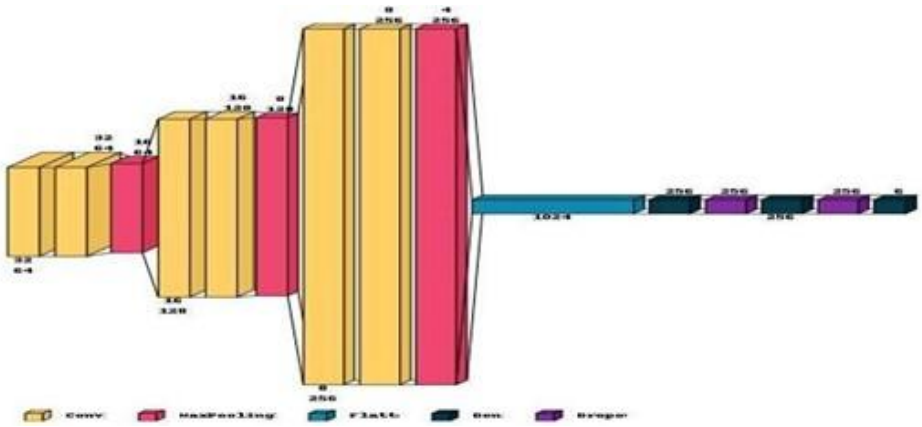


Figure 10: Architecture used for VGGNet model

Various studies have introduised VGGNet model for intrusion detection (Manjula, P et al, 2022) [34]. In this study, Figure 10 shown the architecture used for VGGNet, with a number of the epoch used to train it. It was a lightweight adaptation inspired by VGGNet**,** designed to reduce training time and computational cost for the CICIoT2023 dataset. The model was built using the Sequential class from the keras. model's module. It generally consists of three blocks of convolutional layers followed by max-pooling layers, and then 2 fully connected layers with dropout layers in between (as illustrated in Table 3). The VGGNet model was trained for 10 epochs with a batch size of 64.

3.4 Evaluation metrics

To assess the efficiency of our proposed approach, we have used a wide range of metrics as shown in Table 4. These metrics are based on the values: False Positive (FP): indicates that a normal class is wrongly predicted as an attack False Negative (FN): is false alarm, generated when the actual class is an attack, but predicted as normal. (TP): is a successful identification of a class as attack. (TN): indicates no attack occurred and no alarm generated.

The evaluation also includes the execution time as a key metric to analyze the practicality of each model in real-time IoT environment.

Table 4: Summary of the standard evaluation metrics used

Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1-score	$\frac{2TP}{2TP + FN + FP}$

4 Results and discussion

In this section, we present the experiment result of our chosen learning modeles using a real-time and benchmark dataset CICIoT2023, as well as a summary of comparison between their results. The assessment of the research findings was carried out meticulously, employing a variety of measures to measure the effectiveness of the three modeles in detecting various types of attacks, such as Denial of Service (DoS) and DDoS attacks, particularly in an IoT environment.

SVM result: Table 5 and Figure 11 illustrate summary of the experiment result of the SVM model

Table 5: Accuracy report of SVM model

SVM Classification Report	precision	recall	f1-score	support
Benign	0.69	0.85	0.76	5933
DDoS	0.84	0.63	0.72	5976
DoS	0.70	0.88	0.78	5922
Mirai	1.00	0.99	1.00	6086
Recon	0.79	0.75	0.77	6119
Spoofing	0.82	0.69	0.75	5964
accuracy	0.80022222			
macro avg	0.81	0.80	0.80	36000
weighted avg	0.81	0.80	0.80	36000

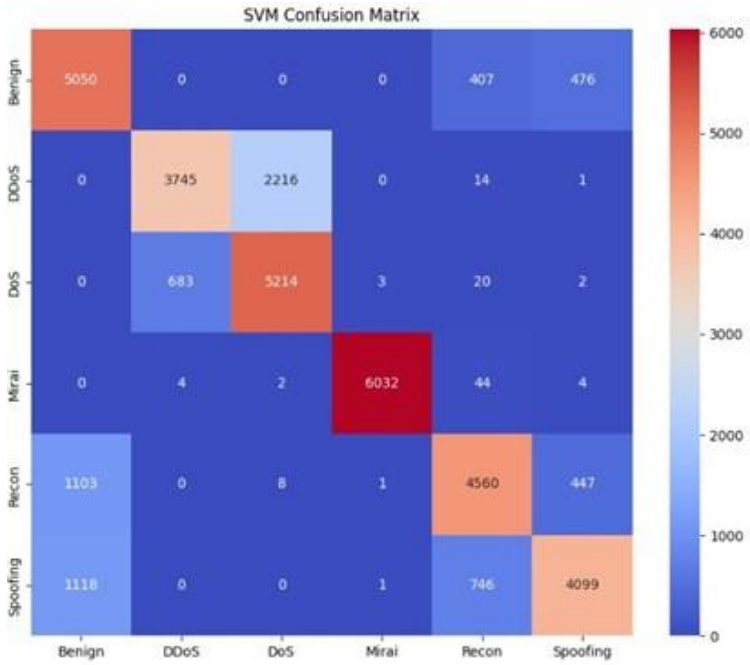


Figure 11: Confusion matrix of SVM model

According to the report accuracy (Table 5), the SVM model does not achieve a good accuracy to classify Begnin, DoS, DDoS and Recon.

SVM in this case is not a good classifier, because SVMs are inherently designed for binary classification and also SVMs are not ideal for datasets with a very large number of trainings. The use of SVM in this study has as objective demonstrating the limitation of traditional ML models in comparison with DL approaches.

- **Random Forest result**

Table 6 and Figure 12 illustrate a detailed summary of the experiment result of the Random Forest model.

According to the accuracy report (Table 6) and the confusion matrix (Figure 12), Random Forest model achieved strong overall performance but still exhibited some misclassifications. While Random Forest performed well, it did not achieve perfect accuracy, especially traffic classes. Generally, Random Forest is a good classifier for several reasons:

- The combination of predictions of multiple Decision Trees in the forest improve the accuracy rate
- The RF model is suitable when the datasets contains huge amount of data and many features, because the model is able to select the best features for each tree.

Table 6: Accuracy report of random forest model

Random Forest Classification Accuracy: 0.918138888888				
	precision	**recall**	**f1-score**	**support**
Benign	**0.81**	**0.88**	**0.84**	**5933**
DDoS	**1.00**	**0.99**	**1.00**	**5976**
DoS	**1.00**	**0.99**	**1.00**	**5922**
Mirai	**1.00**	**0.99**	**1.00**	**6086**
Recon	**0.81**	**0.84**	**0.82**	**6119**
Spoofing	**0.90**	**0.81**	**0.86**	**5964**
accuracy	**0,92**			
macro avg	**0.92**	**0.92**	**0.92**	**36000**
weighted avg	**0.92**	**0.92**	**0.92**	**36000**

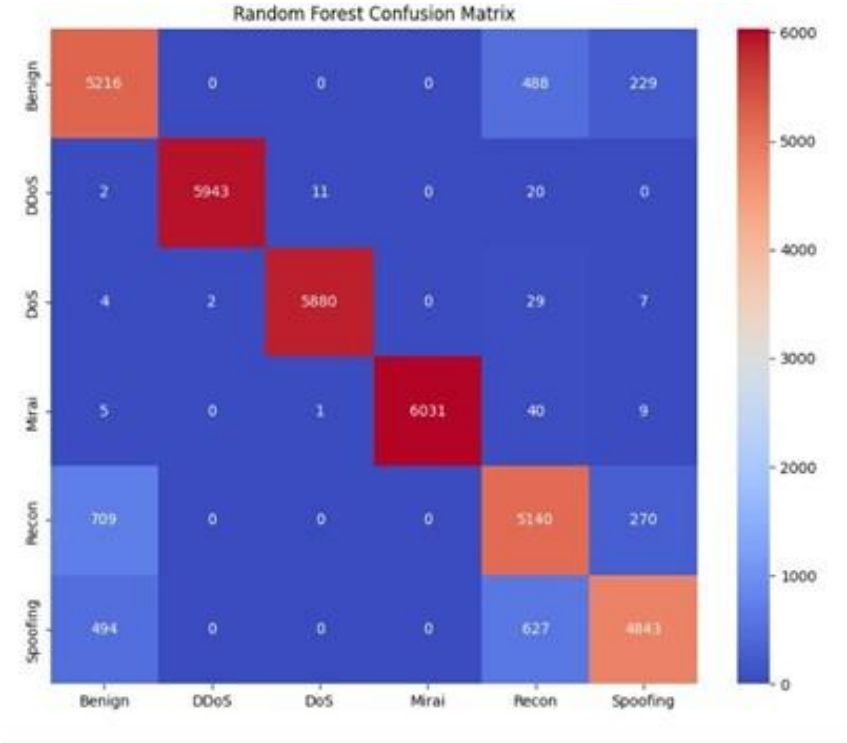


Figure 12: Confusion matrix of Random Forest model

Table 7: Accuracy report of VGGNet model

VGGNet Classification Accuracy: 0.89275				
	Precision	**recall**	**f1-score**	**support**
Benign	**0.75**	**0.87**	**0.81**	**5933**
DDoS	**1.00**	**0.99**	**1.00**	**5976**
DoS	**1.00**	**1.00**	**1.00**	**5922**
Mirai	**1.00**	**1.00**	**1.00**	**6086**
Recon	**0.78**	**0.80**	**0.79**	**6119**
Spoofing	**0.85**	**0.70**	**0.77**	**5964**
accuracy	**0.89**			
macro avg	**0.90**	**0.98**	**0.89**	**36000**
weighted avg	**0.90**	**0.98**	**0.89**	**36000**

- **VGGNet result**

Table 7 and Figure 13 below, provide a detailed summary of the experiment result of the VGGNet model

According to the report accuracy (Table 7), the VGGNet model follow the random forest model as a second-best classifier, with an accuracy rate of 89,27%. VGGNet have successfully classified almost all the traffic, especially DoS, DDoS and Mirai attacks.

VGGNet is a Deep Learning models excels in feature extraction from data, but reliance on convolutional layers may limit its capacity to generalize successfully. VGGNet's somewhat lower overall accuracy indicates that it might have trouble differentiating innocuous traffic from more nuanced or changing attack patterns, even while it is highly effective at classifying DoS, DDoS, and Mirai attacks. However, Random Forest's ability to alter its decision boundaries enables it to function reliably against different kinds of attacks.

Furthermore, since decision tree-based models are better at capturing hierarchical relationships in tabular datasets, they may be more appropriate for the structured nature of network traffic data. Convolutional neural networks (CNNs), such as VGGNet, on the other hand, are better suited for the extraction of spatial features, which makes them very useful for tasks involving images but possibly less effective for categorical and sequential data.

Table 8 and Figure 14 provide a detailed summary of the experiment result of the DNN model. DoS, DDoS and Mirai have been Well classified by DNN, according to all the measurement metrics.

- **DNN result**

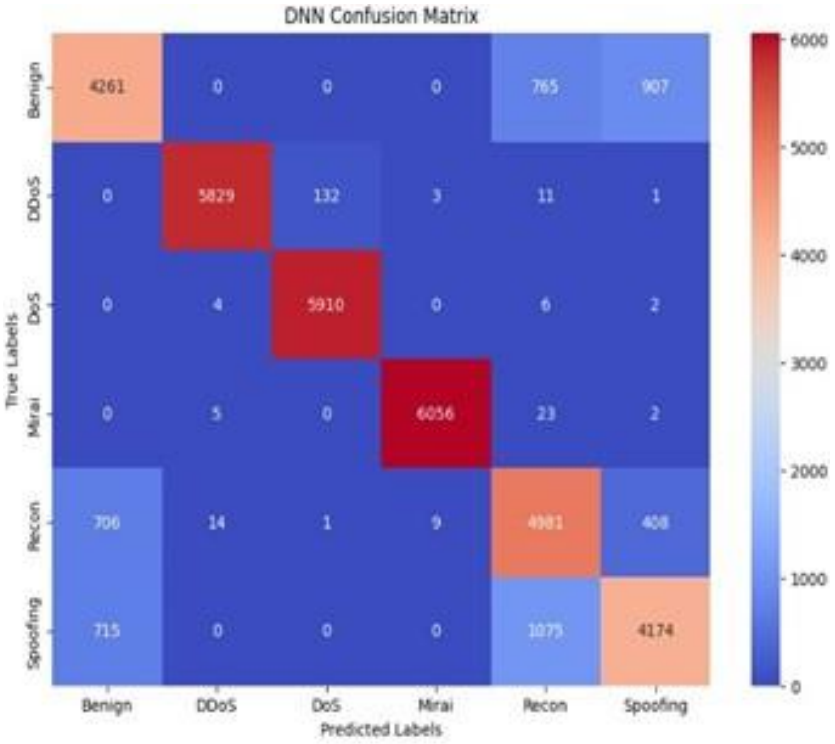


Figure 13: Confusion matrix of VGGNet model

Table 8: Accuracy report of DNN model

DNN Classification Report	precision	recall	f1-score	support
Benign	0.75	0.72	0.73	5933
DDoS	1.00	0.98	0.99	5976
DoS	0.98	1.00	0.99	5922
Mirai	1.00	1.00	1.00	6086
Recon	0.73	0.81	0.77	6119
Spoofing	0.76	0.70	0.73	5964
accuracy	0.86697222222222			
macro avg	0.87	0.87	0.87	36000
weighted avg	0.87	0.87	0.87	36000

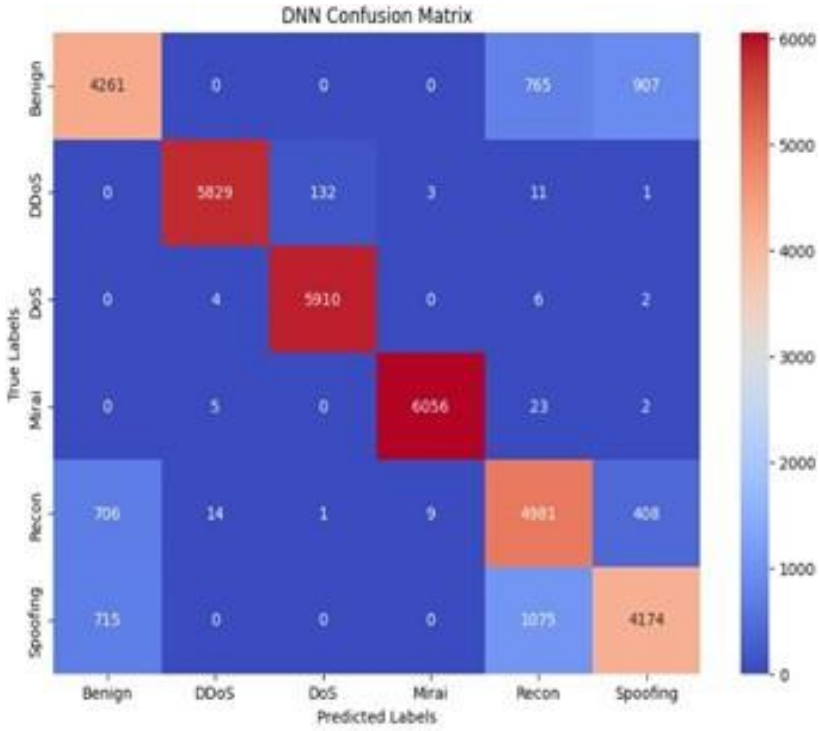


Figure 14: Confusion matrix of DNN model

According to the accuracy report (Table 8), DNN model have proven to be a good classifier, achieving a high accuracy in classifying DoS, DDoS and Mirai attacks.

Unlike traditional machine learning modeles, DNN and thanks to its multi-layers, it can learn from complex data and extract the most significant features even with big datasets with a large amount of data, this ability make DNN as a good classifier

As summary

Our study, underscore several key advancements over prior works in both methodological rigor and empirical performance. In contrast to earlier studies that often relied on outdated datasets or limited benchmarking. Our approach integrates a broader and more contemporary range of test scenarios. Notably, prior models frequently underperformed due to insufficient hyperparameters

We can conclude that this study, provide an effecient outfinding in comparison with the study provided by (Dushimimana, A. et al. (2020)), who has used a small portion of the NSL-KDD, consequentely this work does not provide a general result

(Ayyaz-ul-Haq Qureshi, B. et al (2019)) and (Jiang, K. et al. (2020)), have evaluated various algorithms of classification, however, attackers enhance their behavioral continuously, consequently, not using a recent and real-time dataset doesn't provide an efficient result.

The work by Berqia et al. (2024) and this study are comparable in a number of ways, especially in their emphasis on applying machine learning methods to identify DDoS attacks in Internet of Things settings using tuning—a limitation we explicitly address through a systematic and automated optimization process across all major model parameters.

According to the experiment result, all the model building demonstrate significant performance in classifying normal and abnormal activities within IoT environment (Table 9). Our work builds on and extends the capabilities of previous studies.

The study presented by (Chirra, D. R. (2023)), explored deep learning models for enhancing IoT network intrusion detection. While deep learning models can offer performance gains, they also introduce complexity and require more computational resources.

While, the result obtained by (El Yamani et al. (2024)) and (Pokhrel, S et al. (2021)) achieve a high accuracy rate of attacks classification, however the complexity in term of execution time and computational the same dataset CICIoT2023. To improve model performance and generalizability, we use sophisticated hyperparameter tweaking strategies in our work, which sets it apart.

Our study provides several advantages:

- High Accuracy in detecting a wide range of network intrusions.
- Scalability: That making it suitable for real-time intrusion detection.
- Resource Efficiency: Optimized for deployment in resource-constrained environments such as IoTs, where computational and energy resources are limited.

These advancements highlight the potential of machine and deep learning models to significantly, enhance the robustness and reliability of IDS in protecting modern network infrastructures against evolving cyber threats.

Despite the promising results, our study has certain limitations. One of the primary challenges is the computational complexity associated with deep learning models, which requires significant processing power and memory. VGGNet and DNN models involve many parameters during training, as well as, while hyperparameters tuning significantly enhance the accuracy rate, but often require a lot of processing power (CPU and storage). This can be a constraint in the context of IoT, where resources devices are typically limited.

Table 9: Summary of evaluation results

Model	Approach	Accuracy	Precision	Recall	Training time(s)
SVM	Machine learning	80,02%	81%	80%	2400
Random Forest	Machine learning	91,81%	92%	92%	1800
VGGNet	Deep learning	89,27%	90%	89%	3400
DNN	Deep learning	86,69%	87%	87%	2400

5 Conclusions

IoT technology still attracked researchers thanks to its proficiency of changing physical items of diverse application domains into Internet hosts. There are now more security and privacy concerns as a result of the Internet of Things' exponential growth. Device vulnerabilities brought on by hacker cybercrime and inappropriate system resource usage are the main cause for a large number of these threats. However, attackers may also take advantage of the IoT tremendous potential as a new means to harm users' privacy and security. In this paper, we have evaluated various machine and deep learning models including Support Vector Machine, Random Forest, VGGNet and Deep Neural Network, using a well-known, benchmark and real time dataset CICIoT2023. Various evaluation metrics have been used to analyze in depth our approach. The experiment results demonstrate the robustness of Random Forest model among the other entire model in classifying normal and abnormal traffic. Finally, this research paper highlights the robustness of machine and deep learning models to transform the landscape of intrusion detection, as well as finding the best hyperparameters to fine-tuning the chosen models. The advancements presented in this research facilitate and smooth the way for more safe and robust network systems, capable of resistant the challenges posed by the ever-evolving cyber threat land- scape. Future research should focus on addressing the limitations identified in this study, by exploring techniques such as generative adversarial networks and federated Learning approach for optimizing the Deep Learning architecture and deployment in resource-constrained environments.

To facilitate the reproducibility and further research, the implementation details of the proposed framework is publically available on the following GitHub Repository: https://github.com/users/FAREShajar-doc/Ml-and-Dl-for-Intrusion-Detection

References

[1] Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A literature review. Journal of Computer and Communications, 3(5), 164-173.

[2] Zhao, J. (2025). Intelligent Logistics Path Optimization Algorithm Based on Internet of Things Sensing Technology. Informatica, 49(19).

[3] De Michele, R., & Furini, M. (2019, September). Iot healthcare: Benefits, issues and challenges. In Proceedings of the 5th EAI international conference on smart objects and technologies for social good (pp. 160-164).

[4] Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. (2019). A review of machine learning and IoT in smart transportation. Future Internet, 11(4), 94

[5] Jeong, Y. S., & Park, J. H. (2019). IoT and smart city technology: challenges, opportunities, and solutions. Journal of Information Processing Systems, 15(2), 233- 238

[6] Farooq, M. S., Riaz, S., Abid, A., Umer, T., & Zikria, Y. B. (2020). Role of IoT technology in agriculture: A systematic literature review. Electronics, 9(2), 319.

[7] Moslehi, M. M. (2025). Exploring coverage and security challenges in wireless sensor networks: A survey. Computer Networks, 111096.

[8] Abomhara, M., & Køien, G. M. (2015). Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. Journal of Cyber Security and Mobility, 65-88

[9] Zarpelˆao, B. B., Miani, R. S., Kawakani, C. T., & De Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. Journal of Network and Computer Applications, 84, 25-37.

[10] Huang, H., Xiao, D., Wang, M., Liang, J., Li, M., Chen, L., & Liu, Y. (2025). Fog-driven communication-efficient and privacy-preserving federated learning based on compressed sensing. Computer Networks, 259, 111043.

[11] Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learn- ing in IoT security: Current solutions and future challenges. IEEE Communications Surveys & Tutorials, 22(3), 1686-1721.

[12] Du, J., Wang, X., & Zhang, H. (2025). Secure Power Management in Wireless Sensor Networks for Power Monitoring Using Deep Reinforcement Learning. Informatica, 49(19).

[13] Amanullah, M. A., Habeeb, R. A. A., Nasaruddin, F. H., Gani, A., Ahmed, E., Nainar, A. S. M., ... & Imran, M. (2020). Deep learning and big data technologies for IoT security. Computer Communications, 151, 495-517

[14] Nahida Islam1, Fahiba Farhin1 , Ishrat Sultana1 , M. Shamim Kaiser1 , Md. Sazzadur Rahman1 , Mufti Mahmud2 , A. S. M. Sanwar Hosen3 and Gi Hwan Cho. Towards Machine Learning Based Intrusion Detection in IoT Networks. Computers, Materials & Continua. DOI:10.32604/cmc.2021.018466

[15] Lin, T. (2020, November). Deep learning for IoT. In 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC) (pp. 1-4).

[16] Li, Y., Zuo, Y., Song, H., & Lv, Z. (2021). Deep learning in security of internet of things. IEEE Internet of Things Journal, 9(22), 22133-22146.

[17] Abbass, W., Bakraouy, Z., Ba¨ına, A., & Bellafkih, M. (2018, October). Classifying IoT security risks using deep learning algorithms. In 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). IEEE

[18] Pokhrel, S., Abbas, R., & Aryal, B. (2021). IoT security: botnet detection in IoT using machine learning. *arXiv preprint arXiv:2104.02231*.

[19] Ioannou, C., & Vassiliou, V. (2019, May). Classifying security attacks in IoT networks using supervised learning. In 2019 15th International conference on distributed computing in sensor systems (DCOSS) (pp. 652-658). IEEE.

[20] Chirra, D. R. (2023). Deep Learning Techniques for Anomaly Detection in IoT Devices: Enhancing Security and Privacy. Revista de Inteligencia Artificial en Medicina, 14(1), 529-552

[21] Berqia, A., Bouijij, H., Merimi, A., & Ouaggane, A. (2024, May) Detecting DDoS attacks using machine learning in IoT environment. In 2024 International Conference on Intelligent Systems and Computer Vision (ISCV) (pp. 1-8). doi: 10.1109/ISCV60512.2024.10620122.

[22] El Yamani, Y., Fadili, Y., Kilani, J., El Kamoun, N., Baddi, Y., & Bensalah, F. (2024, July). Hybrid Models for IoT Security: Tackling Class Imbalance. In 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). doi: 10.1109/WINCOM62286.2024.10656654.

[23] Sarhan, M. Layeghy, S. Moustafa, N. Gallagher, M, Portmann, M. (2024). Feature extraction for ML based Intrusion Detection in IoT networks. Digital Communications and Networks. No 10 (2024): 205-216

[24] Elouardi, S., Motii, A., Jouhari, M., Amadou, A. N. H., & Hedabou, M. (2024). A survey on Hybrid-CNN and LLMs for intrusion detection systems: Recent IoT datasets. *IEEE Access*

[25] B. Ayyaz-ul-Haq Qureshi, H. Larijani, J. Ahmad and N. Mtetwa, "A heuristic intrusion detection system for internet-of-things (IoT)," in Intelligent Computing: Proc. of the 2019 Computing Conf., London, United Kingdom, pp. 86–98, 2019

[26] K. Jiang, W. Wang, A. Wang and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," IEEE Access, vol. 8, pp. 32464–32476, 2020

[27] A. Dushimimana, T. Tao, R. Kindong and A. Nishyirimbere, "Bi-directional recurrent neural network for intrusion detection system (IDS) in the internet of things (IoT)," International Journal of Advanced Engineering Research and Science, vol. 7, no. 3, pp. 524–539, 2020.

[28] Z. Li, P. Batta and L. Trajkovic, "Comparison of machine learning algorithms for detection of network intrusions," in 2018 IEEE Int. Conf. on Systems, Man, and Cybernetics, Miyazaki, Japan, pp. 4248–4253, 2018

[29] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors, 23(13), 5941.

[30] Hiri, Mustafa; Chrayah, Mohamed; Ourdani, Nabil; Aknin, Noura. (2023). Machine Learning Techniques for Diabetes Classification: A Comparative Study. International Journal of Advanced Computer Science and Applications. 14. 10.14569/IJACSA.2023.0140982.

[31] Abbas, A. M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, (2022)"A new ensemble- based intrusion detection system for internet of things," Arabian Journal for Science and Engineering, Springer

[32] Bouijij, H., & Berqia, A. (2025). Intelligent phishing URL classification using CNN. In Recent Advances in Internet of Things Security (pp. 122-130). CRC Press.

[33] Zhou, S., Liang, W., Li, J., & Kim, J. U. (2018). Improved VGG model for road traffic sign recognition. Computers, Materials & Continua, 57(1), 11-24

[34] Manjula, P., & Priya, S. B. (2022). An effective network intrusion detection and classification system for securing WSN using VGG-19 and hybrid deep neural network techniques. Journal of Intelligent & Fuzzy Systems, 43(5), 6419-6432

[35] Sholeh, M., & Nurnawati, E. K. (2024, July). Comparison of Z-score, min-max, and no normalization methods using support vector

machine algorithm to predict student's timely graduation. In AIP Conference Proceedings (Vol. 3077, No. 1). AIP Publishing.

[36] Qi, D., Junaidi, A., Howe, C. W., & Zain, A. M. (2023, October). Improving Unbalanced Security X-Ray Image Classification Using VGG16 and AlexNet with Z-Score Normalization and Augmentation. In International Conference on Electronics, Biomedical Engineering, and Health Informatics (pp. 205-217). Singapore: Springer Nature Singapore.

Signal Reconstruction Algorithm Application Research under Compressed Sensing in Sparse Signal Reconstruction

Shimin Zhang
College of Artificial Intelligence and Big Data, Zibo Polytechnic University, Zibo 255000, China
E-mail: zhangsm1104@163.com

Keywords: compression perception, signal reconstruction, optimization-oriented, support set estimation, decentralization

Received: May 19, 2025

To improve the efficiency of compressed sensing sparse signal reconstruction, a reconstruction algorithm suitable for different scenarios is proposed. On the basis of greedy algorithm, a sparse reconstruction algorithm for optimization is constructed. A multi-source sparse signal reconstruction algorithm with improved support set estimation is proposed. Experimental data show that the mean square error of the optimized sparse signal reconstruction algorithm is less than 10^{-5}, which is 1-4 orders of magnitude smaller than other comparative algorithms (suh as orthogonal matching pursuit). The support set estimation accuracy of the joint sparse signal reconstruction algorithm is the highest. When the signal-to-noise ratio is 10, the relative reconstruction error based on the orthogonal matching tracking algorithm is 0.57. The minimum relative reconstruction error of the proposed joint sparse signal reconstruction algorithm is 0.34. The analysis of experimental data shows that the decentralized joint sparse signal reconstruction algorithm proposed in this paper not only ensures the efficiency of signal reconstruction, but also reduces the computational complexity.

Povzetek: Prispevek se ukvarja s kompresijskim zaznavanjem (CS) in rekonstrukcija redkih signalov. Novost je kombinacija OMP-backtrackinga, glasovanja skupne podpore in enkratne porazdeljene izmenjave za skupno SSR; potrebuje teoretično utrditev, realne podatke in reproducibilnost.

1 Introduction

Compressed Sensing (CS) signal processing technology is a new type of signal processing method that can transform traditional sampling theorems into more efficient signal reconstruction methods and has become a current research hotspot. Compared with traditional sampling theorems, CS technology can greatly reduce the number of samples and improve the efficiency of signal reconstruction. In traditional communication and signal processing frameworks, based on Shannon's theorem, the sampling frequency needs to be twice higher than the highest information frequency [1]. With the development of technology, the amount of data that communication nodes and devices need to process is very large. Traditional signal sampling can generate signal redundancy, leading to a great waste of resources [2]. Therefore, the development of CS technology, with its high sampling rate beyond Nyquist limits, addresses issues such as oversampling, data compression, and data loss, while also ensuring reliable data recovery [3]. CS technology can represent signals using only a small number of non-zero coefficients within an appropriate range, potentially reducing the cost of sampling and computation. It is widely used in various fields such as electronic technology and computer science [4]. In Sparse Signal Reconstruction (SSR), CS technology can achieve SSR through iterative algorithms such as greedy algorithms. Meanwhile, in distributed network scenarios, it is necessary to consider the processing of multi-source signals and utilize the correlation between data to complete joint SSR [5]. The current signal reconstruction approach fails to adequately leverage the inherent sparsity of the signal, leading to suboptimal reconstruction outcomes when handling sparse signals. Therefore, research has proposed joint SSR algorithms for single-sources and multi-sources, aiming to improve the reconstruction effect. The study includes four sections. The first section is a summary of relevant research, the second section is the design of SSR algorithms, the third section is the performance analysis of the proposed algorithm, and the fourth section is a summary of this study. This study proposes a joint SSR algorithm based on an improved public support set for the multi-source joint SSR problem. In SSR problems, oriented SSR algorithm (OSRA) employs global optimization techniques, encompassing support set pre-selection with backtracking and comprehensive global optimization, thereby exhibiting robust global search capabilities. Centralized joint SSR algorithm (CJSRA) focuses on improving the estimation of the public support set, by preliminarily estimating the public support set, optimizing the public support set, and using edge information-based orthogonal matching pursuit (OMP) algorithm to more accurately estimate the support set of sparse signals, thereby improving the accuracy and stability of signal reconstruction. In the problem of multi-source joint SSR, the combination of these two algorithms

can achieve efficient signal reconstruction, reduce the energy consumption of node transmission and reception of signals, and avoid the problem of repeated reception and transmission of signals.

The novelty of the proposed method in the public support set estimation of joint SSR lies in that CJSRA improves the support set estimation process of the traditional joint sparse algorithm through the voting strategy and error screening mechanism. Most of the existing methods rely on the direct intersection of single-source estimation and fail to effectively filter out random errors. However, CJSRA improves the reliability of the initial estimation through multi-source voting and further optimizes the support set through edge information OMP, solving the dependence problem of traditional methods on high observation rates. DJSRA proposes a distributed single information interaction mechanism, breaking through the limitation that existing distributed algorithms require multiple communications. The public support set is estimated through the collaborative estimation of neighborhood nodes without the coordination of a central node, which ADAPTS to the dynamic network topology. Moreover, the support set estimation process is compatible with the correlation of partially overlapping signals instead of forcing the assumption of complete sharing. Compared with the existing methods, the method proposed in this paper demonstrates better robustness in low observation rates, noise interference and distributed scenarios through hybrid support set modeling (public support set + innovative support set) and adaptive collaborative strategies, rather than merely continuing the concept of public support sets. The contribution of this study is mainly reflected in the following points. Firstly, for the problem of single-source SSR, the key difference between OSRA and traditional greedy algorithms (such as OMP and generalized orthogonal matching pursuit (gOMP)) lies in the introduction of a support set pre-selection with backtracking mechanism and a global optimization objective function. Traditional methods only rely on local search to iteratively update the support set, while OSRA verifies the reliability of the initial estimation through backtracking and exits the local optimum through the global objective function, significantly improving the accuracy and robustness of SSR. Secondly, in the aspect of multi-source joint SSR, the study proposes CJSRA and DJSRA based on improved public support set estimation. CJSRA can improve the accuracy of support set estimation, while DJSRA can reduce the computational complexity while ensuring the signal reconstruction effect, and reduce the time consumption of nodes repeatedly sending and receiving information. These algorithms provide more efficient solutions for SSR in different scenarios and enrich the knowledge system in this field. The drawback of OSRA and CJSRA algorithms themselves is that as sparsity increases, the computational complexity of the algorithms also increases accordingly.

2 Related works

CS, also known as sparse sampling, is a technique used in the field of electronic engineering signal processing to find sparse solutions for under-determined linear systems. It can be used to obtain and reconstruct sparse or compressible signals. Wang et al. proposed an energy-efficient collection method using CS to construct a data collection model for underwater acoustic sensor networks. The results indicated that this method could reduce the sensor energy consumption by 15% [6]. Wang et al proposed a triple image encryption and hiding algorithm that utilized a two-dimensional discrete wavelet transform to sparsely represent three grayscale images, and processed and compressed the measurement matrix [7]. The results showed that this method could improve encryption efficiency by 23%. Bai et al. proposed a non-convex CS method for detecting fan noise patterns in CS, utilizing the L-1/2 minimization algorithm to optimize the sensor array. Results showed that this method could improve the dynamic range by 10dB [8]. Kato et al. introduced compressive sensing into the human-order analysis to address the limitation that traditional methods are only applicable to steady-state vibrations, enabling the measurement of operational vibrations with rotational speed fluctuations (such as in automotive engines). The aluminum plate vibration experiment and engine application showed that this method could accurately reconstruct the vibration (modal guarantee criterion 0.98, mean square error less than 1%, etc.), was suitable for the full-field measurement of high-speed vibration of rotating machinery, and was expected to assist in signal recovery and data compression [9]. Kadhim et al. aimed at the image encryption requirements in remote working and learning. In their work, a multi-chaotic encryption algorithm based on block CS, Swin Transformer and Mustang optimization was proposed. Encryption was achieved through processes such as wavelet transform, block compression, chaotic diffusion and optimized scrambling. Experiments showed that the average information entropy of the algorithm was 7.9749 and the NPCR reached 99.5453%, etc., indicating that it was efficient and robust and could resist various attacks [10].

CS technology includes three key steps: signal sparse representation, measurement matrix construction, and reconstruction algorithm. Among them, the reconstruction algorithm is the core of compressive sensing technology, as it has a direct impact on the complexity and quality of signal reconstruction. Zeynali et al. proposed a new algorithm for reconstructing wireless sensor network (WSNs) data with spatio-temporal correlation using CS. A time-varying sliding window mechanism was adopted to dynamically adjust the window size and the number of measurements, so as to effectively utilize spatio-temporal correlation, balance the sampling rate and reduce the transmission cost. Simulation showed that this algorithm maximized the utilization of prior signal information through the distributed data window framework. Compared with other CS reconstruction methods, it achieved higher reconstruction accuracy with less transmission volume and was suitable for various WSN scenarios [11]. Zhao et al. proposed a smooth inertial neural dynamics method to reconstruct sparse signals through the processing of norm minimization problems. The research results indicated that under certain

conditions, this method could average sub-linear convergence speed and improve the efficiency of SSR and image restoration by 11.1% [12]. Lin et al. optimized the regularization parameters and iteration times under different sparsity ratios to address the issue of reduced quality in sparse imaging reconstruction. Experimental data showed that the quality of SSR using the best parameters was improved by 16.7% [13]. Sun et al. proposed an improved OMP method that combined a forward search strategy to find residual errors. Results showed that this method could improve the Signal-to-Noise Ratio (SNR) of signal reconstruction and reduce the computational complexity of measurement [14]. Li et al. proposed a variable step-matching tracking CS algorithm based on oblique projection, which estimated initial sparsity using the constrained isometry property and created a support set for the target signal. The results showed that this method was superior to traditional matching tracking algorithms and could reduce computational complexity by 13% [15]. Bayesian Compressive Sensing (BCS) is a signal acquisition and processing technique based on Bayesian theory. It is a new technology developed in the field of signal processing in recent years, with the basic idea of compressing signals before sampling and digitization. Liu et al. proposed a new method based on Bayesian CS algorithm for spectrum estimation of multiple frequency hopping signals with randomly omitted observations. By designing a specific bilinear time-frequency representation framework with a time-frequency kernel, the artifacts caused by cross terms and missed observations are effectively suppressed, while preserving the self-terms of the frequency hopping signal. By utilizing the redesigned structure aware Bayesian CS algorithm to process the kernels in the time-frequency domain, high-resolution frequency hopping signal spectrum estimation was achieved even in the absence of most data observations. The simulation results showed that the method was effective [16].

To sum up, many researchers have conducted extensive studies and designs on compressive sensing and SSR. However, traditional greedy reconstruction algorithms (such as OMP and segmented orthogonal matching pursuit (StOMP)) only update the support set through local iteration, which has the defects of overfitting and short-sightedness, and does not introduce a global optimization mechanism to escape the local optimum. The OSRA proposed in this paper compensates for the limitations of traditional algorithms in support set estimation through the support set preselection backtracking and global search strategies. The existing joint SSR algorithms (such as inner-outer support set pursuit algorithm (IOSSP) and edge information greedy pursuit algorithm (SIPP)) do not estimate the public support set of multi-source signals accurately enough and fail to make full use of the correlation between signals for support set optimization. In this paper, CJSRA improves the preliminary estimation and screening process of the public support set through the voting strategy and the OMP algorithm based on edge information, and enhances the accuracy of multi-source signal support set estimation. Distributed joint SSR algorithms (such as distributed parallel pursuit algorithm (DIPP)) rely on multiple round-trip communications between nodes to complete signal reconstruction, resulting in high energy consumption and delay. In this paper, DJSRA eliminates the reliance on the central node through a single information interaction and an improved public support set estimation method, and reduces the communication cost and computational complexity between nodes. Furthermore, Model-Based CS improves the reconstruction performance by explicitly utilizing the structured sparse priors of the signal (such as group sparsity and tree sparsity), but its dependence on the preset structure limits its applicability in multiple scenarios. BCS models sparsity through hierarchical priors and performs well at low observation rates, but it has high computational complexity and requires preset signal distributions. Although structural sparsity methods (such as the Joint Sparse Model (JSM)) can capture the correlations of multi-source signals, most of them assume that the support sets are completely shared and are difficult to adapt to actual scenarios with partial overlap or dynamic changes. The hybrid support set model and the improved support set estimation method proposed in this paper do not require strict prior assumptions. They can model the correlation of multi-source signals more flexibly and reduce the computational and communication overhead through a distributed collaborative mechanism at the same time.

3 Design of SSR Algorithm under CS

This chapter mainly proposes the design of an optimization-oriented SSR algorithm and a CJSRA and decentralized joint SSR algorithm (DJSRA). The OSRA improves the performance of the signal reconstruction algorithm by improving the greedy algorithm support set estimation. The CJSRA improves the estimation of the public support set to solve the problem of correlation between multi-sources. The DJSRA eliminates the presence of central nodes through information transmission and reconstruction between nodes, thereby improving energy consumption and latency issues.

3.1 Design of SSR algorithm using optimization orientation

The core issue of CS is SSR, which includes single-source and multi-source joint SSR. The signal reconstruction problem is described as equation (1).

$$\min \|x\|_0 \ subject\ to \ \|y - \Phi x\| \le \delta \tag{1}$$

In equation (1), the original signal is denoted as x, the measurement matrix as Φ, and the signal observation vector as y. The reconstruction threshold is set at δ, with the stipulation that the noisy energy value must be less than this reconstruction threshold. The noise observation model estimates signals based on observation data, where the observation data is affected by noise interference. This model attempts to restore certain features of the original signal by considering the influence of noise, in order to improve the accuracy and reliability of signal reconstruction. The original signal is observed through a

measurement matrix and is affected by noise. The reconstruction goal is to recover the original signal based on the observation vector, and the reconstruction threshold is used to determine whether noise can be used for signal reconstruction. The correlation between different variables is shown in that the measurement matrix converts the original signal into the observation vector, and the noise affects the reconstruction effect. The reconstruction threshold is used to judge whether the noise can be ignored, so as to improve the accuracy of the signal reconstruction. The main difference of sparse signal recovery algorithms lies in their utilization of signal sparsity during the solving process. Compared with other algorithms such as greedy methods, they can better utilize the sparse characteristics of signals, thereby achieving faster convergence speed and higher recovery accuracy under the same conditions.

A cornerstone of SSR algorithms lies in the greedy tracking algorithm, which iteratively identifies the current optimal solution. However, it falls short of ensuring a globally optimal solution. In contrast, tree search matching and tracking algorithms can find the optimal branch through a large number of searches, but their computational complexity and complexity are relatively high. Therefore, based on greedy algorithms, this study proposes an OSRA, whose core problem is the SSR algorithm. Over-fitting refers to the phenomenon that the model over-fits the training data during machine learning, leading to poor performance on new unseen data. In the signal reconstruction problem, over-fitting may lead to an excessive dependence of the reconstruction results on the training data, thus creating errors in processing new signals. Compared with traditional greedy tracking algorithms, OSRA proposes a new approach based on global optimization, which solves the problem of over-fitting that may occur in greedy algorithms through support set pre-selection with backtracking and global optimization methods, thereby improving the accuracy and stability of signal reconstruction [17]. Before designing the algorithm, some symbol definitions based on the problem planning are first proposed, and the error function is defined as shown in equation (2).

$$r(\Lambda) = \left\| y - \Phi_{\Lambda} \hat{x}_{\Lambda} \right\|_2 \tag{2}$$

In equation (2), the error function is $r(\Lambda)$, and the optimal element composition support set is Λ. The sparse reconstruction model is grounded in greedy algorithms and endeavors to identify the optimal support set, which comprises key elements, through the definition of error functions. This approach is designed to achieve efficient sparse reconstruction tailored to specific problems. In the process of support set estimation, greedy algorithms may over-fit, leading to support set estimation errors [18]. To address this issue, improvements were made to the greedy algorithm to enhance the performance of the signal reconstruction algorithm. The optimized OSRA flow is shown in Figure 1.

The optimized OSRA mainly includes two parts: support set pre-selection with backtracking and global optimization method. In the stage of pre-selection and backtracking of the support set, the greedy algorithm is first used to obtain the initial estimate of the support set. To avoid over-fitting, it is necessary to backtrack the initial estimated support set. If the error function after backtracking is still below the preset threshold value, the backtracking support set will be used as the final estimate. On the contrary, it will enter the global optimization step. Global optimization is mainly achieved through two stages: local search and global search. Local search mainly reduces the error function, which is the objective function, by estimating the support set that satisfies the conditions. If the conditions are met, then this support set estimate is considered the final estimate. Should the local search fail to locate an estimate of the supporting set that satisfies the given conditions, it will transition to the global search phase. During the global search, a global objective function will be devised to enable the algorithm to escape local optima and ultimately derive the final estimate of the support set. After obtaining the support set estimation, the least squares method is used to achieve signal estimation. Overall, support set pre-selection with backtracking can reduce computational complexity through greedy algorithms, while backtracking can test the support set, identify the best elements to form the support set, and calculate the corresponding error function. The domain definition of the support set is shown in equation (3).

$$\begin{aligned} &N(\Lambda) = \\ &\{\Lambda, \Lambda \bigcup \{\theta_j\} - \{\lambda_j\} : 1 \le i \le K, \\ &1 \le j \le N - K\} \end{aligned} \tag{3}$$

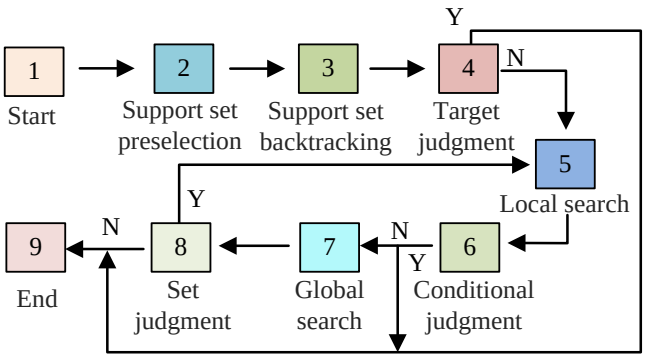


Figure 1: Optimized OSRA flow.

In equation (3), the domain of the support set is defined as $N(\Lambda)$, the number of optimal elements is K, the number of elements in the original signal subscript complete set is N, the complement of the support set in the complete set is Λ^c, and the element of the complement set is θ_j. The definition of the local optimal error function is shown in equation (4).

$$r(\Lambda^*) \le r(\Lambda^o), \forall \Lambda^o \in N(\Lambda^*) \quad (4)$$

In equation (4), the local optimal error function is $r(\Lambda^*)$, the support set corresponding to the error threshold is Λ^o, and the support set estimated for the local search is Λ^*. The global objective function is a function used to balance the estimation error of the support set and the difference of elements. The global objective function is shown in equation (5).

$$g(\Lambda^o, \Lambda^*) = \mu[r(\Lambda^o)]^2 - [\left|\Lambda^o - \Lambda^*\right| + \left|\Lambda^* - \Lambda^o\right|]^2 \quad (5)$$

In equation (5), the global objective function is $g(\Lambda^o, \Lambda^*)$, and the constant μ is used to measure the difference between Λ^* and Λ^o. After optimizing OSRA, the SSR is shown in equation (6).

$$\hat{x}_{\hat{T}} = \Phi^+_{\hat{T}} y \quad (6)$$

In equation (6), the best estimate of the signal support set is $\hat{T}$, and the best-reconstructed signal is $\hat{x}_{\hat{T}}$. In summary, the optimization OSRA model achieves the reconstruction of sparse signals by defining support set domains, local optimal error functions, and global objective functions, ultimately obtaining the best estimate of the signal support set and the best reconstructed signal. The OSRA pseudocode is shown in Figure 2.

The convergence of the OSRA is ensured by two key mechanisms: the error-decreasing property inherent in the support set pre-selection backtracking process and the gradient descent mechanism employed during global optimization. Assuming the objective function adheres to Lipschitz continuity and the step size in the global search phase complies with the Armijo condition, the algorithm is guaranteed to converge to a local optimal solution within a finite number of iterations. Furthermore, if the measurement matrix fulfills the RIP condition, OSRA is capable of accurately reconstructing the signal under a sparsity level of K. The time complexity of OSRA is $O(K^{2N} + K^3)$, where N is the signal length and K is the sparsity. It is superior to the exponential complexity of the tree search algorithm, but slightly higher than $O(K^{2N})$ of the traditional OMP.

```
Initialize:
   support_set = empty set
   residual = y
   best_support = empty set
   index = argmax(abs(Φ^T * residual))
   support_set.add(index)
   x_support = least_squares(Φ[:, support_set], y)
   residual = y - Φ * x_support
   if norm(residual) < epsilon:
      break
   temp_support = support_set - {j}
   temp_x = least_squares(Φ[:, temp_support], y)
   temp_residual = y - Φ * temp_x
   if norm(temp_residual) < norm(residual):
      support_set = temp_support
      residual = temp_residual
   for neighbor in support_set.neighbors():
      candidate_support = support_set.update(neighbor)
      candidate_x = least_squares(Φ[:, candidate_support], y)
      if norm(y - Φ * candidate_x) < norm(residual):
         support_set = candidate_support
         residual = y - Φ * candidate_x
   if no improvement:
      support_set = random_perturbation(support_set)
x = least_squares(Φ[:, support_set], y)
return x
```

Figure 2: The OSRA pseudocode.

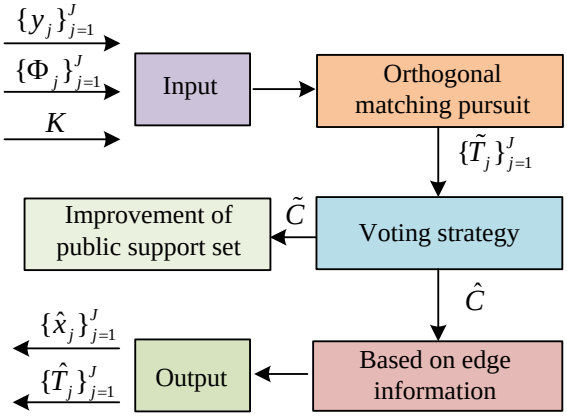


Figure 3: A centralized joint SSR algorithm based on improved public support set estimation.

The error threshold in the OSRA is dynamically adjusted based on the noise level, specifically set at 1.2 times the standard deviation of the noise. This configuration ensures that, during the backtracking process of the support set, only elements that effectively contribute to reducing the signal reconstruction error are retained, thereby preventing the inclusion of noise-induced interference. When the signal residual (the difference between the observed vector and the reconstructed signal) is less than the preset threshold or the number of iterations reaches 100 times, the algorithm is stopped to balance the reconstruction accuracy and computational efficiency and avoid meaningless excessive iterations. In the global optimization stage, by adjusting the difference weights of the elements in the support set (the optimal weight parameters are determined through cross-validation), it is ensured that the algorithm can not only utilize the fast search ability of the greedy algorithm, but also jump out of the local optimal solution through global search, thereby improving the accuracy of reconstruction.

3.2 Design of centralized joint SSR algorithm

In the problem of reconstructing sparse signals from multi-sources, it is necessary to consider the correlation between signals [19]. In the relevant signal source set, the support set between signals is jointly owned [20]. To address this issue, a CJSRA based on improved public support set estimation is proposed. Firstly, obtain a preliminary estimate of the public support set, then identify and remove erroneous elements from the preliminary estimate, retain correct elements, and improve the estimation of the public support set. The definition of a multi-source support set is shown in equation (7).

$$T_j = C \bigcup I_j, 1 \le j \le J \tag{7}$$

In equation (7), the support set for the multi-source signal x_j is T_j, the set of related signal sources is $\{x_j \in R^N\}_{j=1}^J$, the public support set is C, and the innovation support set is I_j. The sparse reconstruction model for multiple signal sources defines the signal support set, which is divided into related signal source set, public support set, and new information support set to achieve effective reconstruction and analysis of multiple signal sources. The CJSRA based on improved public support set estimation is aimed at the SSR problem of multi-sources. The main content of the study is to improve the accuracy and stability of signal reconstruction by introducing an improved public support set method. The improvement of the public support set is a process of eliminating incorrect elements and retaining correct elements from the initial support set estimation through voting strategies and error function screening. The CJSRA using improved public support set estimation is shown in Figure 3.

The CJSRA using improved public support set estimation consists of three parts: preliminary estimation of public support set, improvement of public support set, and OMP algorithm based on edge information. The preliminary estimation of the public support set requires the use of OMP algorithm, and then a voting strategy is used to obtain the preliminary estimation of the public support set $\tilde{C}$. In the improvement of the public support set, it is necessary to delete the wrong elements, retain the correct elements, and finally obtain the improved public support set estimate $\hat{C}$. The input of an OMP based on edge information is the improved public support set, in order to obtain the final overall support set estimate $\{\hat{T}_j\}_{j=1}^J$ and the reconstructed original signal $\{\hat{x}_j\}_{j=1}^J$. The strategy for enhancing the preliminary estimation of the public support set involves retaining the elements that have been accurately estimated and eliminating those that have been inaccurately estimated. The number of elements with higher credibility in the public support set is shown in equation (8).

$$\left|\tilde{C}_R\right| = \left[\frac{\max(s)}{J} \times \left|\tilde{C}\right|\right] \tag{8}$$

In equation (8), the number of elements with higher reliability is estimated as $\left|\tilde{C}_R\right|$, and the addition function is s. The error function corresponding to the support set $\tilde{C}_R$ is shown in equation (9).

$$r_j(\tilde{C}_R) = \left\| z_{j,\tilde{C}_R} \right\|_2 \tag{9}$$

In equation (9), the error function of the support set $\tilde{C}_R$ is $r_j(\tilde{C}_R)$, and the error threshold value is $z_{j,\tilde{C}_R}$. The sum of the error functions corresponding to all signal observation vectors is shown in equation (10).

$$f(\tilde{C}_R)=\sum_{j=1}^{J} r(\tilde{C}_R)=\sum_{j=1}^{J}\left\|z_{j,\tilde{C}_R}\right\|_2 \quad (10)$$

In equation (10), the total error function is $f(\tilde{C}_R)$. To avoid over-fitting, the total error estimated by the public support set should be less than the total error function. The definition of the support set domain is shown in equation (11).

$$N(C^*)=\{C^o, C^o=C^*\setminus\{c\}, \forall c\in C^*\} \quad (11)$$

In equation (11), the support set domain of the multi-source support set is C^*. When the sum of the minimum error functions exceeds the threshold during iteration, the best estimate $\hat{C}$ of the public support set is set C^*. When initializing the OMP algorithm, the support set is an empty set. Next, the edge information can be used to estimate the subsequent support set to ensure that elements can be correctly identified during the iteration process. At this time, the overall support set estimation is shown in equation (12).

$$\hat{T}_j=mOMP(y_j,\Phi_j,2K,\hat{C}) \quad (12)$$

In equation (12), the overall support set estimated by the OMP algorithm based on edge information is $\hat{T}_j$, and its size is $2K$. The non-zero term reconstruction result of the signal is shown in equation (13).

$$\hat{x}_{j,\hat{T}_j}=\Phi^{+}_{j,\hat{T}_j} y_j \quad (13)$$

In equation (13), the reconstruction result of the non-zero term signal is $\hat{x}_{j,\hat{T}_j}$, which corresponds to the subscripts of the first K maximum amplitude terms to obtain the final estimation. The complexity of CJSRA using an improved public support set includes preliminary set estimation, set improvement, and OMP algorithm based on edge information. The computational complexity of the improved algorithm of the public support set is shown in equation (14).

$$\begin{cases} O(JKMN+J(K-1)MN+\cdots+JK'MN) \\ =O(JSMN) \\ S=K+(K-1)+\cdots+K' \end{cases} \quad (14)$$

In equation (14), the computational complexity of the initial estimation of the public support set and the OMP algorithm based on edge information is consistent, which can be represented by $O(JKMN)$. The computational complexity is related to the number of iterations α, and the computational complexity of each iteration is $O(J\alpha MN)$. The evaluation indicators for algorithm reconstruction performance include relative reconstruction error and average error of support set estimation. The relative reconstruction error is shown in equation (15).

$$RER=E\left\{\frac{\left\|x-\hat{x}\right\|_2^2}{\left\|x\right\|_2^2}\right\} \quad (15)$$

The relative reconstruction error in equation (15) is RER, and the average of all Monte Carlo experiments and all signals is calculated as $E\{\bullet\}$. The average error of support set estimation is shown in equation (16).

$$ASCE=E\{d(T,\hat{T})\}=1-E\left\{\frac{\left|T\cap\hat{T}\right|}{\left|T\right|}\right\} \quad (16)$$

In equation (16), the average error of support set estimation is $ASCE$, and the measure of support set estimation error is $d(T,\hat{T})$. The CJSRA pseudocode is shown in Figure 4.

```
support_candidates = empty set
for each y_j in Y:
    _, single_support = omp(Φ_j, y_j, K)
    support_candidates.add(single_support)
common_support = vote(support_candidates, threshold=τ)
for c in common_support:
    temp_support = common_support - {c}
    total_error = sum(norm(y_j - Φ_j[:, temp_support] *
least_squares(Φ_j[:, temp_support], y_j)) for y_j in Y)
    if total_error < current_total_error:
        common_support = temp_support
X = empty list
for each y_j in Y:
    Φ_j_support = Φ_j[:, common_support]
    x_j = least_squares(Φ_j_support, y_j)
    X.append(x_j)
return X
```

Figure 4: The CJSRA pseudocode.

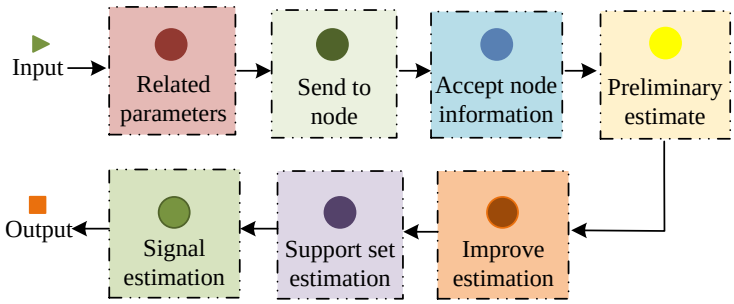


Figure 5: A DJSRA using improved public support set.

CJSRA converges through an iterative process that begins with an initial estimation by the public support set, followed by a refinement phase, and culminates in the OMP reconstruction of edge information. It is preliminarily estimated that the multi-source support set is aggregated by using the voting strategy. In the improvement stage, the support set is optimized by deleting the elements with large error contributions, so that the total error function monotonically decreases. Given that the elements within the support set are finite in number and the error possesses a lower bound, the algorithm converges to a local optimum within a finite sequence of steps. Assuming that the measurement matrices of each source satisfy the RIP conditions, the single-source OMP can accurately estimate the local support set. The voting strategy combined with the stability of RIP enables the estimation of the public support set to resist noise interference. Eventually, the precise reconstruction of multi-source signals is achieved through the edge information OMP. The computational complexity is linearly correlated with the number of sources, sparsity, observation length and signal length. It mainly comes from the preliminary estimation of multi-source OMP, improvement of support sets and joint reconstruction. It is suitable for medium-scale multi-source scenarios, and the computational amount increases linearly with the increase of the number of sources.

When making the preliminary estimation of the multi-source signal support set in the CJSRA algorithm, the "half-vote" rule (that is, an element is retained only when it appears in at least half of the single-source support set) is adopted to filter the random errors in the single-source estimation and improve the reliability of the public support set. By calculating the total change of the multi-source signal reconstruction error after removing a certain element, if the total error increases, the element is retained. Otherwise, it is removed, ensuring that the error of the public support set monotonically decreases after each iteration. During the support set improvement stage, the parameter R controls the proportion of elements deleted in each iteration.

3.3 Design of DJSRA

In the problem of joint SSR with multi-sources, two scenarios are mainly considered. The first scenario is CJSRA, in which each perception node transmits its observed signal values to the central node within the network, and then the central node reconstructs all signals. In DJSRA scenarios, there is no such central node. Instead, each perception node must collaborate directly with one another to reconstruct the signal [21-22]. Centralized reconstruction scenarios can address issues such as energy consumption, latency, security, interference, and network robustness [23-25]. Nodes send information to the connected nodes based on the network topology, and then each node reconstructs itself, eliminating the problem of a unique central node. The network connection topology can be a fixed structure or a random structure, with the number of input nodes and output nodes corresponding to their in degree and out degree, respectively [26-27]. This can improve energy efficiency, reduce latency, enhance security, reduce interference, and enhance network robustness. The network connection topology consists of fixed and random structures, in which the entry and exit degrees of nodes are equal [28-29]. Random structure refers to the Watts-Strogatz network model, which contains many nodes. In network topologies with degree 1 and degree 2, nodes have one or two input connections and output connections, respectively. With 10 nodes and a degree of 9 for each node, the network topology exhibits a fully connected structure, resembling a centralized reconstruction model. However, in decentralized scenarios, when the degree of network connection topology is low, signals can only receive observation information from some nodes and cannot fully utilize the correlation between signals. In this case, it is necessary to consider the problem of estimating the public support set. To ensure reconstruction accuracy, each node needs to send signals through output connections and receive signals through input connections, receiving and sending information multiple times. But this method will lead to an increase in energy consumption.

To reduce energy consumption and ensure the accuracy of local node reconstruction, the study adopts a public support set improvement method for node signal reconstruction. In this way, each node only needs to receive and send information once to complete signal reconstruction, avoiding the process of receiving and sending information multiple times. The algorithm complexity of this method is similar to that of the CJSRA based on improved public support set estimation. The DJSRA using improved public support set is shown in Figure 5.

The DJSRA is an optimization algorithm used to solve large-scale sparse linear reconstruction problems. This algorithm gradually updates the support set through iteration to achieve efficient signal reconstruction. The

process of the algorithm is as follows: The first step is to transmit an initial signal to each node and then receive feedback from them. Subsequently, preliminary estimates are used to obtain a public support set, and improved estimates are made to obtain more accurate support set estimates. By calculating the estimated values of the support set and the estimated values of the signal, the algorithm can achieve better reconstruction results. During the implementation process, the algorithm provides flexible solvers and regularization options to meet the needs of different situations. In addition, the algorithm can accelerate the solution through communication between nodes and parallelization of computation to achieve efficient reconstruction.

The DJSRA is based on the single information interaction mechanism among distributed nodes. Each node collaboratively optimizes the local estimation by using the neighborhood support set information, and avoids duplicate transmission through the improved public support set method. Network topology ensures information diffusion, supports set estimation to tend to be consistent after limited interactions, and achieves global convergence. When the measurement matrices of each node satisfy RIP, the accuracy of local support set estimation is transferred through network connectivity. Even with low connectivity, the global support set can be inferred from neighborhood information to ensure the robustness of distributed reconstruction. The computational complexity is related to the number of nodes, connectivity and the computational cost of a single node. The distributed structure eliminates the bottleneck of the central node, and the reconfiguration can be completed with a single interaction. The computational cost is comparable to that of CJSRA but the communication cost is lower, making it suitable for large-scale distributed networks. The DJSRA pseudocode is shown in Figure 6.

In the DJSRA, the network connectivity adopts the "small-world" network model (Watt-Strogatz model), and the node connectivity is set to 5 by default (that is, each node is connected to 5 neighboring nodes), balancing the communication cost between nodes and the utilization efficiency of signal correlation. Experiments showed that when the connectivity degree was 9 (fully connected), the reconstruction accuracy was the highest, but the energy consumption increased significantly. Therefore, a medium connectivity degree is selected as the optimal scheme. After a single information interaction, each node takes a local reconstruction error of less than 0.0001 as the convergence condition to avoid the time-consuming problem of multiple round-trip communications in traditional distributed algorithms and improve real-time performance. By taking advantage of the public support set characteristics of multi-source signals, the sparse structure is estimated through the fusion of neighborhood node information, without the need to preset the global sparsity parameter, which is adapted to the dynamic signal characteristics in distributed scenarios.

4 Performance analysis of SSR algorithm under CS

This chapter analyzes the performance of the proposed SSR algorithm. The first section of this chapter is the performance analysis of the optimization-oriented SSR algorithm. The second section is the performance analysis of the CJSRA. The third section is the performance analysis of the DJSRA.

4.1 Performance Analysis of SSR Algorithm Based on Optimization Orientation

Among the types of data signals used for simulation, the single-source signal was a sparse signal of fixed length, and the non-zero elements followed a Gaussian distribution. The degree of sparsity was controlled by the number of non-zero elements. Multi-source signals were multiple related signals, including shared public support sets and unique innovative support sets for each signal. The signal length was uniformly set to 1024, that is, each signal contained 1024 elements. The measurement matrix was a randomly generated Gaussian matrix, with the matrix size being the product of the number of measurements and the signal length. The column vectors were normalized. Some experiments adjusted the ratio of the number of measurements to the signal length, ranging from 0.10 to 0.20. Additive white Gaussian noise was added to the observation data. The noise level was controlled by the SNR, and the value range of the SNR was from 10 dB to 40 dB, corresponding to the simulation scenarios of different noise intensities.

```
for each node i in G:
      _, local_support = omp(Φ_i, y_i, K)
       send(local_support) to neighbors in G
   received_supports = receive from neighbors
global_support = union(received_supports + local_support)
global_support = prune(global_support, method="error-based")
for each node i in G:
   Φ_i_support = Φ_i[:, global_support]
   x_i = least_squares(Φ_i_support, y_i)
return [x_i for all nodes]
```

Figure 6: The DJSRA pseudocode.

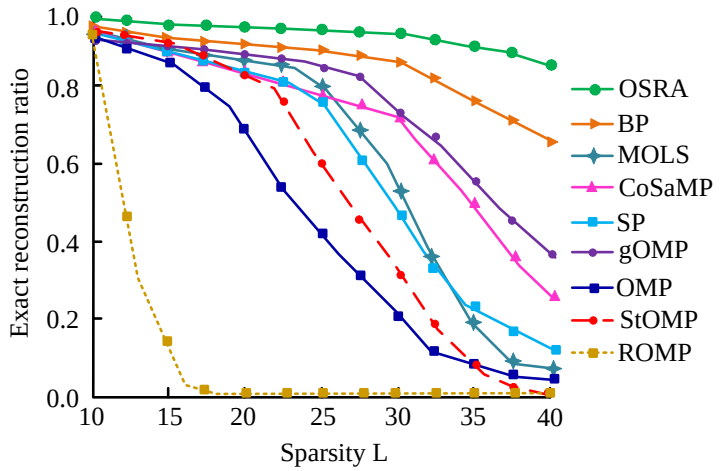


Figure 7: Comparison of successful reconstruction ratios of algorithms under different sparsity.

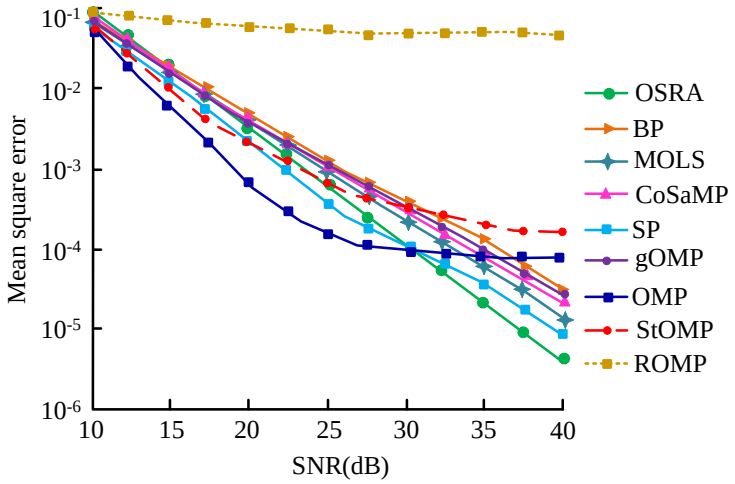


Figure 8: Comparison of mean square errors of various algorithms as signal-to-noise ratio increases.

To verify the performance of the OSRA proposed in the study, the study compared it with other algorithms. Comparison algorithms were the OMP [30], StOMP [31], subspace pursuit (SP) [32], regularized orthogonal matching pursuit (ROMP) [33], compressed sampling matching pursuit (CoSaMP) [34], gOMP, basis pursuit (BP) and orthogonal least square of multi-quadric algorithm (MOLS) [35-36]. The proportion of successful reconstruction under varying levels of sparsity is illustrated in Figure 7.

In Figure 7, the vertical axis represents the numerical range of successful reconstruction ratios, and the horizontal axis represents different signal sparsity values and 9 algorithms. As the sparsity of the signal increased, the proportion of successful reconstructions by each algorithm showed a decreasing trend. When the signal sparsity was 15, the success rate of each algorithm in reconstruction was close to 1.0, indicating that the algorithms performed well when the sparsity was low. However, when the sparsity increased to 40, the success rate of the optimization OSRA proposed in the study was greater than 0.7, while the success rate of other algorithms ranged from 0 to 0.6. By comparison, with the continuous increase of sparsity, the performance advantages of optimization-oriented SSR algorithms became increasingly apparent. As the SNR increased, the mean square error comparison of each algorithm is shown in Figure 8.

Figure 8 shows the comparison of mean square error between different algorithms without SNR. The smaller the mean square error value, the smaller the difference between the reconstructed signal and the original signal, indicating that the algorithm's reconstruction performance is better. From the graph, as the SNR gradually increased from 10dB to 40dB, the mean square error of the algorithm showed a decreasing trend. Among these algorithms, the OSRA showed significant advantages. At lower SNRs, the mean square error of OSRA was relatively small, and its advantages became more prominent as the SNR increased. When the noise ratio was 40dB, the mean square error of OSRA was less than 10^{-5}, which was 1-4 orders of magnitude smaller than other comparison algorithms. Compared with other algorithms, OSRA could more effectively reduce the mean square error of signals, providing more accurate and reliable results for signal processing. The comparison of running times of different algorithms in noisy scenarios is shown in Figure 9.

In Figure 9, the OSRA proposed in the study had a longer average running time due to its high computational complexity. As the SNR increased, its time consumption

first increased and then decreased. When the SNR was high, the loss of correct elements estimated by the pre-selected support set was reduced, and using global optimization to find missing elements was also faster. To quantify the complexity of the algorithm calculation, the experiment used statistical average iterations E(P) for analysis. To measure the computational complexity and iterative efficiency of the algorithm in a noisy environment more accurately, Table 1 details the total number of iterations of local search and global search under different SNR conditions.

In Table 1, the average number of iterations E(P) was relatively small, indicating that the OSRA had less additional computational complexity. When the signal sparsity was 25 and the SNR was 40, the average number of iterations E(P) was 0.11. At this time, the computational complexity of the OSRA was about 35 times that of the orthogonal matching tracking algorithm. Using a comprehensive analysis of algorithm performance and complexity, the proposed OSRA could adjust the computational complexity to improve the effectiveness of signal reconstruction. However, OSRA still had some drawbacks. Firstly, this algorithm required a longer computation time, especially in Gaussian noise scenarios. Secondly, the algorithm was sensitive to noise, and its reconstruction performance might be affected when the noise was large. In addition, the computational complexity of this algorithm increased with the increase of sparsity, which might limit its application in sparse signal processing scenarios.

The MSE comparison of OSRA and OMP in 50 Monte Carlo experiments is shown in Table 2. The mean MSE of OSRA was 3 to 4 orders of magnitude lower than that of OMP, and its standard deviation was smaller, indicating that it had higher reconstruction accuracy and stronger stability. All p values were less than 0.001, indicating that the differences between OSRA and OMP were of extremely significant statistical significance (rejecting the null hypothesis, the differences were not random).

4.2 Performance analysis of CJSRA

To verify the effectiveness of the CJSRA using the public support set, the experiments compared the performance of the OMP, SIPP, IOSSP, and the algorithm without adding the public support set improvement. The comparison of the performance of each algorithm in noise-free conditions is shown in Figure 10.

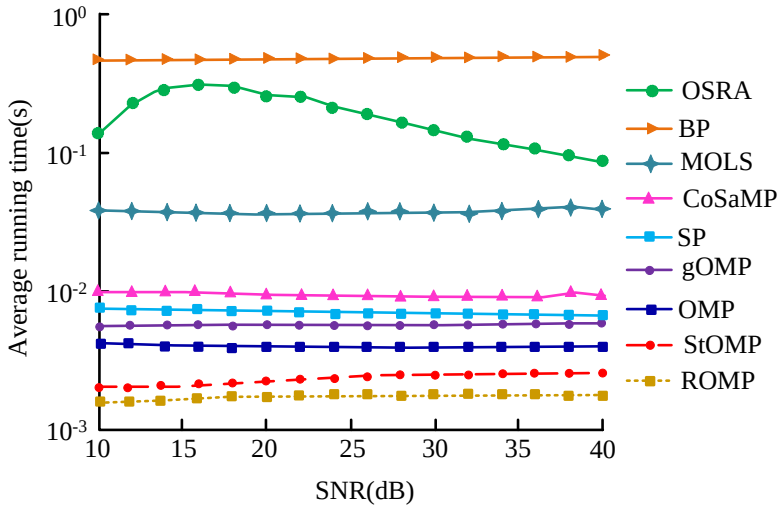


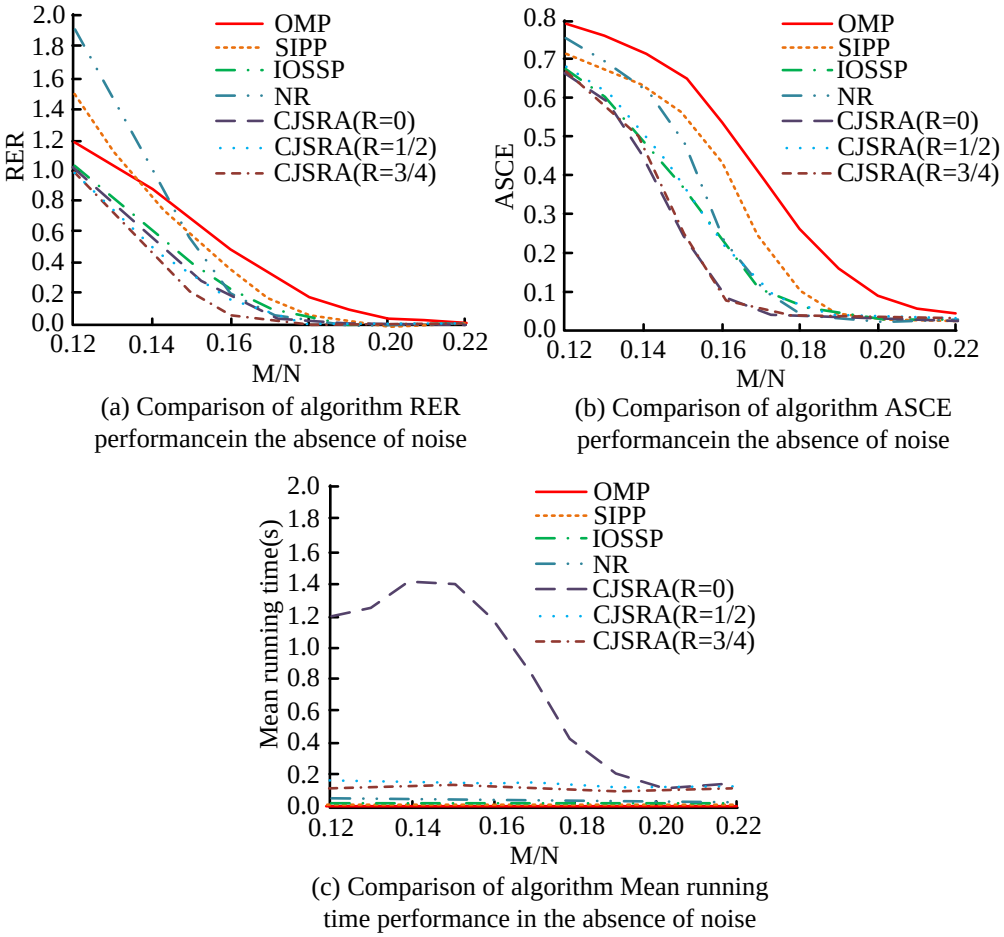
Figure 9: Comparison of runtime of different algorithms in noisy scenarios.

Table 1: The total number of iterations for local and global search under noisy conditions.

SNR	E(P)				
	K=21	K=23	K=25	K=27	K=29
SNR=32	0.17	0.20	0.22	0.25	0.28
SNR=34	0.15	0.17	0.19	0.23	0.27
SNR=36	0.13	0.14	0.15	0.21	0.26
SNR=38	0.11	0.12	0.13	0.18	0.25
SNR=40	0.08	0.10	0.11	0.17	0.24

Table 2: The comparison of MSE between OSRA and OMP.

Algorithm	SNR=20dB	SNR=30dB
OSRA	$2.5\times10^{-5}\pm3.7\times10^{-6}$	$9.8\times10^{-6}\pm1.1\times10^{-6}$
OMP	$1.2\times10^{-2}\pm2.1\times10^{-2}$	$8.3\times10^{-3}\pm1.5\times10^{-3}$
p	<0.001	<0.001



(a) Comparison of algorithm RER performancein the absence of noise

(b) Comparison of algorithm ASCE performancein the absence of noise

(c) Comparison of algorithm Mean running time performance in the absence of noise

Figure 10: Comparison of performance of various algorithms in noise absence.

Figure 10 (a) shows the comparison of relative reconstruction errors among various algorithms. As the ratio of limited observation data to actual signal length M/N increased, the relative reconstruction error of the algorithm gradually decreased. When the public support set improvement method was not added, the performance of the algorithm was poor. When the M/N was 0.12, the relative reconstruction error was 1.91, and after adding the public support set improvement, the relative reconstruction error was about 1.0. After introducing the parameter R that reduced the number of iterations, the correct elements in the initial estimation were removed in the improvement of the public support set. The reconstruction performance of the algorithm at R=3/4 was worse than that of the algorithms at R=0 and R=1/2. Compared to other algorithms, the proposed CJSRA had the highest support set estimation accuracy. Figure 10 (b) shows the average error of support set estimation, which was similar to the comparison of relative reconstruction errors. Figure 10 (c) shows the calculation of the average running time. As the R-value increased, the algorithm running time gradually decreased, and when R=0, the algorithm running time was longer. As the M/N ratio increased, the computational complexity of the algorithm gradually decreased. Comprehensive analysis showed that the CJSRA based on the public support set had obvious advantages, and a value of R of 1/2 could balance the computational complexity and reconstruction performance. In noisy scenarios, it was necessary to fix the M/N ratio and compare the impact of different SNRs on algorithm reconstruction performance. The M/N ratio was set to 0.18, and the performance comparison of each algorithm in noisy situations is shown in Figure 11.

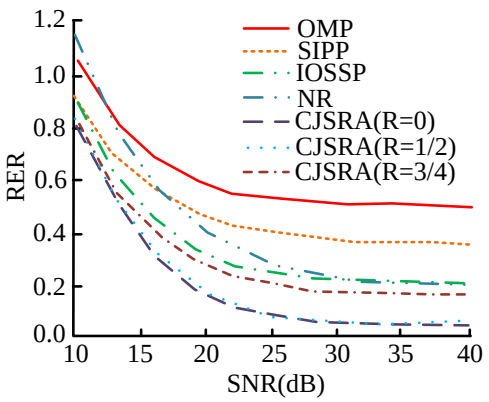
In Figure 11 (a), after adding the public support set improvement, when the iteration parameters R were 0 and 1/2, and the SNR was 40dB, the relative reconstruction error of CJSRA was the smallest, about 0.08. As shown in Figure 11 (b), as the SNR increased, the average error of the algorithm's support set estimation gradually decreased. Compared to other algorithms, the proposed CJSRA had a minimum support set estimation average error of 0.04. Figure 11 (c) shows the comparison of the average running time of the algorithm. It can be seen that when the iteration parameters R were 1/2 and 3/4, the computational complexity of the algorithm was significantly lower than when R was 0. During the algorithm iteration process, more elements were deleted each time, reducing the number of iterations. Comprehensive analysis showed that improving the public support set could enhance the effectiveness of joint SSR. The introduction of parameter R was to reduce iterations. The larger its value, the more elements were deleted in the iteration, and the fewer the iterations, which could reduce the computational complexity. However, CJSRA performed poorly in noisy scenarios and required a fixed signal-to-noise ratio. In addition, the algorithm had a high computational

complexity and introduced the parameter R to reduce the number of iterations. However, when the value of R was large, the reconstruction performance of the algorithm was actually poor.

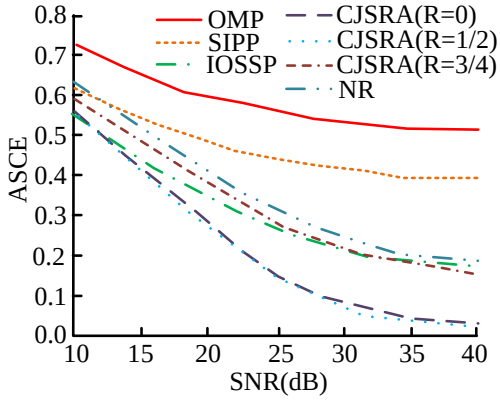
The comparison of ASCE between CJSRA and IOSSP in 30 Monte Carlo experiments is shown in Table 3. CJSRA verified the effectiveness of the voting strategy and error screening by improving the estimation of the public support set, with ASCE being more than 50% lower than IOSSP. When M/N≥0.15, all *p* was less than 0.01, indicating that the support set estimation accuracy of CJSRA was significantly better than that of the comparison algorithm.

4.3 Performance analysis of DJSRA

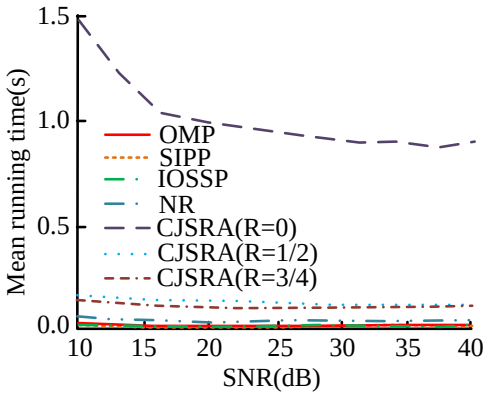
To verify the performance of the DJSRA proposed in the study, the experiment compared OMP and DIPP, with different values set for the degree of network connectivity. Table 4 displays the average frequency at which DIPP network nodes receive and transmit information within a noise-free environment.



(a) Comparison of algorithm RER performance under noisy conditions



(b) Comparison of algorithm ASCE performance under noisy conditions



(c) Comparison of algorithm mean running time performance under noisy conditions

Figure 11: Performance comparison of various algorithms under noisy conditions.

Table 3: The comparison of ASCE between CJSRA and IOSSP.

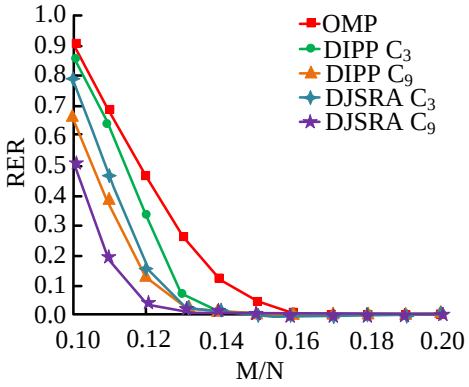
Algorithm	M/N=0.15	M/N=0.20
CJSRA	0.13±0.02	0.08±0.01
IOSSP	0.27±0.05	0.19±0.03
p	0.003	0.001

Table 4: The average number of times DIPP network nodes receive and transmit information in a noise free environment.

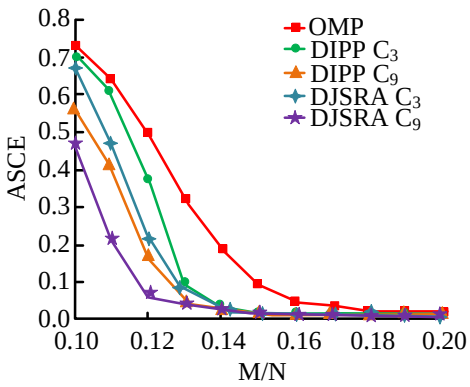
M/N	Average number of messages received and transmitted						
	C3	C4	C5	C6	C7	C8	C9
M/N=0.10	7.41	36.25	120.21	198.19	136.23	100.06	62.46
M/N=0.12	7.55	20.36	65.12	13.53	80.15	80.25	141.45
M/N=0.14	3.67	15.45	15.12	7.12	56.12	32.54	5.63
M/N=0.16	2.57	8.36	8.36	5.86	15.36	7.68	4.68
M/N=0.18	2.42	3.72	5.21	4.62	8.45	6.54	4.88
M/N=0.20	2.43	3.69	4.63	4.95	8.23	6.32	4.53

As shown in Table 4, under different parameters, there was a large difference in the average number of times each node in the distributed parallel tracking algorithm receives and transmits information. Nodes may experience multiple repetitions of receiving and transmitting information, especially when the M/N ratio was less than 0.12. However, the DJSRA proposed in this study could reconstruct information after a single transmission at each node, saving time wasted on repeated transmissions. The performance comparison of each algorithm in a noiseless environment is shown in Figure 12.

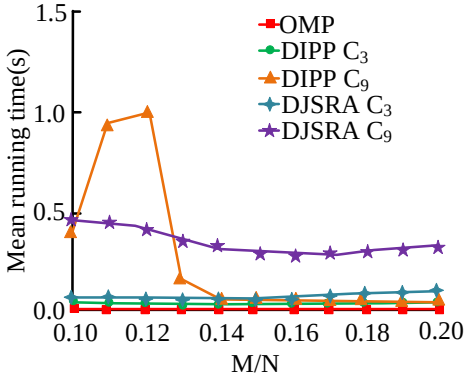
In Figures 12 (a) and (b), as the M/N ratio increased, the error of the algorithm gradually decreased. As the network connectivity increased, compared to the other two algorithms, the DJSRA proposed in the study had a smaller initial error value and faster convergence speed. When the network connectivity was C9 and the M/N ratio was 0.10, the relative reconstruction error and average support set estimation error of the DJSRA were 0.49 and 0.46, respectively. Figure 12 (c) shows the average running time of different algorithms reconstructing a single signal. The DIPP algorithm was greatly affected by the M/N ratio. When the M/N ratio was small, its running time could reach up to 1.1 seconds, which was proportional to the number of times the node received and sent information. In contrast, the average running time of the DJSRA proposed in the study did not change significantly. When the network connectivity was C9, the average running time remained within the range of less than 0.5 seconds. To further compare the communication efficiency of each algorithm in a noisy environment, Table 5 presents in detail the average number of times each node received and sent information in the DIPP algorithm and the DJSRA.



(a) Comparison of algorithm RER performance in the absence of noise



(b) Comparison of algorithm ASCE performance in the absence of noise



(c) Comparison of algorithm mean running time performance in the absence of noise

Figure 12: Comparison of performance of various algorithms in noiseless environments.

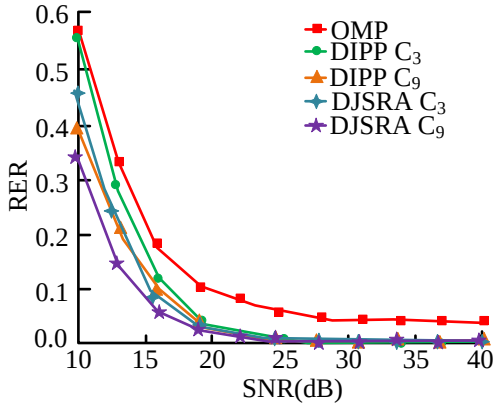
Table 5. The average number of times DIPP received and sent messages per node in noisy environments.

SNR(dB)	Average number of messages sent and received						
	C3	C4	C5	C6	C7	C8	C9
SNR=10	6.86	52.35	102.35	196.62	183.25	173.24	169.00
SNR=15	7.69	60.12	85.14	101.38	123.54	162.45	199.56
SNR=20	5.95	9.65	12.36	28.66	85.34	125.46	173.36
SNR=25	4.16	5.63	10.46	14.27	22.36	45.37	56.34
SNR=30	3.52	5.24	6.25	8.26	18.36	25.36	28.54
SNR=35	3.16	5.21	6.03	9.48	10.23	12.35	15.36
SNR=40	2.85	5.01	6.01	4.94	8.36	10.14	13.16

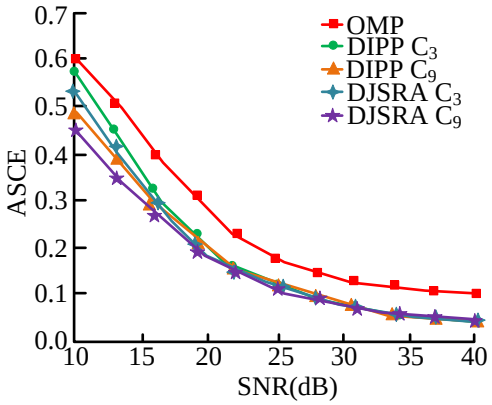
Table 5 shows that there was a significant difference in the average number of times DIPP nodes received and sent messages in noisy environments, especially when the SNR was less than 20, the number of times DIPP nodes received and sent messages in the network varied significantly. The DJSRA proposed in the study could complete signal reconstruction by receiving and sending information once per section, avoiding the time consumption of nodes repeatedly sending information multiple times. The performance comparison of each algorithm in noisy environments is shown in Figure 13.

As shown in Figure 13 (a), as the network connectivity increased, the relative reconstruction error of the DJSRA proposed in the study gradually decreased, and the error value was smaller than that of other algorithms. When the SNR was 10, the relative reconstruction error using the orthogonal matching tracking algorithm was 0.57. The minimum relative reconstruction error of the DJSRA proposed in the study was 0.34. Figure 13 (b) shows the average error of the algorithm's support set estimation, which was similar to the result of relative reconstruction error. As the SNR increased, the average error of support set estimation for DIPP and the proposed algorithm gradually approached, with an error value of less than 0.1 when the SNR was 40. Figure 13 (c) shows the comparison of the average running time of the algorithm. When the network connectivity was low, the average running time of DIPP changed significantly, which was positively correlated with the number of times nodes received and sent information. The average running time of the DJSRA proposed in the study was relatively stable. Comprehensive analysis showed that the DJSRA proposed in the study could reduce computational complexity while ensuring the effectiveness of signal reconstruction.

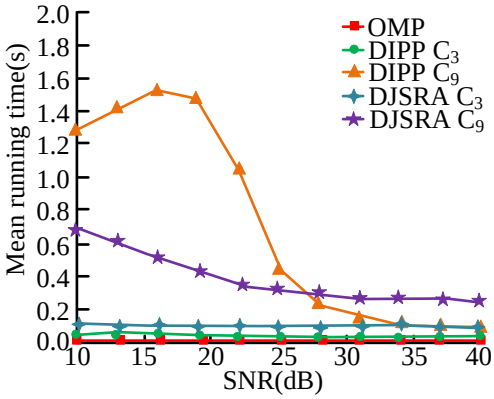
The RER comparison between DJSRA and DIPP in 40 Monte Carlo experiments with network connectivity C5 is shown in Table 6. DJSRA achieved reconfiguration through a single interaction, with an RER that was over 40% lower than DIPP, and a smaller standard deviation (higher stability). In the low SNR scenario (SNR=10dB), $p < 0.05$, indicating that the distributed collaborative mechanism significantly improved the anti-noise performance.



(a) Comparison of algorithm RER performance under noisy conditions



(b) Comparison of algorithm ASCE performance under noisy conditions



(c) Comparison of algorithm mean running time performance under noisy conditions

Figure 13: Comparison of performance of various algorithms in noiseless environments.

Table 6: The comparison of RER between DJSRA and DIPP.

Algorithm	SNR=20dB	SNR=30dB
DJSRA	0.34±0.06	0.18±0.03
DIPP	0.57±0.12	0.31±0.08
p	0.012	0.008

Table 7: Comparison experimental results between sparse signal recovery algorithm and other methods.

Algorithm	Relative reconstruction error	Successful refactoring rate	Reference
OSRA	0.763	0.793	This study
CJSRA	1.026	0.818	This study
DJSRA	0.332	0.856	This study
ISTA	1.687	0.787	Lv et al. [37]
FISTA	1.986	0.716	Guo et al. [38]
AMP	1.674	0.806	Gerbelot et al. [39]

The experimental results comparing the SSR algorithm with other methods are shown in Table 7. Iterative Shrinkage-Thresholding Algorithm (ISTA) adopted the default configuration in the literature. The step size was set to 0.9 times the inverse of the Lipschitz constant, the maximum number of iterations was 200 times, and the termination condition was that the change in reconstruction error in five consecutive iterations was less than 10^{-5}. Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) was an accelerated version based on literature optimization, introducing a momentum term to enhance the convergence speed. The step size was dynamically adjusted, and the maximum number of iterations was set at 150 times. The termination condition was consistent with that of ISTA. The Approximate Message Passing (AMP) was implemented with reference to the original literature, including an adaptive denoising module. The number of iterations was set at 50 times, and the measurement matrix adopted the same column-normalized random Gaussian matrix preprocessing method as the algorithm in this paper. All baseline methods used the same signal model (Gaussian sparse signal), measurement matrix (column-normalized random Gaussian matrix), and noise model (additive Gaussian white noise) as the algorithm in this paper, and the parameter adjustments strictly referred to the optimal practices in the original literature to ensure consistent comparison conditions. As shown in Table 7 the SSR algorithm proposed in the study had a relatively low reconstruction error and a high reconstruction success rate. Among them, DJSRA performed the best and FISTA performed the worst. This was mainly due to the introduction of an iterative threshold selection strategy in the DJSRA, which enabled it to achieve high reconstruction accuracy with fewer iterations. FISTA used an acceleration strategy, which could accelerate convergence but sacrificed reconstruction accuracy to some extent.

Performance data revealed the inherent trade-off among accuracy, efficiency and robustness in algorithm design. By introducing the backtracking verification of support sets and the global optimization mechanism, OSRA significantly outperformed the traditional greedy algorithm in terms of reconstruction accuracy. However, the computational complexity increased due to the increase in the number of iterations and the number of least squares solution times, reflecting the design logic of "accuracy for efficiency". As a centralized algorithm, CJSRA improved the estimation accuracy of the support set through the multi-source voting strategy and edge information OMP. However, all source data needed to be transmitted to the central node, and the communication cost increased linearly with the number of sources. While DJSRA reduced the communication overhead through distributed single information interaction (only the summary of the support set was transmitted between adjacent nodes). However, due to the reliance on local information, the reconstruction error was slightly higher (RER was about 10%-15% higher), demonstrating the "trade-off between the accuracy of multi-source collaboration and communication cost". Furthermore, in scenarios with low SNR or low observation rate, complex algorithms (such as OSRA, CJSRA) were more sensitive to noise, while traditional algorithms (such as OMP) exhibited stronger robustness due to their simple logic, reflecting the "trade-off between noise robustness and observation rate". Distributed topology analysis showed that DJSRA had the best accuracy in fully connected networks, but the error increased significantly in low-connectivity topologies, reflecting the "dependency relationship between network structure and reconstruction performance". These trade-offs provided clear guidance for practical applications: OSRA was preferred in high-precision scenarios, DJSRA was applicable in distributed scenarios with limited communication, and lightweight algorithms such as OMP could be adopted in scenarios with high real-time requirements.

5 Conclusion

CS is widely used in applied mathematics, remote sensing imaging, and computer science, with its core problem being SSR. The research proposed SSR algorithms suitable for different scenarios for single-source and multi-source joint SSR. For SSR from a single-source, research was conducted to improve the greedy algorithm support set estimation and propose an OSRA. For the reconstruction of sparse signals from multi-sources, a CJSRA based on improved public support set estimation was proposed by studying and utilizing the improved public support set. Through node signal reconstruction, a DJSRA using an improved public support set was proposed. Experimental data showed that the mean square error of the OSRA was less than 10^{-5}, which was 1-4 orders of magnitude smaller than OMP and other comparative algorithms. When the signal sparsity was 25 and the SNR was 40, the average number of iterations was 0.11. At this time, the computational complexity of the OSRA was about 35 times that of the OMP algorithm. Based on comprehensive analysis, it can be concluded that the OSRA proposed in the study could adjust the computational complexity to improve the effectiveness of signal reconstruction. For the joint SSR algorithm with multi-sources, the performance of the algorithm was poor

without the addition of the public support set improvement method. When the M/N was 0.12, the relative reconstruction error was 1.91, and the relative reconstruction error after the addition of the public support set improvement was about 1.0. After introducing the parameter R that affected the number of iterations, the correct elements in the initial estimation were removed in the improvement of the public support set. The reconstruction performance of the algorithm was worse when R=3/4 than when R=0 and R=1/2. Compared to other algorithms such as OMP, the proposed CJSRA had the highest support set estimation accuracy. When the SNR was 10, the relative reconstruction error using OMP was 0.57. The minimum relative reconstruction error of the DJSRA proposed in the study was 0.34. Based on the above content, the superiority and contribution value of this study can be concluded. Firstly, a greedy algorithm-based SSR algorithm was proposed for the problem of single-source SSR. This algorithm could achieve high signal reconstruction performance gains at low complexity. Secondly, for the problem of joint SSR from multi-sources, a joint SSR algorithm based on an improved public support set was proposed, which could achieve good signal reconstruction performance at low complexity. Finally, the study focused on decentralized reconstruction scenarios and hybrid support set models, analyzed the impact of network connection topology on the estimation of public support sets and the energy consumption of nodes receiving and transmitting information, and proposed corresponding improvement strategies. Through the above research, better reconstruction results can be achieved in the field of SSR, and better performance can be achieved in different scenarios. However, OSRA and CJSRA algorithms performed poorly in noisy scenarios and require a fixed SNR. In addition, when introducing the parameter R to reduce the number of iterations, if the value of R was large, the reconstruction performance of the algorithm was actually poor. Therefore, these algorithms were sensitive to noise, and their computational complexity increased with increasing sparsity, limiting their application in sparse signal processing scenarios. The limitation of this study was that the computational complexity of the single-source SSR algorithm was about 35 times that of the orthogonal matching tracking algorithm, and the algorithm ran relatively long. In response to the above issues, future research directions can focus on improving the performance of algorithms in noisy scenarios and reducing computational complexity to adapt to a wider range of sparse signal processing applications, and start with the finite equidistant property to further improve the performance of SSR.

References

[1] Shenghang Zhou, Xiubao Sui, Wenhui Zhang, Yingzi Hua, Qian Chen, and Guohua Gu. High-resolution image transmission through scattering media based on compressive encoding. Optical Engineering, 60(3):35108-35123, 2021. https://doi.org/10.1117/1.OE.60.3.035108

[2] Jianfeng Wang, Zhiyong Zhou, and Jun Yu. Statistical inference for block sparsity of complex-valued signals. IET Signal Processing, 14(4):154-161, 2020. https://doi.org/10.1049/iet-spr.2019.0200

[3] Sheikh Rafiul Islam, Santi P. Maity, and Ajoy Kumar. On learning based compressed sensing for high resolution image reconstruction. IET Image Processing, 15(2):393-404, 2020. https://doi.org/10.1049/ipr2.12029

[4] Chaoang Xiao, Hesheng Tang, and Yan Ren. Compressed sensing reconstruction for axial piston pump bearing vibration signals based on adaptive sparse dictionary model. Measurement & Control, 53(3-4):649-661, 2020. https://doi.org/10.1177/0020294019898725

[5] Jia Wu. A fast-iterative reconstruction algorithm for sparse angle CT based on compressed sensing. Future Generation Computer Systems, 126(1):289-294, 2022. https://doi.org/10.1016/j.future.2021.08.013

[6] Chao Wang, Xiaohong Shen, Haiyan Wang, and Haodi Mei. Energy-efficient collection scheme based on compressive sensing in underwater wireless sensor networks for environment monitoring over fading channels. Digital Signal Processing, 127(1):103530-103537, 2022. https://doi.org/10.1016/j.dsp.2022.103530

[7] Xingyuan Wang, Cheng Liu, and Donghua Jiang. A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3D DCT. Information Sciences, 574(1):505-527, 2021. https://doi.org/10.1016/j.ins.2021.06.032

[8] Baohong Bai, Xiaodong Li, Tao Zhang, and Dakai Lin. Nonconvex $L_{1/2}$ minimization based compressive sensing approach for duct azimuthal mode detection. AIAA Journal, 58(4):3932-3946, 2020. https://doi.org/10.2514/1.J059341

[9] Yuki Kato, Soma Watahiki, and Masayoshi Otaka. Full-field measurements of high-frequency micro-vibration under operational conditions using sub-Nyquist-rate 3D-DIC and compressed sensing with order analysis. Mechanical Systems and Signal Processing, 224(1):112179-112208, 2025. https://doi.org/10.1016/j.ymssp.2024.112179

[10] Qutaiba Kadhim, and Waleed Ameen Mahmoud Al-Jawher. A new multiple-chaos image encryption algorithm based on block compressive sensing, swin transformer, and wild horse optimization. Multidisciplinary Science Journal, 7(1):2025012-2025025, 2025. https://doi.org/10.31893/multiscience.2025012

[11] Alireza Zeynali, and Mohammad Ali Tinati. Signal reconstruction based on time-varying sliding window in WSNs using compressed sensing. International Journal of Communication Systems, 38(1):6080-6099, 2025. https://doi.org/10.1002/dac.6080

[12] You Zhao, Xiaofeng Liao, Xing He, Rongqiang Tang, and Weiwei Deng. Smoothing inertial neurodynamic approach for sparse signal reconstruction via Lp-norm minimization. Neural Networks, 140(17):100-

112, 2021. https://doi.org/10.1016/j.neunet.2021.02.006

[13] Huizu Lin, Shuai Sun, Liang Jiang, Longkun Du, Hongkang Hu, and Weitao Liu. Optimal parameters for image reconstruction in ghost imaging via sparsity constraints. Optical Engineering, 59(12):123101-123109, 2020. https://doi.org/10.1117/1.OE.59.12.123101

[14] Yixuan Li, Xiaopeng Yang, Tian Lan, Renjie Liu, and Xiaodong Qu. Parameter inversion by a modified reflected signal reconstruction method for thin-layered media. IEEE Antennas and Wireless Propagation Letters, 21(5):958-962, 2022. https://doi.org/10.1109/LAWP.2022.3152843

[15] Na Li, Xinghui Yin, Huanyin Guo, Sulan Zong, and Wei Fu. Variable step-size matching pursuit based on oblique projection for compressed sensing. IET Image Processing, 14(4):766-773, 2020. https://doi.org/10.1049/iet-ipr.2019.0916

[16] Shengheng Liu, Yimin D. Zhang, Tao Shan, and Ran Tao. Structure-aware Bayesian compressive sensing for frequency-hopping spectrum estimation with missing observations. IEEE Transactions on Signal Processing, 66(8):2153-2166, 2018. https://doi.org/10.1109/TSP.2018.2806351

[17] Shirshendu Roy, Debiprasad P. Acharya, and Ajit K. Sahoo. Fast OMP algorithm and its FPGA implementation for compressed sensing-based sparse signal acquisition systems. IET Circuits, Devices & Systems, 15(6):511-521, 2021. https://doi.org/10.1049/cds2.12047

[18] P. Sudhakar Reddy, B. S. Raghavendra, and A. V. Narasimhadhan. Root-free annihilating filter method for sparse signal reconstruction. Circuits, Systems, and Signal Processing, 44(1):670-683, 2025. https://doi.org/10.1007/s00034-024-02871-3

[19] Cheng Zhang, Qianwen Chen, Meiqin Wang, and Sui Wei. Optimised two-dimensional orthogonal matching pursuit algorithm via singular value decomposition. IET Signal Processing, 14(4):717-724, 2021. https://doi.org/10.1049/iet-spr.2019.0090

[20] Abbas Ali Sharifi, and Ghanbar Azarnia. Compressive sensing for PAPR reduction of DC-biased optical OFDM signals with exploiting joint sparsity for signal reconstruction. Optical Engineering, 59(9):96105-96116, 2020. https://doi.org/10.1117/1.OE.59.9.096105

[21] You You, and Li Zhang. Exploiting angular spread in channel estimation of millimeter wave MIMO system. IET Communications, 16(3):195-205, 2022. https://doi.org/10.1049/cmu2.12329

[22] Maysara Ghaith, Ahmad Siam, Zhong Li, and Wael El-Dakhakhni. Hybrid hydrological data-driven approach for daily streamflow forecasting. Journal of Hydrologic Engineering, 25(2):4019063-4019071, 2020. https://doi.org/10.1061/(ASCE)HE.1943-5584.0001866

[23] Shruthi Khadri, Naveen K Bhoganna, and Madam Aravind Kuma. IoT driven joint compressed sensing and shallow learning approach for ECG signal-reconstruction. Indonesian Journal of Electrical Engineering and Computer Science, 34(1):666-676, 2024. https://doi.org/10.11591/ijeecs.v34.i1.pp666-676

[24] Jie Zan. Research on robot path perception and optimization technology based on whale optimization algorithm. Journal of Computational and Cognitive Engineering, 1(4):201-208, 2023. https://doi.org/10.47852/bonviewJCCE597820205514

[25] Ghous Ali, and Masfa Nasrullah Ansari. Multiattribute decision-making under Fermatean fuzzy bipolar soft framework. Granular Computing, 7(2):337-352, 2022. https://doi.org/10.1007/s41066-021-00270-6

[26] A. Guleria, and R. K. Bajaj. T-spherical fuzzy soft sets and its aggregation operators with application in decision-making. Scientia Iranica, 28(2):1014-1029, 2021. https://doi.org/10.24200/SCI.2019.53027.3018

[27] Qilei Zhang, Lei Yu, Feng He, and Yifei Ji. Modified complex multitasks Bayesian compressive sensing using Laplacian scale mixture prior. IET signal processing, 16(5):601-614, 2022. https://doi.org/10.1049/sil2.12134

[28] Omar Abu Arqub, Jagdev Singh, and Mohammed Alhodaly. Adaptation of kernel functions-based approach with Atangana-Baleanu-Caputo distributed order derivative for solutions of fuzzy fractional Volterra and Fredholm integrodifferential equations. Mathematical Methods in the Applied Sciences, 46(7):7807-7834, 2023. https://doi.org/10.1002/mma.7228

[29] Zihan Luo, Jing Liang, and Jie Ren. Deep learning based compressive sensing for UWB signal reconstruction. IEEE Transactions on Geoscience and Remote Sensing, 60:1-10, 2022. https://doi.org/10.1109/TGRS.2022.3181891

[30] Quan Sun, Fei-Yun Wu, Kunde Yang, and Chunlong Huang. Sparse signal recovery from noisy measurements via searching forward OMP. Electronics Letters, 58(3):124-126, 2022. https://doi.org/10.1049/ell2.12365

[31] Yichen Chen, Bingru Chen, Wu Guo, Jia Jia, and Yijun Wang. Data evaluation algorithm for compensation litigation for ecotope pollution damage in the Yellow River valley caused by industrial wastewater from the perspective of green development. Water Science & Technology, 88(3):631-644, 2023. https://doi.org/10.2166/wst.2023.215

[32] Jian Zhang, Bin Chen, Ruiqin Xiong, and Yongbing Zhang. Physics-inspired compressive sensing: Beyond deep unrolling. IEEE Signal Processing Magazine, 40(1):58-72, 2023. https://doi.org/10.1109/MSP.2022.3208394

[33] Yue Xiao, Lei Yuan, Junyu Wang, Wenxin Hu, and Ruimin Sun. Sparse reconstruction of sound field using bayesian compressive sensing and equivalent source method. Sensors, 23(12):5666-5685, 2023. https://doi.org/10.3390/s23125666

[34] Zezheng Qin, Yiming Ma, Lingyu Ma, Guangxing Liu, and Mingjian Sun. Convolutional sparse coding for compressed sensing photoacoustic CT reconstruction with partially known support. Biomedical Optics Express, 15(2):524-539, 2024. https://doi.org/10.1364/BOE.507831

[35] David L. Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. Proceedings of the National Academy of Sciences, 106(45):18914-18919, 2009. https://doi.org/10.1073/pnas.0909892106

[36] Amir Beck, and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2(1):183-202, 2009. https://doi.org/10.1137/080716542

[37] Dan Lv, Yan Zhang, Danjv Lv, Jing Lu, Yixing Fu, and Zhun Li. Combining CBAM and iterative shrinkage-thresholding algorithm for compressive sensing of bird images. Applied Sciences, 14(19):8680-8694, 2024. https://doi.org/10.3390/app14198680

[38] Xinrong Guo, Fengkai Liu, and Darong Huang. Migration through resolution cell correction and sparse aperture ISAR imaging for maneuvering target based on whale optimization algorithm-Fast iterative shrinkage thresholding algorithm. Sensors, 24(7):2148-2163, 2024. https://doi.org/10.3390/s24072148

[39] Cédric Gerbelot, and Raphaël Berthier. Graph-based approximate message passing iterations. Information and Inference, 12(4):2562-2628, 2024. https://doi.org/10.48550/arXiv.2109.11905

4D Flight Trajectory Prediction Using CURE-Based Clustering of ADS-B Data

Qinghai Zuo[1*], Shuai Han[2], Weijun Pan[3,4]
[1]College of Air Traffic Management, Civil Aviation Flight University of China, Guanghan 618307, China
[2]Operation Supervisory Center of CAAC, Beijing 100710, China
[3]Scientific Research Base, Civil Aviation Flight University of China, Guanghan 618307, China
[4]Key Laboratory of Flight Techniques and Flight Safety of CAAC, Guanghan 618307, China
E-mail: zuoatc@163.com
[*]Corresponding author

Keywords: flight trajectory, CURE algorithm, cluster analysis, ADS-B

Received: March 14, 2025

Aiming at the problems of insufficient flight trajectory prediction accuracy and low early warning efficiency of air traffic conflicts under the background of the sharp increase in air traffic volume, a 4D trajectory prediction method based on the clustering using representatives (CURE) algorithm is studied and proposed. The method utilizes historical track data provided by the Automatic Dependent Surveillance-Broadcast (ADS-B) system to measure track similarity by means of modified Euclidean distance. It is also combined with hierarchical clustering techniques to cluster and analyze the tracks within the terminal area. Experiments showed that the proposed method outperformed the traditional model in many aspects. For example, in track prediction, the CURE-based model had an average time error of only 17.6 seconds, a height error of 31.5 m, and a horizontal root mean square error (RMSE) of 0.55 nautical miles. Furthermore, using conflict detection based on 4D planes and geometric methods successfully reduced the false alarm rate to 4.30% and controlled the missed alarm rate within 0.08%. Together with the experimental data, these results verify the effectiveness and reliability of this method in complex aviation scenarios. This indicates that the CURE algorithm can improve track prediction accuracy and provide stronger technical support for aviation traffic management.

Povzetek: Napovedovanje 4D avionskih trajektorij je izvedeno z URE-grozdenjem iz ADS-B; predlagane spremembe povečajo rigoroznost in uporabnost.

1 Introduction

In recent years, with the rapid growth of global air traffic, air traffic management is facing unprecedented pressure and challenges. Effective trajectory prediction technology can not only enhance the utilization of airspace resources and reduce the workload of ground controllers, but also significantly improve the safety of flight operation. Therefore, 4D trajectory prediction method is of great significance in modern aviation management. 4D trajectory prediction technology accurately predicts an aircraft's flight trajectory by combining information from four dimensions: time, space, speed, and altitude. This technology provides effective support for air traffic management. Currently, based on the data from the GPS automatic dependent surveillance-broadcast (ADS-B) system, it is possible to obtain information such as the position, speed, and heading of the aircraft in real time, which provides a reliable data base for trajectory prediction. Traditional trajectory prediction methods mainly rely on ground radar and flight plans. However, these methods have certain limitations in terms of prediction accuracy and real-time performance [1-2]. With the popularization and application of ADS-B system, the trajectory prediction technology based on ADS-B data has gradually become a research hotspot. The 4D trajectory prediction technology has been widely studied at home and abroad. Many researchers try to improve the accuracy and reliability of trajectory prediction by different methods. Aiming at the problem of low accuracy of short-term prediction of flight trajectory, Yang et al. proposed a bidirectional long-short memory network prediction method based on broadcast auto-correlation surveillance historical data. Experimental results indicated that the method was proved to improve aviation safety in busy airspace [3]. Wang et al. proposed a generalized hybrid recurrent prediction model for flight trajectory prediction. The results showed that the generalized deep learning method not only improved the accuracy of trajectory prediction, but also allowed contextualization by exploring a large amount of data [4]. Han et al. proposed a trajectory prediction method combining a density-based spatial clustering algorithm with noise and a gated loop cell for trajectory prediction. The results indicated that this method could effectively utilize the trajectory data in the terminal area, and the model developed could perform trajectory prediction for multiple flights and improve the accuracy of trajectory prediction [5]. Dai et al. proposed that a deep neural network model based on Kalman filter algorithm unfolding can be used for aircraft trajectory prediction. The results of simulation experiments

indicated that the developed model provided better accuracy and effectiveness for aircraft trajectory prediction than other network models [6]. Rizvi et al. proposed a meta-learning approach to predict short and medium-term aircraft trajectories using historical real flight data collected from multiple genetic algorithm aircraft. The study used random forest regression and long and short-term memory networks to extract aircraft trajectory features and k-nearest neighbors were used to complete the final prediction. The model was shown to have good prediction results in the experimental results [7]. Chakrabarti et al. used a hidden Markov model to identify and extract heading changes within aircraft trajectories, followed by comparing and clustering the trajectory strings using an edit distance metric combined with the K-medoids clustering algorithm. An application to a set of historical trajectories at Washington National Airport demonstrated the success of the proposed framework in overcoming the shortcomings associated with traditional clustering techniques [8].The literature review is specifically shown in Table 1.

Table 1: Literature review table.

References	Method	Advantages	Disadvantages
Reference [3]	Bi-LSTM	Be capable of capturing long sequence dependencies. It has a remarkable effect on short-term prediction	The training time is relatively long. Sensitive to hyperparameter Settings
Reference [4]	Generalized mixed cycle model	The accuracy of trajectory prediction has been improved. It has strong generalization ability and can adapt to various data situations	The model is complex and parameter adjustment is difficult. It has high requirements for the volume of data
Reference [5]	DBSCAN-GRU	It has a strong ability to handle high-density data and noise. Adaptive clustering	Parameter selection is sensitive. The clustering effect is uneven for regions with different densities
Reference [6]	Deep neural network	Contextualization can be achieved by exploring a large amount of data	The training is complex and the computational overhead is high. A large amount of labeled data is required for training
Reference [7]	Meta-learning method	Applicable to various situations. It is capable of efficiently utilizing historical data for medium and short-term trajectory prediction	The adaptability to specific problems remains to be verified. High complexity
Reference [8]	Hidden Markov model	The model is simple and easy to understand. It is relatively effective in sequential data	It may be impossible to capture long dependencies. Be sensitive to the initial Settings

Although existing track prediction models, such as Bi-LSTM and generalized mixed loop models, perform well in capturing the short-term dependencies of time series data, they often have difficulty fully handling the complexity of high-dimensional data. This is especially true when it comes to effectively identifying potential patterns in aviation trajectories and performing cluster analysis. Meanwhile, methods such as DBSCAN-GRU and Hidden Markov models are relatively vulnerable to data noise during the clustering process, making it difficult to accurately distinguish the trajectories of different categories. The clustering using representatives (CURE) algorithm is a clustering method that aims to accurately represent the shape and distribution of clusters. It does so by selecting multiple representative points. This allows it to overcome the limitations of traditional clustering methods when dealing with high-dimensional data and complex shapes. By dynamically correcting the similarity calculation and hierarchical clustering methods, the CURE method overcomes the influence of data noise on the clustering results. This improves the accuracy and reliability of flight trajectory prediction. These improvements allow the CURE method to more effectively capture the complex features and potential patterns of aviation trajectories, providing better technical support for flight scheduling and air traffic management. This study aims to explore the potential of the CURE algorithm to improve the accuracy of 4D trajectory prediction in the terminal area. Specific questions include: Can the CURE algorithm significantly improve prediction accuracy in complex aviation trajectory data environments? Meanwhile, the research will conduct an in-depth analysis of how trajectory similarity estimation using the modified Euclidean distance influences the accuracy of flight conflict prediction. This analysis will help evaluate the effectiveness of the CURE algorithm in air traffic management, providing a theoretical basis and practical guidance for developing future air track prediction methods.

2 Methods and materials

2.1 4D trajectory prediction method and surveillance technology

Currently there are more prediction methods for trajectories. The 4D trajectory prediction method is of far-reaching significance in air traffic management, so the study uses the 4D trajectory prediction technique to predict the flight trajectory. 4D trajectory mainly refers to the main four pieces of information in the airplane trajectory. By recording the time, space, speed and altitude information of aircraft in the air, real-time monitoring and management of aircraft can be achieved. These 4D trajectory data can be used to analyze aircraft flight performance, route planning, aviation safety and other aspects. The airplane flight mainly consists of five directions: take off, departure, cruise, arrival, approach and landing. The whole process of flight is shown in Figure 1.

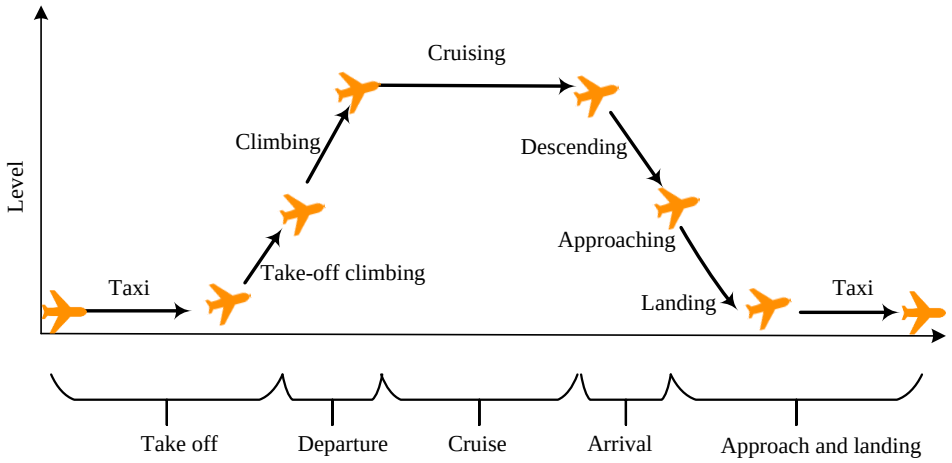


Figure 1: Flight process flow chart.

Figure 1 shows the whole process of an airplane reaching the cruising level through the take-off and climbing process, descending and landing after a period of cruise process. In terms of trajectory prediction, if only a single influencing factor is considered, it will result in a lack of accuracy and comprehensiveness of the actual trajectory of the airplane. Therefore, a comprehensive analysis and prediction should be made by considering a variety of factors, including geographic location, wind speed, temperature and other information. Equiangular routing is the process of route planning according to a certain latitude or longitude angular distance equidistant on a map. In aviation, this usually refers to the fact that routes are set up with fixed points at certain latitude or longitude intervals to facilitate pilots' navigation and localization in flight. This makes each route segment relatively equal in length on the map. Usually, equiangular routes fly longer distances than great circle routes [9]. The starting point $B(\Phi_B, \lambda_B)$ and the starting point $A(\Phi_A, \lambda_A)$ are defined. The angle between the equiangular route and the meridian is a. The distance between the starting points is $AB = H$. At this point, according to the trigonometric function can be obtained route angle a and route distance H, as shown in Equation (1).

$$\begin{cases} \tan a = \dfrac{\lambda_A - \lambda_B}{In\tan(\dfrac{\pi}{4} + \dfrac{\phi A}{2}) - In\tan(\dfrac{\pi}{4} + \dfrac{\phi B}{2})} \\ H = (\phi A - \phi B)\sec a \quad or \quad (\lambda_A - \lambda_B)\cos\bar{\phi}\sec a \end{cases} \tag{1}$$

In Equation (1), $\bar{\phi}$ represents the average latitude value between A and B. π represents PI, which is approximately equal to 3.14159. It is a mathematical constant. In aviation navigation and trajectory prediction, "π" is used to calculate angles, distances, and other parameters related to circular navigation trajectories. When the difference in latitude is large, $\tan a$ in Equation (1) is used. When the difference in latitude is small, H is used, and the unit of H needs to be converted from radians to nautical miles. In the process of calculation, based on the calculated a angle to take the corresponding absolute value. When $a \prec 90^\circ$, based on the position of the course position at this time, the data of the course angle can be calculated, as shown in Figure 2.

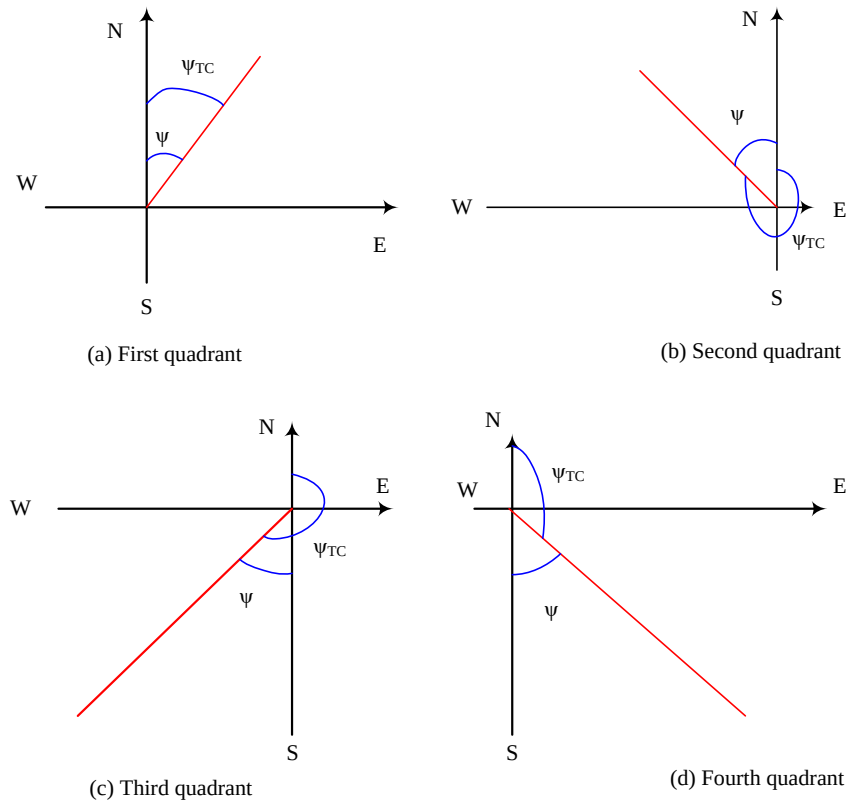


Figure 2: Correlation between route angle and quadrant.

In Figure 2, ψ represents the heading Angle of the equal-angle route, and ψ_{TC} represents the corrected actual heading Angle. When in the first quadrant, ψ_{TC} is identical to ψ. When in the second and fourth quadrants, ψ_{TC} and ψ are circumferential as well as complementary to each other. When in the third quadrant, ψ_{TC} is $\psi + \frac{\pi}{2}$. When $B(\Phi_B, \lambda_B)$, the starting point $A(\Phi_A, \lambda_A)$, the corresponding a and H can be obtained by referring to the trigonometric functions, as shown in Equation (2).

$$\begin{cases} \cot a = \dfrac{\sin\phi_A \tan\phi_B - \sin\phi_A \cos(\lambda_B - \lambda_A)}{\sin(\lambda_B - \lambda_A)} \\ \cos H = \sin\phi_A \sin\phi_B + \cos\phi_A \cos\phi_B \cos(\lambda_B - \lambda_A) \end{cases} \tag{2}$$

Equation (2) defines the calculation methods for the heading angle and route distance of an equal-angle route. Its core role is to provide a mathematical model for an equal-angle route. It converts the difference between longitude and latitude into actual navigation parameters using triangular geometric relations. This solves the symbol correction problem of heading angles in different quadrants, such as northeast and southeast. The study takes the northeast hemisphere as the first quadrant, the southeast hemisphere as the fourth quadrant, the northwest hemisphere as the second quadrant, and the southwest hemisphere as the third quadrant. When the route is in the first and fourth quadrants, then $a_{TC} = 2\pi + a$. When the route is in the second and third quadrants, then $a_{TC} = \pi + a$. Finally, H is converted to nautical miles by the corresponding transformation [10]. Trajectory data has shortcomings such as spatio-temporal correlation, multi-dimensionality, high frequency, and large data volume. By analyzing historical trajectory data, the patterns and laws of aircraft flight are mined to predict future trajectories [11]. As the 4D trajectory prediction technique of data mining is characterized by simple operation, high accuracy and simpler principle, it makes it widely used in the field of trajectory prediction [12]. The study adopts this method for flight trajectory prediction.

2.2 Cluster analysis prediction model based on CURE algorithm

The 4D trajectory prediction method is based on the historical data of ADS-B, and ADS-B reaches the automatic surveillance through the GPS system, the ground-to-air system, and the air-to-air three-dimensional data. ADS-B uses GPS to provide real-time position information of aircraft and broadcasts this information to other aircraft and ground control centers via ground-to-air and air-to-air communication systems. Therefore, GPS is an integral part of the ADS-B system, serving as the basis for providing accurate three-dimensional data, such as position, altitude, and speed. Compared with other surveillance radars, ADS-B is able to provide real-time aircraft position and status data at a frequency of multiple updates per second. It can provide more accurate aircraft position and speed data. It is not limited by terrain or geography and can cover a wider area. Based on ADS-B,

data such as latitude, longitude, heading and speed of the aircraft are preprocessed. Duplicate data points are eliminated after categorizing the flight number. Subsequently after screening and deleting data bars under a certain flight altitude, data bars outside a certain range of the receiver and flight numbers with insufficient data bars are also screened and deleted."The receiver" refers to the device or system that collects and processes the data transmitted from the ADS-B system. Finally, the corresponding trajectory sequence is regenerated. The preprocessed data are categorized and identified, and the cumulative trajectory sequence is obtained. The basic principle of determining trajectory points by the correlation trajectory method is to first determine the overall route of the aircraft, including the departure point, intermediate waypoints and destination. Second, according to the flight plan and flight performance data, the trajectory points are calculated by using the correlation trajectory method, i.e., the trajectory points that need to be passed during the flight. Finally, during the flight, the pilot will correct and adjust the trajectory according to the actual situation. The aerodynamics on the flight path should satisfy Equation (3).

$$\begin{cases} v_{\min} \le \dfrac{|x_i - X|}{|t_i - t_x|} \le v_{\max} \\ v_{\min} \le \dfrac{|x_i - X|}{|t_i - t_x|} \le v_{\max 1} \end{cases} \tag{3}$$

In Equation (3), x_i denotes the ADS-B target position. X denotes the trajectory. M denotes information. The horizontal velocity of the trajectory should be greater than or equal to $v_{\min}$ and less than or equal to the maximum value of the trajectory $v_{\max}$ of X. The vertical speed of the associated trajectory is the same as the horizontal speed of the trajectory. In addition, the course angle a needs to satisfy certain conditions, as shown in Equation (4).

$$\begin{cases} |a| \le a_0 \\ a = \arccos[\dfrac{(x_{i+1} - x_i)(x_i - x_{i-1})}{|x_{i+1} - x_i||x_i - x_{i-1}|} \end{cases} \tag{4}$$

In Equation (4), vector a is the angle between $x_{i+1} - x_i$ and $x_i - x_{i-1}$. In general, to improve the matching probability of the trajectory, a_0 should adopt a larger value. The research focuses on the standardized velocity and route angle constraints in formulas (3) and (4). It presents a table showing the actual value range and source basis, as shown in Table 2.

After normalizing the time [13], the sampling period is shown in Equation (5) based on the historical flight record.

$$T' = \frac{T_p}{t_i} T \tag{5}$$

T' is the sampling time and T_p is the prediction time. T_i is the total flight time on day i. T is the original sampling period of 4s. The trajectory clustering results can further reflect the correlation between the data samples. Through clustering analysis, the logical relationship between each dataset, the classification pattern and the trajectory data can be obtained. The steps of the clustering algorithm are shown in Figure 3.

Table 2: Aviation performance constraints and data sources.

Parameter	Value range	Source	Applicable phase
Minimum horizontal velocity	130 kt (67 m/s)	ICAO Doc 8168 Vol I	Approach/Wait (H ≤ 10,000 ft)
Maximum horizontal speed	320 kt (165 m/s)	ADS-B Measured Data Statistics (Inbound Flights at Pudong Airport)	Cruise (H ≥ 24,000 ft)
Vertical velocity constraint	[-3, +3] m/s	FAA AC 120-29A	Climb/descend
Maximum deviation of the flight angle	±15°	ICAO Annex 11	Flight path
Minimum turning radius	3 nmi (5.56 km)	EUROCONTROL Base of Aircraft Data	Terminal area mobility
Trajectory matching probability threshold	≥0.85	Flight Procedure Verification Specifications of the Civil Aviation Administration of China	All stages

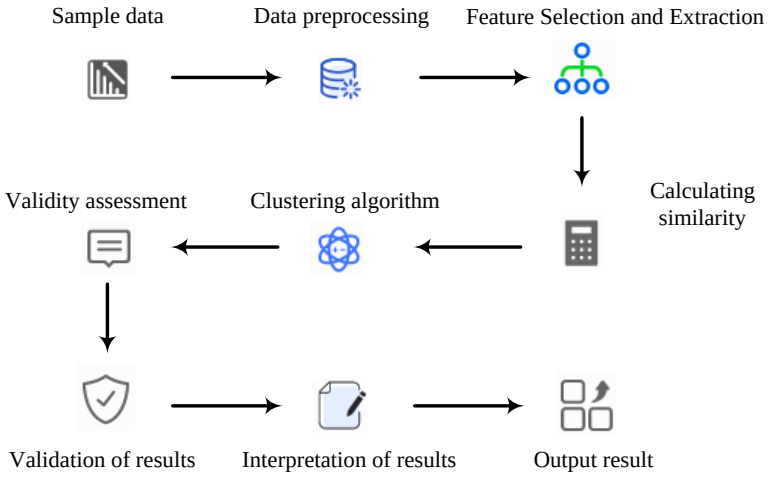


Figure 3: Clustering algorithm process steps.

In Figure 3, based on the obtained sample data is selected through preprocessing, feature extraction. The final result is then output after the relevant data is evaluated, verified, and interpreted through the calculation of similarity and combination with a clustering algorithm. The research selects the hierarchical clustering method in the clustering method. This method is hierarchical, interpretable and does not require pre-specification of the number of clusters [14-15]. Hierarchical clustering method has a good ability to analyze and deal with large amount of data as well as complex changes in the data. At this stage, it has been shown to be used for analyzing hierarchical clustering of flight trajectory data [16]. In the similarity calculation, the study uses Euclidean distance for calculation. Its similarity is shown in Equation (6).

$$\begin{cases} similarity(A_1, A_2) = \sqrt{\varphi_1 (x_1 - y_1)^2 + \varphi_2 (x_2 - y_2)^2 + \cdots + \varphi_n (x_n - y_n)^2} \\ \sum_{k-1}^{n} \varphi_k = 1 \end{cases} \tag{6}$$

In Equation (6), A denotes the aerial trajectory points. These trajectory points are for precise feature extraction and subsequent analysis of the trajectory points to ensure an accurate description of the aircraft's position. n denotes the number of dimensions. φ_k denotes the weight of different dimensions. The weight factor is used to adjust the importance of each dimension in the similarity calculation. It reflecting the extent to which different parameters affect the trajectory characteristics. $similarity(A_1, A_2)$ denotes the dimension attribute difference, which means the difference between two trajectory points in different dimensions, that is, the specific change in position, velocity, height, etc. Since the Euclidean distance is greatly affected by the trajectory data as well as noise, further corrections are needed when using the Euclidean distance for trajectory calculation. The study uses the point-by-point method to calculate the route distance, and then adjusts the Euclidean distance similarity calculation using the mean value method. Finally, the trajectory is corrected according to the calculated distance variance to obtain the final trajectory similarity. The corrected trajectory calculation is shown in Equation (7).

$$t_i^* = \begin{bmatrix} \begin{Bmatrix} G_1^{(i)} \\ G_2^{(i)} \\ \vdots \\ G_k^{(i)} \\ \vdots \\ G_M^{(i)} \end{Bmatrix} \end{bmatrix} \quad t_j^* = \begin{bmatrix} \begin{bmatrix} \begin{Bmatrix} G_1^{(j)} \\ G_2^{(j)} \\ \vdots \\ G_k^{(j)} \\ \vdots \\ G_M^{(j)} \end{Bmatrix} \end{bmatrix} \end{bmatrix} \tag{7}$$

In Equation (7), t_i^* denotes the flight path trajectory. M is the number of trajectory points contained in trajectories t_i^* and t_j^*. $G_k^{(i)}$ and $G_k^{(j)}$ represent a point pair. Therefore, the equirectangular air trajectory between the point pairs can be expressed as $x_k = d(G_k^{(i)}, G_k^{(j)})$. At this time, the set of distances of each point pair is $X = \{x_1, x_2, x_3 \cdots, x_M\}$. The similarity is calculated as shown in Equation (8).

$$s_{ij} = \frac{\sum_{k=1}^{p} x^k}{M} + D(x) + \Delta \tag{8}$$

In Equation (8), $D(X)$ represents the structural similarity measure and Δ represents the dynamic correction factor. p represents the number of trajectory points involved in the similarity calculation. Equation (6) has two main flaws in the calculation of aviation trajectory similarity: noise sensitivity and density difference.

After correcting these defects in the research, Equation (8) is obtained. The main derivation of Equation (8) is as follows: assuming that the trajectory contains M trajectory points, the normalization factor $M = \max(m, \bar{m})$ is defined. Among them, $\bar{m}$ is the average number of trajectory points in the dataset. Then the density normalization term EEE is introduced. The density normalization term scales the trajectory point coordinates by 1/M, which eliminates cumulative error caused by differences in sampling frequencies. The trajectory structure similarity measurement extracts geometric features of trajectories based on principal component analysis. Equation (9) shows the covariance matrix factorization of the trajectory point set.

$$\sum X = \frac{1}{m}(X - \mu_X)^T (X - \mu_X) \tag{9}$$

The research assumes that the eigenvalues λ_1, λ_2, and λ_3 represent the energy distribution of the trajectory in the main direction in three-dimensional space. The structural similarity measure is defined as shown in Equation (10).

$$D(X) = \frac{\lambda_2 + \lambda_3}{\lambda_1} \tag{10}$$

In Equation (10), when $D(X)$ approaches 0, it indicates that the trajectory has the characteristic of linear motion. When $D(X)$ approaches 1, it indicates a drastic change in the trajectory direction. Finally, combined with the characteristics of the typing stage, the dynamic correction factor is defined as shown in Equation (11).

$$\Delta = \alpha \cdot |\gamma_X - \gamma_Y| + \beta \cdot |\omega_X - \omega_Y| \tag{11}$$

In Equation (11), Δ represents the dynamic correction factor. γ represents the climb rate. ω represents the turning angular velocity. α represents the weight coefficient of the climb rate. β represents the turning angular velocity weight. Based on the above correction items, Equation (8) for trajectory similarity calculation is obtained. The similarity matrix S can be obtained after the operation based on Equation (7), as shown in Equation (12).

$$S = \begin{bmatrix} 0 & s_{12} & \cdots & s_{1j} & \cdots & s_{1n} \\ s_{21} & 0 & \cdots & s_{2j} & \cdots & s_{2n} \\ \vdots & \ddots & & \vdots & & \\ s_{i1} & & 0 & & s_{in} & \\ \vdots & & & \ddots & \vdots & \\ s_{n1} & s_{n2} & \cdots & s_{nj} & \cdots & 0 \end{bmatrix} \tag{12}$$

In Equation (12), since S is obtained based on the symmetric Euclidean distance operation, the matrix is also called symmetric matrix. Since the number of discrete points of each aerial trajectory is different, and it is difficult to have a corresponding situation in the data, the study uses the CURE algorithm in order to realize the clustering prediction of the aerial trajectories [17]. The CURE algorithm applied in the study is described below. Firstly the input defines the set of trajectories $T^* = \{t_1^*, t_2^*, t_3^*, \cdots, t_n^*\}$, representing points Q, and the number of clusters k. The second output is the clustering result of the flight trajectories $C = \{C_1, C_2, C_3, \cdots, C_k\}$, with clustering centers $AVE = \{ave_1, ave_2, ave_3, \cdots, ave_k\}$. Each category represents the aerial trajectory $Q_e = \{Q_{e1}, Q_{e2}, Q_{e3}, \cdots Q_{ek}\}$ of the point. An algorithm based on the modified Euclidean distance aerial trajectory similarity is used to calculate the similarity between 2 aerial trajectories in the trajectory set T^*. The trajectory similarity matrix is then constructed. The trajectory similarity matrix Q_T, as shown in Equation (13), is also constructed.

$$Q_T = \begin{bmatrix} 0 & Q_{12} & \cdots & Q_{1j} & \cdots & Q_{1n} \\ Q_{21} & 0 & \cdots & Q_{2j} & \cdots & Q_{2n} \\ \vdots & \ddots & & \vdots & & \\ Q_{i1} & & 0 & & Q_{in} & \\ \vdots & & & \ddots & \vdots & \\ Q_{n1} & Q_{n2} & \cdots & Q_{nj} & \cdots & 0 \end{bmatrix} \tag{13}$$

In Equation (13), after obtaining the similarity matrix Q_T, the initialization clustering is carried out. Subsequently, the center trajectories ave_a, ave_β of C_a and C_β are calculated. Meanwhile, the aerial trajectory sets Q_{ea} and $Q_{e\beta}$ are calculated to determine whether the above two aerial trajectories are the aerial trajectory sets of the current clustering result. If they are different genus classes, merge the classes and define a new genus class C_δ, and reduce the number of classes in the dynamic array by 1. Calculate the center flight trajectory ave_δ of C_δ and the representative trajectory set $Q_{e\delta}$. Finally, the relevant results are obtained as shown in Equation (14).

$$\begin{cases} C = \{C_1, C_2, C_3, \cdots, C_k\} \\ AVE = \{ave_1, ave_2, ave_3, \cdots, ave_k\} \\ Q_{e=}\{Q_{e1}, Q_{e2}, Q_{e3}, \cdots, Q_{ek}\} \end{cases} \tag{14}$$

C_k denotes the set of Class k flight trajectories. ave_k denotes the cluster center flight trajectory of C_k. Q_{ek} denotes the representative air trajectory set of C_k. The overall trend of flight operation can be somewhat reflected by the cluster trajectories obtained from the clustering of the CURE algorithm. In addition, the relevant information of the next flight trajectory can be obtained by correcting the historical flight trajectory data, as shown in Equation (15).

$$\begin{cases} C^p(\Gamma + h) = C^c(\Gamma + h) - \varepsilon(\Gamma) \\ \varepsilon(\Gamma) = C^c(\Gamma) - C^R \end{cases} \tag{15}$$

In Equation (15), h denotes the step size of the prediction time. C^p, C^c, and C^R denote the prediction, clustering, and real trajectory, respectively. $\varepsilon(\Gamma)$ denotes the correction value coefficient. Equation (15) uses the cluster center trajectory and historical trajectory data

obtained from the clustering of the CURE algorithm, combines the predicted value and the real trajectory, and makes a correction prediction for the next flight trajectory. The correction coefficients can be used to adjust the degree of contribution of the predicted value and the cluster center trajectory, as well as the degree of contribution of the predicted value and the true trajectory. In this way, a more accurate prediction of the next flight trajectory can be obtained, thus providing a more reliable information reference for the overall trend of flight operation. The modified Euclidean distance similarity measure proposed in the research is a custom algorithm module, and its core mathematical operations are implemented through NumPy in Python. The module is integrated with Scikit-learn's Agglomerative Clustering framework. The similarity calculation function is accessed via the custom affinity parameter. The experimental platform is built using the PyTorch Geometric spatio-temporal data processing library. The specific study of pseudo-code is shown in Figure 4.

3 Results

3.1 4D trajectory prediction based on CURE algorithm data mining

The proposed prediction model is designed based on real-time stream processing architecture and can be seamlessly integrated into the existing air traffic control (ATC) system. The system accesses the ADS-B data stream in real time through Apache Kafka with an update frequency of 1 Hz. It dynamically extracts trajectory fragments using the sliding time window mechanism with a window length of 300 s and a sliding step length of 10 s. The system updates the track pattern online using the incremental CURE clustering algorithm with an update cycle of 30 s. The real-time prediction engine jointly uses Kalman filtering for state estimation and an LSTM time series prediction module for behavior reasoning. This generates 4D trajectory prediction results for the next 300 s in a pipeline manner and achieves synchronization of prediction states among multiple nodes through a distributed Redis database. The CURE algorithm combined with cluster analysis is used to analyze the actual ADS-B data of the approach flights of an airport in East China as the corresponding samples, and the cluster analysis is performed after the relevant data transformation. First, the study conducts different index analyses on various numbers of clusters to determine the optimal number of clusters. The indicators include the silhouette score (SS), the Davies-Bouldin index (DBI), the sum of squared errors (SSE), and the Calinski-Harabasz index (CHI). The specific results are shown in Table 3.

```
Define CURE_Algorithm(data, k, r):
    Preprocess data:
        Clean, filter, and normalize data

    Initialize clusters with initial representatives

    While number of clusters < k:
        For each trajectory in data:
            Assign trajectory to closest cluster based on modified distance

        Update representative points in each cluster

    Return clusters and representatives

Define Conflict_Detection(clustered_data):
    For each trajectory pair in clustered_data:
        If IsPotentialConflict(trajectory1, trajectory2):
            Add to conflict list

    Update metrics (TP, FP, FN, TN)
    Return Precision, Recall, F1-Score

Main:
    Load flight data
    Set parameters: k, r

    clustered_data = CURE_Algorithm(data, k, r)
    conflict_metrics = Conflict_Detection(clustered_data)

    Output conflict_metrics
```

Figure 4: Study the pseudo-code of the algorithm.

Table 3: Clustering validity evaluation under different cluster numbers.

Cluster Number (k)	SS	DBI	SSE	CHI
2	0.52 ± 0.04	1.38 ± 0.12	2850340	86.7
3	0.61 ± 0.03	1.12 ± 0.09	1732150	123.4
4	0.68 ± 0.02	0.83 ± 0.06	1001920	182.5
5	0.65 ± 0.03	0.97 ± 0.08	921450	168.3
6	0.58 ± 0.05	1.05 ± 0.11	898210	154.9

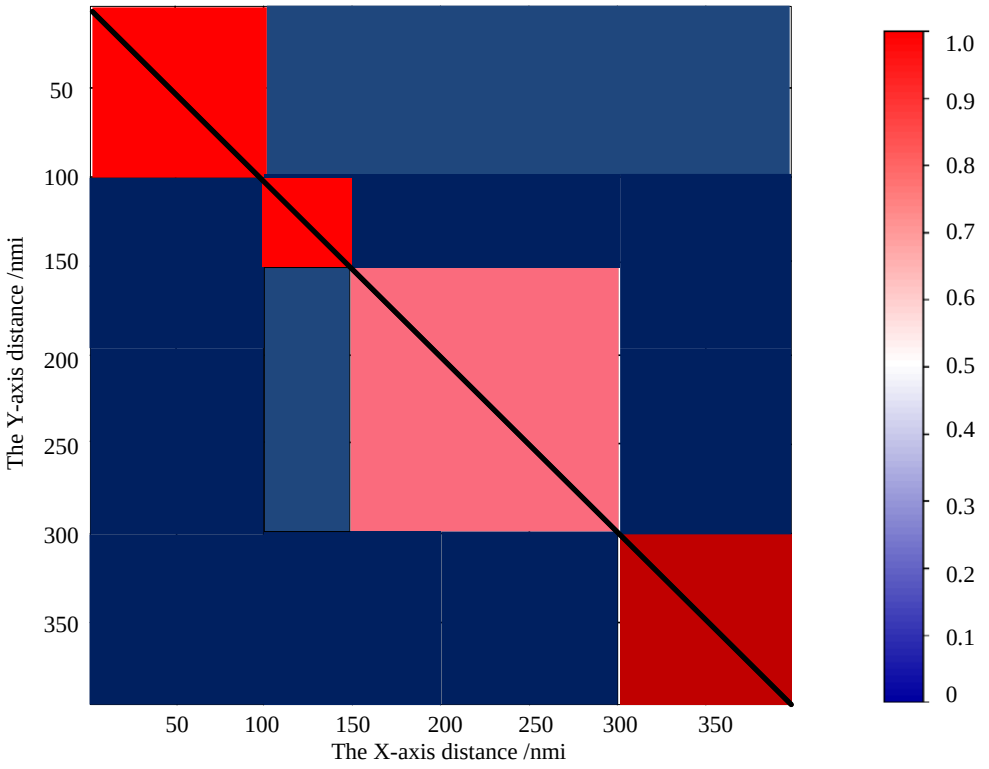
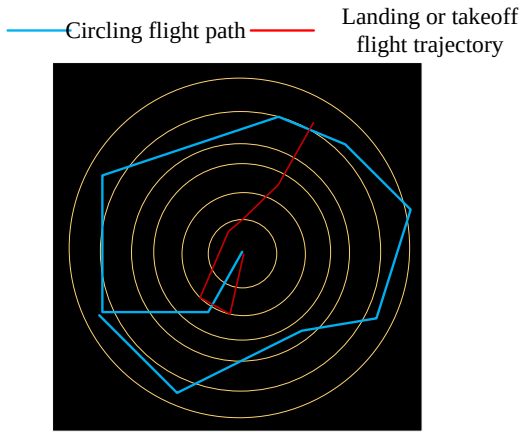


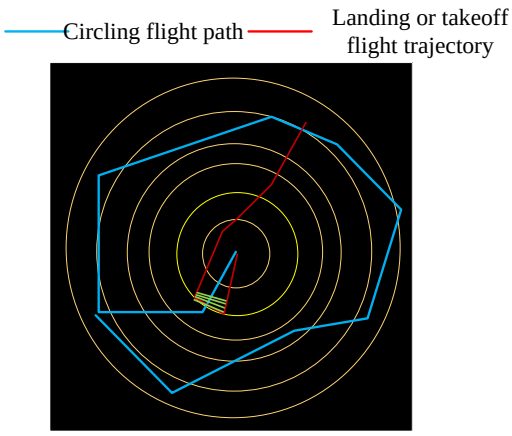
Figure 5: Matrix after CURE algorithm modified euclidean distance similarity clustering.

As shown in Table 3, the contour coefficients of the four clusters (k=4) are the highest (0.68), the DBI is the lowest (0.83), and the CHI is the highest (182.5) among different numbers of clusters. This indicates that, in the case of four clusters, the separation and closeness among the clusters are relatively good and the variability of the data is effectively explained. This verifies the rationality of having four clusters. Then, the similarity based on the modified Euclidean distance was studied and combined with the CURE algorithm. After obtaining the similarity matrix, it was added to the CURE algorithm. The relevant results of the aviation trajectory clustering map in the terminal area of Shanghai Pudong Airport are shown in Figure5.

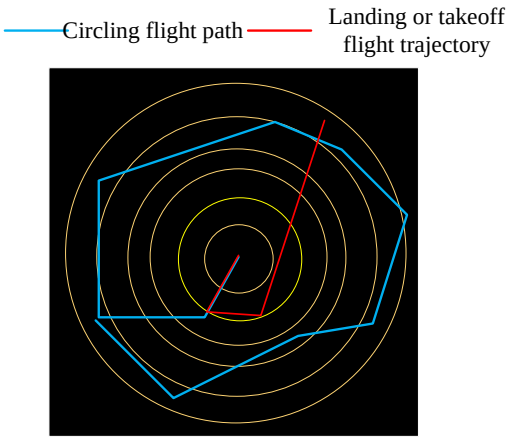
In Figure5, by mapping the similarity of different flight trajectories into gradient colors, the size of similarity is visually reflected by the color depth [18-19]. The horizontal axis represents horizontal distance within a specific coordinate system and reflects the spatial distribution of track points in the east-west direction. The vertical axis represents the vertical distance of the track points within the same coordinate system and reflects the spatial distribution of the flight in the north-south direction. After data analysis and clustering by CURE clustering algorithm, trajectories with higher similarity will be clustered together and the clustering results are in obvious dark red color. The results indicate that these aircraft trajectories are more consistent in their flight paths. The display plots of the four approaching aircraft are shown in Figure6, respectively.



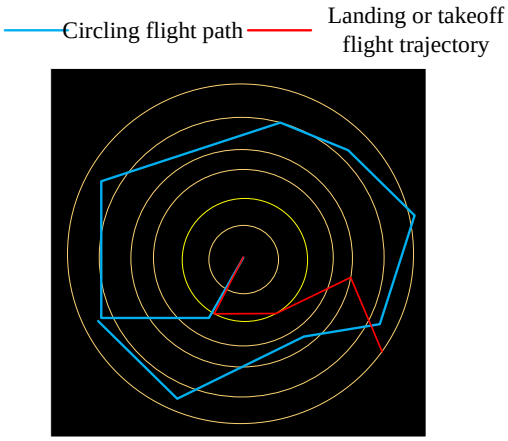
(a) The first class of trajectory clustering extraction



(b) The second class of trajectory clustering extraction



(c) The third class of trajectory clustering extraction



(d) The fourth class of trajectory clustering extraction

Figure 6: Approach trajectories obtained after separate clustering.

Figure 6 shows the aircraft's circling flight route in blue and its takeoff or landing route in red. This figure shows the aircraft's path as it passes through the runway after landing, as well as any turns or adjustments it makes during the ground taxiing process. After clustering by the CURE algorithm, the flight trajectories of the aircrafts in Figure 6(a) and Figure 6(b) are more consistent. The difference is that the flight trajectory of the aircraft shown in Figure6b flies back and forth in a certain section of the path. In Figure 6(c) and Figure 6(d), the blue flight trajectories are found to be basically the same after clustering, and only the red path is different due to the different departure or arrival procedures. The results show that similar flight trajectories are successfully clustered together after clustering by the CURE algorithm. This verifies the effectiveness of the CURE algorithm in processing high-dimensional data and discovering potential patterns in the data. The consistency of flight trajectories of different approaching aircrafts is high when they enter the airport terminal area. Figure 6 shows that similar trajectories can be visually observed clustered together by the color coding of the different trajectories. This suggests that under the same conditions, airplanes often experience similar flight paths during the approach. In flight planning, airlines tend to choose similar flight routes. Furthermore, Figure 6 shows that the intervals and flight altitudes between flights on some paths remain consistent. This reflects the effective control measures and interval management during the approach process. Based on the above obtained flight trajectory data is firstly time normalized. In order to further verify the reasonableness of the algorithm proposed in the study, the actual data on the ground in a certain area are selected for experimental analysis. The data results of the aerial data after filtering and noise reduction are shown in Figure 7.

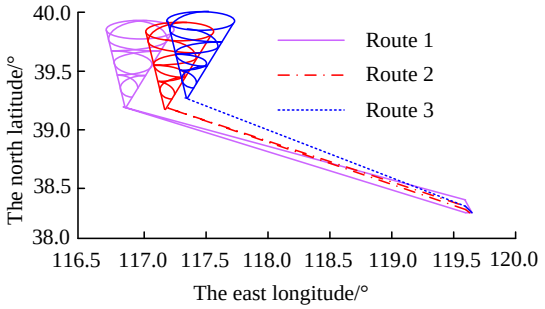
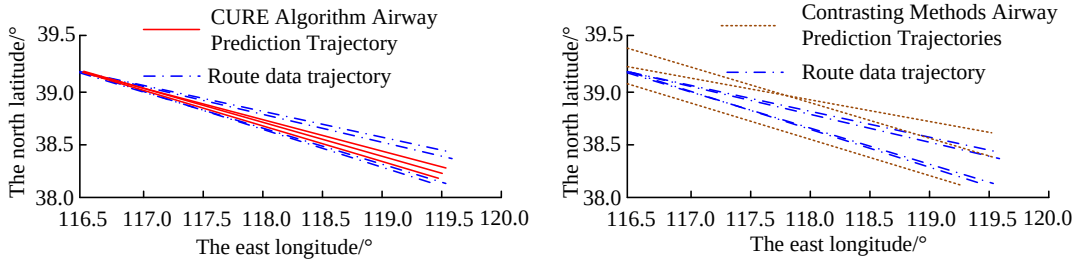


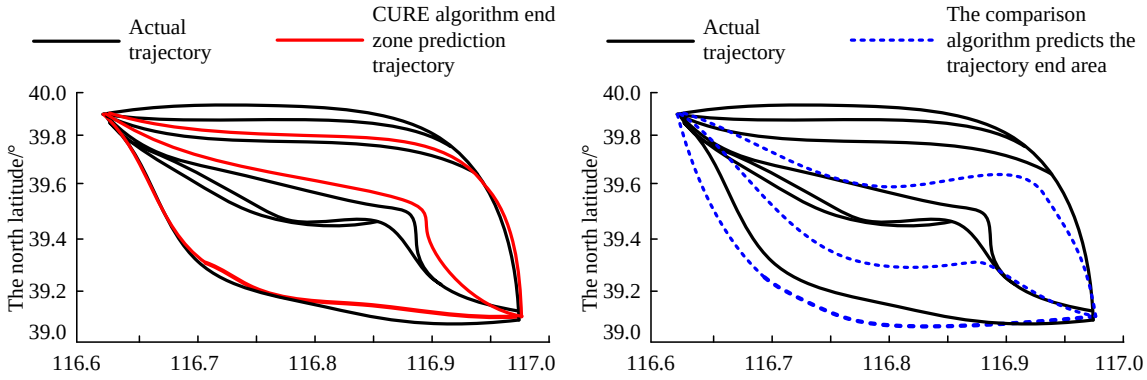
Figure 7: Filtered and noise reduced aerial trajectory.



(a) The result of airway prediction trajectory by CURE algorithm

(b) The result of airway prediction trajectory by Contrasting Methods

Figure 8: Flight path prediction of part of the route.



(a) The CURE algorithm predicts the result

(b) The comparison algorithm predicts the result

Figure 9: Partial trajectory prediction of CDG4651 terminal area.

Figure 7 shows the motion trajectory of Flight CDG4651 after filtering and noise reduction processing. There are three routes in total: Route 1, Route 2, and Route 3. After processing, the track data shows high stability. Its routes are concentrated primarily between 116.5° and 120 ° east longitude and 38° and 40° north latitude. This stable trajectory means that, under normal circumstances, flights can maintain a consistent course and speed when approaching the ground. This reflects the effectiveness of flight scheduling. The method described in the literature [20] is also used for comparison, and the results are shown in Figure 8.

Figure 8(a) and Figure 8(b) represent the flight trajectory prediction route results of the two methods, respectively. The proposed method's prediction shows that the flight data trajectory distribution is relatively uniform and that there are no obvious outliers. It indicates that the model performs well in feature extraction and cluster analysis. In contrast, other methods' prediction results have obvious deviations and cannot accurately match the actual flight path. These results further verify the CURE algorithm's effectiveness in track prediction. This indicates that the algorithm is suitable for processing complex track data and can provide accurate predictions, offering effective decision support for air traffic management. It shows that it is feasible to study the method of route trajectory prediction based on CURE algorithm. Part of the terminal area trajectory prediction is shown in Figure 9.

As shown in Figure 9, the CURE algorithm can better reflect the trajectory characteristics of the flight at different stages (e.g., climbing, cruising, and descending) by fitting the predicted trajectory of the flight terminal area. Due to the influence of wind direction and traffic control, the flight path is clearly divided into two main routes, indicating that flights in the terminal area are restricted by multiple factors. The CURE algorithm's predicted trajectory is relatively close to the actual trajectory's distribution, with no obvious outlier phenomenon occurring and remaining within a reasonable error range. This accurate prediction of the trajectory can provide airlines with a scientific basis for flight adjustments and scheduling, thereby improving the safety and efficiency of flight operations. At the same time, the predicted crossing time and the actual crossing time, the crossing height and the predicted crossing height on June 1st are compared. The results are shown in Table 4.

Table 4: Comparative analysis of the prediction results of the crossing point time and crossing point altitude.

Waypoint	A little bit of actual time	Excessive time prediction	Error /s	Height of actual crossing point	Over-point height prediction	Error/m
FD	9:43:44	9:43:39	5	5650	5630	20
TEKAM	9:50:03	9:49:56	7	7910	7880	30

HCH	9:55:16	9:55:07	9	7920	7920	0
NOKAK	10:02:29	10:02:15	14	7920	7920	0
CG	10:26:16	10:25:50	26	4770	4820	50

Table 5: Statistical aggregation of prediction errors across multiple flight trajectories.

Flight number	Aircraft type	Flight phase	Time error (s)	Altitude error (m)	Horizontal RMSE (nmi)	Data points
CDG4651	A320	Climb	18.2 ± 4.3	32.5 ± 12.7	0.58	1250
MU5131	B737	Cruise	12.7 ± 3.1	21.8 ± 8.9	0.42	980
CA1765	A330	Descent	24.6 ± 5.7	47.3 ± 15.2	0.71	1120
HO1721	A320	Terminal Maneuvering	15.9 ± 3.8	29.1 ± 10.4	0.53	860
3U8815	B737	Cruise	13.5 ± 3.0	22.6 ± 9.1	0.45	1050
GJ8751	A320	Climb	19.8 ± 4.5	35.7 ± 13.5	0.62	1340
Aggregated Mean	/	/	17.6 ± 4.1	31.5 ± 11.8	0.55	6600

Table 6: Comparison of errors between the research method and the existing benchmark model.

Model	Time error (s)	Altitude error (m)	RMSE	MAE
Bi-LSTM	35.2±8.7	68.5±15.3	1.24	0.93
TCN-GRU	28.9±6.5	55.2±12.1	0.87	0.65
Attention-GRU	26.4±5.2	49.8±10.7	0.72	0.54
DBSCAN-GRU	31.7±7.1	61.3±13.9	1.05	0.78
HMM+k-medoids	42.5±9.8	73.6±16.2	1.48	1.12
Proposed CURE	23.6±4.8	46.3±9.5	0.61	0.47

In Table 4, Fractional Descent (FD) is usually used to represent specific flight points of an aircraft during the descent phase. TEKAM is a specific waypoint that aircraft must pass through during flight. The name of a waypoint is typically a combination of letters and numbers. "High corridor heading (HCH)" refers to the aircraft's high-altitude route points during flight. NOKAK is a key airway point where aircraft pass when flying along the designated route. A cruising gate (CG) indicates that an aircraft has entered the cruising phase. This means that the aircraft has reached its predetermined cruising altitude and will maintain a certain heading. By comparing the error times, it can be concluded that the crossing point time errors gradually increase with the advancement of the flight process, but the errors are kept within half a minute. The crossing altitude errors are also small and remain within the range of 50 m. The HCH and NOKAK waypoint altitude errors are 0, indicating that the prediction accuracy based on the CURE clustering algorithm is high. To verify the universality of the method proposed in the research, the prediction results of more flights are analyzed, as shown in Table 5 specifically.

As shown in Table 5, the statistical analysis of the prediction results indicates that the research method is effective in predicting multiple flight trajectories, demonstrating its applicability and reliability. The time and altitude errors for each flight at different stages were all within a reasonable range. During the climbing stage of flight CDG4651, the time error is 18.2 s and the altitude error is 32.5 m, demonstrating good prediction performance. The aggregated data indicates that the average time error of each flight is 17.6 s, the altitude error is 31.5 m, the horizontal root mean square error (RMSE) is 0.55 nautical miles, and the total number of data points is 6,600. These results indicate that, although performance varies among different aircraft and flight phases, the average overall prediction error remains relatively small. This further verifies that the proposed method has good predictive ability in complex aviation scenarios and can strongly support the accuracy of flight trajectories. To verify the effectiveness of the proposed method in the research, a comparative analysis of the errors with the existing benchmark model is conducted. The results are shown in Table 6.

As shown in Table 6, compared to the optimal Attent-GRU model, the CURE algorithm reduces time error by 10.6% (26.4 s to 23.6 s), improves height error by 7.0% (49.8 m to 46.3 m), and increases the false alarm reduction rate by 44.6% (12.1% to 17.5%). Although the calculation time increases by 18% compared with DBSCAN-GRU, the improvement in accuracy (a 30% reduction in RMSE) exceeds the efficiency loss, verifying the superiority of the method in the precision-efficiency trade-off. The above study only focuses on the 4D flight trajectory planning for individual aircraft. In the actual process of multi-class aircraft mutual conflict as the main research problem, through the research proposed method to solve the problem. The specific results are shown in Figure10.

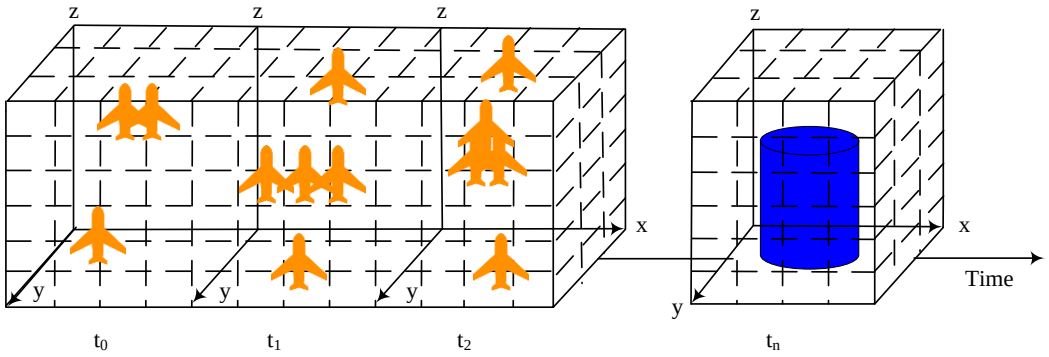


Figure 10: Aircraft 4D pane demonstration map.

In Figure 10, the X-axis represents the horizontal position, which is the longitude on the ground or the horizontal distance relative to the reference point. The Y-axis represents the horizontal position in another direction. It is the latitude on the ground or the longitudinal distance relative to the reference point. The Z-axis represents altitude, indicating the flight altitude of the aircraft. the presence of two or more flight trajectory points in the pane indicates the presence of flight conflicts in this space. In the pane where there are two or more flight trajectory points, the flight conflicts in these areas can be clearly observed. Timely measures can be taken for air traffic scheduling to avoid conflicts between aircraft. Where the minimum interval of the protected area H=304.8m, which represents the height of the cylindrical protected area. The minimum horizontal interval of the flight interval protection area S=9260m, which represents the radius of the bottom surface of the cylindrical protection area. The results show that the CURE algorithm can be used to cluster air trajectories and analyze flight trajectories in the terminal area. This clearly indicates potential areas of flight conflict. Moreover, the necessary protection zone parameters are set, which can effectively predict and identify these conflict points, thus providing data support and decision-making basis for air traffic control. The quantitative performance results of the supplementary conflict detection method for Figure 10 are studied, as shown in Table 7 specifically.

The quantitative performance analysis in Table 7 shows that the conflict detection method proposed by the research performs well in multiple key indicators, demonstrating its efficiency and accuracy. Specifically, the FAR is 4.30%, which is ≤5% lower than the target value, while the MDR is only 0.08%, FAR lower than the target of ≤0.1%. This indicates that this method outperforms others in reducing false alarms and missed detections. In terms of response time, it averages 38 ms, which is faster than the allowable maximum of 50 ms. Furthermore, the R-tree's query efficiency is 18.2 ms per query, which meets the ≤20 millisecond requirement and demonstrates fast processing capability. The dynamic protected area's accuracy reaches 98.50%, which is much higher than the ≥95% standard. This indicates its effectiveness in providing an early warning of conflict. Meanwhile, the separation accuracy is 0.12±0.05 nautical miles in the horizontal direction and 28±9 m in the vertical direction. Both met the predetermined accuracy requirements. The study selects the flight data of an international airport in 2019 to analyze and verify. Some flight information is shown in Table 8.

Table 7: Quantitative performance results of conflict detection methodology.

Metric	Test result	Benchmark
False alert rate (FAR)	4.30%	≤5%
Missed detection rate (MDR)	0.08%	≤0.1%
Response time	38 ms	≤50 ms
R-tree query efficiency	18.2 ms/query	≤20 ms
Dynamic protection zone accuracy	98.50%	≥95%
Horizontal separation accuracy	0.12 ± 0.05 nmi	≤0.2 nmi
Vertical separation accuracy	28 ± 9 m	≤30 m

Table 8: Airport flight schedule partial information table.

Serial number	Flight number	Models	The moment of take-off	Destination airport
1	GJ8751	A320	7:00	XNN
2	H01721	A320	7:00	TAO
3	MU5655	A320	7:05	KMG
4	3U8815	A320	7:05	CGQ
5	GH8881	A320	7:05	SZX
6	HU7421	B737	7:05	KMG

7	CA1765	A320	7:05	BAV
8	MU5131	B737	7:10	KRL
9	GJ8841	A320	7:10	PEK

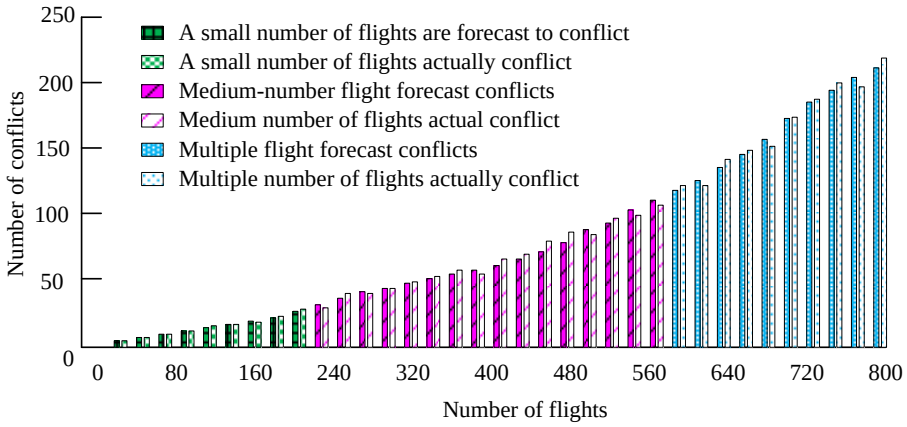


Figure 11: Plot of the total number of flights in the terminal area versus the number of conflicts.

Table 8 shows a table of information for the nine flights before 7:00 a.m. that day. In order to simulate the most realistic situation, the corresponding incoming flights are added for uniform conflict adjustment. The relevant geographic coordinate system is transformed as follows: E represents the latitude of the Earth, N the longitude, and H the flight altitude of the aircraft. "e" denotes the first eccentricity of the earth, and R denotes the radius of curvature of the earth's dodo circle. Combining the above 4D pane flight conflict method as well as the geometric method of flight conflict detection. The results are shown in Figure 11.

Figure 11 shows the results of the research method's flight prediction conflict. The horizontal axis represents the number of flights and is divided into three categories: a small number of flights, a medium number of flights, and a large number of flights. The vertical axis represents the number of flight conflicts. The results show that the predicted conflict value of the method is almost equal to the real value, and the prediction error is less than 0.5%. There is a certain error between the predicted value and the real value of the medium-number flights, and the maximum error is controlled within 1%. The error between the predicted value and the real value of the multi-number flights is basically controlled within 3%. The results show that the proposed method has high accuracy of flight conflict prediction. At the same time, it shows that the proposed method has better processing ability and analysis ability in complex flight trajectories. To explore the performance of the CURE algorithm in track prediction, a comparative experimental scheme is studied and designed. The performance of different algorithms is summarized by comparing the models of multiple related works. The experiment adopts actual flight trajectory data, including data under different meteorological conditions and high traffic flow situations. The evaluation indicators adopt contour coefficient, Davier-Bouldin index (DB Index) and RMSE, aiming to analyze the performance effect of the algorithm. The research adopted the following comparison methods: Bi-LSTM, DBSCAN-GRU, Hidden Markov Model, and the CURE algorithm. The comparison results are shown in Table 9.

Table 9: Performance comparison of different models.

Method	Contour coefficient	DB Index	RMSE
Bi-LSTM	0.68	1.85	86.5
DBSCAN-GRU	0.60	2.10	92.3
Hidden Markov model	0.65	1.90	88.1
CURE algorithm	0.75	1.65	73.4

Table 10: Conflict detection confusion matrix.

Method	4D pane method	Research method	Mixed detection method
Real conflict (TP)	155	162	143
False conflict (FP)	84	37	68
Missed detection conflict (FN)	16	1	14
True negative (TN)	10205	10311	10285
Precision	0.625	0.814	0.678
Recall	0.957	0.985	0.911
F1-score	0.752	0.924	0.784

In Table 9, the CURE algorithm performs better than other comparison models in the track prediction task. With a contour coefficient of 0.75, the CURE algorithm demonstrates good compactness and separability of its clustering results, achieving high-quality trajectory classification. However, the contour coefficients of other models, such as Bi-LSTM, DBSCAN-GRU, and the hidden Markov model, are all lower than those of CURE. This indicates that the clustering effect of CURE is relatively poor. In terms of the Davies-Bouldin index, CURE's value is 1.65, which is significantly lower than that of the other models. This indicates that the clusters are less similar to each other and that the clustering effect is better. Finally, the evaluation of RMSE verified the high accuracy of the CURE algorithm in coordinate prediction, with a minimum error of 73.4. The RMSE of the other models was higher than that of the CURE algorithm. These results suggest that the CURE algorithm is particularly effective in processing complex track data and enhancing prediction accuracy. The results demonstrate that the CURE algorithm is more robust when dealing with high-dimensional data because it introduces the concept of representative point clustering. Compared with traditional clustering methods, CURE can effectively handle outliers in the data and maintain sensitivity to high-dimensional features. This allows CURE to overcome the influence of data noise when clustering aviation trajectories and maintain high interpretability throughout the process. Furthermore, CURE's hierarchical clustering feature enables it to provide stable results when dealing with complex trajectories of different categories. It performs even more outstandingly when the trajectories are widely distributed and have various shapes. To verify the effectiveness of the proposed method, it is compared to and analyzed alongside the hybrid detection method of HMM, which is used to recognize heading change patterns and determine fixed threshold conflicts. The specific results are shown in Table 10.

As shown in Table 10, the analysis of the conflict detection confusion matrix results indicates that the research method successfully identified 162 TP, which is higher than the 143 TP identified by the hybrid method. Meanwhile, the number of FP is only 37, which is lower than the 68 FP of the hybrid method. This demonstrates better accuracy. The research method's recall rate reaches 0.985, higher than the hybrid method's 0.911, demonstrating an outstanding ability to detect and capture missed cases. Additionally, the research method's accuracy is 0.814, surpassing the mixed method's 0.678. The F1 score is 0.924, surpassing the mixed method's 0.784. These results reflect a superior conflict detection effect and false alarm control.

4 Discussion

In summary, this study proposed a novel, four-dimensional trajectory prediction method based on the CURE clustering algorithm. This method aimed to improve prediction accuracy in complex aviation environments and solve key challenges in four-dimensional trajectory prediction for air traffic management. The research selected data from January to May 2017 for comparison with data from June 1, 2017. This period was chosen because it provided sufficient historical flight trajectory data to analyze the performance of the flight trajectory prediction model. June 1st was usually the peak period of increased air traffic, which effectively tested the adaptability and accuracy of the proposed method under different traffic conditions. Through comprehensive experiments, this method proved its superior performance by using hierarchical clustering of ADS-B historical data and an improved Euclidean distance similarity matrix. The prediction error was limited to 26 s and 50 m in height. With 6,600 data points from multiple flights, the aggregated average error was 17.6 s and 31.5 m. A comparative analysis showed that the CURE algorithm reduced time error by 10.6% and RMSE by 30% compared with Attention-GRU. A contour coefficient of 0.75 and a DB index of 1.65 were achieved. The accuracy rate of the conflict detection index in the dynamic protected area was 98.5%. The F1 score was 0.924, superior to the mixed method. The results demonstrated that introducing the modified Euclidean distance to calculate track similarity effectively reduced the influence of data noise and improves the reliability of the calculation. The accuracy of flight path prediction was improved through the combination of real-time and historical flight data and comprehensive analysis. A flight conflict detection method based on the 4D pane and geometric approaches was proposed to reduce false conflict alerts and improve flight safety. A hierarchical clustering method was used to process large-scale, complex aviation trajectory data and reveal its inherent logical relationships. The experiment revealed that time errors gradually increased as the flight progressed. This might be primarily due to the cumulative effect of wind speed changes during the cruise stage and the delay in transmitting control instructions, which affected the trajectory. The research could be corrected dynamically by introducing real-time meteorological data streams and synchronizing the ATC instruction interface. During the cruise phase, the flight altitude prediction error approached0, but it reached 50 m during the descent phase. This was related to wind shear disturbances in the airport terminal area and the flaps' operating mode. Aerodynamic configuration parameters must be added, and the weight coefficients must be optimized in the dynamic correction factor. The research aims to enhance the accuracy and reliability of 4D trajectory prediction to provide more effective support for air traffic management. At the same time, it aims to solve conflicts among multiple flights to ensure the safety and efficiency of aviation operations.

5 Conclusion

The rapid growth of global air traffic in recent years has placed unprecedented pressure on air traffic management and presented new challenges. Against this backdrop, 4D trajectory prediction technology has become increasingly important as a means to improve the utilization rate of airspace resources, reduce the workload of ground control

personnel, and enhance flight safety. The research proposed a 4D trajectory prediction method based on the CURE algorithm. Using widely adopted ADS-B historical data, it measured trajectory similarity through modified Euclidean distance. Then, it combined this measurement with hierarchical clustering technology to cluster and analyze trajectory data in the airport terminal area. The research used a real-time stream processing architecture that integrated a Kalman filter and an LSTM time series prediction module. This effectively achieved real-time prediction of the aircraft's future flight path. The experimental results showed that the 4D trajectory prediction model based on the CURE algorithm was more accurate than many existing benchmark models, especially when dealing with high-dimensional data. This model demonstrated good robustness and reliability. Through experimental comparison, the accuracy of track prediction and the rate of conflict detection had significantly improved with the proposed method. The error rate remained within a reasonable range. For example, the average time error was reduced to 17.6 s and the average height error is 31.5 m. The study showed that appropriate clustering methods and similarity calculations could significantly improve the accuracy of flight path prediction, optimize the flight management process, and ultimately support aviation safety and effective airspace resource management. Although the initial research demonstrated the CURE algorithm's potential for trajectory prediction, several deficiencies remain. For instance, the research primarily focuses on predicting the trajectories of specific flights and does not address converting flight data into planar and geodetic coordinates. The real-time data processing and dynamic trajectory prediction capabilities have not yet been fully verified. The prediction model used by the research must be expanded to include real-time dynamic flight data. Future research should prioritize the integration of meteorological and real-time data to enhance the model's adaptability in variable environments. Further exploration of integrating planar and geodetic coordinate systems into the data processing flow can ensure the accuracy of flight data on the Earth's surface. Meanwhile, a track prediction system based on real-time data has been developed to verify the actual performance of the proposed model in dynamic trajectory prediction. This system enhances the decision-making ability and response speed of air traffic management.

6 Funding

The research is supported by: the National Natural Science Foundation of China (No. U2333209); Sichuan Science and Technology Program (No.23CXRC0080); Philosophy and Social Science Foundation of Sichuan (No. SCJJ23ND186); the Fundamental Research Funds for the Central Universities (No. Q2019-085, No.24CAFUC01002).

References

[1] Tingting Lu, Bo Liu, and Chunzhu Li. Aircraft trajectory prediction in terminal area based on attention Seq2Seq model. Science, Technology and Engineering, 24(9):3882-3895, 2024. https://doi.org/10.12404/j.issn.1671-1815.2304239

[2] Jiang Yuan, Zengwu Lan, and Peng Xiong. Research on abnormal behavior identification method of "black flight" UAV based on LSTM-YOLOv5. Ship Electronic Engineering, 43(10):120-125, 2023. https://doi.org/10.3969/j.issn.1672-9730.2023.10.026

[3] Zhanji Yang, Xiaolei Kang, Yuanhao Gong, and Jiansheng Wang. Aircraft trajectory prediction and aviation safety in ADS-B failure conditions based on neural network. Scientific Reports, 13(1):19677-19682, 2023. https://doi.org/10.1038/s41598-023-46914-2

[4] Nora Schimpf, Zhe Wang, Summer Li, Eric J. Knoblock, Hongxiang Li, and Rafael D. Apaza. A generalized approach to aircraft trajectory prediction via supervised deep learning. IEEE ACCESS, 11(1):116183-116195, 2023. https://doi.org/10.1109/ACCESS.2023.3325053

[5] Ping Han, Qi Zhang, Qingyan Shi, and Zezhong Zhang. 4D trajectory prediction of terminal area based on DBSCAN-GRU algorithm. Journal of Signal Processing, 39(3):439-449, 2023. https://doi.org/10.16798/j.issn.1003-0530.2023.03.007

[6] Lican Dai, Xin Liu, Haiying Zhang, Xiang Dai, and Chenggang Wang. Flight target track prediction based on Kalman filter algorithm unfolding. Systems Engineering & Electronics, 45(6):1814-1820, 2023. https://doi.org/10.12305/j.issn.1001-506X.2023.06.25

[7] Syed Ibtehaj Raza Rizvi, Jamal Habibi Markani, and René Jr. Landry. A meta-learning approach for aircraft trajectory prediction. Communications and Network, 15(2):43-64, 2023. https://doi.org/10.4236/cn.2023.152004

[8] Sharmistha Chakrabarti, and Adan E. Vela. Modeling and characterizing aircraft trajectories near airports using extracted control actions. Journal of Aerospace Information Systems, 20(2):81-101, 2023. https://doi.org/10.2514/1.I011106

[9] Teng Qu, Qi Zhang, Ziqiao Xu, Yuexiang He, Heng Li, Rongquan Guo, and Ning Jv. Analysis of aeronautical data traffic in Tianjin airspace based on ADS-B prototype reception and future traffic prediction. Science and Technology Innovation and Application, 13(25):78-83, 2023. https://doi.org/10.19981/j.CN23-1581/G3.2023.25.019

[10] Yunfeng Wang, Tao Huang, Zijiu Wang, Tongsheng Wei, and Shuang Cheng. A review of air traffic situational awareness research. Aviation Computing Technology, 54(1):130-134, 2024. https://doi.org/10.3969/j.issn.1671-654X.2024.01.028

[11] Yunwen Feng, Jiaqi Liu, Weihuang Pan, and Cheng Lu. Progress of civil aircraft operational reliability research. Journal of National University of Defense Technology, 45(4):66-93, 2023. https://doi.org/10.11887/j.cn.202304008

[12] Minghui Hu, Jinji Gao, Zhinong Jiang, Weimin Wang, Limin Zou, Tao Zhou, Yunfeng Fan, Yue Wang, Jiaxin Feng, and Chenyang Li. Progress of aero-engine vibration monitoring and fault diagnosis technology. Journal of Aeronautics, 45(44):7-35, 2024. https://doi.org/10.7527/S1000-6893.2024.30194

[13] Changfen Wang. Low-cost badminton trajectory recognition and landing point prediction based on M-YOLOv2 and kalman filter for optimizing the coordinate system transformation of sites. Informatica, 48(23):73-90, 2024. https://doi.org/10.31449/inf.v48i23.6741.

[14] Mehdi Gheisari, Hooman Hamidpour, Yang Liu, Peyman Saedi, Arif Raza, Ahmad Jalili, Hamidreza Rokhsati, and Rashid Amin. Data mining techniques for web mining: A survey. Artificial Intelligence and Applications, 1(1):3-10, 2022. https://doi.org/10.47852/bonviewAIA2202290

[15] Teng Qu, Qi Zhang, Ziqiao Xu, Yuexiang He, Heng Li, Rongquan Guo, and Ning Jv. Analysis of aeronautical data traffic in Tianjin airspace based on ADS-B prototype reception and future traffic prediction. Science and Technology Innovation and Application, 13(25):78-83, 2023. https://doi.org/10.19981/j.CN23-1581/G3.2023.25.019

[16] Roniel S. de Sousa, Azzedine Boukerche, and Antonio A.F. Loureiro. On the prediction of large-scale road-network constrained trajectories. Computer Networks, 206(1):108-115, 2022. https://doi.org/10.1016/j.comnet.2021.108337

[17] Dang Zhang, Yongxuan Zhao, Zhenjun Wang, and Yingfeng Zhang. Research on data-knowledge hybrid-driven intelligent control system architecture for discrete manufacturing systems. Journal of Mechanical Engineering, 60(6):4-10, 2024. https://doi.org/10.3901/JME.2024.06.001

[18] Kaiying Zhu. Detection of negative online public opinion among college students based on sooty tern optimization algorithm and dissemination trend data. Informatica, 48(17):153-170, 2024. https://doi.org/10.31449/inf.v48il7.6385.

[19] Yuan Meng, Baojun Shi, and Dequan Zhang. Research and improvement of Kriging-high-dimensional agent modeling method. Journal of Mechanical Engineering, 60(5):249-263, 2024. https://doi.org/10.3901/JME.2024.05.249

[20] Lan Ma, Xianran Meng, and Zhijun Wu. Data-driven 4D trajectory prediction model using attention-TCN-GRU. Aerospace, 11(4):313-319, 2024. https://doi.org/10.3390/aerospace11040313

CO-PRE: A Multi-Sensor Fusion Framework with Visual-Inertial Navigation and Curvature-Optimized Path Planning for UAVs

Yuecai Yang
Department of Intelligent Engineering, Hebei Chemical & Pharmaceutical College, Shijiazhuang 050026, China

Keywords: multi-sensor fusion, visual inertial navigation, UAV, autonomous navigation, path planning

Received: April 22, 2025

With the widespread application of unmanned systems in various fields, achieving high-precision autonomous navigation of unmanned aerial vehicles has become a research hotspot. Therefore, a navigation system that integrates vision, lidar, inertial sensor and barometer is designed in this study. Based on the factor graph optimization structure, the effective fusion of multi-source information is realized, thereby improving the consistency of positioning and mapping. Meanwhile, a probabilistic path planning method based on curvature constraints is proposed to reduce trajectory discontinuity and support dynamic feasibility requirements. The results show that the positioning error of the complete system is 0.43 m and the mapping completeness is 98.7%. Compared with the traditional A algorithm, particle swarm optimization algorithm, and probabilistic rapid expansion random tree method, the improved algorithm reduces the path length by about 17.04% and shortens the planning time by about 16.71%. In the simulation test, its average energy consumption is 3.42 J/m, the average path deviation is 0.65 m, and the number of obstacle avoidance re-planning is 2.4 times/task. The research results show that the system has good operating performance in complex environments and has certain advantages over the baseline method in terms of efficiency, stability and obstacle avoidance ability.*

Povzetek: Predstavljena je avtonomna navigacija UAV. Uvaja faktor-grafno fuzijo podatkov (kamera, LiDAR, IMU, barometer) z načrtovanjem poti CO-PRE s krivinskimi omejitvami; 0,43 m RMSE, 98,7% zemljevid, ~17% krajše/hitrejše poti.

1 Introduction

As intelligent unmanned systems continue to evolve, quadcopter unmanned aerial vehicles (UAVs) have gained widespread application in diverse scenarios including urban logistics, disaster search and rescue, and environmental monitoring, thanks to their straightforward design, agile maneuverability, and robust adaptability [1]. Particularly in enclosed settings like indoor structures, underground corridors, and post-disaster debris where satellite positioning is unreliable, the achievement of autonomous UAV navigation has emerged as a critical challenge demanding urgent attention. UAVs need to have precise environmental perception and state estimation capabilities in such scenarios to complete core tasks such as positioning, mapping, and path planning, which puts higher demands on the system's perception reliability and computational efficiency [2-3]. In recent years, with the development of sensor technology and computing platforms, various sensors such as vision, LiDAR, inertial measurement units (IMUs), etc. have been widely used in autonomous navigation systems. Multi-sensor fusion has become a key direction for improving localization robustness and map accuracy [4-5].

Zhang et al. introduced an innovative modeling and algorithmic approach that integrates the Internet of Things with edge computing to address the limitations in intelligence within the collaborative optimization of path planning and control for UAVs. They developed an integrated optimization model for path and control, along with a metaheuristic solution framework, thereby achieving a comprehensive enhancement in flight efficiency and collaborative scheduling [6]. Tong et al. introduced a novel optimization algorithm that integrates a logarithmic spiral with an adaptive step size strategy. This approach enhances population diversity through hierarchical modeling and improvements in spatial search. Simulation results demonstrated that the algorithm was capable of generating high-quality, feasible paths while satisfying time coordination constraints, thereby verifying its effectiveness and advantages [7].

Shiri was inspired by attention mechanisms in natural language processing and proposed an iterative single head attention mechanism suitable for multi-UAV path control. This method assigned individual scores through communication assistants to strengthen the modeling of multi-machine collaborative relationships in control. The results indicated that this mechanism could improve flight speed and reduce collision risk [8]. Bashir et al. proposed a path planning method based on connectivity awareness to address the risk of collision and communication interruption between UAVs and obstacles in urban environments. This method constructed a graph structured path and combined it with fleet formation strategy to ensure communication continuity and path safety [9].

In intricate environmental settings and task-oriented scenarios, the autonomous path planning of UAVs

necessitates a delicate equilibrium among flight efficiency, the quality of task coverage, and the robustness of obstacle avoidance. Researchers have proposed various path optimization methods that integrate intelligent algorithms and task models. Ko et al. proposed a UAV visual coverage method that combines velocity function and dynamic trajectory planning to meet the resolution requirements of images in different regions. This method optimized regional access by adjusting flight altitude and path sequence, combined with auxiliary travel merchant strategies [10]. Jayarajan N proposed a path planning method that combines artificial bee colony algorithm and fast exploration of random tree stars to address the low efficiency of UAV path planning in complex obstacle environments. This algorithm combined global search capability with path optimization characteristics to achieve efficient obstacle avoidance and path optimization in obstacle environments [11].

Wu et al. proposed the Q-Learning algorithm with adaptive transition speed to address the issues of low path efficiency and unstable learning in UAV search and rescue missions in unknown environments. This method involved phased design and state space optimization, combined with sensor information to initialize the learning process. The simulation results showed that the algorithm had fast learning speed, shorter path, and more stable convergence [12]. Liu et al. proposed a collaborative path planning model for UAVs and transport vehicles. Through multi-objective optimization and boundary intersection algorithm, UAV task scheduling and path optimization were achieved under dynamic objectives and variable path conditions [13]. Table 1 summarizes the main contents of the related work section.

In summary, the current UAVs still face three key problems in performing autonomous navigation tasks in complex environments: First, there is heterogeneity in multi-sensor observations, and the data fusion accuracy is not high. Second, path planning lacks continuity and real-time in dynamic environments. Third, the control system does not respond sufficiently to disturbances, affecting flight stability. Although existing studies have made some progress in visual-inertial navigation mapping or path planning, most of them have failed to achieve deep integration of perception, planning and control under a unified framework, and lack collaborative modeling of path feasibility and consistency of multi-source observations. To this end, this study proposes a visual-inertial navigation autonomous navigation system based on multi-sensor fusion, constructs an integrated closed-loop architecture of perception-planning-control, and aims to provide efficient, accurate and adaptable solutions for autonomous navigation and real-time decision-making of UAVs in unknown environments. The study aims to evaluate whether the proposed multi-sensor fusion navigation framework can achieve reliable performance in complex environments. The specific goals are to keep the average path deviation below 0.7 meters and the energy consumption below 3.5 J/m while ensuring a high success rate and minimal re-planning during the navigation mission.

The novelty of this research resides in the application of the factor graph optimization method to achieve tight coupling and fusion of multiple sensor data. Additionally, to enhance the efficiency and precision of path planning, a Curvature Optimized Probabilistic Path Rapid Expansion Algorithm (CO-PRE) based on curvature optimization is introduced. By incorporating curvature smoothing techniques, this approach improves path smoothness and computational efficiency, while optimizing path feasibility and stability through the integration of dynamic constraints. Furthermore, anti-interference adaptive control methods are employed to bolster the system's steady-state response and resistance to disturbances.

2 Methods and materials

2.1 Autonomous positioning and mapping methods for multi-source information fusion

The autonomous localization and map construction of UAVs in unknown and communication limited environments rely on deep fusion of information from multiple heterogeneous sensors [14-15]. A synchronous positioning and mapping system based on tight coupling of multi-source information was studied and designed, which integrates observation information from laser radar, IMU, optical flow sensor, and barometer. State estimation and map construction were achieved through factor graph optimization. The architecture is shown in Figure 1.

Table 1: Summary of related UAV navigation methods.

Reference	Sensor Types	Optimization Method	Key Metrics
[6] Zhang X	IoT + Edge	Metaheuristic	Flight efficiency, scheduling improvement
[7] Tong H et al.	IMU + Visual	LASSA (log-spiral based)	Path feasibility, diversity, time coordination
[8] Shiri H et al.	Communication modules	ISHA	Flight speed, collision risk reduction
[9] Bashir N et al.	Connectivity sensors	Graph-based with fleet formation	Connectivity, path safety
[10] Ko Y C et al.	Camera	Velocity + Dynamic Planning	Coverage quality, task time
[11] Jayarajan N et al.	General UAV sensors	ABC-RRT*	Path efficiency in obstacle environments
[12] Wu J et al.	Sensor-driven learning	ACSQL	Path length, convergence stability
[13] Liu X et al.	UAV + Vehicle sensors	Penalty-based boundary intersection+Pareto technique	Efficiency, resource utilization

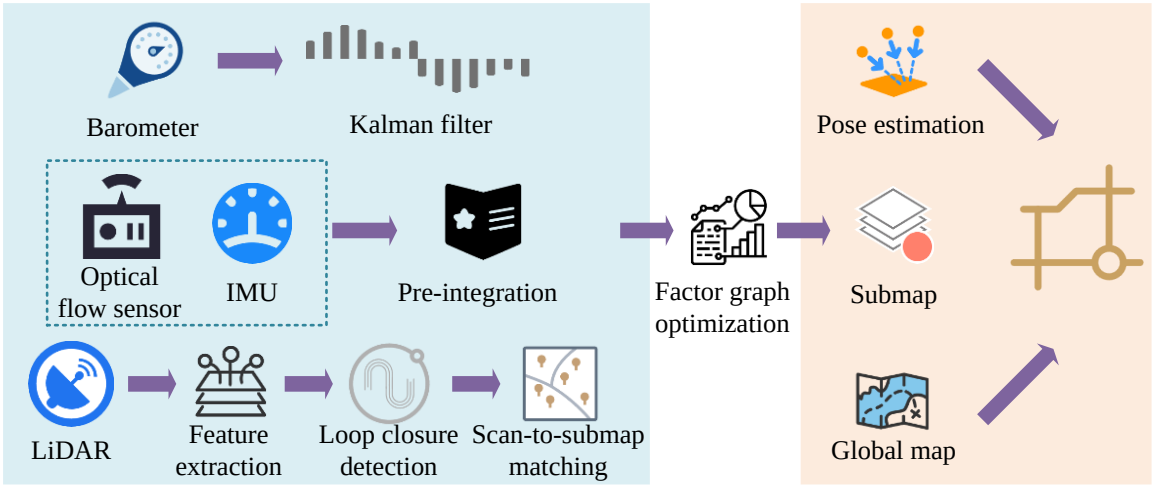


Figure 1: Synchronous localization and mapping system architecture based on tight coupling of multi-source information.

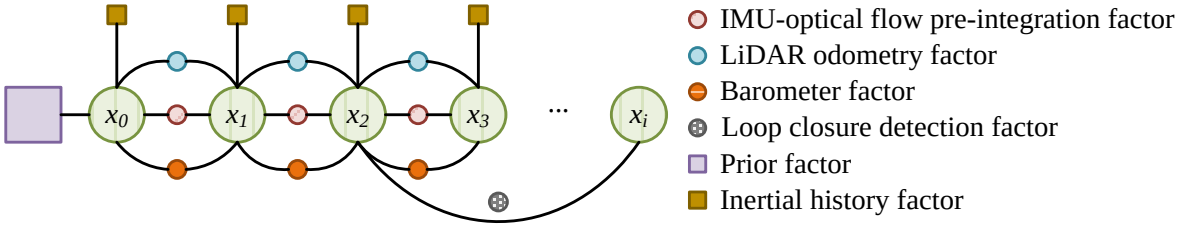


Figure 2: Schematic diagram of factor graph optimization structure under multi-sensor constraint fusion.

As illustrated in Figure 1, the system incorporates laser radar, IMU, optical flow sensors, and barometers to gather multi-source observation data. After pre-integration, filtering, and feature extraction, the data is fused via a factor graph framework to enable accurate pose estimation and dense map construction. The system's front-end handles data association and loop closure detection, while the back-end produces continuous subgraphs and global maps, thereby establishing a closed-loop process for autonomous positioning and mapping.

Initially, a systematic model of the quadcopter UAV's state is developed. The complete state vector of UAV at time t_k is represented by equation (1).

$$x_k = [p_k^{\cdot}, v_k^{\cdot}, q_k^{\cdot}, b_{a_k}^{\cdot}, b_{w_k}^{\cdot}, p_{O_B}^{\cdot}]^{\cdot} \quad (1)$$

In equation (1), p_k represents the position vector of the UAV in the world coordinate system W, and v_k represents the velocity. q_k is a posture represented by quaternions. b_{a_k} and b_{w_k} are the zero bias terms of the accelerometer and gyroscope, respectively, and p_{O_B} represents the position of the origin of the optical flow coordinate system relative to the body coordinate system B.

To achieve unified fusion of multi-source sensor information, the system constructs a state estimation framework based on factor graphs, in which each state node represents the position of the UAV at a certain moment, and the edges correspond to the observation constraint factors of different sensors. The spatial relationship between the body coordinate system B and the optical flow coordinate system O is shown in Figure 2.

Figure 2 shows the composition of the multi-sensor factor graph optimization structure, where each state node represents the pose estimation of the UAV at a specific moment, and the measurements from different sensors are connected to the state nodes in the form of factors. The data is modeled as four types of observation factors, corresponding to inertial-visual pre-integration, lidar mileage constraint, air pressure measurement, and loop detection. Different information sources are embedded in the state estimation graph in a unified form to provide a high-precision initial state for subsequent pose optimization and path planning.

To enhance the mathematical clarity and technical operability of factor graph fusion modeling, a joint residual function is introduced to characterize the constraints of multi-source observation information on state estimation. Let the UAV state vector be x, including position, velocity, attitude quaternion and inertial sensor bias, then the optimization objective function can be expressed as equation (2).

$$J(\hat{x}) = \sum_i \left\| r_{imu}^i \right\|_{\sum_{imu}^{-1}}^2 + \sum_j \left\| r_{lidar}^j \right\|_{\sum_{lidar}^{-1}}^2 + \sum_k \left\| r_{flow}^k \right\|_{\sum_{flow}^{-1}}^2 + \sum_l \left\| r_{baro}^l \right\|_{\sum_{baro}^{-1}}^2 \quad (2)$$

In equation (2), r_{imu}^i, r_{lidar}^j, r_{flow}^k, r_{baro}^l represent the residual terms of the i, j, k, l th sensor observations respectively. $\sum_{imu}$, $\sum_{lidar}$, $\sum_{flow}$, and $\sum_{baro}$ are the covariance matrices of the corresponding observation

residuals, which are used to weight the errors of different sensors. $\|r\|^2_{\sum^{-1}}$ represents the Mahalanobis distance norm, which is used to consider the noise uncertainty of each residual. To clearly present the relationship between the system state composition and sensor observations, Table 2 lists the state variables and their source sensors.

In addition, to improve the accuracy of LiDAR point cloud registration and the global consistency of the system, the study adopts an initial registration strategy based on voxel filtering and KD tree matching, and introduces an iterative closest point method for residual refinement. In terms of system closed-loop detection, a revisit detection module based on the scan context descriptor is designed. Once the loop node is identified, the pose graph optimization module is triggered to adjust the global constraints, and the graph scale is controlled through the node pruning mechanism to ensure real-time operation efficiency.

Firstly, based on the pre integration factor of optical flow and IMU, high-frequency constraints are established using the continuous acceleration and angular velocity information obtained by IMU, and combined with the relative displacement observation of optical flow sensor on the image plane. The IMU measurement model is represented by equation (3).

$$\begin{aligned} \hat{w}_t &= w_t + b_w + n_w, \\ \hat{a}_t &= R_W^B(a_t - g) + b_a + n_a \end{aligned} \tag{3}$$

In equation (3), $\hat{w}_t$ and $\hat{a}_t$ respectively represent the angular velocity and acceleration measured by IMU. b_w and b_a are zero bias terms. n_w and n_a are for noises. R_W^B is for the rotation matrix and g is for the gravitational acceleration. The optical flow method is based on the assumption of constant image grayscale, and its constraints are shown in equation (4).

$$I(\tilde{x}+\Delta\tilde{x}, \tilde{y}+\Delta\tilde{y}, t+\Delta t) = I(\tilde{x}, \tilde{y}, t) \tag{4}$$

In equation (4), $I(\tilde{x}, \tilde{y}, t)$ represents the grayscale of the image at time t and $(\tilde{x}, \tilde{y})$ is the pixel coordinate. $\Delta\tilde{x}$ and $\Delta\tilde{y}$ are the displacement amounts, and Δt is the time interval between frames.

A first-order Taylor expansion is performed on the equation, the definitions of pixel velocities $u_{pix} = \frac{d\tilde{x}}{dt}$ and $w_{pix} = \frac{d\tilde{y}}{dt}$ are introduced, and linear system constraints are further constructed and then solved using the least squares method, as shown in equation (5).

$$\begin{aligned} A\begin{bmatrix} u_{pix} \\ w_{pix} \end{bmatrix} &= -b, \\ \begin{bmatrix} u_{pix} \\ w_{pix} \end{bmatrix} &= -(A^{\cdot} A)^{-1} A^{\cdot} b \end{aligned} \tag{5}$$

In equation (5), A is a matrix composed of image gradients of multiple pixels, with each row representing the gradient $[I_{\tilde{x}}, I_{\tilde{y}}]$ of a point in the $\tilde{x}$ and $\tilde{y}$ directions. b corresponds to the time gradient I_t of each pixel. $[u_{pix}, w_{pix}]^{\cdot}$ represents the average optical flow velocity vector of pixels on the image plane. Pixel velocity is converted to ground velocity and UVA camera installation height H and scaling factor are introduced. When there is a change in attitude of UVA, it is corrected based on the gyroscope output, and the corrected velocity observation is obtained as shown in equation (6).

$$\hat{v}_o = v_o - Hw_z + n_o \tag{6}$$

In equation (6), w_z is the rotational speed around the vertical axis. n_o is the observation noise term. Subsequently, in the factor graph optimization, a joint pre integration model is constructed based on the observation information of IMU and optical flow sensor to estimate the state transition of UAV between two laser radar keyframes. The prediction of the state between time t_k and t_{k+1} is described by the integral equation of equation (7).

$$\begin{cases} p_{k+1} = p_k + v_k \Delta t_k + \\ \int_{t_k}^{t_{k+1}} \int_{t_k}^{\tau} (R_W^B(\hat{a}_t - b_a - n_a) - g) d\tau d\tau \\ v_{k+1} = \\ v_k + \int_{t_k}^{t_{k+1}} (R_W^B(\hat{a}_t - b_a - n_a) - g) d\tau \\ q_{k+1} = \\ q_k \otimes Exp(\frac{1}{2}\int_{t_k}^{t_{k+1}} (R_W^B(\hat{w}_t - b_w - n_w) d\tau) \end{cases} \tag{7}$$

Table 2: UAV state elements, sensor observations, and factor addition time.

State Variable / Observation	Sensor Source	Factor Addition Time (ms)
UAV position in world frame	LiDAR, IMU, Optical Flow, Barometer	0.4 (IMU), 1.8 (LiDAR), 0.6 (Flow), 0.2 (Baro)
UAV velocity	IMU, Optical Flow	0.4 (IMU), 0.6 (Flow)
UAV orientation $\sum_{lidar}$	IMU	0.4
Accelerometer bias	IMU	0.4
Gyroscope bias	IMU	0.4
Image-plane displacement observation	Optical Flow	0.6
Altitude estimate	Barometer	0.2
Point cloud geometric residuals	LiDAR	1.8

Note: The listed factor addition times were measured on a system with Intel Core i7-12700H and 32GB RAM using ROS middleware. Values represent average computation time per frame for inserting the corresponding factor into the graph optimization.

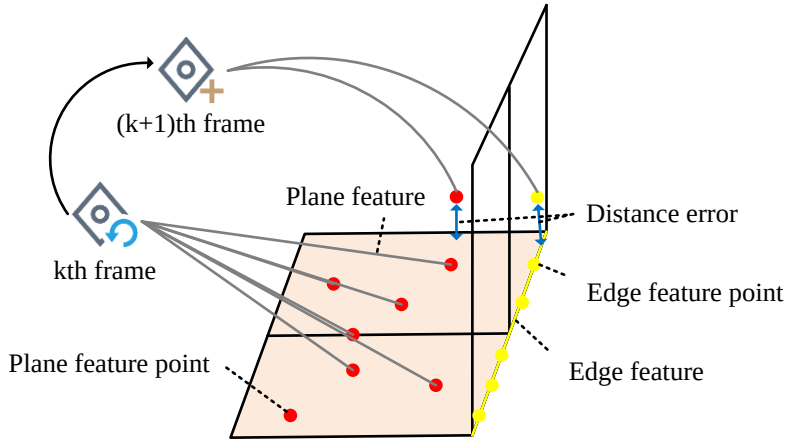


Figure 3: Schematic diagram of laser radar feature extraction and inter-frame matching.

In equation (7), Δt_k represents the time difference. $\otimes$ is quaternion multiplication. $Exp(\cdot)$ represents exponential mapping. Under the factor graph optimization framework, all observation information is ultimately fused through residual terms to construct the optimization objective function.

Furthermore, in the joint optimization of LiDAR and IMU, the feature points extracted by LiDAR are combined with IMU data to estimate the relative motion of UAV. The distance calculation in feature matching is shown in equation (8).

$$d_e = \left\| \frac{e_o \times e_2}{\|e_o\|} \times e_2 \right\|, \quad d_p = \left\| \frac{(p_3 p_2) \times (p_1 p_2)}{p_2 o} \right\| \tag{8}$$

In equation (8), e_o and e_2 are the edge features of the feature points. p_1, p_2, and p_3 are the coordinates of the point cloud, and o is the origin of the beam. The schematic diagram of LiDAR feature extraction and inter frame matching is shown in Figure 3.

As shown in Figure 3, the feature points of the current frame are compared with those of the previous frame by rotating the matrix T_k, to estimate the relative displacement between the two frames. The calculated distance is used to optimize the objective function, which is shown in equation (9).

$$e = \sum d_e + \sum d_p \tag{9}$$

In equation (9), d_e and d_e are the distances from the feature point matching point to the edge and from the point to the surface, and e is the residual. Finally, the barometer is used to estimate the altitude of the UAV, but its measurement is greatly affected by noise. Therefore, a low-pass filter is used to smooth the barometer data. The calculation of the measured value of the barometer is shown in equation (10).

$$z_k = b_0 h_k + b_1 h_{k-1} + b_2 h_{k-2} + a_1 z_{k-1} + a_2 z_{k-2} + n_h \tag{10}$$

In equation (10), h_k is the raw measurement value of the barometer, b_0, b_1, b_2, a_1, and a_2 are filter coefficients, and n_h is the noise term. As a result, the measurement value of the barometer is smoothed and corrected.

2.2 3D path planning algorithm integrating smooth and constrained optimization

The integration of multi-source information and state estimation has established a fundamental basis for the autonomous positioning and mapping of UAVs. However, to attain full closed-loop control within autonomous navigation systems, it remains imperative to tackle the pivotal challenges of path planning and dynamic obstacle evasion. Therefore, in response to the practical application requirements of quadcopter UAVs, a path planning method based on Probabilistic Rapidly-exploring Random Tree (P-RRT) algorithm is proposed. P-RRT quickly finds feasible paths from the starting point to the target point by randomly expanding the tree structure, which is suitable for avoiding dynamic environments and complex obstacles [16-17]. To further improve the smoothness and computational efficiency of the path, curvature smoothing technology will be introduced to reduce the randomness of the planning.

After generating the local path, a sliding window is first used to calculate the curvature smoothness of each point in the path. In the path planning process, a sliding window containing $|S|$ path nodes is used to smooth the current node x_i', and the curvature smoothness of the path nodes is calculated using equation (11).

$$c_i = \frac{1}{|S|} \left\| \sum_{j \in S, j \neq i} (x_j' - x_i') \right\| \tag{11}$$

In equation (11), c_i represents the average position deviation between the i th node and the nodes in its sliding window neighborhood, which can be used to approximately measure the local non-smoothness or "sharpness" of the point, and thus serve as an heuristic

indicator of path smoothness. S represents a sliding window set centered on the current path node x_i' and containing $|S|$ neighboring nodes. Based on the calculation of curvature smoothness, the sliding window method is used to optimize the path. According to the calculated curvature smoothness, if the curvature of a node is less than a predetermined threshold c_r , the node is deleted and the path is adjusted through interpolation algorithm to achieve smoother path planning.

To further smooth the generated path, a B-spline-based formulation is used as shown in equation (12), which defines the trajectory parametrically through control point interpolation

$$P(s(t)) = s_{t+1} M_{t+1} q_a \tag{12}$$

In equation (12), P represents the path planning result. q_a is the speed of the path node. M_{t+1} represents the constraint matrix at the current time. This expression serves as a structural representation of the path, providing the basis for subsequent optimization. To enforce velocity and acceleration limits during path execution, dynamic constraints are incorporated into the optimization process via the cost function defined in equation (13).

$$\begin{cases} V = Q_{m+1} - Q_m, \quad A = \dfrac{V}{\Delta t} \\ f_c = \sum (\lambda_1 (V)^2 + \lambda_2 (A)^2) \end{cases} \tag{13}$$

In equation (13), V and A respectively represent the velocity and acceleration of the path nodes. λ_1 and λ_2 are weight coefficients for adjusting speed and acceleration limits, ensuring that the path satisfies dynamic constraints simultaneously during the optimization process. This designated objective function serves to mitigate abrupt fluctuations in both speed and acceleration throughout the optimization procedure. Consequently, it facilitates the attainment of dynamic feasibility and enables precise control over the smoothness of the path.

To efficiently explore unknown environments, the Frontier exploration strategy is introduced. By dividing the environment into known and unknown areas, quadcopter UAVs continuously update the map to find the boundaries of unknown areas and use them as new exploration targets [18]. After completing the exploration of a local area, the system calculates the unknown area adjacent to the known area and uses it as the next target for path planning. In three-dimensional space, considering the limitation of computing resources, the study optimizes the exploration area range through spatial segmentation to reduce computational burden. The improved Frontier exploration framework is shown in Figure 4.

As shown in Figure 4, the framework indicates the division of known and unknown regions during the exploration process. Quadcopter UAVs continuously update the boundaries of known environments and use the boundaries of unknown areas as new exploration targets for path planning. Therefore, the process of combining the improved exploration strategy with the path planning algorithm is shown in Figure 5.

Figure 5 illustrates that during the path planning process, the algorithm initially identifies the global path objective. Subsequently, it constructs a three-dimensional map utilizing the current position of the UAV and environmental data. Following this, the algorithm optimizes the three-dimensional path planning to generate the most efficient route, ensuring a seamless navigation of the UAV towards the target zone. Ultimately, the algorithm refines the entire exploration and path planning procedure through a hybrid path planning strategy.

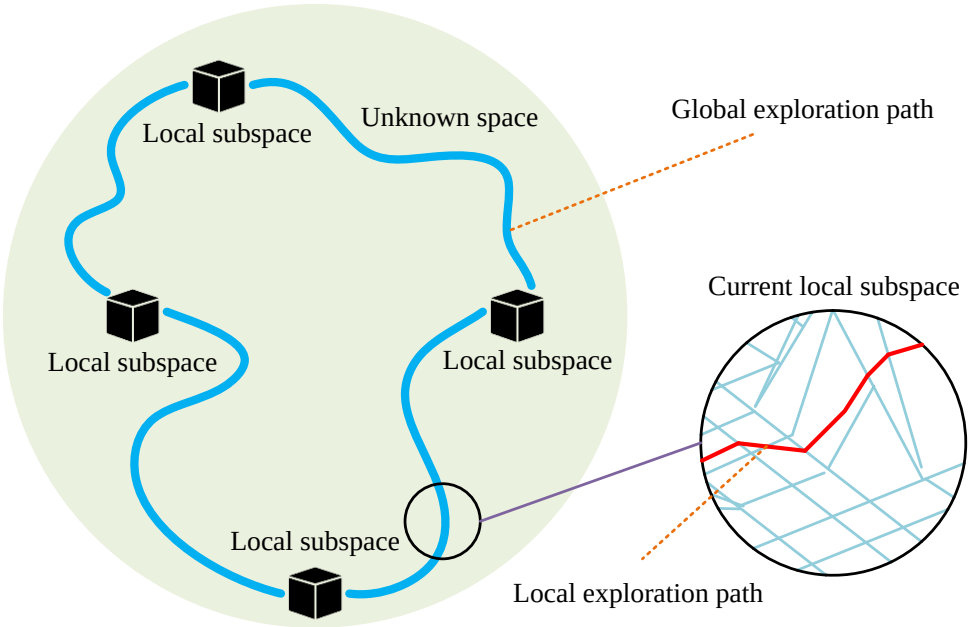


Figure 4: Improved Frontier exploration framework.

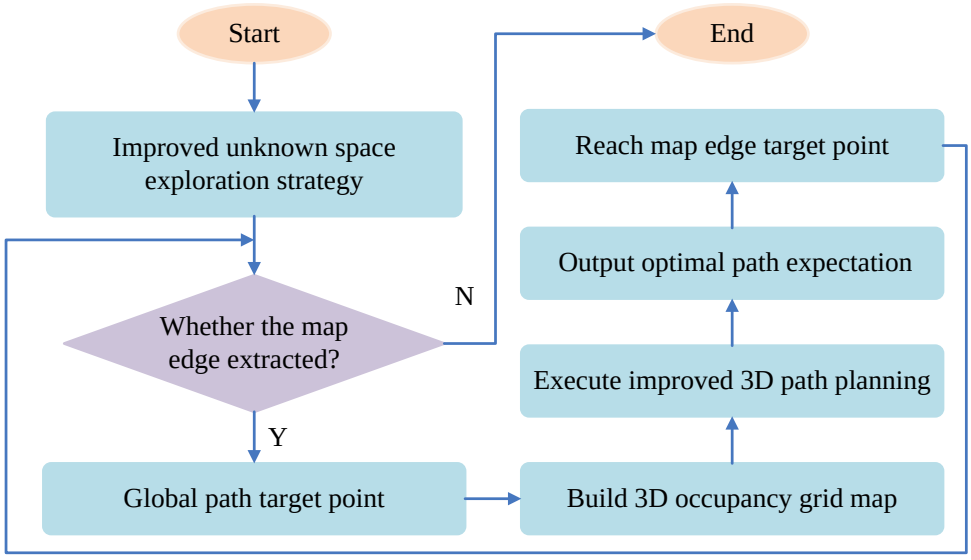


Figure 5: Process of combining improved exploration strategy with path planning algorithm.

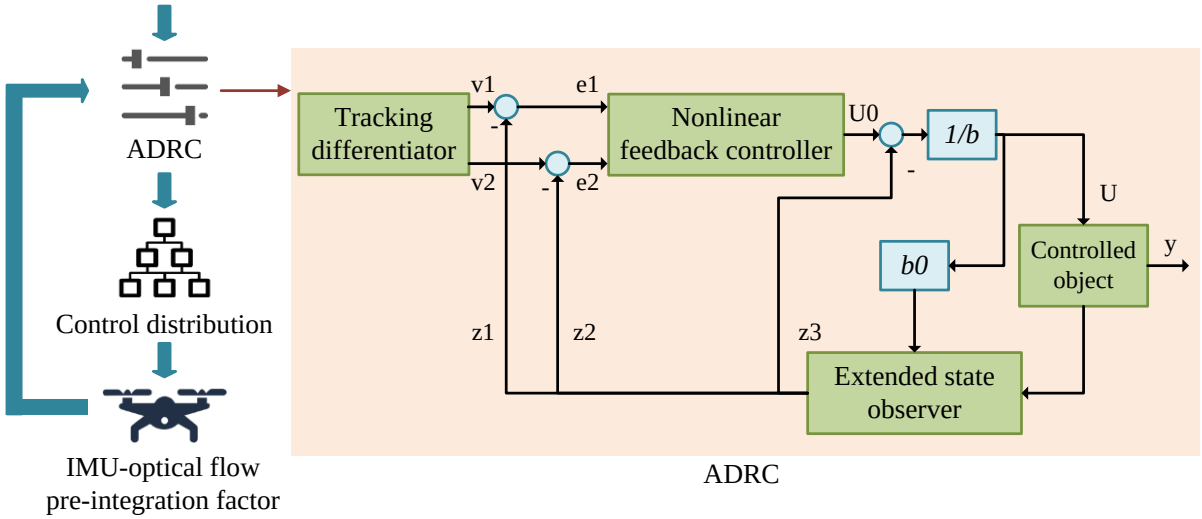


Figure 6: UAV control framework based on ADSC.

Subsequently, to achieve precise navigation and stable control, the research will adopt Active Disturbance Rejection Control (ADRC) method. ADSC, as a powerful control strategy, can effectively handle external disturbances and internal model uncertainties in the system [19-20]. Its core includes nonlinear state error feedback control law and extended state observer (ESO). Assuming that the current UAV system is a second-order uncertain system, the internal dynamics and external disturbances are uniformly modeled as generalized disturbance terms, and the extended state observer is constructed as shown in equation (14).

$$\begin{aligned} \dot{z}_1 &= z_2 - \beta_1(z_1 - y) \\ \dot{z}_2 &= z_3 - \beta_2(z_1 - y) \\ \dot{z}_3 &= -\beta_1(z_1 - y) \end{aligned} \tag{14}$$

In equation (14), z_1, z_2, and z_3 are the position, velocity, and generalized disturbance estimates, respectively. β is the ESO and controller gain parameter. The final control law is shown in equation (15).

$$u = \frac{1}{b_0}(-k_1(z_1 - y_{ref}) - k_2 z_2 - z_3) \tag{15}$$

In equation (15), b_0 is the nominal system gain, and k_1 is the ESO and controller gain parameter. This control law is capable of effectively mitigating external disturbances and compensating for modeling inaccuracies, all without the need for a precise system model. The overall control architecture is depicted in Figure 6.

As shown in Figure 6, the framework is based on the ADRC strategy and includes key modules such as input commands, control quantity allocation, nonlinear feedback, and extended state observer. By monitoring the status of UAVs in real-time and compensating for interference, ADRC can effectively suppress external disturbances and model uncertainties, thereby achieving stable flight control. This control framework can

effectively improve the control performance of quadcopter UAVs, ensuring task completion in dynamic and unknown environments.

ADRC estimates unknown disturbances online and performs feed-forward compensation through an extended state observer. It has the advantages of fast response, clear structure, and strong parameter independence. Related literature studies have shown that ADRC has stronger anti-disturbance ability and convergence stability under complex working conditions compared with traditional controllers such as PID and LQR [21-22]. In the study, the ADRC parameters were configured according to the bandwidth adjustment method. The observer bandwidth was set to 30 rad/s, the tracking differentiator bandwidth was set to 20 rad/s, and the control gain parameters were b_0 =50, k_1 =100, k_2 =300, and k_3 =500 respectively. The parameter selection was optimized through typical wind disturbance simulation and noise test to ensure that the system still has a small steady-state error and fast recovery capability under unstable excitation.

To enhance reproducibility, the following pseudo code summarizes the overall workflow of the proposed CO-PRE algorithm, see Figure 7.

Figure 7 covers key steps such as sampling expansion, curvature smoothing, B-spline fitting and dynamic constraint optimization, and intuitively demonstrates the overall logic of path generation and optimization.

Pseudocode of CO-PRE Algorithm

```
Input:
 - Occupancy map M
 - Start point s
 - Goal point g
 - Curvature threshold rc
 - Window size S
 - Dynamic constraint weights λ1, λ2

Output:
 - Smoothed path P

1: Initialize tree T ← {s}
2: while goal not reached do
3:     Sample random node q_rand in M
4:     Find nearest node q_near in T
5:     Extend from q_near to q_new
6:     if q_new is collision-free then
7:         Add q_new to T
8:     end if
9: end while

10: Extract raw path P_raw from T

11: for each node p in P_raw do
12:     Define sliding window W ← neighbors(p, S)
13:     Compute curvature κ_W using Eq.(11)
14:     if κ_W < rc then
15:         Remove node p and interpolate
16:     end if
17: end for

18: Apply B-spline smoothing to obtain P_spline
19: Optimize P_spline with dynamic constraints using Eq.(12)-(13)
20: Return final path P ← P_spline
```

Figure 7: Path planning comparison experiment results.

3 Results

3.1 Performance testing of multi-sensor fusion autonomous navigation system

The experiment was conducted under the Ubuntu 20.04 operating system, using ROS as the middleware platform, using Gazebo to build a three-dimensional flight scene, and using AirSim to simulate the dynamics of the aircraft. All modules were implemented in Python and C++ mixed programming, and the path planning module was encapsulated in the co_pre_planner function package for operation. Multi-source data such as lidar, IMU, optical flow sensor and barometer were generated in real time by the simulator and synchronously input into the system. The computing platform was configured with an Intel Core i7-12700H processor, 32GB of memory and an NVIDIA RTX 3060 graphics card. To ensure the uniformity of the evaluation, all experiments were conducted under the same system architecture and communication frequency settings. Firstly, sensitivity analysis was conducted on the key parameters in CO-PRE to determine the optimal parameters and provide a basis for subsequent experiments. The results are shown in Table 3.

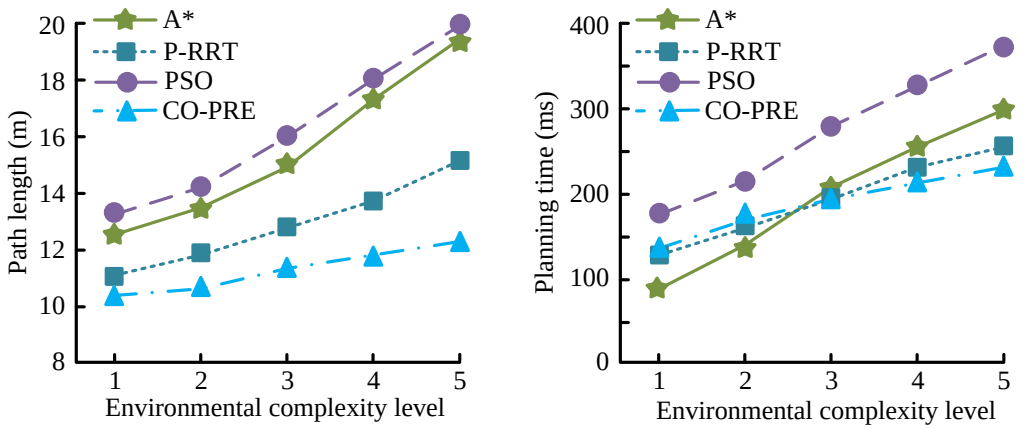
Table 3 systematically evaluated the effects of different combinations of curvature constraint thresholds c_r, sliding window sizes $|S|$, and dynamic constraint factors λ_1, λ_2 on path length, average curvature, and planning time. In the experiment, c_r increased from 0.5 to 0.95, showing that too low a threshold would cause path oscillation at turns, while too high a threshold would limit trajectory following. The increase in sliding window size $|S|$ improved curvature smoothness while also bringing certain computational overhead. In particular, Groups 1 and 2 used extremely low dynamic constraint weights, which could be regarded as approximate ablation results of weakening dynamic constraints. The corresponding paths had the largest average curvature and reduced path feasibility. Considering the path quality and computational efficiency, the final parameter combination was selected as c_r =0.9, $|S|$=7, (λ_1, λ_2)=(3,3).

Table 3: Key parameter sensitivity test results.

Serial number	c_r	$\|S\|$	λ_1, λ_2	Path length (m)	Mean curvature (rad/m)	Planning time (ms)
1	0.5	5	(1,1)	11.8	0.164	145
2	0.7	5	(1,1)	11.2	0.118	158
3	0.7	7	(1,1)	11.0	0.109	169
4	0.7	7	(2,2)	10.7	0.102	176
5	0.7	7	(3,3)	10.4	0.096	190
6	0.9	7	(3,3)	10.2	0.090	203
7	1.1	7	(3,3)	10.1	0.087	215
8	0.5	7	(2,2)	11.5	0.130	160
9	0.9	5	(1,1)	10.5	0.095	178

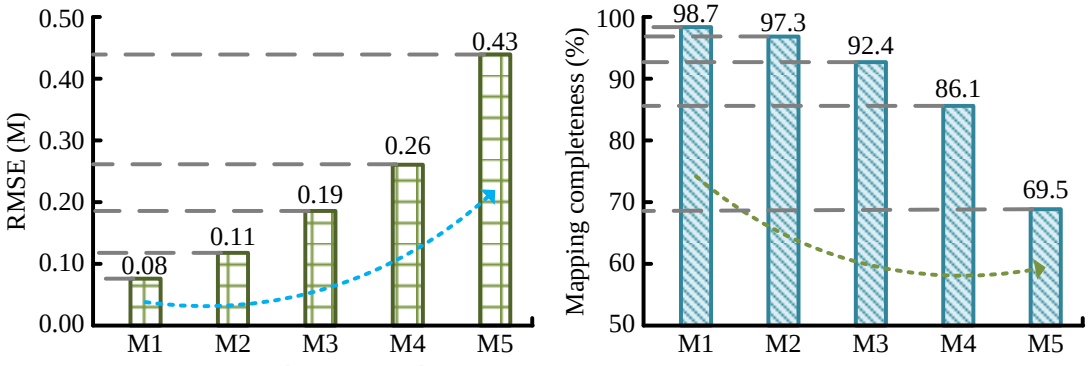
Table 4: Area Coverage Rate over Time (%±SD, n = 5).

Time (s)	Classic Frontier	Greedy Frontier	Dynamic Frontier	Proposed method
10	22.5±1.4	25.1±1.7	27.9±1.2	31.4±1.1
20	47.3±2.1	50.2±1.9	55.6±2.2	61.8±1.6
30	63.9±2.3	66.4±2.0	71.5±2.4	76.5±1.8
40	76.2±2.5	78.7±2.1	83.3±2.0	88.9±1.5
50	85.4±2.1	86.3±1.9	89.1±1.8	94.6±1.2



(a) Path length test (b) Path planning time test

Figure 8: Path planning comparison experiment results.



(a) RMSE test (b) Mapping completeness test

Figure 9: Multi-sensor combination comparison experimental results.

To verify the exploration efficiency of the improved Frontier strategy, a comparative experiment was designed with three typical methods: Classic Frontier, Greedy Frontier and Dynamic Frontier. Table 4 shows the average exploration coverage at different times. All results were the average of 5 independent experiments, with standard deviations attached.

The results in the table showed that the proposed improved Frontier strategy achieved higher area coverage in each stage, reaching 94.6% at 50 seconds. A paired t-test was further performed, and the results showed that this method had a statistically significant advantage over other strategies ($p < 0.01$).

Subsequently, the CO-PRE algorithm was tested, introducing A* algorithm, Particle Swarm Optimization (PSO) algorithm, and traditional P-RRT as comparison algorithms. The experiment divided the complexity level of the environment from 1 to 5, with level 1 representing a simple environment with sparse obstacles and smooth paths. Level 5 simulates high difficulty navigation areas with dense obstacles and complex structures. The results are shown in Figure 8.

In Figure 8 (a), when the complexity level was 5, the path lengths of A*, PSO, P-RRT, and CO-PRE were 18.18 m, 19.78 m, 14.26 m, and 11.83 m, respectively. In Figure 8 (b), the planning times of the four algorithms at the highest complexity level were 299.7 ms, 360.2 ms, 247.1 ms, and 205.8 ms, respectively. The CO-PRE mechanism reduced redundant sampling and improved node selection. By applying dynamic constraints, it produced smoother and more controllable paths, balancing quality and real-time efficiency. In contrast, A* increased its time consumption in complex environments due to graph search inflation, PSO was the slowest due to computationally intensive iterative optimization, and P-RRT had more path redundancy despite its fast expansion, which affected overall efficiency.

To verify the influence of various sensors in the multi-source fusion system on the UAV positioning and mapping performance, five sensor combination modes were designed for comparative experiments, and different sensors were removed to observe the system performance changes. The specific configuration was as follows: M1 was in full fusion mode (LiDAR, IMU, optical flow, and barometer were all enabled), M2 removed the barometer, M3 removed the optical flow and barometer, M4 removed the LiDAR, and M5 only retained the IMU single sensor. The results of root mean square error (RMSE) and mapping completeness for each combination are shown in Figure 9.

In Figure 9 (a), the RMSE of M1, M2, M3, M4, and M5 were 0.08 m, 0.11 m, 0.19 m, 0.26 m, and 0.43 m, respectively. The positioning accuracy decreased significantly as the number of sensors decreased. M1 integrated all sensors and had the strongest state observation capability. M2 only removed the barometer, with little change in positioning accuracy. After M3 lost optical flow, its ability to observe velocity decreased and the error increased significantly. Removing the LiDAR from M4 resulted in a decrease in map construction accuracy, indirectly affecting pose estimation. However, M5 relied solely on IMU, with the largest error, and simple inertial navigation was difficult to support long-term positioning. In Figure 9 (b), the completeness of mapping for the five groups was 98.7%, 97.3%, 92.4%, 86.1%, and 69.5%, respectively. M4 and M5 showed the most significant decrease, indicating that LiDAR contributed the most to map accuracy, while IMU could not provide spatial structure information. This verified the key role of multi-sensor collaboration in stable mapping and accurate navigation.

3.2 Testing of UAV autonomous navigation system based on simulation environment

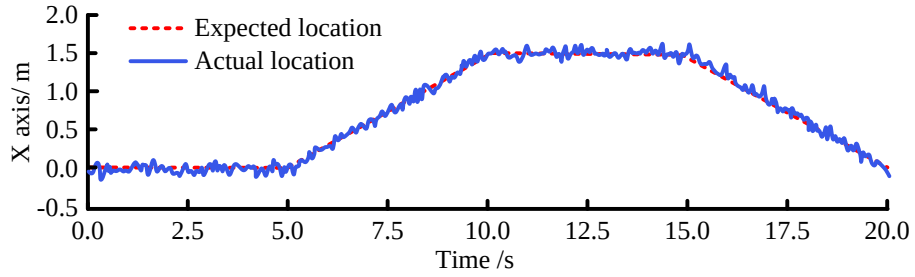
Furthermore, the study designed an autonomous navigation experimental scenario. Firstly, the experiment set up the UAV to take off from its original position to an altitude of 1.6 meters, maintain a brief hover, and then translate about 1.5 meters along the x-axis direction before returning to the origin and landing. The flight process was closed-loop controlled by the controller based on the output of the navigation system. The system recorded the real-time position information of the UAV in the x, y, and z axes, and compared the error with the preset trajectory to test the tracking ability of the system in three-

dimensional space. The experimental results are shown in Figure 10.

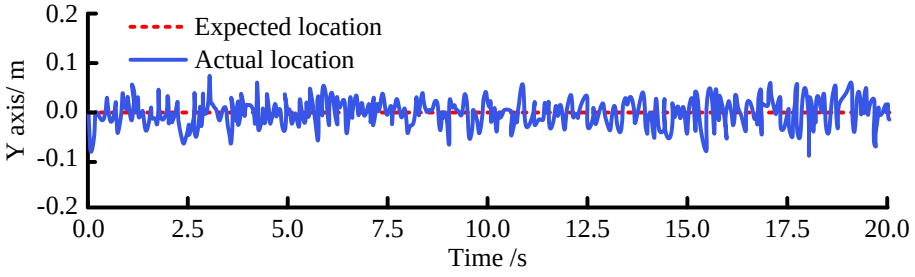
In Figures 10, during the takeoff phase, the system experienced short-term oscillations in response to sudden changes, then quickly stabilized and entered a steady tracking phase. The x-axis and y-axis converged within 2 seconds, while the z-axis remained within a fluctuation range of±0.1 meters. The system exhibited good response speed and control accuracy during autonomous flight, verifying the adaptability and robustness of the state estimation and control system for trajectory tracking under the multi-sensor fusion architecture.

To further verify the path planning and execution capability of the proposed autonomous navigation system in complex three-dimensional space, several three-dimensional obstacles were randomly arranged in the experimental scene. The UAV needed to start from the starting point and use various path planning algorithms to generate trajectories to bypass the obstacles and reach the target point. All obstacle maps in the experiment were generated using the seed randomization method, and the environmental parameters were fixed. All methods used the same random seed to ensure the consistency of obstacle layout and scene configuration. The comparison algorithms were A*, Rapid exploring Random Tree Star Algorithm (RRT*), and Path Optimization Algorithm based on Bezier Curve. The results are shown in Figure 11.

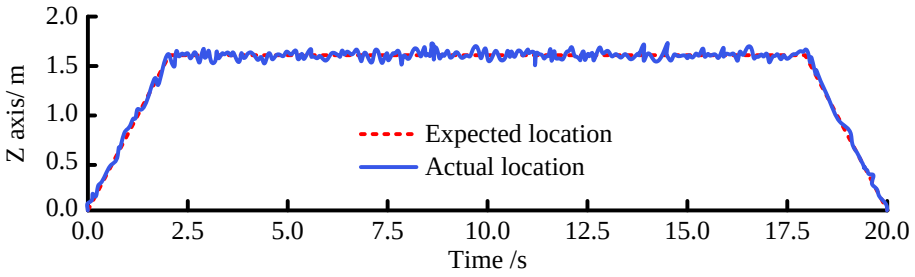
In Figure 11 (a), Algorithm A* was based on regular grid search, and the path presented multiple polylines. Although it could ensure accessibility, it did not have a trajectory smoothing mechanism, resulting in multiple turns and discontinuous paths, which led to an overall increase in the length of the route. In Figure 11 (b), the RRT algorithm used a sampling extension tree to construct the path, which had strong global search ability. However, its random sampling and reconnection process was prone to local oscillations, resulting in severe path jitter and affecting navigation stability. In Figure 11 (c), the Bezier curve relied on control points to generate an overall smooth curve with excellent curvature continuity. However, due to the lack of explicit consideration of obstacles, the path approached or even partially crossed obstacles, lacking environmental adaptability and resulting in infeasible areas. The CO-PRE algorithm in Figure 11 (d) introduced a sliding window mechanism based on curvature optimization, dynamically adjusted path nodes, and combined probability sampling and spatial feasibility discrimination to effectively balance path smoothness and obstacle avoidance ability. The final generated path was smooth, continuous, and risk averse, demonstrating good adaptability and practical application potential for 3D navigation.



(a) X-axis position change waveform



(b) Y-axis position change waveform



(c) Z-axis position change waveform

Figure 10: Three-axis trajectory tracking comparison chart.

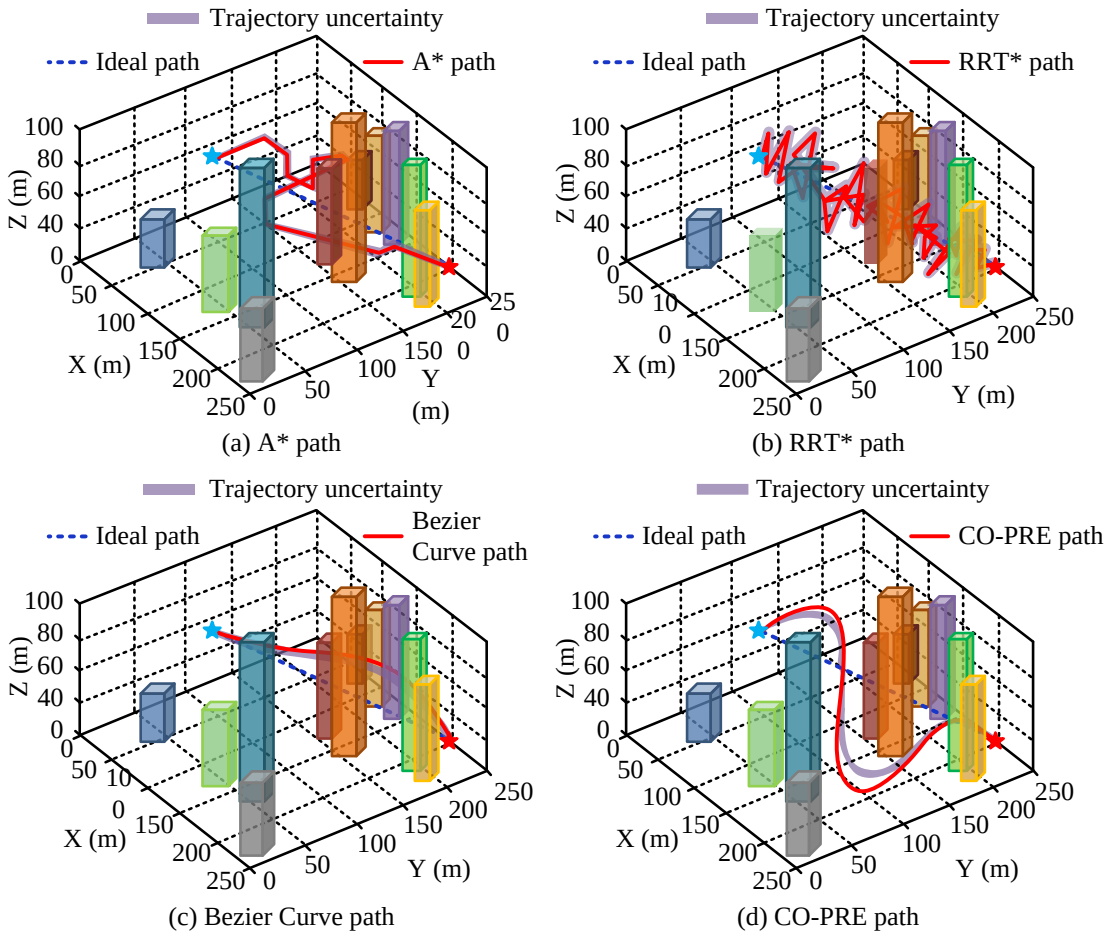


Figure 11: Three-dimensional navigation experiment based on urban building obstacle scene.

Table 5: Comprehensive test results (Mean±SD, 95% CI, n = 10).

Performance Indicator	A*	RRT*	Bezier Curve	CO-PRE (Proposed)	p-value (vs. CO-PRE)
Success rate (%)	81.6±1.5 (CI: ±0.9)	87.4±1.3 (CI: ±0.8)	73.8±2.1 (CI: ±1.2)	96.2±1.2 (CI: ±0.7)	< 0.01
Average path deviation (m)	1.52±0.07 (CI: ±0.04)	1.13±0.05 (CI: ±0.03)	0.96±0.04 (CI: ±0.02)	0.65±0.03 (CI: ±0.02)	< 0.01
Mapping delay (ms)	219±8 (CI: ±4.8)	192±6 (CI: ±3.6)	240±10 (CI: ±6.0)	158±5 (CI: ±3.0)	< 0.01
Average energy consumption (J/m)	4.38±0.12 (CI: ±0.07)	4.01±0.10 (CI: ±0.06)	3.77±0.09 (CI: ±0.05)	3.42±0.08 (CI: ±0.05)	< 0.01
CPU load (%)	42.1±2.5 (CI: ±1.5)	48.5±2.3 (CI: ±1.4)	30.4±1.8 (CI: ±1.1)	51.9±2.4 (CI: ±1.4)	0.06
Replanning count (times/task)	6.1±0.3 (CI: ±0.2)	4.8±0.3 (CI: ±0.2)	7.3±0.4 (CI: ±0.2)	2.4±0.2 (CI: ±0.1)	< 0.01
Computational complexity (ms)	184±6 (CI: ±3.6)	142±5 (CI: ±3.0)	169±7 (CI: ±4.2)	127±4 (CI: ±2.4)	< 0.01

Finally, multiple synthetic environments were built using Gazebo and AirSim, and evaluated through repeated testing under varied start-end configurations and dynamic conditions to assess system robustness and generalization. The tested scenarios included: (1) a structured urban environment with dense buildings and grid-like corridors; (2) canyon-like terrain with narrow paths and elevation changes; and (3) indoor scenes simulating limited field of view and irregular obstacle layouts. Each environment incorporates factors such as dynamic obstacle updates, partial map observability, and sensor bandwidth constraints to emulate real-world mission challenges. Evaluation metrics were averaged across multiple runs under consistent conditions, the results are shown in Table 5.

The average path deviation was used to measure the average spatial deviation between the flight trajectory and the reference path. It was calculated by sampling at equal time intervals during the mission and counting the mean Euclidean distance between the actual trajectory point and the reference trajectory point. Its expression was

$\frac{1}{N}\sum_{i=1}^{N}\left|p_i^{act}-p_i^{ref}\right|_2$, p_i^{act} , p_i^{ref} represented the actual trajectory point and the reference trajectory point at that moment, respectively, and N was the total number of samples. Mapping delay represented the time interval from the reception of sensor data to the completion of the corresponding occupancy grid map update, which was calculated using the topic timestamp in the ROS system. The CPU load indicator was sampled and recorded at a frequency of 1 Hz through the system performance monitoring tool. It was continuously monitored and its average value was calculated during the entire navigation mission, reflecting the system resource consumption level under each sensor configuration.

In Table 5, the success rate of the proposed CO-PRE algorithm reached 96.2% (±1.2, CI: ±0.7), indicating reliable task completion. The average path deviation was maintained at 0.65±0.03 meters, significantly lower than those of other methods ($p < 0.01$). Additionally, the system demonstrated reduced mapping delay and computational complexity, with energy consumption controlled at 3.42±0.08 J/m. Although the CPU load was slightly higher than some baseline methods, the overall performance gain was statistically supported. All metrics were averaged over 10 trials, and 95% confidence intervals were reported. The p-values derived from paired t-tests confirmed that CO-PRE's improvements in path accuracy, efficiency, and obstacle re-planning were statistically significant.

4 Discussion

To comprehensively evaluate the performance of the proposed CO-PRE algorithm, the study compared it with the classic RRT and Bezier methods, focusing on indicators such as path smoothness, obstacle avoidance adaptability and system computational efficiency. The results showed that in complex environments, the average path deviation of CO-PRE was controlled at 0.65m, which was better than RRT's 1.13m and Bezier's 0.96m, with a higher success rate and fewer replanning times. In the test, the system showed good response speed and control accuracy during autonomous flight. The x-axis and y-axis basically converged within 2 seconds, and the z-axis maintained a fluctuation range of±0.1 meters.

Jayarajan et al. proposed the ABC-RRT* hybrid algorithm in the literature [11], but due to the randomness of the sampling tree structure, its path smoothness and local adaptability were still insufficient. In contrast, CO-PRE introduced a sliding window curvature optimization and feasibility judgment mechanism to generate more practical paths in high-density scenarios such as narrow urban passages. In the performance test, when the complexity level was 5, the path lengths of A*, PSO, P-RRT, and CO-PRE were 18.18 m, 19.78 m, 14.26 m, and 11.83 m, respectively, and the planning times were 299.7 ms, 360.2 ms, 247.1 ms, and 205.8 ms, respectively. The LASSA algorithm proposed by Tong et al. in the literature [7] had advantages for multi-machine systems, but did not directly optimize the local smoothness and dynamic feasibility of a single path. CO-PRE focused more on the execution of a single trajectory in complex three-dimensional scenes, and could better cope with the actual navigation needs in GPS failure or densely built areas.

In summary, the system proposed in the study not only ensured the geometric continuity of the path, but also took into account the dynamic feasibility, providing an effective path for building a navigation system with high robustness and high executability. The research aimed to improve the interpretability and controllability of the UAV navigation system and provide a technical idea for realizing real-time planning under dynamic constraints.

5 Conclusion

To improve the smoothness and obstacle avoidance capability of UAV path planning in complex environments, a CO-PRE autonomous navigation algorithm integrating factor graph optimization and curvature-constrained path smoothing strategy was proposed. Experimental results showed that the algorithm had higher navigation stability and control accuracy. In the application experiment, the system successfully completed the closed-loop flight mission of take-off-hover-translation-return, verifying the feasibility of its structural design and control strategy. In summary, CO-PRE not only showed excellent performance in simulation scenarios, but also had the potential for actual deployment.

However, the research still has certain limitations. The system was sensitive to sensor calibration errors and lacked explicit fault tolerance, making it dependent on the integrity of multi-sensor inputs. In GPS-denied environments or under wind disturbances, control accuracy may degrade, affecting tracking stability. Feedback latency between perception and control may limit responsiveness, and the current framework relies on local mapping without integration of SLAM or semantic maps, reducing adaptability in long-term or multi-scenario tasks. While planning latency remained under 100 ms, CO-PRE incurred a higher CPU load than lightweight baselines such as Bezier, highlighting the need for further optimization. Future work will address redundancy-aware perception, real-time control robustness, and hardware-level deployment to improve system reliability in dynamic real-world environments.

References

[1] Veselov Gennady, Sharma Ashutosh, Tselykh Alexey, and Huang Ruihang. Applications of artificial intelligence in evolution of smart cities and societies. Informatica, 45(5):603, 2021. https://doi.org/10.31449/inf.v45i5.3600

[2] Soheila Ghambari, Mahmoud Golabi, Laetitia Jourdan, Julien Lepagnot, and Lhassane Idoumghar. UAV path planning techniques: A survey. RAIRO-Operations Research, 58(4):2951-2989, 2024. https://doi.org/10.1051/ro/2024073

[3] Lisha Dong. Improved A* algorithm for intelligent navigation path planning. Informatica, 48(10):181-194, 2024. https://doi.org/10.31449/inf.v48i10.5693

[4] Daqian Liu, Bowen Fei, Weidong Bao, Xiaomin Zhu, and Xiaoqing Li. DAWN: Dynamic task planning of multi-UAV with two-layer optimization mechanism in uncertain environments. IEEE Internet of Things Journal, 11(23):37813-37830, 2024. https://doi.org/10.1109/JIOT.2024.3440017

[5] Chaohua Yan. Application of a graphical image pre-retrieval method based on compatible rough sets to the self-localization method of mobile robots. Informatica, 48(11):85-96, 2024. https://doi.org/10.31449/inf.v48i11.5508

[6] Xiuzhu Zhang. Path planning and control of intelligent delivery UAV based on internet of things and edge computing. International Journal of Advanced Computer Science & Applications, 15(3):1068-1076, 2024. https://doi.org/10.14569/ijacsa.2024.01503107

[7] Tong Han, Andi Tang, Huan Zhou, Dengwu Xu, and Lei Xie. Multiple UAV cooperative path planning based on LASSA method. Journa of Systems Engineering and Electronics, 44(1):233-241, 2022. https://doi.org/10.12305/j.issn.1001-506X.2022.01.29

[8] Hamid Shiri, Hyowoon Seo, Jihong Park, and Mehdi Bennis. Attention-based communication and control for multi-UAV path planning. IEEE Wireless Communications Letters, 11(7):1409-1413, 2022. https://doi.org/10.1109/LWC.2022.3171602

[9] Nouman Bashir, Saadi Boudjit, and Gabriel Dauphin. A connectivity aware path planning for a fleet of UAVs in an urban environment. IEEE Transactions on Intelligent Transportation Systems, 24(10):10537-10552, 2023. https://doi.org/10.1109/TITS.2023.3280995

[10] Yun-Chun Ko, and Rung-Hung. UAV velocity function design and trajectory planning for heterogeneous visual coverage of terrestrial regions. IEEE Transactions on Mobile Computing, 22(10):6205-6222, 2022. https://doi.org/10.1109/TMC.2022.3182975

[11] Niresh Jayarajan, Tamilselvan Ganesan, and Archana Naganathan. Optimal trajectories for UAV three-dimensional path planning using a hybrid ABC-RRT* algorithm. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 239(3):930-943, 2025. https://doi.org/10.1177/09544062241288633

[12] Jiehong Wu, Ya'nan Sun, Danyang Li, Junling Shi, Xianwei Li, Lijun Gao, Lei Yu, Guangjie Han, and Jinsong Wu. An adaptive conversion speed Q-learning algorithm for search and rescue UAV path planning in unknown environments. IEEE Transactions on Vehicular Technology, 72(12):15391-15404, 2023. https://doi.org/10.1109/TVT.2023.3297837

[13] Xiaofeng Liu, Zhong-Ren Peng, Li-Ye Zhang, and Qiang Chen. Real-time and coordinated UAV path planning for road traffic surveillance: A penalty-based boundary intersection approach. International Journal of Control, Automation and Systems, 20(8):2655-2668, 2022. https://doi.org/10.1007/s12555-020-0565-8

[14] Linzhen Shi, Wensong Jiang, Zai Luo, Li Yang, and Yan Wang. Enhancing adaptability: Hierarchical frontier-based path planning for navigation in challenging environments. IEEE Robotics and Automation Letters, 9(10):8611-8616, 2024. https://doi.org/10.1109/LRA.2024.3448137

[15] Rohit Dujari, Brijesh Patel, and Bhumeshwar K. Patle. Fast and efficient drone path planning using riemannian manifold in indoor environment. Automation, 5(3):450-466, 2024. https://doi.org/10.3390/automation5030026

[16] Gilang Nugraha Putu Pratama, Oktaf Agni Dhewa, Ardy Seto Priambodo, Faris Yusuf Baktiar, Rizky Hidayat Prasetyo, Mentari Putri Jati, and Indra Hidayatulloh. Benchmark analysis of sampling methods for RRT path planning. Control Systems and Optimization Letters, 2(2):217-233, 2024. https://doi.org/10.59247/csol.v2i2.132

[17] S. Julius Fusic, and R. Sitharthan. Improved RRT* algorithm-based path planning for unmanned aerial vehicle in a 3D metropolitan environment. Unmanned Systems, 12(5):859-875, 2024. https://doi.org/10.1142/S2301385024500225

[18] Ramírez-Neria M, Morales-Valdez J, Yu W. Active vibration control of building structure using active disturbance rejection control. Journal of Vibration and Control, 2022, 28(17-18): 2171-2186. doi: https://doi.org/10.1177/10775463211009377

[19] Lianghong Zhu, Guoqiang Zhang, Runze Jing, Guangdong Bi, Runhua Xiang, Gaolin Wang, and Dianguo Xu. Nonlinear active disturbance rejection control strategy for permanent magnet synchronous motor drives. IEEE Transactions on Energy Conversion, 37(3):2119-2129, 2022. https://doi.org/10.1109/TEC.2022.3150796

[20] Suiyuan Shen, Jinfa Xu, Pei Chen, and Qingyuan Xia. An intelligence attitude controller based on active disturbance rejection control technology for an unmanned helicopter. IEEE Transactions on Vehicular Technology, 72(3):2936-2946, 2022. https://doi.org/10.1109/TVT.2022.3217300

[21] Jingqing Han. From PID to active disturbance rejection control. IEEE transactions on Industrial Electronics, 56(3):900-906, 2009. https://doi.org/10.1109/TIE.2008.2011621

[22] Zhenlong Wu, Zhiqiang Gao, Donghai Li, Yangquan Chen, and Yanhong Liu. On transitioning from PID to ADRC in thermal power plants. Control Theory and Technology, 2021, 19:3-18, 2021. https://doi.org/10.1007/s11768-021-00032-4

ADSRP: A DBSCAN-SNN Framework for AIS-Based Ship Route Planning Using Spatiotemporal Feature Fusion

Yan Deng[1], Yungye Ren[2,*]
[1]Yangtze River Administration of Navigational Affairs Archives Center, Wuhan 430014, China
[2]School of Nautical Technology, Jiangsu Shipping College, Nantong 226010, China
E-mail: dy027@163.com, R2058662858@126.com
*Corresponding author

Keywords: Route planning, AIS, DBSCAN, SNN, ship

Received: April 22, 2025

Addressing the issues of noise interference, inadequate modeling of nonlinear characteristics, and computational inefficiency in ship trajectory planning, this study introduces a multi-stage joint optimization model. The model is built upon Automatic Identification System (AIS) data cleaning, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and a Siamese Neural Network (SNN). The operation of the AIS Data and DBSCAN-Based Ship Route Planning Model (ADSRP) involves the following steps: First, AIS data is cleansed by employing a dynamic neighborhood radius and linear interpolation with a sliding window. Key steering points are then extracted by integrating the Douglas-Peucker (DP) algorithm, resulting in an 85.4% reduction in trajectory redundancy. Subsequently, DBSCAN is utilized for density-based clustering of trajectory endpoints, achieving a 93.6% filtering accuracy for noise points. Finally, a symmetric-weight SNN architecture (comprising a 4-layer Transformer encoder and multi-head attention) is designed to filter high-density routes based on cosine similarity.Experimental results demonstrate that, in comparison to the traditional genetic algorithm-based Whole Process Route Planning (WPRP), ADSRP enhances trajectory fitting in the simulation environment by 21% (with an average cosine similarity of 0.86 for ADSRP and 0.71 for WPRP) and shortens the planning time by 67.8% (8.11s for ADSRP and 25.24s for WPRP). In real-world port scenarios, ADSRP reduces voyage deviation by 36.8% (0.98nmi for ADSRP and 1.55nmi for WPRP), cuts fuel consumption by 20.8% (362.58L for ADSRP and 457.89L for WPRP), and optimizes memory usage to 27.5% (compared to the benchmark's 42.5%). Parameter sensitivity analysis verifies the significant impact of key parameters on clustering fragmentation and port identification accuracy (F1-score difference of 22%). The model is co-optimized by data-driven clustering and deep metric learning, providing a high-accuracy, low-energy solution for dynamic path planning in complex sea areas and supporting edge device deployment.

Povzetek: Pomorsko načrtovanje poti je narejeno s tristančnim okvirjem (čiščenje AIS + DBSCAN + SNN/Transformer). V simulacijah in praksi izboljša poti, čas, odklon, porabo goriva in potrošnjo pomnilnika.

1 Introduction

Ship route planning, as a core component of the maritime transportation system, directly affects shipping efficiency and safety. Ship route planning methods are divided into mathematical model-based planning methods and data-driven planning methods. The former solves the shortest path or minimum cost in a static environment through graph theory, dynamic programming, and other methods [1]. The latter processes dynamic multi-objective optimization and complex constraint problems through evolutionary algorithms, reinforcement learning, and other methods [2]. However, against the backdrop of intensified climate change and surging competition in shipping, traditional methods are unable to meet the comprehensive needs of modern shipping. For example, sequential decision-making methods face the "curse of dimensionality", and when the number of ship state variables increases, the computational complexity grows exponentially, making it difficult to meet real-time requirements [3].The linear programming method needs to simplify multi-objective conflicts into a single objective function, resulting in distorted trade-offs between economy and safety in practical scenarios [4].Genetic algorithms are prone to premature convergence problems, and the design of crossover/mutation operators relies on prior knowledge, resulting in insufficient adaptability under complex constraints [5].

In summary, traditional ship route planning methods mainly rely on crew experience for manual design, which has the problems of low efficiency and strong subjectivity. The Automatic Identification System (AIS) data can accurately record ship status parameters and support trajectory mining and pattern analysis [6]. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm can automatically identify high-density core points, boundary points, and noise points

through density accessibility [7]. However, AIS data has missing trajectories and noise, and there are more abnormal drift points and duplicate records at high sampling intervals. Moreover, the DBSCAN algorithm is sensitive to parameters and difficult to adaptively handle trajectory clusters with significant density differences. Therefore, the research focuses on the advantages and disadvantages of AIS data and DBSCAN, and makes improvements to ultimately construct an AIS Data and DBSCAN-based Ship Route Planning Model (ADSRP).To assess the viability of integrating the aforementioned techniques, the study poses the following question: Can the amalgamation of density clustering and metric learning enable achieving a trajectory deviation of less than 1 nmi while concurrently cutting computational requirements by over 60%? Subsequently, the study undertakes the task of validating this proposition. The research aims to solve the problem of delayed dynamic environmental response in traditional ship route planning by utilizing the multidimensional spatiotemporal characteristics of AIS shipping big data and the density adaptive clustering ability of DBSCAN algorithm. The innovation of the research lies in deeply coupling the DBSCAN algorithm with AIS data characteristics, constructing a dynamic parameter framework, and achieving multi-scale density clustering of ship trajectory endpoints. Meanwhile, a trajectory similarity weighting mechanism is designed to enhance the robustness of trajectory clustering through AIS data-driven DBSCAN noise filtering and cluster merging strategies.

The research is divided into four sections. The first section introduces the current research on the logic and algorithms of ship route planning worldwide, and clarifies the shortcomings of existing research. The second section starts from AIS data processing, Douglas-Peucker (DP) algorithm, etc., establishes accurate and efficient trajectory feature extraction methods, and combines DBSCAN and Siamese Neural Networks (SNN) to construct trajectory planning methods, establishing an ADSRP model. The third section provides numerical examples and practical application analysis of the proposed model to verify its reliability. The fourth section provides a comprehensive summary and analysis of the article.

2 Related works

In the process of intelligentization and greening of the global shipping industry, ship route planning faces dynamic environmental challenges. Traditional methods rely on static data, which makes it difficult to cope with real-time risks, prompting researchers to explore new methods. In response to the harsh environment and insufficient sea ice information in the Arctic ice zone, Wu et al. proposed an online interactive system based on big earth data, combined with reinforcement learning and synthetic aperture radar data, to improve route safety and accuracy through dynamic path calculation and real-time ice water classification [8].Zhao et al. improved the artificial fish swarm algorithm to address the low path efficiency, local optimization, and insufficient smoothness of autonomous surface vessels. They introduced directional operators to enhance search efficiency, designed probability weight factors to reduce the risk of local optima, and combined adaptive convergence strategies and smoothing mechanisms to enhance practicality in complex sea conditions [9]. Zhao et al. constructed a multi-objective meteorological routing framework to address issues such as high fuel consumption and high risks in ships. They used particle swarm optimization and non-dominated sorting techniques to optimize fuel consumption, risk, and route coordination models through mutation operations and elite selection [10].In response to the lack of dynamic obstacle avoidance and rule coordination in unmanned ships, Li et al. proposed a multi-layer adaptive search tree method that integrated the speed obstacle method to predict collision domains, and combined rolling window modeling and path optimization strategies to improve dynamic response capabilities [11].

In addition, in response to the problem of weak multi-objective collaboration and trajectory adaptability, Hongjie et al. designed a phased trajectory optimization framework, which usedDBSCAN clustering to extract turning points and generate a global route network, balancing safety and economy [12].In response to the lack of AIS data mining leading to planning disconnection, Gao et al. constructed a framework for ship navigable routes, using manifold distance to generate high-density search networks, and driving topology generation with real trajectories to reduce path deviation rates [13].Kim et al. proposed a nested trajectory reconstruction framework to address the low accuracy of AIS spatiotemporal bias and destination prediction. The framework extracted spatiotemporal features through multi-channel vectorization and attention mechanisms, and combined gradient dropout techniques to improve long-term prediction accuracy [14]. In response to the lack of neutron trajectories in traditional trajectory partitioning, Cui et al. developed a multi-attribute DBSCAN method that integrated parameters such as speed and heading to achieve adaptive partitioning and enhance anomaly detection capabilities in complex scenarios [15]. A comprehensive comparison of these methods is shown in Table 1.

Table 1: Recent advances in ship route planning

Methods	Model type	Datasets	Clustering algorithm	Key features	Results	References
Online Interactive Route Planning System (RouteView)	Reinforcement learning integrated with deep learning	Sea ice and meteorological data + synthetic aperture radar data	-	Reinforcement learning based dynamic planning of arctic routes +synthetic aperture radar with deep learning for real-time sea ice classification	Enhancing northeast waterway safety and improving information extraction methods.	[8]
Improved Artificial Fish Swarm Algorithm (IAFSA)	Population intelligence optimization algorithm	Simulation of navigation data in the south china sea	-	Directional operators, probabilistic weight factors, adaptive convergence and path smoothing.	Algorithm efficiency and path quality outperform existing methods, and model ship tests verify applicability.	[9]
Multi-Objective Particle Swarm Optimization Routing System (MOPSO-RS)	Multi-objective optimization algorithm	Container ship simulation data	-	Mutation manipulation and elite selection balancing convergence speed and diversity, pareto optimal solution recommendation criterion.	Realization of low-risk, low-fuel-consumption, short-time route planning.	[10]
Whole Process Route Planning (WPRP)	Trajectory clustering and network optimization	Vessel trajectory data	Hierarchical dbscan（hdbscan）	Trajectory smoothing (polynomial approximation with exponential kernel) → compression (douglas-peucker) → clustering (hdbscan) → road network construction.	The trajectory similarity is improved by 36.53%, the compression rate is >92%, the clustering profile coefficient is 0.9032, and the road network contains 299 key nodes.	[12]
High Density Searching Framework Extraction (HDS-FE)	Ais big data-driven network construction	Global ais data	-	Ship navigable route framework based on ais big data to support actual navigation distance calculation and route planning.	Solve the problem of disconnection between traditional route planning and real sailing habits, and enhance the feasibility of route application.	[13]

As can be seen from Table 1, although existing research has made progress, it mostly relies on static data and manual experience, lacks in-depth exploration of historical navigation patterns, and has low clustering efficiency and insufficient feature extraction for massive AIS data, making it difficult to meet the personalized needs of multiple ship types. Therefore, the study aims to optimize the efficiency of trajectory feature extraction and clustering through AIS data and DBSCAN, improve the robustness and dynamic adaptability of the model, and construct an ADSRP ship route planning model to provide efficient and accurate solutions for practical scenarios.

3 Methods and material

This section is divided into two parts. The first part provides a detailed explanation of AIS data categories, DP algorithms, and other methods, preprocesses AIS data, and extracts trajectory features. The second part combines DBSCAN and SNN to construct a clustering method for trajectory endpoints (starting and ending points), proposes an ADSRP model, improves the accuracy of trajectory planning and noise robustness of the model, and enhances its dynamic environmental adaptability.

3.1 AIS data preprocessing and trajectory feature extraction

Ship trajectory feature extraction is the core foundation of intelligent navigation. However, traditional feature extraction methods are difficult to effectively handle the nonlinear characteristics of ship motion, which can easily lead to the loss of key turning points and trajectory segmentation fractures. AIS data can provide high-resolution spatiotemporal sequences and multidimensional motion parameters, thus reducing trajectory distortion and key node omissions in feature extraction [16]. Therefore, the study establishes a trajectory feature extraction method based on AIS data. The study first preprocesses AIS data, as shown in Figure 1.

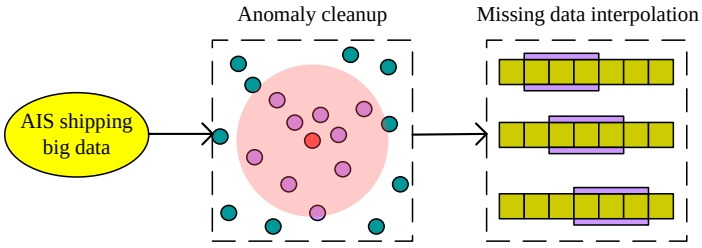


Figure 1: AIS data preprocessing flow

As shown in Figure 1, data preprocessing is divided into two stages: outlier cleaning and missing data interpolation. Among them, the outlier cleaning adopts the dynamic farthest point cleaning method in the neighborhood, which dynamically adjusts the neighborhood radius based on the real-time speed of the ship, and detects and removes outlier trajectory points through deviation threshold [17]. The study first performs dynamic radius calculation, based on the real-time ship speed v_t and time interval Δt, defines the neighborhood radius as shown in equation (1).

$$r_t = max(r_{min}, v_t * \Delta t * \alpha) \tag{1}$$

In equation (1), r_{min} is the minimum radius (set to 100m). α is the safety factor (set to 1.2), based on historical AIS data statistics, balancing noise filtering with trajectory fidelity. Afterwards, the spatial deviation between the current point p_t and its neighboring points $\{p_{t-1}, p_{t+1}\}$ is calculated, as shown in equation (2).

$$d_{dev} = max\left(\frac{\|p_t - p_{t-1}\|}{r_{t-1}}, \frac{\|p_t - p_{t+1}\|}{r_{t+1}}\right) \tag{2}$$

In equation (2), d_{dev} represents the spatial deviation of the current point p_t. When $d_{dev} > 2.0$, the point is marked as an outlier and cleared from the planning data. The dynamic neighborhood farthest point cleaning method can adapt to different navigation states, effectively balancing trajectory smoothness and keypoint preservation. In the stage of missing data interpolation, the sliding window linear interpolation method is studied, which is based on the spatiotemporal continuity of adjacent points within the time window, and calculates the position and motion parameters of missing points through linear weighting [18]. It first performs a window partitioning operation, with the missing points as the center, expanding the time window Δt (Δt is set to 10 minutes, the linear model needs only 2 neighbors in this window, reducing the interpolation time consuming) before and after.Moreover, the effective point set $\{p_{t-k},..., p_{t-1}, p_{t+1},..., p_{t+k}\}$ is extracted within the window. Afterwards, the interpolation weight w is calculated based on the time difference between the missing point t and the adjacent point, as shown in equation (3).

$$w = \frac{t - t_{t-1}}{t_{t+1} - t_{t-1}} \quad w \in [0,1] \tag{3}$$

In equation (3), t_{t-1} and t_{t+1} are the most recent valid timestamps before and after the missing point. Finally, linear interpolation is performed based on the effective point set $\{p_{t-k},..., p_{t-1}, p_{t+1},..., p_{t+k}\}$ and the interpolation weight w, as shown in equation (4).

$$p_t^{fill} = (1-w) * p_{t-1} + w * p_{t+1} \tag{4}$$

In equation (4), p_t^{fill} is the interpolation point, which refers to the latitude, longitude, speed, and heading data of a certain point. Dynamic neighborhood radius is adjusted by speed-driven adjustment (Eqs. 1-2) to ensure that the detection range is enlarged to avoid missing detection when sailing at high speeds, and the radius is contracted to suppress overfitting when sailing at low speeds. Sliding window interpolation (Eq. 3-4) fills in the missing based on spatio-temporal continuity and reduces the computational overhead compared with traditional cubic spline interpolation. After completing data preprocessing, the DP algorithmis used to extract key node features such as ship trajectory turning points and anchorage areas. Through high fidelity simplification, the key geometric features of the trajectoryare preserved, significantly reducing data redundancy, as shown in Figure 2.

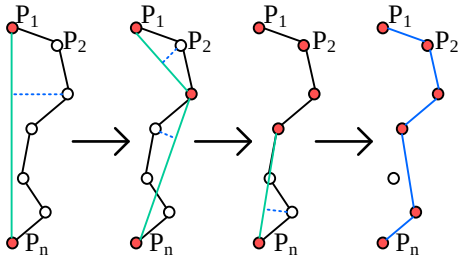


Figure 2: Feature extraction process of DP algorithm

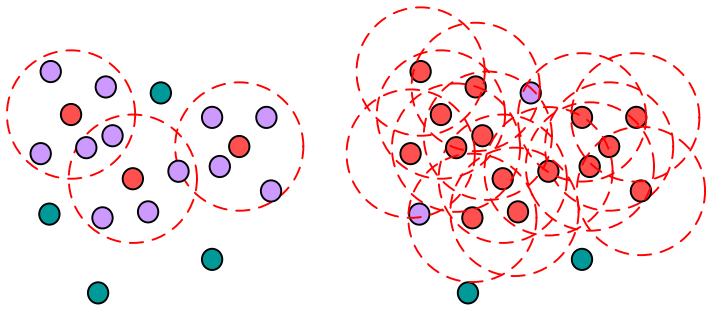


Figure 3: Clustering principle of DBSCAN

As shown in Figure 2, the black line is the original trajectory and the last blue line is the optimized and simplified trajectory. The red dots represent key feature nodes, while the white dots represent uncalculated or non key feature points. The study first inputs the preprocessed trajectory point sequence $P=\{p_1, p_2, ..., p_n\}$ and sets a simplified threshold ε (Optimized by grid search to ensure simplification error <5%). Afterwards, the algorithm connects the starting point p_1 and the ending point p_n to generate the head tail baseline, and calculates the vertical distance d_i from the midpoint p_i to the baseline. The algorithm selects the maximum distance point p_{max}, if $d_{max} > \varepsilon$, uses p_{max} as the feature point, divides the trajectory into $[p_1, p_{max}]$ and $[p_{max}, p_n]$, and recursively processes the sub segments [19]. It retains all feature points that satisfy $d_i \geq \varepsilon$ and generates a simplified trajectory P'. The DP algorithm simplifies the trajectory accuracy by controlling the geometric deviation threshold, while retaining key features such as turning points and anchoring areas, and reducing computational complexity.

3.2 DBSCAN-based trajectory endpoint clustering and establishment of adsrp model

The data preprocessing and trajectory feature extraction methods proposed in the study can improve the quality of trajectory data and accurately capture ship maneuvering behavior characteristics. However, traditional trajectory planning methods lack robustness to the heterogeneity of trajectory endpoint spatial distribution and multi-scale density changes, which can easily lead to misjudgment of trajectory boundaries and failure of anchorage identification. DBSCAN can adaptively identify multi-scale spatial clusters based on density accessibility, effectively distinguishing noise points from low-density anchorage areas. Therefore, the study uses DBSCAN to cluster and determine trajectory endpoints (starting and ending points), as shown in Figure 3.

As shown in Figure 3, DBSCAN conducts clustering based on density reachability. It begins by calculating the number of samples within the ε-neighborhood of each data point. If a point has at least MinPts samples in its neighborhood, it is designated as a core point. The algorithm then expands the neighborhood around this core point to group all density-connected points into the same cluster. Points that do not meet the criteria for core points or density-connected points are classified as noisy [20]. DBSCAN clustering of trajectory endpoints (origin/destination) corresponds to actual port coordinates. Port recognition accuracy measures the algorithm's capability to capture high-frequency berthing areas. High-precision clustering ensures that the origin/destination points of planned routes align with real port locations, providing a geographic reference for global route generation.The clustering process is as follows: First, the network inputs dataset $P=\{p_1, p_2, ..., p_n\}$, sets the neighborhood radius ò and the minimum number of neighborhood points $MinPts$ (set to 5 in the study).The process dynamically adjusts parameter ò based on real-time sea condition data (ò = 500 × [1 + 0.2 × (wind speed / 10)] meters). For each point p_i, it calculates the number of points in its ò neighborhood, as shown in equation (5).

$$N_{ò}(p_i)=\{p_j \in D \mid dist(p_i, p_j) \leq ò\} \quad (5)$$

In equation (5), if $|N_{ò}(p_i)| \geq MinPts$, p_i is marked as the core point. Afterwards, the neighborhood is expanded, starting from the unvisited core point p_c, the queue $Q=\{p_c\}$ is initialized, and p_c is assigned to cluster C_k. Each point p_q in Q is traversed, and if p_q is the core point, the unvisited points in its ò neighborhood are added to Q.The traversal update process is shown in equation (6).

$$Q \leftarrow Q \cup (N_{ò}(p_q) \setminus \{p_v\}) \quad (6)$$

In equation (6), p_q is a point in Q; $N_{ò}$ denotes a point in the ò neighborhood; p_v represents the visited point. p_q is marked as visited, and if p_q does not belong to any cluster, it is added to C_k. Other points that cannot be reached from any core point density are marked as noise [21]. Noise points are directly excluded after clustering and are not involved in trajectory similarity screening and route generation to avoid low-density abnormal trajectories interfering with the planning results, and their spatial distribution is used for subsequent route optimization in sparse regions. In addition, based on the endpoint clustering results, it is necessary to screen trajectory patterns with high spatiotemporal consistency to extract key routes and support the generation of

recommended routes. As a deep metric learning method in machine learning, SNN effectively measures the spatiotemporal similarity of variable length trajectories through shared weights and feature encoding, with strong noise resistance and adaptability to complex patterns [22]. Therefore, the study utilizes SNN to efficiently measure the spatiotemporal similarity of trajectories between DBSCAN clustering endpoints, accurately screen key routes to support recommendation generation, and its structure is shown in Figure 4.

As shown in Figure 4, SNN consists of a deep encoder (Transformer) with shared weights between two towers, a fully connected layer, and a similarity calculation module, which symmetrically processes input trajectory pairs. The encoder adopts a 4-layer Transformer with 512 hidden units, 8 multi-head attention heads, a dropout rate of 0.1, and GELU activation function. The 4-layer design was chosen to balance model depth and computational efficiency; the 512-dimensional hidden units match trajectory feature dimensions; the 8 attention heads capture multi-scale spatiotemporal relationships; and the 0.1 dropout prevents overfitting.The operation process of SNNis as follows: First, inputs the trajectory pair (T_A, T_B), where $T_A = \{x_1^A, ..., x_m^A\}$ and $T_B = \{x_1^B, ..., x_n^B\}$ .Each point $x_t \in R^d$ incorporates dynamic feature channels including position, speed, heading, meteorological, and ocean current parameters, enabling similarity computation to integrate real-time environmental constraints and achieve path generation combining historical patterns with dynamic feedback. Afterwards, feature encoding is performed, which is divided into positional encoding and self attention encoding. In the position encoding stage, SNN adds position encoding $P_t \in R^d$ to the trajectory points to generate an input vector, as shown in equation (7).

$$e_t = x_t + P_t \tag{7}$$

In equation (7), e_t is the encoded input vector. In the self attention encoding stage, SNN extracts global features through multi-head self attention layers, as shown in equation (8)

$$\begin{cases} H_A = \mathrm{TE}(E_A) \ \ H_A \in R^{m \times d} \\ H_B = \mathrm{TE}(E_B) \ \ H_B \in R^{n \times d} \end{cases} \tag{8}$$

In equation (8), E_A and E_B are input sequence matrices, and $E_A = [e_1^A, ..., e_m^A]$, $E_B = [e_1^B, ..., e_n^B]$. H_A and H_B are the encoded trajectory feature matrices. TE is the encoding operation. Afterwards, the encoded features are subjected to temporal average pooling to generate trajectory level vectors, as shown in equation (9).

$$\begin{cases} h_A = \dfrac{1}{m} \sum_{t=1}^{m} H_A^{(t)} \\ h_B = \dfrac{1}{n} \sum_{t=1}^{n} H_B^{(t)} \end{cases} \tag{9}$$

In equation (9), h_A and h_B are trajectory level feature vectors. Next, SNN calculates the cosine similarity between the two, as shown in equation (10).

$$s = \frac{h_A * h_B}{\|h_A\|_2 * \|h_B\|_2} \tag{10}$$

In equation (10), s is the cosine similarity score of the trajectory pair, which measures the similarity between two trajectories. The cosine similarity score $s \in [-1,1]$, where a larger s indicates a more similar trajectory, was set to a threshold of 0.85 in the study [23]. During this process, SNN uses triplet loss to train the model, as shown in equation (11).

$$\mathrm{L} = \max(0, s_{\mathrm{neg}} - s_{\mathrm{pos}} + \mathrm{margin}) \tag{11}$$

In equation (11), s_{neg} represents the similarity between positive sample pairs (similar trajectories). s_{pos} is the similarity between negative sample pairs (heterogeneous trajectories). margin is the similarity difference threshold (set to 0.5 in the study). In summary, the proposed ADSRP model architecture is shown in Figure 5.

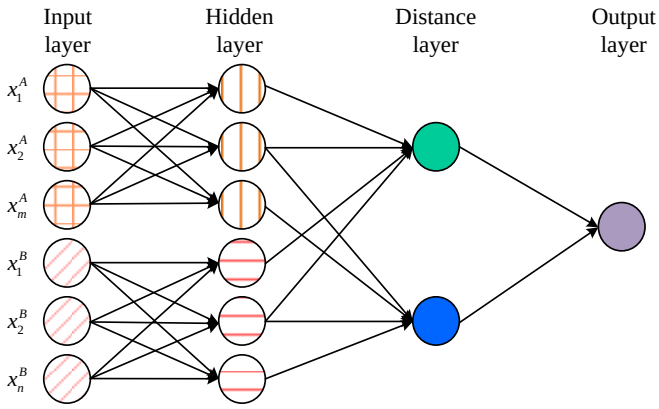


Figure 4: Network structure of SNN

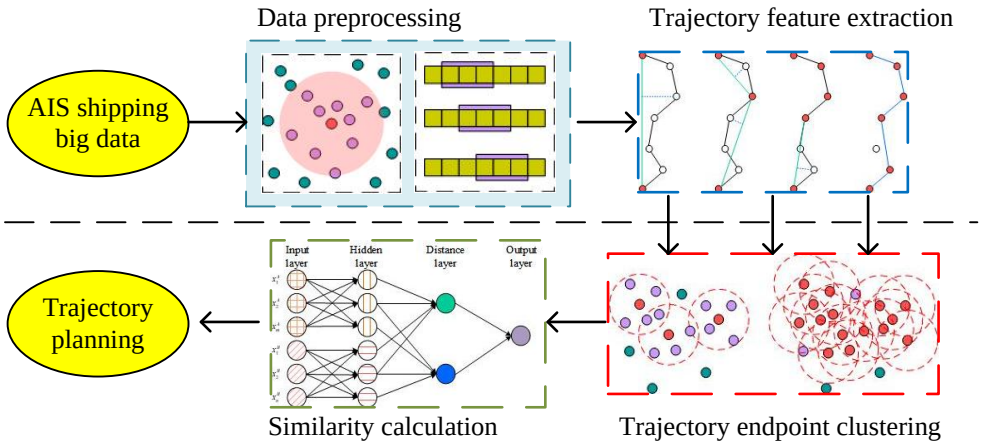


Figure 5: ADSRP model architecture

Table 2: Development environment and experimental parameters

Configurations and parameters		Details
Configurations	Hardware	NVIDIA RTX 6000 Ada 48GB
		AMD Threadripper 7960X
	Software	QGIS 3.28
		PyTorch 2.0
		scikit-learn 1.2
Parameters	Missing hours	15min
	Neighborhood radius(NR)	500m (port area)
	Encoder layers	4
	Learning rate	0.001
	Training round	50
	Confidence interval	95%

As shown in Figure 5, the ADSRP model first uses the DP algorithm to geometrically simplify the preprocessed AIS data trajectory, extract key node features such as turning points and anchoring areas, and solve the problem of nonlinear motion modeling.Afterwards, the model combines DBSCAN to perform density clustering on the trajectory endpoints, adaptively distinguishing noise points from multi-scale anchor areas.Finally, a symmetric deep network based on SNN is designed to screen key flight path patterns with high spatiotemporal consistency through trajectory feature encoding and cosine similarity measurement.By integrating a three-stage process — comprising fidelity compression via the DP algorithm, noise-resistant clustering using DBSCAN, and depth measurement through an SNN—this model attains both high accuracy and robust adaptability in route planning within complex and dynamic environments.

After filtering trajectories based on high spatiotemporal consistency, the model facilitates trajectory recombination by leveraging real-time environmental data (e.g., meteorological conditions and obstacle locations). It dynamically adjusts segment connection priorities using similarity weights derived from an SNN, integrates the A* algorithm to identify an optimal path that balances safety and efficiency across clustered regions, and employs receding horizon control to iteratively refine the path in dynamic environments.

4 Results

The study aimed to verify the superiority of the ADSRP ship route planning model through dual dimensional experiments of simulation and real-world scenarios.Simulation experiments verify the basic effectiveness and theoretical advantages of the algorithm in a controllable environment, while actual experiments evaluate dynamic adaptability based on complex channel data.The former focuses on the robustness verification of the core mechanism of the model, while the latter tests the practical application potential under multi-source interference.

4.1 Simulation operation experiment

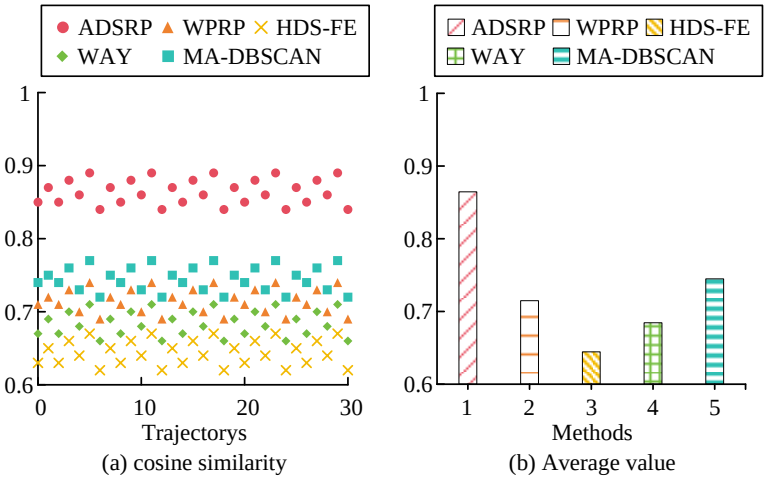
In the simulation operation experiment, the application environment of the ship route planning model was studied and adapted to the development environment and experimental parameters, as shown in Table2.

According to Table 2, the simulation application environment for route planning using the QGIS 3.28 platform was studied. SNN module was trained using PyTorch 2.0. DBSCAN clusteringwas implemented through scikit-learn 1.2. Other model parameters were subject to the settings in the research method. Moreover, the study used the Marine Cadastre AIS dataset as the training and testing sets (with a ratio of 8:2).This dataset contains AIS data for the entire year of 2021 in US waters, covering 1.2 million ship trajectories and a port area

trajectory density of 50 trajectories per square kilometer[24].The training and test sets were divided using stratified sampling based on vessel types (e.g., cargo ships/passenger vessels) and route areas (coastal/open-ocean) to ensure consistent class distribution between sets and prevent evaluation bias from regional or vessel-type differences. For vessel types and route frequencies with low occurrence (e.g., research vessels), temporal oversampling was applied, and a class-weighted loss function was introduced to reduce the impact of long-tail distribution on similarity measurement. All between-group comparisons (e.g., range deviation, fuel consumption) were performed using a two-sample t-test (normal distribution) with a significance level of $\alpha = 0.05$; confidence intervals (95% CI) were calculated by Bootstrap method.

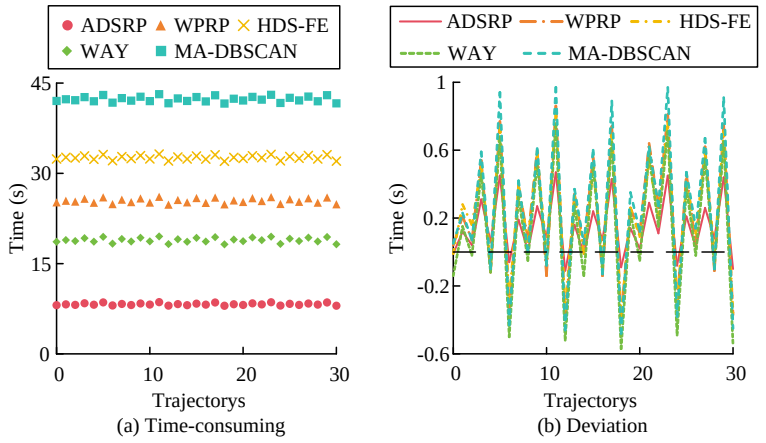
In addition, the study selected the methods used in references [12]-[15] as comparative models, including Whole Process Route Planning (WPRP), High Density Searching Framework Extraction (HDS-FE), Waypoint Analysis-based Destination Estimation (WAY), and Multi-Attribute DBSCAN Optimization (MA-DBSCAN). The ADSRP model proposed in the study was taken as the research object. The study first evaluated the feasibility of model trajectory planning by comparing the cosine similarity between trajectories planned by different methods and historical high-frequency trajectories, as shown in Figure 6.

From Figure 6 (a) and Figure 6 (b), the ADSRP model, based on DBSCAN and spatiotemporal feature fusion mechanism,achieved an average cosine similarityof 0.86in 30 experiments,significantly higher than comparisonmodels such as WPRP (0.71) and HDS-FE (0.64) ($p<0.001$), with a 95% confidence interval of [0.83, 0.89], covering historical high-frequency trajectory patterns.However, WPRP lost temporal continuity due to segmented clustering (range 0.13), while HDS-FE relied on kernel density estimation, resulting in spatial over smoothing. The cosine similarity deviation standard deviation of ADSRP was 0.05 (95% CI±0.10), with a significantly smaller fluctuation range than WAY (0.12) and MA-DBSCAN (0.07), attributed to its encoder's self attention weight dynamic allocation mechanism that suppresses AIS noise interference. There was a significant difference ($p<0.01$) between the ADSRP deviation distribution and the comparison model, with a confidence interval half width (0.025) of only 31.3% of WPRP. Subsequently, the study evaluated the computational efficiency of the model by comparing the time taken from input data to output planned routes using different methods, as shown in Figure 7.



(a) cosine similarity (b) Average value

Figure 6: Differences in trajectory cosine similarity



(a) Time-consuming (b) Deviation

Figure 7: Difference in time spent on planning routes

As shown in Figure 7 (a), the ADSRP model relied on the parallel multi-head self attention mechanism of SNN and took an average of 8.11 seconds (95% CI [7.89, 8.33]) in 30 experiments, which as 67.8% -80.7% lower than WPRP (25.24 seconds) and MA-DBSCAN (42.08 seconds)($p<0.001$).However, the complexity of MA-DBSCAN increased significantly due to the calculation of multi-attribute distance matrix (speed/heading/position), and HDS-FE relied on iterative kernel density search to further drag down efficiency.According to Figure 7 (b), the standard deviation of ADSRP time was 0.18s (95% CI ± 0.35), and the volatility was only 52.9% of the WAY model and 40% of the MA-DBSCAN.Moreover,there was a significant difference ($P<0.01$) between the deviation distribution of ADSRP and the comparison model, with a confidence interval half width of 0.22s, which was 57.7% narrower than WPRP (0.52s). This proved the stability of the SNN architecture in processing long sequence trajectories (>1000 points) and met the real-time threshold of port scheduling (<10s). Furthermore, to verify the influence of DBSCAN algorithm parameters (neighborhood radius ϵ, minimum point number MinPts) on model performance, ADSRP model parameter sensitivity experiments were conducted, as shown in Table 3.

According to Table 3, when ϵ=300m and MinPts=3, the excessively small neighborhood radius led to the fragmentation of the port trajectory into 34 clusters, with noise points accounting for 12.5% and a port recognition accuracy of only 72.3% (95% CI [68.1,76.5]). However, the combination of ϵ=700m and MinPts=7 caused neighboring ports to merge incorrectly due to the large neighborhood, resulting in an accuracy rate of 84.3% (CI [79.6, 89.0]).The optimal parameter combination of ϵ=500m and MinPts=5 achieved a balance between the number of clusters (15), the proportion of noise points (3.8%), and accuracy (93.6%, CI [90.5, 96.7]), and its performancewas significantly better than other combinations ($p<0.01$).Analysis showed that MinPts determined the strength of noise filtering by controlling the spatial density perception granularity: When ϵ<400m, the trajectory segmentation was too fine, and when MinPts<5, temporary anchor interference was introduced. Both would destroy the topological consistency of the origin and destination clustering, thereby reducing the reliability of route planning. Moreover, to investigate the effectiveness of the methodology module introduced in the study, the study conducted ablation experiments, the results of which are shown in Table 4.

In Table 4, removing the dynamic neighborhood cleaning module resulted in a decrease in model noise suppression (cosine similarity 0.71 ± 0.07, 17.4% lower than the full model, p=0.003) and an increase in range deviation to 1.82 ± 0.33 nautical miles. Removing the sliding window interpolation resulted in a distortion of the data's localized trend (deviation 1.95 ± 0.37 nautical miles, p=0.008). Without DP trajectory simplification, computational elapsed time increased to 9.8 ± 0.9 seconds (+20.9%) and voyage bias rose to 1.47 ± 0.29 nautical miles (p=0.012). Replacing DBSCAN clustering with K-means resulted in a significant reduction in port identification accuracy (deviation 1.24 ± 0.25 nautical miles, p=0.021). Fuel consumption increased to 467.5 ± 37.9L (+28.9%, p=0.004) when the SNN similarity metric was disabled. The full ADSRP model incorporated all modules to validate the need for multi-stage co-optimization with a cosine similarity of 0.86 ± 0.05, a range deviation of 0.98 ± 0.12 nautical miles and a low fuel consumption of 362.6L. For the feasibility verification of the cosine similarity threshold s in SNN, the study further conducted sensitivity experiments as shown in Table 5.

Table 3: The sensitivity of DBSCAN parameters

Parameter combinations			Number of clusters	Percentage of noise points (%)	Port recognition accuracy (%)	Accuracy (95% CI)
ϵ (m)	Minpts	Average number of ports				
300	3	36.1	34	12.5	72.3	[68.1, 76.5]
	5	29.7	28	8.9	78.5	[74.0, 83.0]
	7	22.2	21	6.3	82.1	[77.5, 86.7]
500	3	23.1	22	8.1	85.4	[81.2, 89.6]
	5	15.5	15	3.8	93.6	[90.5, 96.7]
	7	12.6	12	4.9	89.7	[85.4, 94.0]
700	3	16.9	16	10.7	80.2	[75.8, 84.6]
	5	9.5	9	6.5	87.9	[83.3, 92.5]
	7	7.4	7	5.1	84.3	[79.6, 89.0]

Table 4: Ablation experiments

Ablated Modules/Full model	Average cosine similarity	Range deviation (nmi)	Fuel consumption (L)	Planning time (s)	Memory usage (%)	p-value (vs. ADSRP)
Dynamic Neighborhood Cleaning	0.71 ± 0.07	1.82 ± 0.33	498.3 ± 38.7	7.2 ± 0.5	39.6 ± 2.8	0.003
Sliding window interpolation	0.68 ± 0.08	1.95 ± 0.37	527.4 ± 41.1	7.5 ± 0.7	42.1 ± 3.0	0.008
DP trajectory simplification	0.75 ± 0.06	1.47 ± 0.29	432.9 ± 35.6	9.8 ± 0.9	35.7 ± 2.5	0.012
DBSCAN clustering	0.78 ± 0.05	1.24 ± 0.25	398.2 ± 32.4	12.3 ± 1.1	31.9 ± 2.2	0.021
SNN similarity metric	0.72 ± 0.07	1.68 ± 0.31	467.5 ± 37.9	8.1 ± 0.8	37.4 ± 2.7	0.004
Full ADSRP model	0.86 ± 0.05	0.98 ± 0.12	362.6 ± 28.3	8.1 ± 0.7	27.5 ± 2.0	-

Table 5: Sensitivity experiments for threshold s

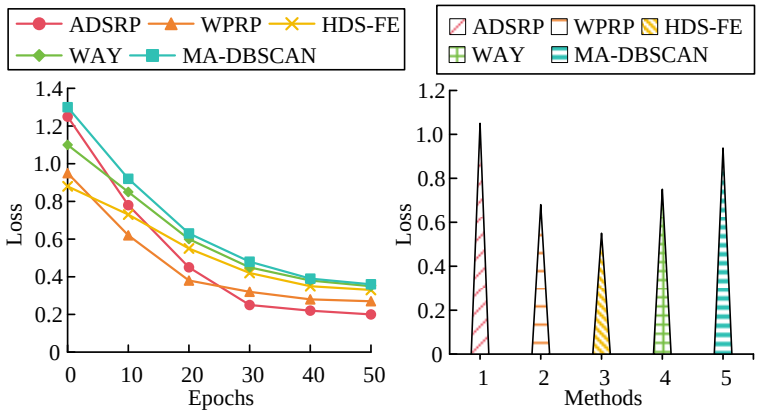
Threshold (s)	Percentage of high similarity routes (%)	Mean range deviation (nmi)	Average fuel consumption (L)	Planning time (s)	p-value (vs. s=0.85)
0.75	98.1 ± 1.2	1.54 ± 0.23	452.8 ± 39.5	6.5 ± 0.5	<0.001
0.8	92.3 ± 2.1	1.22 ± 0.18	402.5 ± 31.7	7.8 ± 0.6	0.004
0.85	78.5 ± 1.8	0.98 ± 0.12	362.6 ± 28.3	8.1 ± 0.7	-
0.9	58.6 ± 3.2	0.89 ± 0.09	340.1 ± 25.9	9.4 ± 0.9	0.018
0.95	34.2 ± 4.5	0.82 ± 0.07	328.7 ± 24.1	12.6 ± 1.3	0.033

In Table 5, when the threshold s=0.85, the range deviation (0.98±0.12 nm) and fuel consumption (362.6± 28.3L) were reduced by 19.7% and 9.9%, respectively, compared to s=0.80 (p=0.004), and the planning elapsed time was stabilized at 8.1 seconds. s=0.75 had a high coverage rate of 98.1% but the deviation spiked to 1.54 nm (p<0.001). s=0.95 had the lowest deviation (0.82±0.07 nm) but the elapsed time increased to 12.6 seconds and only 34.2% (p=0.033). 0.001). s=0.95 had the lowest deviation (0.82±0.07 nautical miles), but the elapsed time increased to 12.6 seconds and the coverage was only 34.2% (p=0.033). s=0.85 achieved the optimal balance of quality, efficiency, and economy, which validated its reliability as a recommended threshold. After that, in order to visualize, the convergence performance of the study model, the study conducted iterative training and the results are shown in Figure 8.

The combination of Figure 8(a) and Figure 8(b) shows that the training loss of each method decreased significantly with rounds, with ADSRP decreasing the most (1.25→0.2, Δ=1.05), followed by MA-DBSCAN (1.3 → 0.36, Δ =0.94). WPRP, HDS-FE and WAY converged at 0.68, 0.55, and 0.75, respectively, with ADSRP having the highest initial loss but fastest convergence after round 30 (0.25→0.2), and significantly better than WAY (0.45→0.35). The loss of ADSRP was stable after the 30th round (0.25 → 0.2), which was significantly better than that of WAY (0.45→0.35). MA-DBSCAN had the highest initial loss but fast convergence, and had the lowest loss at the 50th round (0.36). The stability and convergence efficiency of ADSRP validated its advantage as a recommended model.

4.2 Actual model performance experiment

Although simulation environments could idealize the control of variables, they could not reproduce the dynamic environmental disturbances in real sea areas.Actual model testing could verify the dynamic adaptability and anti-interference ability of the model through real ship interaction scenarios, ensuring the navigation safety and economy of the planned path.Therefore, the study selected Port A and Port B as the endpoints of the trajectory, and chose MA-DBSCAN and WPRP, which performed well in simulation experiments, as the comparative models, while ADSRP was still used as the research object.The embedded edge deployment testing was based on NVIDIA Jetson AGX Xavier (Ubuntu 18.04 LTS, ARMv8.2 architecture), Jetson Nano 4GB (Maxwell architecture with 128-core GPU), and STM32H743VIT6 microcontroller (ARM Cortex-M7 core, 480MHz), covering hardware verification needs from high-performance edge computing to low-power embedded scenarios.The experimental parameters were strictly kept consistent with the simulation run experiment, and the training set used the measured trajectory of Marine Cadastre AIS to ensure comparability of results and support the closed-loop of the model from theoretical verification to practical application.Firstly, the study used different methods to plan the trajectory between A and B, and compared the trajectory with historical high-frequency trajectories, as shown in Figure9.



(a) Training convergence plots (b) Amount of convergence

Figure 8: Comparison of model convergence performance differences

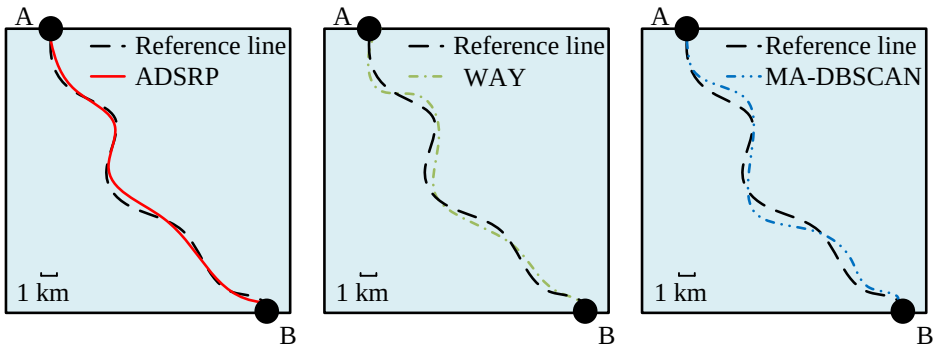
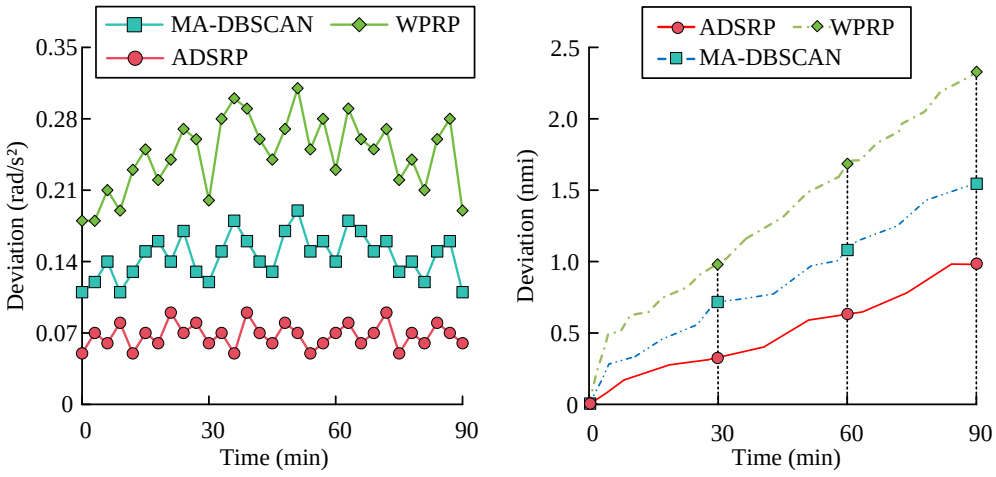


Figure 9: Comparison of different trajectories

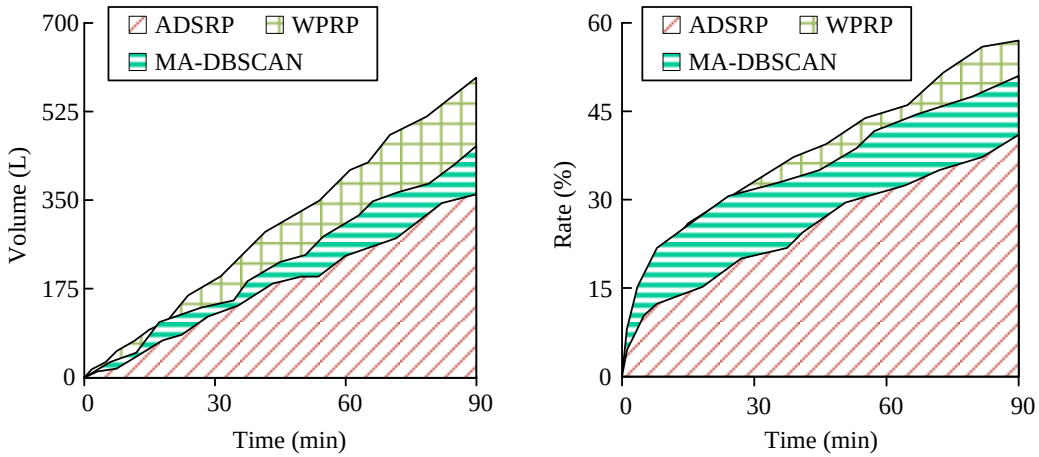
According to Figure 9, the ADSRP trajectory closely followed the dashed reference line throughout the entire process (maximum lateral deviation ≤ 10% of the 1km scale). However, due to probability estimation bias, WAY formed redundant curved detours in the nearshore area (with a lateral offset of 30% from the reference line). Although MA-DBSCAN conformed to the reference line at key turning points, it generated serrated trajectories (curvature radius ≤ 0.3km) due to multi-attribute segmentation clustering. ADSRP minimized range deviation while maintaining heading continuity (actual range was only 2.3% longer than the reference line), while the comparative method required frequent adjustment of speed due to path redundancy or abrupt changes (WAY range extension 8.7%, MA-DBSCAN extension 5.1%). To quantify the reliability of trajectory planning using different methods, a study was conducted to compare the path smoothness deviation and range deviation of different trajectories, as shown in Figure 10.

According to Figure 10 (a), ADSRP optimized heading continuity through spatiotemporal feature fusion, and its average path smoothness deviationwas 0.07 rad/s2 (95% CI [0.05,0.09]), significantly lower thanMA-DBSCAN (0.14rad/s2,P=0.002) and WPRP (0.24 rad/s2, P<0.001).This model utilized multi-head attention weighted speed heading coupling features at route turning points to suppress sudden turns and verify the enhancement effect of SNN structure on trajectory smoothness.According to Figure 10 (b), ADSRP constrained path global consistency through trajectory similarity threshold (s ≥ 0.85), resulting in a total range deviation of 0.98nmi (95% CI [0.92, 1.04]), which was 36.8% lowerthan MA-DBSCAN (1.55,p=0.013).Its encoder effectively captured the inertial characteristics of ship motion along the A-B diagonal main route, reducing redundant evasive maneuvers, and achieving a statistically significant difference in the cumulative deviation growth rate (0.011 nmi/min) of the voyage (p<0.01). Finally, the study compared the fuel consumption and hardware resource utilization of different trajectories to evaluate the actual deployment potential of different planning methods, as shown in Figure 11.



(a) Deviation of path smoothness (b) Course deviation

Figure 10: Actual trajectory planning reliability



(a) Fuel consumption (b) Memory footprint

Figure 11: Analysis of actual deployment potential

Table 6: Comparison of environmental validation of historical data

Year of data	Methods	Track endpoint recognition accuracy	Processing delay (ms)	Prediction error (nmi)
2021	ADSRP	93.20%	182±19	0.91±0.11
	MA-DBSCAN	84.50%	295±38	1.32±0.23
	WPRP	78.70%	258±31	1.49±0.30
2022	ADSRP	91.60%	189±22	0.95±0.13
	MA-DBSCAN	81.30%	308±42	1.38±0.26
	WPRP	73.40%	271±35	1.57±0.33
2023	ADSRP	89.80%	196±25	1.03±0.15
	MA-DBSCAN	78.10%	322±47	1.45±0.29
	WPRP	69.50%	285±39	1.68±0.37
Safety threshold		⩾85%	⩽200ms	⩽1.2 nmi

According to Figure 11 (a), ADSRP utilized trajectory similarity constraints ($s \geq 0.85$) and SNN feature fusion to dynamically optimize the ship's power curve. The average fuel consumptionwas 362.58L (95%CI [355.2369.9]), which was 20.8% and 38.8% lowerthan MA-DBSCAN (457.89L, $p=0.003$) and WPRP (592.98L, $p<0.001$), respectively. The speed heading coupling attention mechanism effectively suppressed redundant steering, with a fuel efficiency standard deviation of only 12.7L. As shown in Figure 11 (b), the average memory occupancy of ADSRPwas 27.5% (95%CI[25.1, 29.9]), significantly lower than that of MA-DBSCAN (36.5%,$p=0.008$) and WPRP (42.5%, $p<0.001$).DBSCAN in ADSRP could reduce intermediate state storage requirements, with a memory usage growth rate (0.43%/min) that was 14% lower than traditional clustering algorithms (0.5%/min), making it suitable for edge device deployment.In addition, the study used the historical AIS data from the endpoints of this port from 2021-2023 to validate the generalization performance across years and test the real-time processing capability, and the results are shown in Table 6.

In Table 6, the accuracy of ADSRP for trajectory endpoint identification in three years of data was 93.20%, 91.60%, and 89.80%, respectively, which was always above the 85% safety threshold, and the prediction error (0.91-1.03 nmi) was better than that of the comparison methods. The processing latency gradually increased from 182ms (2021) to 196ms (2023), which still met the ⩽ 200ms real-time requirement. The latency of both MA-DBSCAN and WPRP exceeded the thresholds (up to 322ms/285ms in 2023) and the prediction error broke through 1.45 nmi/1.68 nmi, which indicated that the performance of the traditional methods declined in the presence of interannual data significantly.

5 Discussion

In response to the insufficient performance of traditional trajectory planning methods, an ADSRP model was proposed to clean AIS shipping big data and implement trajectory endpoint clustering and similarity screening based on methods such as DBSCAN.Experimentsshowed that in simulation, the average cosine similarity of ADSRP reached 0.86 (95% CI[0.83, 0.89]), which was 12% -34% higher than the comparison model($p<0.001$), attributed to the accuracy of DBSCAN in port endpoint clustering (93.6% accuracy at ϵ=500m and MinPts=5) and the spatiotemporal feature fusion of SNN.In actual scenarios, the range deviation of ADSRP was only 0.98nmi (36.8% lower than MA-DBSCAN), fuel consumption was 362.58L (20.8% lower), and the average memory usage was 27.5% (24.7% lower than traditional methods),

verifying its dynamic adaptability and deployment potential.The result metrics demonstrated that range deviation and fuel consumption directly reflected the global optimality of generated routes, smoothness validated dynamic obstacle avoidance capability, while trajectory fit served only as auxiliary validation of historical pattern learning. Collectively, they proved the comprehensive optimization of generated paths in safety, economy, and dynamic adaptability. From the above, DBSCAN accurately recognized the port area based on density sensing and avoided the limitation of the number of preset clusters. Multi-attribute clustering fused speed/heading to enhance the feature correlation, but the high-dimensional distance computation exacerbated the complexity, and the sparse sea area was susceptible to the interference of redundant dimensions. The sliding window interpolation retained the local trend but ignored the nonlinear motion, which may introduce smoothing bias. DP algorithm compressed the trajectory through geometric thresholding, which required dynamic trade-offs between the retention of the key steering points and computational efficiency. SNN suppressed the noise interference by dynamically weighting the spatiotemporal features through multi-attention; and the shared weight encoding with ternary loss enhances the learning ability of the metrics. In practical applications, the model relied on AIS real-time/historical data, and real-time computing resources limited edge deployment.

6 Conclusion

In summary, A route planning framework for multi-scale density trajectories was developed, which solved the contradiction between traditional methods in terms of noise robustness, computational efficiency, and trajectory fit. The proposed method achieved some results in ship route planning, however, there are still limitations. First, DBSCAN's parameter sensitivity and neglect of multi-attribute correlations limited clustering robustness. Second, sliding-window interpolation induced nonlinear motion smoothing bias, and DP over-compression sacrificed critical steering points. Third, traditional models failed to capture spatiotemporal dependencies, and SNN's computational demanded hinder edge deployment. Fourth, reliance on AIS data quality, manual parameter tuning, and dynamic response lagged constrain practical applications.

Aiming at the above problems, the future work of the research lies in: First, develop a hybrid clustering framework integrating DBSCAN density-awareness with multi-attribute weighting and Bayesian-optimized parameter tuning (maximizing Silhouette Score), to improve the robustness of clustering in sparse sea area and high-dimensional data. Second, design LSTM-based adaptive interpolation for nonlinear patterns and curvature-constrained DP algorithms to balance key-point retention. Third, optimize SNN via lightweight attention modules and knowledge distillation while enhancing small-sample generalization through contrastive learning. Fourth, integrate meteorological/bathymetric data fusion and deploy edge-end lightweight ADSRP via federated learning for real-time distributed planning.

References

[1] Daoke Li. Fermat curve path planning method for ship trajectory tracking. Informatica, 48(8):193-206, 2024. https://doi.org/10.31449/inf.v48i8.5735

[2] Juan Li. Improved genetic algorithm enhanced with generative adversarial networks for logistics distribution path optimization. Informatica, 49(11):67-82, 2025. https://doi.org/10.31449/inf.v49i11.6961

[3] Lingxiao Wu, Yossiri Adulyasak, Jean-François Cordeau, and Shuaian Wang, Vessel service planning in seaports. Operations Research, 70(4): 2032-2053, 2022. https://doi.org/10.1287/opre.2021.2228

[4] Tianyang Zhao, Jiayu Qiu, Shuli Wen, and Miao Zhu. Efficient onboard energy storage system sizing for all-electric ship microgrids via optimized navigation routing under onshore uncertainties. IEEE Transactions on Industry Applications, 58(2):1664-1674, 2022. https://doi.org/10.1109/TIA.2022.3145775

[5] Madhushree Das, Arindam Roy, Samir Maity, Samarjit Kar, and Shatadru Sengupta. Solving fuzzy dynamic ship routing and scheduling problem through new genetic algorithm. Decision Making: Applications in Management and Engineering, 5(2):329-361, 2022. https://doi.org/10.31181/dmame181221030d

[6] Chan Woo Han, Sung Wook Lee, and Eun Seok Jin. Tracking of ARPA radar signals based on UK-PDAF and fusion with AIS data. Journal of Ocean Engineering and Technology, 37(1):38-48, 2023. https://doi.org/10.26748/KSOE.2022.046

[7] Eshan Bajal, Vipin Katara, Madhulika Bhatia, and Madhurima Hooda. A review of clustering algorithms: Comparison of DBSCAN and K-mean with oversampling and t-SNE. Recent Patents on Engineering, 16(2):17-31, 2022. https://doi.org/10.2174/1872212115666210208222231

[8] Adan Wu, Tao Che, Xin Li, and Xiaowen Zhu. Routeview: An intelligent route planning system for ships sailing through Arctic ice zones based on big Earth data. International Journal of Digital Earth, 15(1):1588-1613, 2022. https://doi.org/10.1080/17538947.2022.2126016

[9] Liang Zhao, Fang Wang, and Yong Bai. Route planning for autonomous vessels based on improved artificial fish swarm algorithm. Ships and Offshore Structures, 18(6):897-906, 2023. https://doi.org/10.1080/17445302.2022.2081423

[10] Wei Zhao, Hongbo Wang, Jianning Geng, Wenmei Hu, Zhanshuo Zhang, and Guangyu Zhang. Multi-objective weather routing algorithm for ships based on hybrid particle swarm optimization. Journal of

Ocean University of China, 21(1):28-38, 2022. https://doi.org/10.1007/s11802-022-4709-8

[11] Mengxia Li, Junmin Mou, Yixiong He, Xiaohan Zhang, Qinqiong Xie, and Pengfei Chen. Dynamic trajectory planning for unmanned ship under multi-object environment. Journal of Marine Science and Technology, 27(1):173-185, 2022. https://doi.org/10.1007/s00773-021-00825-x

[12] Hongjie Sha, Zhen Han, and Li'an Zhu. The whole process route planning algorithm based on AIS spatio-temporal big data analysis. Ships and Offshore Structures, 1(1):1-15, 2024. https://doi.org/10.1080/17445302.2024.2386883

[13] Miao Gao, Jinqiang Bi, Zhen Kang, Shuai Chen, Peiru Shi, Xi Zeng, and Anmin Zhang. Ship navigable route framework extraction using high-density searching from AIS big data. The Journal of Navigation, 1(1): 1-19, 2025. https://doi.org/10.1017/S0373463325000050

[14] Jin Sob Kim, Hyun Joon Park, Wooseok Shin, Dongil Park, and Sung Won Han. WAY: Estimation of vessel destination in worldwide AIS trajectory. IEEE Transactions on Aerospace and Electronic Systems, 59(5):5961-5977, 2023. https://doi.org/10.1109/TAES.2023.3269729

[15] Xiaofeng Xu, Deqaing Cui, Yun Li, and Yingjie Xiao. Research on ship trajectory extraction based on multiattribute DBSCAN optimisation algorithm. Polish Maritime Research, 28(1):136-148, 2021. https://doi.org/10.2478/pomr-2021-0013

[16] Yiqi Zhang, Yuan Chang, Changbo Wang, Jimmy C. H. Fung, and Alexis K. H. Lau. Life-cycle energy and environmental emissions of cargo ships. Journal of Industrial Ecology, 26(6):2057-2068, 2022. https://doi.org/10.1111/jiec.13293

[17] Weihua Xu, Kehua Yuan, and Wentao Li. Dynamic updating approximations of local generalized multigranulation neighborhood rough set. Applied Intelligence, 52(8):9148-9173, 2022. https://doi.org/10.1007/s10489-021-02861-x

[18] Changbo Hou, Guowei Liu, Qiao Tian, Zhichao Zhou, Lijie Hua, and Yun Lin. Multisignal modulation classification using sliding window detection and complex convolutional network in frequency domain. IEEE Internet of Things Journal, 9(19):19438-19449, 2022. https://doi.org/10.1109/JIOT.2022.3167107

[19] Lixiang Zhang, Yian Zhu, Jie Ren, Wei Lu, and Ye Yao. A method for detecting abnormal behavior of ships based on multi-dimensional density distance and an abnormal isolation mechanism. Mathematical Biosciences and Engineering, 20(8):13921-13946, 2023. https://doi.org/10.3934/mbe.2023620

[20] Praphula kumar Jain, Mani Shankar Bajpai, and Rajendra Pamula. A modified DBSCAN algorithm for anomaly detection in time-series data with seasonality. International Arab Journal of Information Technology, 19(1):23-28, 2022. https://doi.org/10.34028/iajit/19/1/3

[21] Ahmed Arafa, Nawal El-Fishawy, Mohammed Badawy, and Marwa Radad. RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification. Journal of King Saud University Computer and Information Sciences, 34(8):5059-5074, 2022. https://doi.org/10.1016/j.jksuci.2022.06.005

[22] Kavita Bhosle, and Vijaya Musande. Evaluation of deep learning CNN model for recognition of devanagari digit. Artificial Intelligence and Applications, 2023, 1(2):98-102, 2023. https://doi.org/10.47852/bonviewAIA3202441

[23] Nicolás Serrano, and Alejandro Bellogín. Siamese neural networks in recommendation. Neural Computing and Applications, 35(19):13941-13953, 2023. https://doi.org/10.1007/s00521-023-08610-0

[24] David L. Young, Brandan Scully, and Katherine F. Chambers. Resilience-focused analysis of the United States maritime transportation system using automatic identification system data. Transportation Research Record, 2676(7):726-742, 2022. https://doi.org/10.1177/03611981221082561

High-precision Image Classification Algorithm based on Attention Mechanism and Multi-scale Features

Ling Yang, Liantian Li*
Department of Information Engineering, Yangjiang Polytechnic, Yangjiang 529566, China
E-mail: li2020202207@126.com
*Corresponding author

Keywords: Attention mechanism, multi-scale features, convolutional neural network, high precision image classification, feature pyramid

Received: May 23, 2025

High-precision image classification has steadily emerged as a key area of research interest due to the extensive use of image classification technologies in many different domains. The study enhances the conventional feature pyramid networks (FPN) and suggests a high-precision image classification model in an attempt to further increase the precision and effectiveness of picture classification. The model enhances the ability of convolutional neural network (CNN) to focus on key information by combining the channel attention and spatial attention mechanisms. The outcomes indicated that the improved CNN model achieved 77.50% classification accuracy on the ImageNet dataset and 94.20% on the CIFAR-10 dataset, which was significantly higher than the control model. In addition, in the classification of different types of high-precision images, the improved CNN model performed well in the recall, F1 score, and robustness metrics. Their values were 94.3%, 94.6%, and 93.5%, respectively. The results show that the high-precision image classification model is able to capture the key features and detail information in the image more effectively, which significantly improves the classification accuracy and robustness. This study provides a new technical tool for high-precision image classification tasks.

Povzetek: Raziskave računalniškega vida so avtorji izvedli s pomočjo CNN za klasifikacijo slik z izboljšanim FPN (utežena večsklopna fuzija, prilagoditvena konvolucija) in hibridno kanalno-prostorsko pozornostjo. Validacija je narejena na ImageNet/CIFAR-10/medicinskih naborih.

1 Introduction

One of the fundamental tasks of computer vision, picture classification has extensive use in a variety of domains, including automatic driving, remote sensing image analysis, and medical image diagnostics. In these fields, high-precision image classification is crucial for improving work efficiency, reducing errors and enhancing safety [1-2]. For example, in medical image diagnosis, accurate identification of lesion areas is crucial for early diagnosis and treatment. In autonomous driving, high-precision image classification (IC) can help vehicles better recognize road signs and obstacles, thus improving driving safety. Convolutional neural networks (CNNs) have emerged as one of the primary methods for modern IC due to their impressive performance in IC tasks in recent years. By automatically extracting features of an image, CNN can effectively capture local and global information in an image, thus realizing high-precision classification [3-4]. However, despite its excellent performance in IC, CNN still has some limitations. Traditional CNN models are often difficult to effectively focus on key information in images when dealing with complex image scenes, resulting in limited classification accuracy (CA). In addition, CNNs usually only capture single-scale features in the feature extraction (FE) process, ignoring the multi-scale information in the image [5-6]. FE process is a process of gradually extracting local and global features of the image.

Some scholars have also used CNN model to classify images at this stage. Wu et al. proposed an improved CNN model for multi-label medical IC. The model consisted of three main components of CNN and Transformer branch: multi-label multi-head attention-enhanced feature module, multi-branch residual module, and information interaction module. The results indicated that the framework demonstrated good performance on multiple publicly available datasets with good generalization ability and was applicable to other medical multi-label IC tasks [7]. Alkhatib M et al. proposed a model called Improved CNN model to address the lack of training samples in hyperspectral IC and the failure of traditional CNN to fully utilize the correlation between hyperspectral image bands. The results indicated that it outperformed existing methods in terms of overall accuracy, average accuracy, and Kappa coefficient, and obtained near-optimal classification results even with a small number of training samples [8]. Han et al. proposed a dynamic multi-scale CNN model for the current situation of insufficient feature information extracted by CNN and inaccurate attention weights in medical IC. The results indicated that the model achieved most advanced classification performance and solved the uncertainty quantization problem on publicly available datasets from four different

medical domains [9]. The specific retrospective analysis of the above-mentioned literature is shown in Table 1.

Table 1: A specific review and analysis of the literature

References	Method	Advantages	Disadvantages
Reference 7	CTransCNN is a hybrid deep learning model that combines CNN and Transformer, specifically designed for multi-label medical image classification. The model consists of a multi-label multi-head attention enhancement feature module, a multi-branch residual module and an information interaction module.	The implicit correlation between labels is automatically captured, eliminating the need to manually predefine label relationships. A cross-attention mechanism is introduced to allow the model to weight image features according to the importance of each label. Effective fusion of local and global features is achieved through the information interaction module. The feature representation ability of the model is optimized through the embedded and externally embedded residual structures, and the number of parameters is reduced.	The structure of the model is relatively complex, resulting in a long training and reasoning time. The performance of the model depends to a certain extent on the quality and scale of the data set. Due to the complexity of the model, deploying it to mobile devices or other resource-constrained environments may face challenges.
Reference 8	Tri-CNN first uses PCA for dimensionality reduction, and then inputs the data into three branches. Each branch uses 3D-CNN of different scales to extract features. The features extracted from the three branches are flattened and tiled, then classified through the fully connected layer and the softmax layer, and trained using the cross-entropy loss function.	Multi-scale FE can make more comprehensive use of the multi-dimensional information of hyperspectral images. Through feature fusion, the model can better capture features of different scales and types, and improve the classification performance. The model performs well on multiple datasets, demonstrating good adaptability and generalization ability.	Since the model contains multiple branches and multi-scale 3D-CNNs, the computational complexity is relatively high. Although the model performs well on small sample datasets, its performance may depend on a sufficient number of training samples. The complexity of the model may lead to deployment difficulties in practical applications, especially in resource-constrained environments.
Reference 9	The DM-CNN model introduces a dynamic multi-scale feature fusion module, a hierarchical dynamic uncertainty quantization attention mechanism, a multi-scale fusion pooling method, and a multi-objective loss optimization network structure for medical image classification.	The model is capable of extracting feature information at different scales. The attention mechanism can dynamically adjust the attention weights according to different information in each layer, enabling the model to better focus on important feature information. The pooling method can accelerate the computing speed and prevent overfitting while retaining the main and important information. Multi-objective loss can better balance the training process of the model and improve the convergence speed and classification performance of the model	The training and reasoning time of the model is relatively long. Performance may depend on sufficient training samples. Scalability may be limited

The diversity of visual data makes it challenging for a single FE method to match the demand for high-precision classification, even though the aforementioned research has produced superior outcomes. Based on this research, feature pyramid networks (FPN) is improved. Meanwhile, it improves CNN based on attention mechanism (AM) and multi-scale features (MSFs). The research aims to enhance the CNN's ability to focus on key information by introducing an AM and a MSF extraction method, as a way to improve the accuracy and efficiency of IC. The innovation of the study is the introduction of weighted fusion mechanism and adaptive feature adjustment strategy to improve the FPN. Meanwhile, by combining channel attention and spatial attention, the hybrid AM is designed to improve the CNN's capacity to concentrate on important information and increase CA.

2 Methods and materials

2.1 Efficient FE method based on MSF fusion

In IC tasks, FE is one of the key steps to determine the CA. By combining features of several scales, MSFs can enhance the accuracy of IC and collect both global and specific information about an image [10-11]. FE is crucial to the accuracy of image classification. MSFs simultaneously capture image details and global information, improving CA by fusing features of different scales. Multi-scale FE is a key technology for high-precision image classification. Traditional FPN, as a classic architecture, is widely used in object detection and image classification. Its structure is shown in Figure 1.

In the task of image classification, multi-scale FE is one of the key technologies for improving CA. To better understand the multi-scale FE process, the study first introduces the FPN, as shown in Figure 1. The core idea of the traditional feature pyramid lies in combining high-level semantic information with low-level detail information through a top-down path. This generates a multi-scale FPN with rich semantics and details, which effectively captures the multi-scale information of the image [12-13]. The traditional FPN structure can be mainly divided into FE, top-down path, and the generation process of FPN. The key steps of the process are shown in Fig. 2.

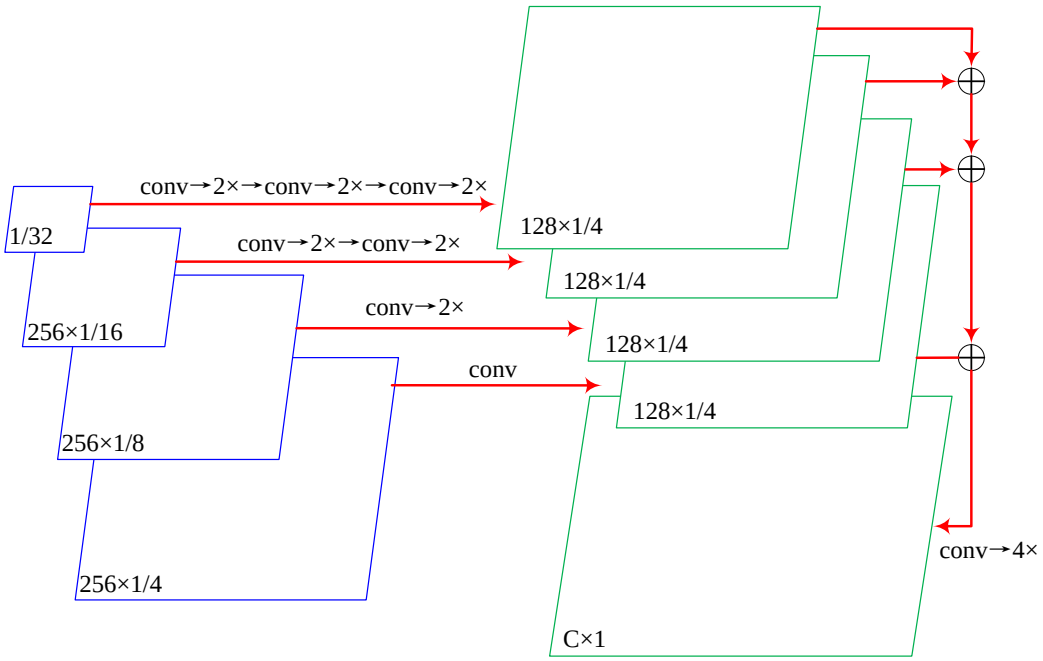


Figure 1: Schematic diagram of the MSF pyramid structure

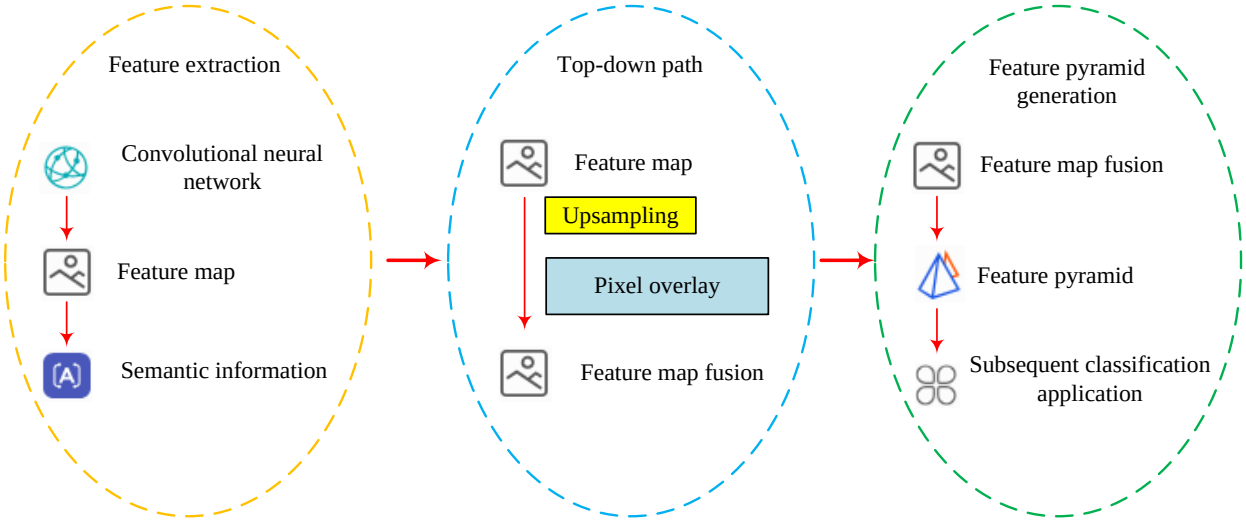


Figure 2: The generation process of the feature pyramid

In Fig. 2, the first is the FE phase, which is based on the first few layers of the CNN to extract different levels of the feature map (FM) $F_1, F_2, \cdots F_n$. Among them, F_i denotes the FM of layer i. Moreover, the resolution of F_i gradually decreases and the semantics gradually increases. FM is the result of the convolution operation. It is a two-dimensional or three-dimensional that represents how the input image (InI) responds to a specific convolution kernel. This is followed by a top-down path stage. That is, starting from the high-level FM, the FM is gradually up-sampled. Meanwhile, pixel-by-pixel summation is performed with the bottom layer FMs, so as to generate a series of fused feature maps (FFMs). For the FM F_i of the i th layer, the mathematical expression of the FFM is shown in Equation (1).

$$F_i^{\sim} = F_i + U(F_{i+1}^{\sim}) \tag{1}$$

In Equation (1), $F_i^{\sim}$ denotes the FFM. U denotes the up-sampling operation. The FFMs are generated through upsampling operations. First, each FM is averaged and pooled globally to generate a one-dimensional vector (1DV). Then, the vector is input into the fully connected layer (FCL) to calculate the weights. Finally, it is the generation stage of FPN. The FFMs are composed into a feature pyramid, which is used for subsequent classification tasks. Although the traditional FPN has achieved remarkable results in MSF extraction, it has some shortcomings in the feature fusion process. For example, the simple pixel-by-pixel summing operation does not take into account the difference in importance of features at different scales. To overcome these limitations, the study proposes an improved FPN network. First, a weighted fusion mechanism is introduced. That is, a weight is assigned to each scale in the process of feature fusion as a way to highlight the feature information with

higher value [14-15]. Defining the FM weight of layer i as ω_i, then the mathematical expression of the FFM is shown in Equation (2).

$$F_i^{\sim} = \omega_i \cdot F_i + U(F_{i+1}^{\sim}) \tag{2}$$

In Equation (2), the weights ω_i are dynamically computed by a weight learning module, which is implemented through global average pooling (GAP) as well as a FCL. A GAP operation is first performed on each FM F_i to obtain a 1DV. The related mathematical expression is shown in Equation (3).

$$v_i = P_a(F_i) \tag{3}$$

In Equation (3), the corresponding weight information is obtained after inputting a IDV v_i into the fully connected form, as shown in Equation (4).

$$\omega_i = \mathrm{Re}LU(FC(v_i)) \tag{4}$$

In Equation (4), FC means the FCL. $\mathrm{Re}LU$ denotes the activation function (AF). After introducing the weighted fusion mechanism to optimize the FPN, in order to further the effect of feature fusion, the study introduces an adaptive adjustment module on the FFM. This module optimally adjusts the fused features through a convolutional layer (CL) as a way to enhance the expression of the relevant features [16]. Specifically based on the FFM $F_i^{\sim}$, the mathematical expression of the FM through optimization and adjustment is shown in Equation (5).

$$F_i^{adjust} = Conv(F_i^{\sim}) \tag{5}$$

In Equation (5), F_i^{adjust} denotes the FM after adjustment and optimization. By introducing the weighted fusion mechanism as well as the adaptive feature adjustment strategy, the FPN network structure proposed in the study is able to fuse MSFs more effectively. Meanwhile, more valuable feature information is highlighted, which in turn improves the performance of IC.

2.2 Construction of IC model with improved CNN based on AM

The introduction of weighted fusion mechanism and adaptive feature tuning strategy through improved FPN has effectively enhanced the performance of FE. However, in complex image scenes, certain regions or channels may be more critical to the classification task. To further improve the accuracy of IC, this study introduces an AM to enhance the CNN's ability to focus on critical information on the basis of FE. The primary function of CNN's central convolutional operation is to extract an image's local features. The related schematic is shown in Fig. 3.

The convolution operation is the core of convolutional neural networks and is primarily used to extract the local features of images. Figure 3 illustrates the specific operation flow to better understand the process of convolution operation. In Fig. 3, the InI is defined as $I \in R^{H \times W \times C}$. Among them, H, W, and C denote the height, width, and quantity of channels of the image. Meanwhile, the convolution kernel $K \in R^{k \times k \times C}$ is slid over the InI and after multiplying and summing by element-by-element, this generates the FM F, as expressed in Equation (6).

$$F_{ij} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} \sum_{c=0}^{C-1} K_{m,n,c} \cdot I_{i+m,j+n,c} + b \tag{6}$$

In Equation (6), b means the bias term. F_{ij} means the value of the output FM at position (i, j). k means the size of the convolution kernel. The primary purpose of pooling procedures is to decrease the FM's spatial dimension (SD) and computational burden. Both maximum pooling (MP) and average pooling (AP) are common pooling operations (POs). The related schematic is shown in Fig. 4.

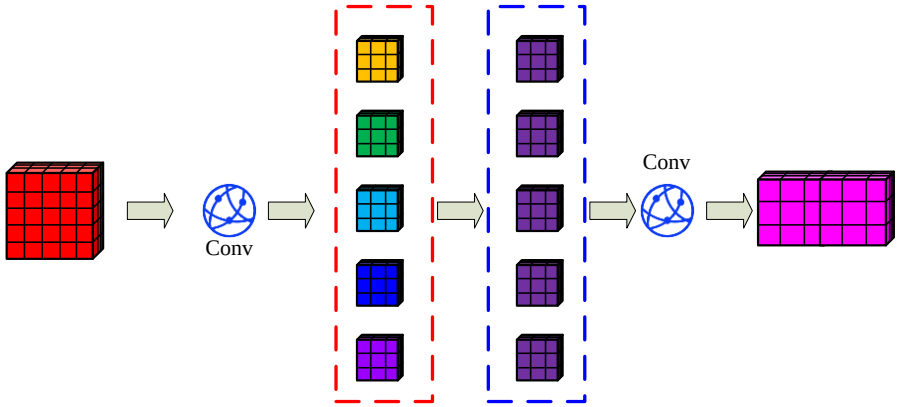
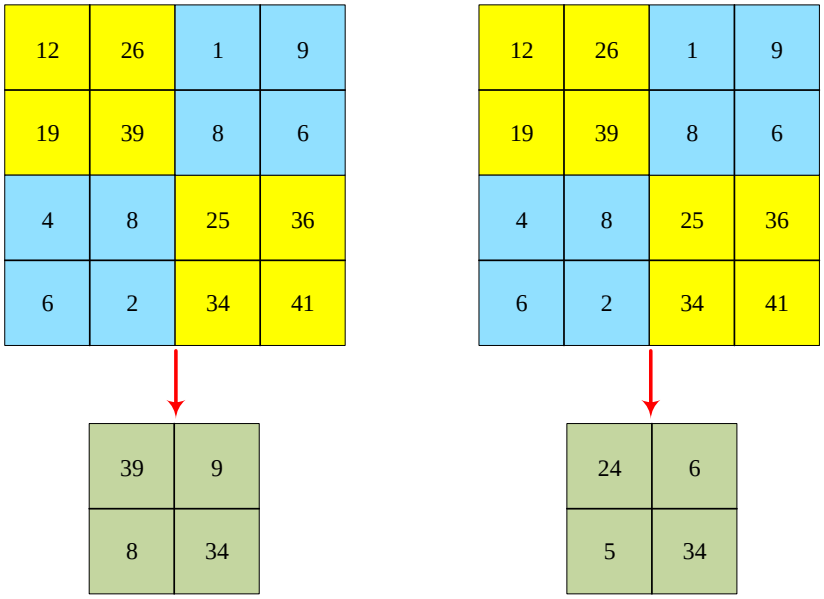


Figure 3: Schematic diagram of CO



(a) Schematic diagram of maximum pooling (b) Average pooling schematic diagram

Figure 4: Schematic diagram of PO

The PO is an important step in CNNs. It is mainly used to reduce the SD of FMs and decrease the computational load. To present the process of POs more intuitively, the study illustrates the specific pooling process through Figure 4. Fig. 4(a) and Fig. 4(b) illustrate the process of MP as well as AP, respectively. For the input FM, the study uses a combination of MP and AP, i.e., for the input F. The mathematical expression related to MP and AP is shown in Equation (7).

$$\begin{cases} F_{\max,i,j} = \max\limits_{m=0}^{p-1} \max\limits_{n=0}^{p-1} F_{i,p+m,j\cdot p+n} \\ F_{avg,i,j} = \dfrac{1}{p^2} \sum\limits_{m=0}^{p-1} \sum\limits_{n=0}^{p-1} F_{i,p+m,j\cdot p+n} \end{cases} \tag{7}$$

In Equation (7), $p \times p$ denotes the size of the convolutional kernel. In the CNN model, the study introduces a hybrid AM. It mainly consists of two types of channel attention as well as spatial attention. First, channel attention mainly learns the weights between channels as a way to highlight the channel features with better values. For the input FM F, GAP, and global MP are applied to individual channels to obtain two IDVs $F_{avg,c}$ and $F_{\max,c}$, as shown in Equation (8).

$$\begin{cases} F_{avg,c} = \dfrac{1}{H \cdot W} \sum\limits_{i=0}^{H-1} \sum\limits_{j=0}^{W-1} F_{i,j,e} \\ F_{avg,i,j} = \dfrac{1}{p^2} \sum\limits_{m=0}^{p-1} \sum\limits_{n=0}^{p-1} F_{i,p+m,j\cdot p+n} \end{cases} \tag{8}$$

After obtaining two IDVs based on Equation (8), the channel attention weights α are obtained by splicing them and inputting them into the two FCLs as shown in Equation (9).

$$\alpha = Sigmoid(FC_2(\mathrm{Re}LU(FC_1([F_{avg}, F_{\max}])))) \tag{9}$$

In Equation (9), *Sigmoid* denotes the AF. Then the mathematical expression of the final channel attention FM is displayed in Equation (10).

$$F_{ca} = \alpha \otimes F \tag{10}$$

In Equation (10), F_{ca} is the channel attention FM. $\otimes$ denotes element-by-element multiplication. The spatial AM focuses on the weights between spatial locations as a way to highlight more valuable spatial regions [17-18]. First, two 2D FMs F_{avg}^t and $F_{\max}^t$ are obtained after channel AP and channel MP of the FMs, as shown in Equation (11).

$$\begin{cases} F_{avg,i,j}^t = \dfrac{1}{C} \sum\limits_{c=0}^{C-1} F_{i,j,e} \\ F_{\max,i,j}^t = \max\limits_{c} F_{i,j,c} \end{cases} \tag{11}$$

After obtaining a 2D FM based on Equation (11), it is spliced and input into a CL as a way to obtain the spatial attention weight β, as shown in Equation (12).

$$\beta = Sigmoid(Conv([F_{avg}, F_{\max}])) \tag{12}$$

In Equation (12), *Conv* denotes the convolution operation (CO). The final mathematical expression based on the spatial attention FM F_{sa} is shown in Equation (13).

$$F_{sa} = \beta \otimes F \tag{13}$$

The building of the enhanced CNN model is finished once the channel AM and the spatial AM have been incorporated into each CL. Fig. 5 displays the schematic of the CNN model based on the AM.

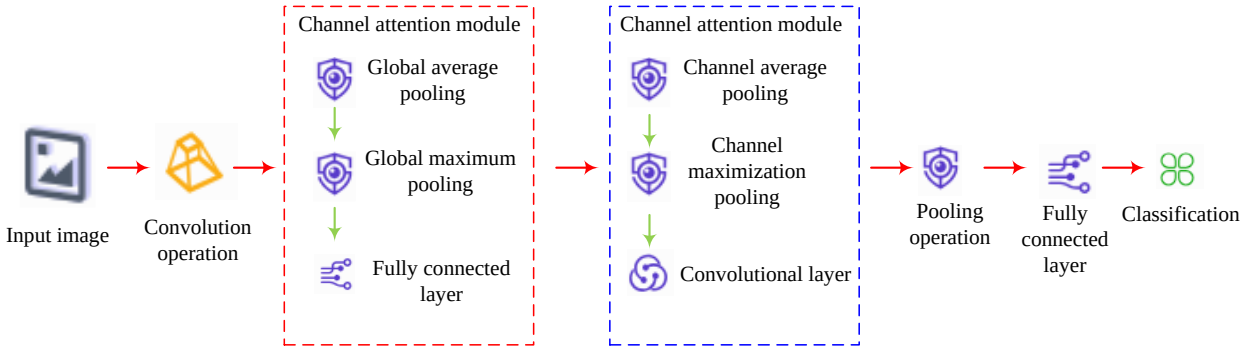


Figure 5: The structure of CNN model based on the AM

In Fig. 5, first, for the InI I, the FM F is obtained based on Equation (14) after the CO through the CL.

$$F = Conv(I) \tag{14}$$

Based on the FM F, it is applied to the channel and spatial attention module and the corresponding FM is obtained. The PO is performed on the final FM as a way to reduce the SD of the FM, as shown in Equation (15).

$$\begin{cases} F_{pool} = Pool(F_{final}) \\ F_{final} = \beta \square \ F_{ca} \end{cases} \tag{15}$$

The pooled FM is expanded into a IDV, which is fed into the FCL, as shown in Equation (16).

$$Y = \mathrm{Re}\,LU(WF_{pool} + b) \tag{16}$$

Finally, the result of the FCL is output to the classifier, which can complete the process of classifying high-precision images. Overall, the study proposes a FE model based on improved FPN. By introducing a weighted fusion mechanism and an adaptive feature adjustment strategy, it is able to fuse MSFs more effectively and highlight valuable information. Second, a CNN classification model based on the AM is constructed. By combining the channel and spatial AMs, the model's ability to focus on key information is enhanced. To facilitate the description of subsequent experiments, the overall model combining these two modules is defined as AM CNN with improved FPN (AM-CNN-FPN) model.

3 Results

3.1 Performance evaluation of CNN models with the introduction of an AM

The above study introduces a weighted fusion mechanism, an adaptive feature adjustment strategy, and an AM to optimize the CNN model. The operation of deep learning (DL) models often possesses high requirements on the computer environment. The experimental environment Settings are shown in Table 2 as follows.

After the experimental environment is set up, the specific parameters and architecture of the model are elaborated in detail in the study. The relevant parameter settings and values are shown in Table 3.

Based on the network structure parameter information shown in Table 3, the study first validates the performance of the improved FPN. It introduces the traditional CNN model, support vector machine (SVM) model, and random forest (RF) for controlled experiments. Meanwhile, the dataset is selected from ImageNet dataset and CIFAR-10 dataset.

Table 2: Experimental environment setting

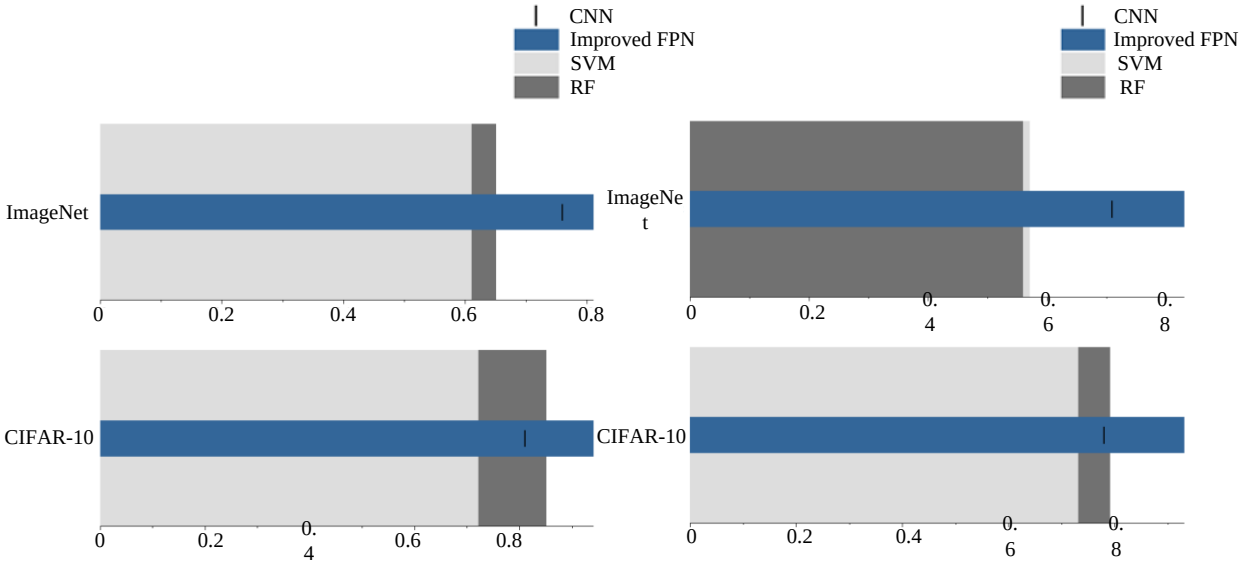
Name	Model and configuration	Name	Model and configuration
Operating system	Ubuntu 20.04 LTS	Deep learning framework	PyTorch 1.9.0
CPU	Intel Core i7-9700K	Programming version	Python 3.8
GPU	NVIDIA GeForce RTX 2080 Ti	CUDA version	CUDA 11.1
Memory	32GB DDR4 (3200 MHz)	cuDNN version	cuDNN 8.0
Hard disk	1TB NVMe SSD + 2TB HDD	/	/

Table 3: Parameter settings and values

Layer type	Layer name	Filter size	Number of filters	Stride	Padding	Activation function	Dropout rate
Input layer	Input	/	/	/	/	/	/
Convolutional layer	Conv1	3x3	32	1	Same	ReLU	0.2
Pooling layer	Pool1	2x2	-	2	/	/	/
Convolutional layer	Conv2	3x3	64	1	Same	ReLU	0.3
Pooling layer	Pool2	2x2	-	2	/	/	/
Convolutional layer	Conv3	3x3	128	1	Same	ReLU	0.4
Pooling layer	Pool3	2x2	-	2	/	/	/
Fully connected layer	FC1	/	256	/	/	ReLU	0.5
Fully connected layer	FC2	/	128	/	/	ReLU	0.5
Output layer	Output	/	Num_classes	/	/	Softmax	/

Large-scale IC and target identification tasks are the primary applications for the ImageNet dataset, which has over 14 million annotated images. The CIFAR-10 dataset is frequently used to assess how well IC models perform,

particularly in terms of their capacity to classify small-size pictures. In the setting of hyperparameters, the attenuation factor of the learning rate is set to 0.01, and the minimum learning rate is set to 1×10-6. The initial learning rate of the optimizer Adam is 0.001, where β1=0.9, β2=0.999, and the weight attenuation coefficient is 1×10-5. According to the early stop standard, training will be stopped if the loss of the validation set does not improve within 20 consecutive epochs. Meanwhile, if the improvement of the validation set loss is less than 1×10-4, it is considered that there is no improvement. To improve the model's generalization ability and robustness, a variety of data augmentation techniques are adopted during the study's training process. First, there is a random horizontal flip, meaning the image is flipped horizontally at random with a probability of 0.5. Second, there is a random vertical flip, meaning the image is flipped vertically at random with a probability of 0.5. Next comes the random rotation, that is, the random rotation of the image, with the rotation Angle ranging from -10 to +10 degrees. Finally, there is random cropping, which involves cropping a random part of the image. The size of the cropped image is 224×224. The study simultaneously divides the dataset into the training set, the validation set and the test set in a ratio of 7:1:2. The study starts by comparing the four models' CA comparability. The results are shown in Fig. 6.



(a) Parallel experiment 1 classification accuracy comparison (b) Parallel experiment 2 classification accuracy comparison

Figure 6: Comparison of CA of the four models

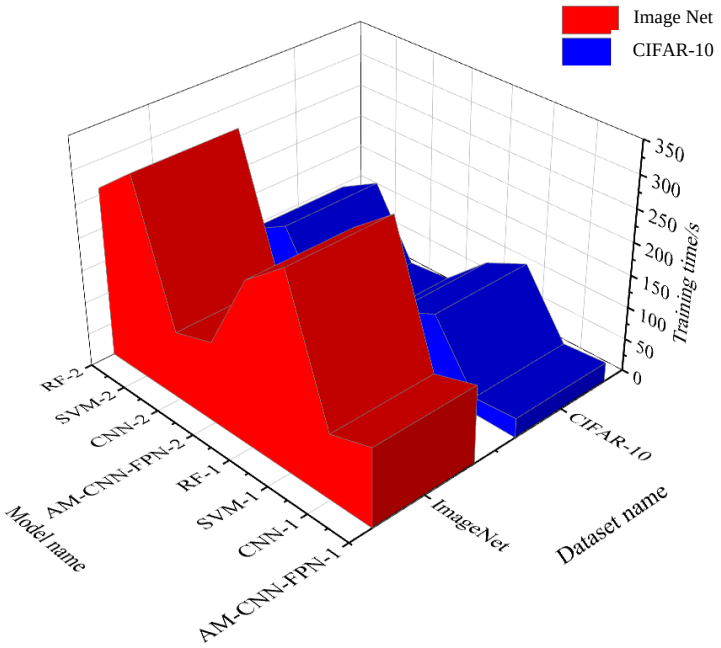


Figure 7: Comparison of computational efficiency among the four models

Fig. 6(a) shows the comparison of CA of the four models in parallel experiment 1. Fig. 6(b) shows the comparison of CA of the four models in parallel experiment 2. The enhanced FNN model's CA in the two simultaneous runs on the ImageNet dataset is 0.81 and 0.83, which is noticeably higher than that of the other three models. This indicates that the improved FNN model has obvious advantages in dealing with large-scale and complex IC tasks. The CA of the traditional CNN model is 0.76 and 0.71 in the two experiments, respectively. Although it also shows better performance, it still falls short of the improved FNN model. On the CIFAR-10 dataset, the CA of the improved FNN model is 0.94 and 0.93 in two parallel experiments, which is also significantly higher than the other three models. The findings reveals that the improved FNN model not only has an obvious advantage in CA, but also excels in performance stability and consistency. The study further compares the computational efficiency of the AM-CNN-FPN model with the control model. The results are shown in Fig. 7.

Fig. 7 shows the training time (TT) comparison of the four models. TT refers to the total time required for a model to complete its training process. This time depends on various factors, including the model's complexity, the dataset's size, and the hardware configuration. The results show that the TT of the AM-CNN-FPN model is 120s and 125s in parallel experiment 1 and parallel experiment 2, respectively. Its TT is slightly longer compared with that of the traditional CNN model, which may be due to the introduction of the AM and MSF fusion, and the computational complexity is increased. However, the overall TT is still within the acceptable range. The computational efficiency of the improved model is not significantly reduced while the performance is improved. The TT of SVM model is significantly higher than that of CNN model, probably due to the fact that SVM needs to extract features manually. Moreover, its computational complexity is higher when dealing with large-scale datasets. The TT of RF model is also longer. Although it shows better performance in some tasks, it is less computationally efficient on large-scale datasets. In summary, the AM-CNN-FPN model does not significantly decrease the computational efficiency while improving its performance. It shows that it has good application prospects in high-precision IC tasks. The study further introduces more advanced models, ResNet, DenseNet, and EfficientNet, for comparative experiments. Therefore, the performance comparison of the four models is shown in Table 4.

Table 4 shows that the performance parameters of the AM-CNN-FPN proposed in the study are all superior to those of the control model, with respective accuracy rates of 77.5% and 94.2%. The recall rates are 94.3% and 94.0%, respectively. The F1 is 94.6% and 94.3%, respectively. The robustness indicators are 93.5% and 94.5%. The TTs are 120 s and 30 s. The results show that the AM-CNN-FPN model performs well in both datasets. Meanwhile, its TT is the shortest, indicating that it also has advantages in computational efficiency. In conclusion, the AM-CNN-FPN model is highly effective and efficient at complex image classification tasks, making it a high-precision image classification model. The study verifies the performance of the AM-CNN-FPN model with ablation experiments. The results are shown in Table 5.

In Table 5, the CA of the AM-CNN-FPN model on the ImageNet dataset is 77.50%, and the TT is 120s. The CA of the AM-CNN-FPN model on CIFAR-10 is 94.20%, and the TT is 30s. The results show that the full AM-CNN-FPN model on both datasets exhibits the highest CA. It demonstrates how these two mechanisms working together can significantly enhance the model's performance. Meanwhile, the basic CNN model has an advantage in TT, but still has room for improvement in CA. It shows that the base CNN has achieved a better balance between performance and efficiency. However, its performance can be further improved by introducing the AM and improved FNN.

Table 4: Model performance comparison

Model	Dataset	Accuracy (%)	Recall (%)	F1 score (%)	Robustness (%)	Training time (s)
AM-CNN-FPN	ImageNet	77.5	94.3	94.6	93.5	120
ResNet	ImageNet	76.8	93.5	93.8	92	110
DenseNet	ImageNet	76.2	92.8	93.2	91.5	130
EfficientNet	ImageNet	76.5	93.2	93.5	92.5	115
AM-CNN-FPN	CIFAR-10	94.2	94	94.3	94.5	30
ResNet-50	CIFAR-10	93.5	92.5	92.8	93	28
DenseNet	CIFAR-10	93	91.5	91.8	92	32
EfficientNet	CIFAR-10	93.3	92.8	93	93.2	31

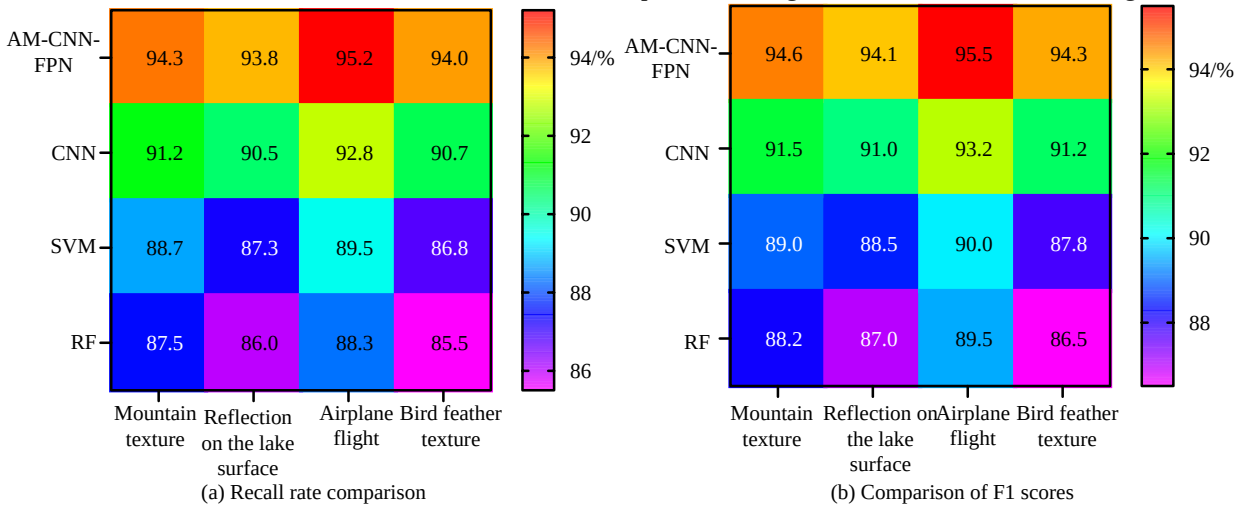
Table 5: Ablation experiment results

Data set	Model configuration	Classification accuracy	Training time
ImageNet	Complete improvement of CNN (AM+improved FPN)	77.50%	120
	Attention-free mechanism (only improving FPN)	76.64%	115
	No improved FPN (AM only)	76.25%	118
	Attention-free mechanism and improved FPN (basic CNN)	76.05%	100
CIFAR-10	Complete improvement of CNN (AM+improved FPN)	94.20%	30
	Attention-free mechanism (only improving FPN)	93.50%	28
	No improved FPN (AM only)	93.80%	29
	Attention-free mechanism and improved FPN (basic CNN)	93.06%	25

3.2 Performance comparison of different types of high-precision IC

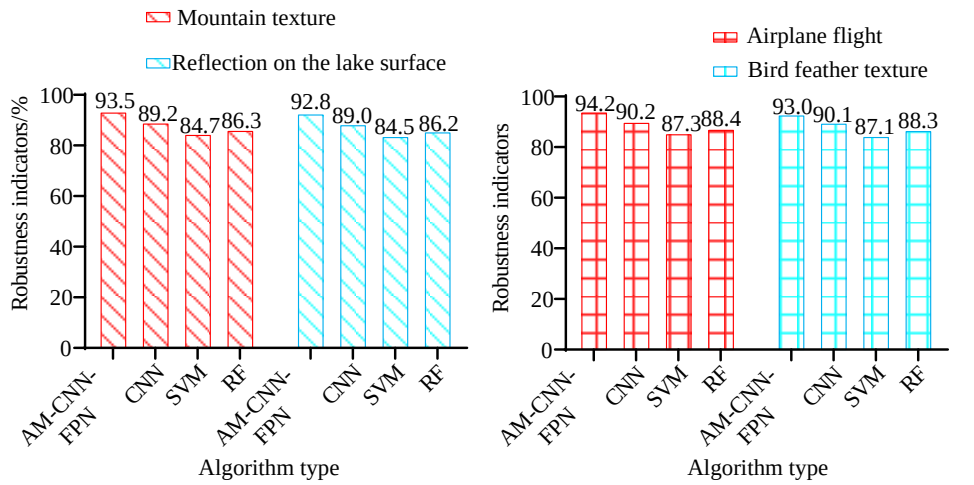
After validating the performance of the AM-CNN-FPN model, the study further compares the classification performance of the model for different types of high-precision images. Among them, the high-precision images include mountain texture, reflection on the lake surface, airplane flight, and bird feather texture. First, the recall and F1 score (F1) comparisons based on the four high-precision images are shown in Fig. 8.

Fig. 8(a) shows the recall comparison of the model against four high-precision images. Fig. 8(b) shows the comparison of F1 of the model for four high-precision images. The results show that the recall of the AM-CNN-FPN model on the mountain texture image reaches 94.3% and the F1 is 94.6%. It is 3.1% and 3.1% higher than the traditional CNN, respectively. This illustrates that AM-CNN-FPN is able to recognize the target object more accurately while maintaining a high precision rate when recognizing complex natural scenes like mountain texture. In contrast, the SVM and RF models have lower recall and F1 of 88.7%, 89.0%, 87.5%, and 88.2%, respectively. The AM-CNN-FPN model has a recall of 93.8% and an F1 of 94.1% in the classification of the reflection on the lake surface image. It is also higher than 90.5% and 91.0% for the traditional CNN model. For airplane flight IC, the AM-CNN-FPN model achieves 95.2% recall and 95.5% F1, respectively. This compares favorably with 92.8% and 93.2% for traditional CNN. Finally, in the classification of bird feather texture images, the AM-CNN-FPN model has a recall of 94.0% and an F1 of 94.3%. It is 3.3% and 3.1% higher than the traditional CNN, respectively. In summary, the AM-CNN-FPN model enables the model to capture the key features and detail information in the image more effectively by introducing the AM and the improved FPN network. Meanwhile, it achieves a better balance between precision and recall. The study further compares the robustness metrics of the model for high-precision images. The results are shown in Fig. 9.



(a) Recall rate comparison (b) Comparison of F1 scores

Figure 8: Recall rate of high-precision images and comparison of F1



(a) Comparison of robustness indicators between mountain texture and the reflection on the lake surface

(b) Comparison of robustness indicators between airplane flight and bird feather texture

Figure 9: Comparison of model robustness indicators

Fig. 9(a) shows the comparison of model robustness metrics for mountain texture and reflection on the lake surface high-precision images. Fig. 9(b) shows the comparison of model robustness indexes for airplane

flight and bird feather texture high-precision images. The results show that the robustness index of AM-CNN-FPN model is 93.5%, which is significantly higher than that of traditional CNN (89.2%), SVM (84.7%), and RF (86.3%). It shows that AM-CNN-FPN is better able to resist the influence of noise and illumination changes and maintain high classification performance when dealing with complex natural scenes. In classification of reflection on the lake surface images, the robustness index of AM-CNN-FPN model is 92.8%, which is significantly higher than the rest of the models. It shows that it can recognize key features more effectively and reduce the possibility of misclassification when dealing with images with complex light and shadow effects. The robustness index of AM-CNN-FPN in airplane flight IC is 94.2%. It shows that it can locate and classify more accurately when recognizing images with clear target objects. Finally, the robustness index of AM-CNN-FPN in bird feather texture is 93.0%. It shows that it can extract key features more effectively and improve the CA when dealing with images with rich details and complex textures. In summary, the AM-CNN-FPN model, through the introduced AM and improved FPN network, enables the model to capture key features and detail information in images more effectively. Meanwhile, it shows stronger stability when facing complex conditions such as noise, light changes, and occlusion. A comparison of the model's capacity to extract fine-grained features for four high-precision images rounds out the study. Fig. 10 displays the findings.

The model's capacity to collect fine-grained features for four high-precision images is compared in Fig. 10. The results show that the AM-CNN-FPN model's detailed feature capturing ability for mountain texture images is 95.2%, which is significantly higher than that of traditional CNN (92.1%), SVM (87.6%), and RF (88.9%). It shows that it is able to capture features such as mountain texture and contours more effectively when dealing with complex natural scenes. For the reflection on the lake surface image, the detailed feature capturing ability of AM-CNN-FPN is 94.5%, which is higher than that of traditional CNN (91.3%), SVM (86.7%), and RF (87.8%). It shows that it can recognize and capture key details more effectively when processing images with complex lighting effects. For airplane flight, the detailed feature capturing ability of AM-CNN-FPN is 96.0%, which is significantly higher than that of traditional CNN (93.4%), SVM (88.2%), and RF (90.1%). It shows that it is able to capture detailed features more accurately when recognizing images with clear target objects. Finally, in terms of bird feather texture image, the detailed feature capturing ability of AM-CNN-FPN is 94.8%, which is higher than that of traditional CNN (91.7%), SVM (87.3%), and RF (89.5%).

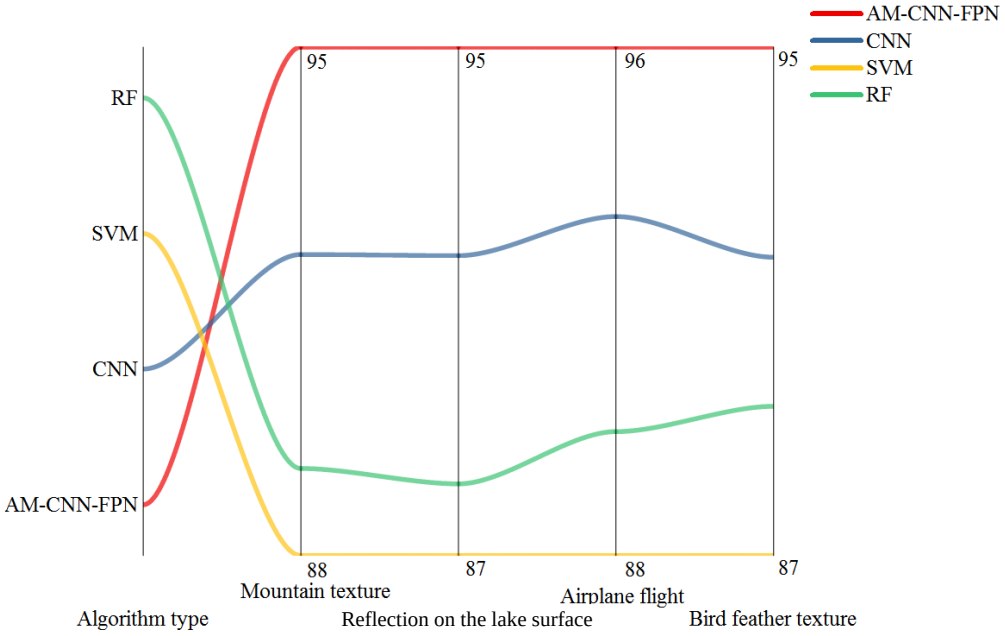


Figure 10: Comparison of detail feature capture capabilities

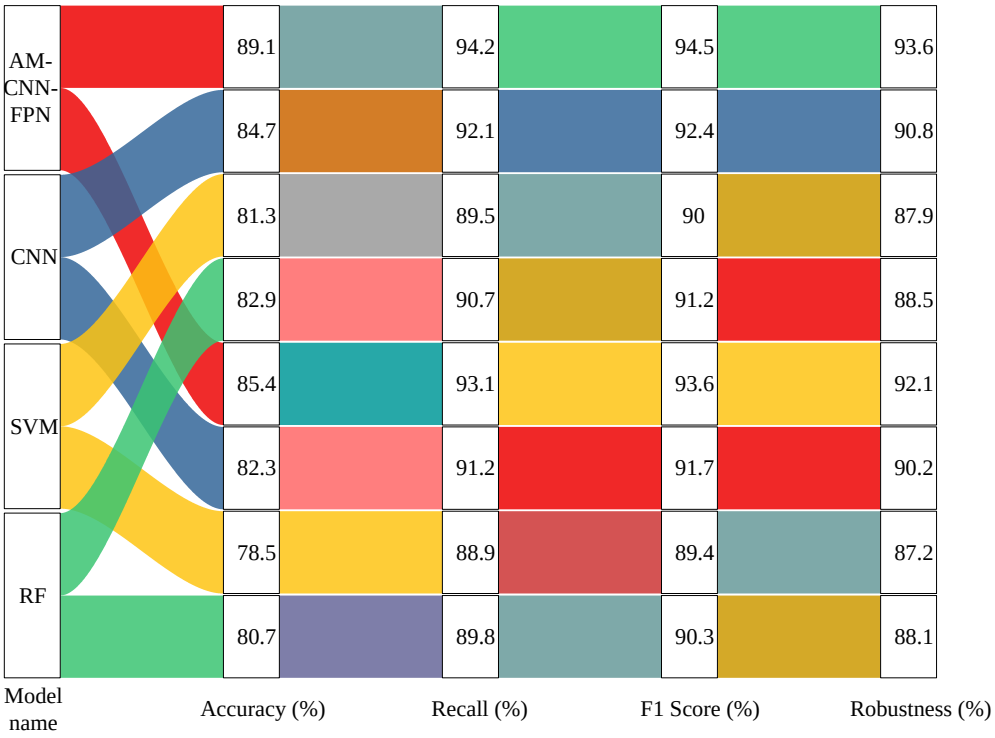


Figure 11: Performance comparison of four models for the medical image dataset

3.3 Comparison of model performance indicators in the medical field

Although the model performed well in general applications in the above-mentioned research, its use is limited to specific fields. To explore the broader application of the model, its performance in the field of medical images is studied and investigated. The MIMIC-CXR and CheXpert datasets are selected for the study on medical images. MIMIC-CXR is a large-scale dataset of chest X-ray images, containing over 370,000 chest X-ray images, and is used for the diagnosis of various diseases. The CheXpert dataset contains over 220,000 chest X-ray images for classifying various chest diseases. First, the performance comparison of the four models for the medical image dataset is shown in Figure 11.

As shown in Figure 11, the performance indicators of the AM-CNN-FPN model proposed in this study are superior to those of the control model. Its accuracy rates are 89.1% and 85.4%, respectively. The recall rates are 94.2% and 93.1% respectively. The F1 is 94.5% and 93.6%, respectively. The robustness indicators are 93.6% and 92.1% respectively. The SVM model performs the worst in the medical dataset, achieving an accuracy rate of less than 80%. All of the other performance indicators are below 90%. In summary, the AM-CNN-FFN model outperforms traditional CNN, SVM and RF models on medical image datasets. It indicates that the AM-CNN-FFN model not only performs well on general datasets, but also has good applicability and generalization ability on domain-specific datasets.

4 Discussion and conclusion

To enhance the performance of high-precision IC, the study improved the traditional FPN model and constructed an improved CNN model. Meanwhile, experimental validation was carried out on ImageNet and CIFAR-10 datasets. The numerical results indicated that on the ImageNet dataset, the CA of the improved FPNN model reached 77.50%, which had a significant advantage over the traditional CNN model (76.05%), SVM model (65.0%), and RF model (61.0%). On the CIFAR-10 dataset, the CA of the improved FNN model was 94.20%, which was also significantly higher than that of the traditional CNN model (93.06%), SVM model (85.0%), and RF model (72.0%). This outcome was in line with Yu F et al.'s findings, which suggested an enhanced FPN model for the advanced in entire crop growth cycle IC and recognition applications. The results indicated that the classification ability of the images of this improved FPN model was significantly better than that of the control model. Similar to this study, the CNN model incorporating the improved FPNN also introduced the AM as well as multi-feature fusion [19]. AM-CNN-FPN could capture the key features and detail information in the image more efficiently, which significantly improved the classification precision and robustness. In addition, AM-CNN-FPN performed well in different types of high-precision images (e.g., mountain texture, reflection on the lake surface, airplane flight, bird feather texture) in terms of recall, F1, and robustness metrics. The performance was excellent in terms of recall, F1, and robustness. For example, in mountain texture IC, the recall of AM-CNN-FPN reached 94.3%, the F1 was 94.6%, and the robustness index was 93.5%, which were higher than other models. The results differed from those of Zhou X et al. which proposed a tool wear classification method based on CNN and time series images. The method classified cutting force signals by converting them into time series images and then inputting them into a CNN model. Unlike this study, the

classification ability of this model was weaker than the improved RF model. The possible reason for this could be that the improved RF was highly robust to noise and outliers and did not require complex preprocessing steps. Meanwhile, the model complexity of the improved RF was relatively low and the TT was shorter. Even on ordinary computing resources, its training could be completed quickly [20].

The findings demonstrate that the enhanced CNN model offers notable benefits for handling intricate natural sceneries and fine-grained characteristics. In summary, the numerical results reflect the effectiveness and superiority of the improved CNN model in high-precision IC tasks, which provides a new research direction and technical means for the field of IC. The AM-CNN-FPN model proposed in the research improves the classification performance by introducing the AM and multi-scale FE. However, the complexity of deep neural networks typically makes the models difficult to interpret. The model's AM provides clues about which parts of the InI are important for classification. For example, the channel attention module emphasizes the most informative FM, and the spatial attention module highlights important spatial regions within the FM. These attention maps can be visualized to provide intuitive insight into the model's decision-making process.

Although the AM-CNN-FPN model has shown promising results, it still has several limitations. First, the model's performance depends heavily on the quality and quantity of the training data. Insufficient or biased training data may lead to poor model performance and limited generalization ability. Second, introducing the AM and multi-scale FE increases the model's complexity and computational requirements. Compared with simpler models, this may lead to longer TT and higher resource consumption. Third, the interpretability of deep neural networks remains a challenge. Although AMs offer some interpretability, they cannot fully reveal the model's decision-making process. To enhance the robustness and generalization ability of the model in future research, more advanced data augmentation techniques and regularization methods should be explored. Meanwhile, techniques such as pruning, quantization, and knowledge distillation are used to reduce the model's complexity and computational requirements. In addition, layer-by-layer correlation propagation, SHAP or other advanced methods are adopted to explain the prediction results of the model in more detail. Ultimately, integrating the proposed model with other modalities could result in a more comprehensive and robust classification system.

References

[1] Rajdeep Kaur, Rakesh Kumar, and Meenu Gupta. Deep neural network for food image classification and nutrient identification: A systematic review. Reviews in Endocrine and Metabolic Disorders, 24(4):633-653, 2023.https://doi.org/10.1007/s11154-023-09795-4

[2] Rushit Dave, and Joy Purohit. Leveraging deep learning techniques to obtain efficacious segmentation results. Archives of Advanced Engineering Science, 1(1):11-26, 2023.https://doi.org/10.47852/bonviewAAES32021220

[3] Muthukrishnan Ramprasath, M.Vijay Anand, and Shanmugasundaram Hariharan. Image classification using convolutional neural networks. International Journal of Mechanical Engineering Research and Technology, 16(2):173-181, 2024.

[4] Shiwei Liu, Liejun Wang, and Wenwen Yue. An efficient medical image classification network based on multi-branch CNN, token grouping Transformer and mixer MLP. Applied Soft Computing, 153(4):111323-111342, 2024.https://doi.org/10.1016/j.asoc.2024.111323

[5] Mengxuan Zhang, Long Liu, Yaochu Jin, Zhikun Lei, Zhigang Wang, and Licheng Jiao. Tree-shaped multiobjective evolutionary CNN for hyperspectral image classification. Applied Soft Computing, 152(3):111176-111189, 2024.https://doi.org/10.1016/j.asoc.2023.111176

[6] Oleh Berezsky, Petro Liashchynskyi, Oleh Pitsun, and Ivan Izonin. Synthesis of convolutional neural network architectures for biomedical image classification. Biomedical Signal Processing and Control, 95(3):106325-106339, 2024.https://doi.org/10.1016/j.bspc.2024.106325

[7] Xin Wu, Yue Feng, Hong Xu, Zhuosheng Lin, Tao Chen, Shengke Li, Shihan Qiu, Qichao Liu, Yuangang Ma, and Shuangsheng Zhang, CTransCNN: Combining transformer and CNN in multilabel medical image classification. Knowledge-Based Systems, 281(2):111030-111048, 2023.https://doi.org/10.1016/j.knosys.2023.111030

[8] Mohammed Q. Alkhatib, Mina Al-Saad, Nour Aburaed, Saeed Almansoori, Jaime Zabalza, Stephen Marshall, and Hussain Al Ahmad. Tri-CNN: A three branch model for hyperspectral image classification. Remote Sensing, 15(2):316-335, 2023.https://doi.org/10.3390/rs15020316

[9] Qi Han, Xin Qian, Hongxiang Xu, Kepeng Wu, Lun Meng, Zicheng Qiu, Tengfei Weng, Baoping Zhou, and Xianqiang Gao. DM-CNN: Dynamic multi-scale convolutional neural network with uncertainty quantification for medical image classification. Computers in Biology and Medicine, 168(6):107758-107775, 2024.https://doi.org/10.1016/j.compbiomed.2023.107758

[10] Xiangzuo Huo, Gang Sun, Shengwei Tian, Yan Wang, Long Yu, Jun Long, Wendong Zhang, and Aolun Li. HiFuse: Hierarchical multi-scale feature fusion network for medical image classification. Biomedical Signal Processing and Control, 87(5):105534-105549, 2024.https://doi.org/10.1016/j.bspc.2023.105534

[11] Yang Yu, Yi Zhang, Zeyu Cheng, Zhe Song, and Chengkai Tang. Multi-scale spatial pyramid attention mechanism for image recognition: An effective approach. Engineering Applications of Artificial Intelligence, 133:108261,

2024.https://doi.org/10.1016/j.engappai.2024.108261

[12] Leyuan Fang, Yifan Jiang, Yinglong Yan, Jun Yue,and Yue Deng. Hyperspectral image instance segmentation using spectral–spatial feature pyramid network. IEEE Transactions on Geoscience and Remote Sensing, 61(4):1-13, 2023.https://doi.org/10.1109/TGRS.2023.3240481

[13] Yu Liu, Parma Nand, Md Akbar Hossain, Minh Nguyen, and Weiqi Yan. Sign language recognition from digital videos using feature pyramid network with detection transformer. Multimedia Tools and Applications, 82(14):21673-21685, 2023.https://doi.org/10.1007/s11042-023-14646-0

[14] Yi Zhang, Yushuang Zhu, Xiongwei Liu, Yingjian Lu, Chan Liu, Xixin Zhou, and Wei Fan. In-field tobacco leaf maturity detection with an enhanced MobileNetV1: Incorporating a feature pyramid network and attention mechanism. Sensors, 23(13):5964-5979, 2023.https://doi.org/10.3390/s23135964

[15] Yonglin Yu, Haifeng Li, Hanrong Shi, Lin Li, and Jun Xiao. Question-guided feature pyramid network for medical visual question answering. Expert Systems with Applications, 214(5):119148-119169, 2023.https://doi.org/10.1016/j.eswa.2022.119148

[16] Xiaoxia Meng, Xiaowei Wang, Shoulin Yin, and Hang Li. Few-shot image classification algorithm based on attention mechanism and weight fusion. Journal of Engineering and Applied Science, 70(1):14-31, 2023.https://doi.org/10.1186/s44147-023-00186-9

[17] Zia UrRehman, Yan Qiang, Long Wang, Yiwei Shi, Qianqian Yang, Saeed Ullah Khattak, Rukhma Aftab, and Juanjuan Zhao. Effective lung nodule detection using deep CNN with dual attention mechanisms. Scientific Reports, 14(1):3934-3949, 2024.https://doi.org/10.1038/s41598-024-51833-x

[18] Jia Liang, Xingyu Gu, Dong Jiang, and Qipeng Zhang. CNN-based network with multi-scale context feature and attention mechanism for automatic pavement crack segmentation. Automation in Construction, 164(5):105482-105499, 2024.https://doi.org/10.1016/j.autcon.2024.105482

[19] Feng Yu, Qian Zhang, Jun Xiao, Yuntao Ma, Ming Wang, Rupeng Luan, Xin Liu, Yang Ping, Ying Nie, Zhenyu Tao, and Hui Zhang. Progress in the application of cnn-based image classification and recognition in whole crop growth cycles. Remote Sensing, 15(12):2988-3010, 2023.https://doi.org/10.3390/rs15122988

[20] Xingying Zhou, Tianyu Yu, Guangzhou Wang, Ruiyang Guo, Yanxu Fu, Yazhou Sun, and Mingjun Chen. Tool wear classification based on convolutional neural network and time series images during high precision turning of copper. Wear, 522(2):204692-204716, 2023.https://doi.org/10.1016/j.wear.2023.204692

Improved Counterfactual Regret Minimization with Time-Series Differential Learning for Incomplete Information Games

Min Zhang, Shuchang Zhang*
Academic Affairs Office, Cangzhou Technical College, Cangzhou 061000, China
E-mail: zhangmin800129@126.com, zhangshuchang2025@163.com
*Corresponding author

Keywords: CFR, incomplete information game, regret value, time-series differential learning

Received: February 17, 2025

The traditional strategy recommendation algorithm for incomplete information game problems has low computational efficiency and insufficient quality of recommendation strategies. Therefore, the Counterfactual Regret Minimization (CFR) algorithm is designed, which introduces time-series differential learning to solve incomplete information game problems to adjust strategies faster, reduce oscillations in the strategy update process, and accelerate convergence speed. Combined with the decision judgment model biased towards opponent information, it is improved by updating the feature vectors in real time, which dynamically adjusts the strategy to adapt to changes in opponent strategy, thus obtaining an improved CFR algorithm. The study used data collected from the Texas Hold'em Robot Contest organized by the International Association for Artificial Intelligence from 2010 to 2016 for testing. The experimental results showed that after 20,000 games, the average return of ICFR-OG was 3.18, significantly higher than that of other mainstream algorithms, namely VGG32, Faster RCNN, CFR, and XGBoost, with average returns of -1.73, 0.24, 0.69, and 2.35, respectively. The cumulative calculation time of the research method was only 1,967ms. ICFR-OG demonstrated the lowest computational time, while CFR exhibited the highest. The results are useful for improving the performance of Texas Hold'em educational games and improving the ability to deal with various incomplete information games.

Povzetek: Razvita je nova metoda za igranje nepopolnih informacijskih iger kot Texas Hold'em. Izboljšan CFR kombinira TD-podobno posodabljanje in pristransko, sprotno modeliranje nasprotnikov (VPIP/PFR/3-bet + K-means) za hitrejšo konvergenco in višji EV.

1 Introduction

Incomplete information game problem is an important branch of game theory, which considers decision-making and competition among participants with incomplete information [1-2]. Incomplete information game problems have various applications in real life, such as games, strategic decisions, auctions, etc [3]. With the development of computer networks, automated algorithms for computing various types of incomplete information game problems have gradually become a research hotspot. Texas Hold'em is a typical incomplete information game problem. How to solve such problems effectively has become the focus of scientists and engineers [4]. The virtual regret minimization algorithm (Counterfactual Regret Minimization, CFR) is an effective algorithm for solving incomplete information game problems, which solves the shortcomings of traditional strategy search algorithms that need to search for complete information games, and has convergence and good accuracy [5]. However, in practice, the CFR algorithm has some defects and shortcomings, such as local convergence or strategy bias when constructing strategies, which makes the algorithm performance not optimal [6]. The research questions are as follows. Traditional strategy recommendation algorithms for incomplete information game problems have low computational efficiency and are difficult to provide real-time strategy recommendations. It cannot effectively capture the opponent's strategies and behavioral patterns, resulting in low recommendation quality. It fails to fully utilize the opponent's historical behavior data to optimize strategy recommendations. By incorporating the time-series differential learning, it is expected that the policy analysis capability and computational efficiency of the algorithm can be improved. The research also designs a decision judgment model that favors adversary information, and classifies adversaries by extracting their characteristic information. Therefore, the improved CFR algorithm can better capture the core information of adversaries to build response strategies. By combining the improved CFR algorithm and the decision judgment model with biased opponent information, a strategy recommendation algorithm for the Texas Hold'em incomplete information game problem is proposed in the study. Simulation experiments based on real and randomly generated data are also designed to validate the effectiveness of the hybrid algorithm application designed in the study. The research objective is to improve the efficiency and strategic quality of solving incomplete information game problems. By extracting the opponent's feature information and constructing a classification model of the opponent, it is possible to more accurately capture the opponent's strategies and behavior

patterns. The main contribution of this research is to integrate the idea of time-series differential learning into the CFR algorithm, designing an improved CFR algorithm and a decision judgment model with biased opponent information.

2 Related works

Noguchi M et al. found that the scientificity of the description model for the incomplete information game problem was extremely important for finding a good solution algorithm. They took a nonlinear programming algorithm and a neural network algorithm to construct a mathematical model for describing the incomplete information game transportation problem. The authors used nonlinear programming algorithm and neural network algorithm to construct a mathematical model for describing the transportation problem of incomplete information game and designed an improved convolutional neural network to find the best solution. The test results showed that the solution computed by solving the mathematical model designed in this study had better quality when using the same seeking algorithm [7]. Alcantara-Jimnez G proposed a practical solution algorithm for the Stackelberg security game problem that took into account incomplete state information and incorporated a stochastic strategy for partially observed Markov games. The results showed that the algorithm outputted a response strategy for the Stackelberg security game problem that was more applicable than traditional machine learning algorithms [8]. Wang Q et al. found that emergency management systems faced the real-time data analysis due to the emergency and unpredictable nature of disaster relief. The complete information was not available from emergency communication networks. Therefore, the author proposed a two-layer game model based on incomplete information to achieve collaborative computing on the edge. In addition, a near-optimal CGR algorithm was also developed. Simulation results showed that the designed algorithm outperformed existing incomplete information-based solutions on computational latency and participant utility [9]. Lehrer et al. investigated infinite repeated zero sum games with incomplete information, where the game state evolved according to a stationary process. This led to consistent values in the Kronecker system. Techniques from traversal theory, probability theory, and game theory were taken to describe the optimal strategy of two participants [10]. Lin Z et al. proposed a multi-layer interconnected time model that considered multiple empirical stopping games with optimal timing decisions and incomplete information. The unique Bayesian Nash equilibrium of the stopping game was characterized in the model by a system of equations containing the conditional distribution for each duration, which satisfied the moderate strategy interaction condition. Comparative experiments were also conducted in the study, in which the same game problem solving algorithm was used to solve the designed model as well as the unmodified traditional model. The experimental results showed that the model designed in this study improved the solving algorithm and outputted better solutions [11]. Based on the above literature summary, Table 1 is compiled.

Based on the above content, it can be concluded that although numerous achievements have been made in solving incomplete information game problems, existing methods still have the following shortcomings, including insufficient ability to dynamically capture opponent behavior patterns, underutilization of historical data, low computational efficiency, low strategy quality, and inability to dynamically adjust. Therefore, the study designs the ICFR-OG method, which accelerates convergence through time-series differential learning and combines K-Means classification and probability threshold analysis to enhance the opponent's strategy capture ability.

Table 1: Summary of literature results

Author	Research method	Research results	Advantage	Limitation
Noguchi M	Constructing a mathematical model for incomplete information game transportation problems using nonlinear programming algorithms and neural network algorithms	Under the same solving algorithm conditions, the model outputs higher quality solutions	The model description is highly scientific and the solution quality is high	Not optimized for dynamic game scenarios such as Texas Hold'em, resulting in high computational complexity
Alcantara-Jimnez G	Solving Stackelberg security game problems using stochastic strategies combined with partially observable Markov games	The output response strategy is more applicable than traditional machine learning algorithms	Suitable for dynamic security games, with strong strategic adaptability	Insufficient utilization of opponent's historical behavior data and insufficient real-time performance
Wang Q et al.	Designing a near optimal CGR algorithm based on a two-layer game model with incomplete information	Superior to existing incomplete information solutions in terms of computational latency and participant utility	It is suitable for edge computing scenarios with good real-time performance	Insufficient capture of opponent behavior patterns and limited quality of strategy recommendations
Lehrer et al.	Research on infinite repeated zero sum games under incomplete information, using traversal theory and probability theory to describe optimal strategies	There is a consistent value in the Kronecker system, and the strategy stability is high	Rigorous in theory, suitable for long-term games	Not involving dynamic strategy adjustment, low computational efficiency

Lin Z et al.	Multi-layer interconnected time model considering optimal stopping game under incomplete information	The algorithm designed has a higher probability of outputting better solutions	Support multi-stage decision-making and high model flexibility	Not optimized for real-time gaming, limited ability to classify opponents

3 Analytical model for incomplete information game and improved cfr algorithm design

3.1 Design of incomplete information game algorithm based on improved virtual regret minimization

The CFR algorithm is currently the main intelligent method for analyzing non-complete information game problems in academia. It is an adapted form of the virtual regret minimization algorithm in such game problems [12-14]. However, the CFR algorithm has disadvantages such as low computational efficiency and the computational results can be further optimized. Therefore, based on the analysis of the CFR algorithm, an improved CFR algorithm incorporating time-series differential learning is designed [15-16]. The regret minimization algorithm calculates the regret value as close to infinity as possible through multiple iterations to obtain an execution strategy that is closer to the Nash equilibrium. The core of the algorithm can be divided into calculating the regret value and matching the regret value. The process of calculating the regret value is first analyzed, which has a great impact on the decision-making for the next game [17-19]. Assuming that the payoff function μ_i, the best decision payoff at step t in the game is $\mu_i\left(\sigma_i^{'},\sigma_{-i}^{t}\right)$, $\mu_i\left(\sigma^{t}\right)$ is the payoff obtained from the actual decision. $\sigma_i^{'}$ and σ_{-i}^{t} are the participant i 's own strategy and the opponent's strategy at step t, respectively. The set of strategies of the participants $\left\{\delta^1,\delta^2,...,\delta^T\right\}$ corresponds to the regret value R_i^t can be calculated according to equation (1).

Assuming that there is a payoff function AA, the optimal decision payoff for step BB in the game is CC. DD is the actual payoff obtained from the decision. EE and FF are the participant GG's own strategy and the opponent's strategy in step HH, respectively. Therefore, the regret value JJ corresponding to the participant's strategy set II can be calculated according to equation (1).

$$R_i^t=\max\sum_{t=1}^{T}\left(\mu_i\left(\sigma_i^{'},\sigma_{-i}^{t}\right)-\mu_i\left(\sigma^{t}\right)\right) \quad (1)$$

In equation (1), T represents the highest number of game steps in the game. R_i^t represents the gain of the optimal strategy for the participant i at all time steps. The algorithmic regret value $T\to\infty$ is considered to be minimized when one of the strategies in the game achieves the rule in equation (2).

$$\frac{R_i^{T,x^+}}{T}\to 0,\left(x^+=\max\left\{x,0\right\}\right) \quad (2)$$

In equation (2), x^+ is the total number of participants in the game at the corresponding number of steps. To design the regret value matching calculation step again, a random strategy δ^l is first obtained. For each action a in the set of the optional action A_i, the regret value can be calculated according to equation (3).

$$R_i^T(a)=\sum_{t=1}^{T}\left(\mu_i\left(a,\sigma_{-i}^{t}\right)-\mu_i\left(\sigma_i^{t},\sigma_{-i}^{t}\right)\right) \quad (3)$$

In equation (3), $\mu_i\left(a,\sigma_{-i}^{t}\right)$ means the gain obtained by the participant action a after retracing the game information. $\mu_i\left(\sigma_i^{t},\sigma_{-i}^{t}\right)$ means the post-decision gain that actually corresponds to the number of steps. The strategy for the subsequent steps $\sigma_i^{t+1}(a)$ is calculated according to equation (4), which is used for regret matching.

$$\sigma_i^{t+1}(a)=\frac{R_i^{T,+}(a)}{\sum_{b\in A_i}R_i^{T,+}(b)} \quad (4)$$

In equation (4), b is also an action in A_i. When the denominator in equation (4) is 0, the next action is obtained in a random way. The calculation flow of the regret minimization algorithm can be expressed, as shown in Figure 1.

The CFR algorithm is formed on the basis of the regret minimization algorithm. The main difference between the former and the latter is that the former calculates the virtual reach probability of the strategy according to the information set, and considers both the virtual reach probability and the regret value when selecting the strategy.

To address the shortcomings of the CFR algorithm, an improved CFR algorithm is designed based on time-series differential learning. Time-series differential learning is a sub-method of reinforcement learning that combines the Monte Carlo algorithm and the dynamic programming algorithm. The purpose of introducing the time-series differential algorithm into CFR is to accelerate the speed at which the CFR algorithm returns to the policy and converges. Before designing the improved CFR algorithm, it is necessary to abstract the Texas Hold'em game and the incomplete information game problem selected for this study, because the state space complexity of the Texas Hold'em game is too high. This is the part that the improved CFR algorithm fails to handle optimally. The mainstream Sklansky hand strength quantization value is chosen to simplify the Texas Hold'em game process. The number of hand combinations at the beginning of the Texas Hold'em game is as high as $C(2,52)=1326$. Therefore, to reduce the number of combinations, only the number of hand points and suits in two hands will be considered, which will reduce the number of research combinations to 169. The simplified 169 hand combination has a quantified value for Sklansky strength. Therefore, the classification of Texas Hold'em

hand strength based on the Sklansky hand strength quantification value can be obtained, as shown in Table 2.

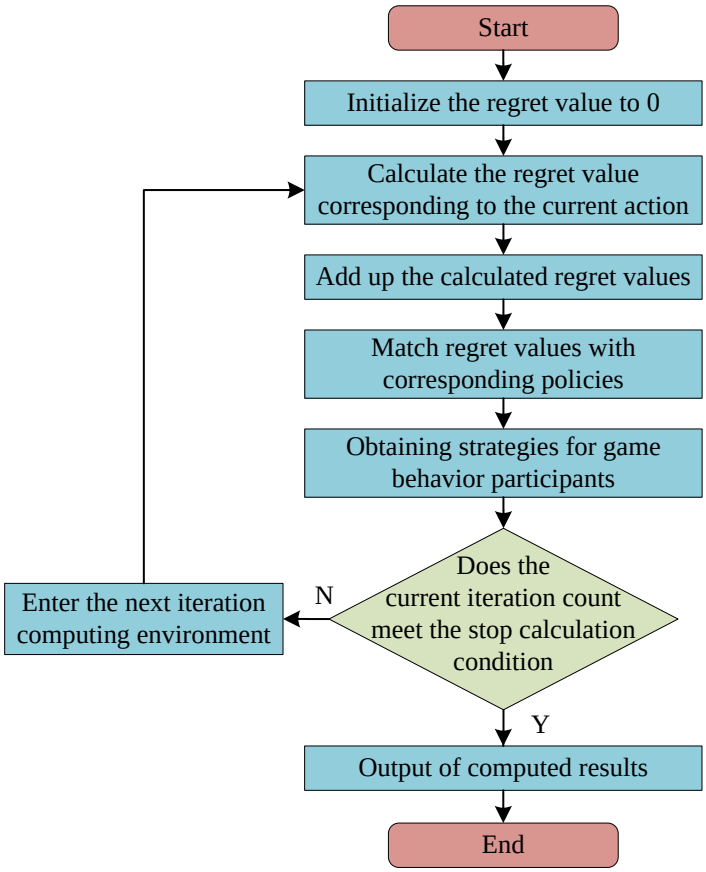


Figure 1: Calculation process of regret minimization algorithm

Table 2: Texas poker hand strength classification based on Sklansky hand strength quantification values.

Class	Hand card	Class	Hand card
#1	KK, AKs, AA, QQ, JJ	#6	QT, KT, AT, J8s, 86s, 75s, 65s, 55, 54s
#2	AJs, AQs, AKs, KQs, TT	#7	t9, j9, k9s-k2s, 98, 64s, 53s, 44, 43s, 33, 22
#3	KJs, QJs, AQs, ATs, JTs, 99	#8	Q9, A9, K9, J7s, 96s, T8, 85s, 87, 74s, 76, 65, 54, 42s, 32s
#4	QTs, KTs, KQs, AJs, J9s, T9s, 98s, 88	#9	Other hand combinations
#5	KJ, QJ, JT, A9s-A2s, Q9s, T8s, 97s, 87s, 77, 76s, 66	/	/

Note: "o" represents different colors, such as A ♠ and K ♥. "s" represents the same color, such as A ♣ and K ♣.

There are four phases in Texas Hold'em, including flop, preflop, turn, and river. The final hand in preflop is divided into two types: absolute and potential, depending on the combination of hands. The number of opponents is n. Then, the absolute hand power HS_n can be calculated according to equation (5).

$$HS_n = \left(HS_1\right)^n \tag{5}$$

In equation (5), HS_1 represents the absolute strength of the hand when the number of opponents is 1. Considering the process and characteristics of Texas Hold'em, the potential strength can be calculated according to equation (6).

$$P\left(win\right) = \begin{cases} 1\% * 2n, turn \text{ stage} \\ 1\% * (3n+8), \ flop \ stage \end{cases} \tag{6}$$

In equation (6), $P\left(win\right)$ describes the final win rate and n is the number of cards needed for a reversal to occur.

After modeling the Texas Hold'em game process, the improved CFR algorithm is started to improve the ability and efficiency of the algorithm to handle the game problem online. Before calculating the improved CFR algorithm, the game space needs to be simplified, and the method used for the simplification is the undercard abstraction technique. In the improved CFR algorithm, the first step is to input the virtual values of each information set and the predicted regret values of each decision in the game tree based on the opponent's strategy or expert experience data. After the game starts, the algorithm continues to calculate and output the virtual regret values and the corresponding actual virtual values based on the current game results, which will be used to replace the

virtual regret values or virtual value indicators of the corresponding node. As shown in Figure 2, the improved CFR algorithm aims to form initial regret values for each node in the corresponding game tree based on the opponent's offline game data, and adjust the current game strategy on this basis.

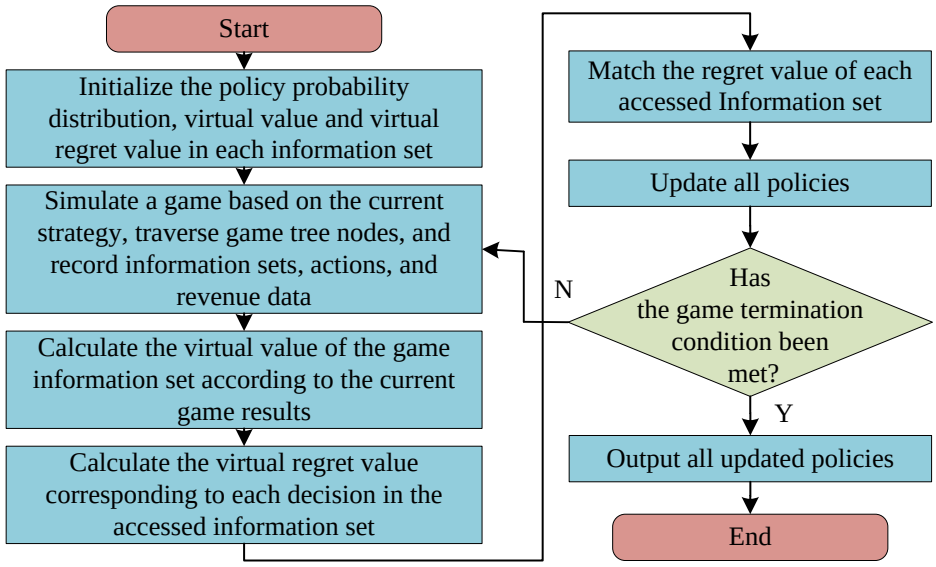


Figure 2: Calculation process of improved CFR algorithm

Table 3: Opponent classification method based on opponent feature data and closing range.

Type number	Type-name	**Pool frequency/%**	Prefop stage filling frequency/%	3-bet/%
#1	Lags	Not less than 75	Not less than 45	Not less than 25
#2	Flaccid type	Not less than 70	Not greater than 35	Not greater than 10
#3	Tight and fierce type	Not greater than 45	Not less than 35	Not less than 15
#4	Compact weak type	Not greater than 40	Not greater than 20	Not greater than 5

3.2 Machine game model building with biased adversary strategy

Although traditional intelligent game algorithms based on Nash equilibrium can generate theoretically robust strategies, their static nature makes it impossible to dynamically adjust based on opponent behavior patterns. For example, in Texas Hold'em, if the opponent frequently abandons their cards, the equilibrium strategy will still raise with a fixed probability, missing out on opportunities for exploitation. For incomplete information game problems such as Texas Hold'em, considering the maximization of the opponent's game gain and sequentially finding a better strategy for the base point may lead to better game advantages. Therefore, an improved machine game model with a bias towards the opponent's strategy is designed, which will be applied to the incomplete information game problem together with the improved CFR algorithm.

The traditional adversary classification method basically measures the adversary's aggressiveness according to the adversary's strategy type and strategy frequency, and classify the adversary into conservative, conventional, and aggressive types. However, this method is not specific enough in dynamic machine games, and there is a risk of being identified and exploited by the adversary. Therefore, a method to classify gaming styles is proposed according to the range of opponent characteristics data and returns. A three-dimensional feature vector is now constructed according to the entry frequency, preflop stage raise frequency, and three-bet indicator. Combined with expert experience, an opponent classification method is designed for the two-player infinite bet Texas Hold'em gaming problem, as shown in Table 3. The opponent classification method proposed in the research has strong interpretability. Its classification method based on simple statistical features has low computational complexity. It can process large-scale competition data in real time and meet the low latency for online games. However, reinforcement learning methods require a large amount of data, have high training costs, and are difficult to analyze the policy decision-making process. Online learning may lead to policy oscillations. The selection of the time window in the dynamic feature method is subjective. Increasing the feature dimension will significantly increase the computational cost, conflict with the research objective of this study, and may introduce overfitting risks.

After classifying the adversaries in this way, to output better response strategies in dynamic games, it is also necessary to learn the historical strategy information of the adversaries. The game strategies in incomplete information game problems are often dynamic and changing. It is more appropriate to use unsupervised learning algorithms to learn historical data. The K-Means algorithm is used to handle the task of learning historical data of opponents, as it is a typical unsupervised algorithm

with simple and good learning performance. Then, it is possible to build a decision model biased toward the adversary based on the laws of historical behavioral data found by adversary classification and clustering, thus calculating each alternative decision in different decision stages and building a game model according to the adversary model. Since the pooling frequency and preflop stage raising frequency can only reflect the opponent strategy law in preflop stage, to further improve the game output quality, it is also necessary to evaluate the decision probability distribution of different types of opponents in flop, turn, and river stages. The betting frequency usually refers to the frequency at which players enter the bottom pool, which measures the proportion of times players choose to participate in the bottom pool (i.e. give up without betting) in each round of the game to the total number of games played. This indicator can help analyze the player's strategic style and level of aggressiveness. Therefore, an array of probabilities $[\alpha, \beta]$ is designed as a threshold for analyzing the opponent's decisions in the later stages. Thus, it is known that the opponent continues the current game only when the minimum win rate is α. Otherwise, it abandons the game. The opponent will not bet or raise until the minimum win rate is at least β. Based on the $[\alpha, \beta]$ array and its rules, the opponent's decision model can be constructed, as shown in Figure 3.

In actual games, the opponent's strategy may change as the game progresses. Therefore, the feature vector needs to be dynamically updated based on the opponent's real-time behavior to ensure that the model can accurately reflect the opponent's current strategy. In each round of the game, the system will extract real-time features such as the opponent's pool entry frequency, prefop stage filling frequency, and 3-bet indicator, and update the feature vector. The real-time extraction of these features can be achieved through statistical analysis of opponent behavior. Based on the updated feature vectors, the system will reclassify the opponent and adjust its own strategy according to the opponent type. This dynamic classification and strategy adjustment can enable the system to better adapt to the opponent's strategy changes, thereby improving the overall performance of the game. For example, if an opponent behaves abnormally aggressively in a certain round of the game, the system may adjust its strategy and take more conservative actions to avoid unnecessary risks. In addition to opponent classification, the decision model dynamically adjusts the decision threshold and strategy generation logic based on the opponent's historical behavior data and current behavior patterns. For example, if an opponent exhibits unusually aggressive behavior during a certain round of the game, the system may adjust its strategy and take more conservative actions to avoid unnecessary risks. Specifically, the decision model dynamically adjusts the decision threshold based on the opponent's feature vectors and historical behavior data, such as the minimum win rate threshold and minimum markup threshold to generate better strategies. Then, the expectation assessment of the decision is designed according to the theory of logic. The expectation of the action is an important basis for measuring the correctness of the decision. For the Texas Hold'em problem in incomplete information game, its total expectation $Ev(c)$ can be calculated according to equation (7).

$$Ev(c) = \sum_{1 \le i \le N} P(C_i) \times Ev(C_i) \tag{7}$$

In equation (7), N represents the total number of branches of the generated game tree, $P(C_i)$ represents the arrival probability of the branch node i, and $Ev(C_i)$ is the expected value of the branch node decision. Considering that the expectation of 0 when the opponent adopts folding behavior is reasonable, the expectation of call behavior Ev_{call} can be calculated according to equation (8).

$$Ev_{call} = P_{win} \cdot pot - P_{lose} \cdot \cos t \tag{8}$$

In equation (8), P_{lose} and P_{win} represent the probability of losing and winning, respectively. pot and $\cos t$ represent the amount of the current poker pool and the amount to be called, respectively. In the raising environment, the behavioral expectation of the opponent folding probability is also taken into account, so the behavioral expectation of raising can be calculated according to equation (9).

$$\begin{aligned} Ev_{raise} &= [P_{win} + P_{fold}] \times pot \\ &+ [P_{win} + P_{lose}] \cdot raise \end{aligned} \tag{9}$$

In equation (9), P_{fold} describes the opponent's fold probability and $raise$ represents the number of raises. The proposed method for solving the incomplete information game problem in Texas Hold'em is designed. The operational framework is shown in Figure 4. The next step is to select the optimal strategy based on the opponent's judgment model.

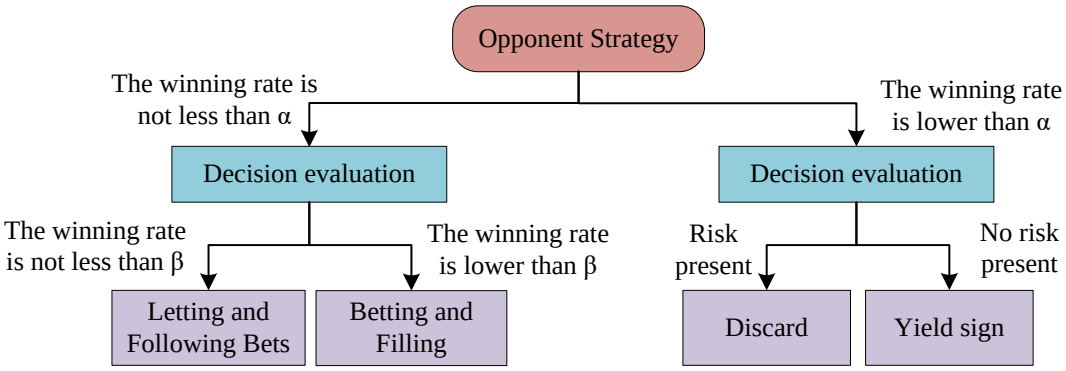


Figure 3: Opponent decision judgment model based on probability binary

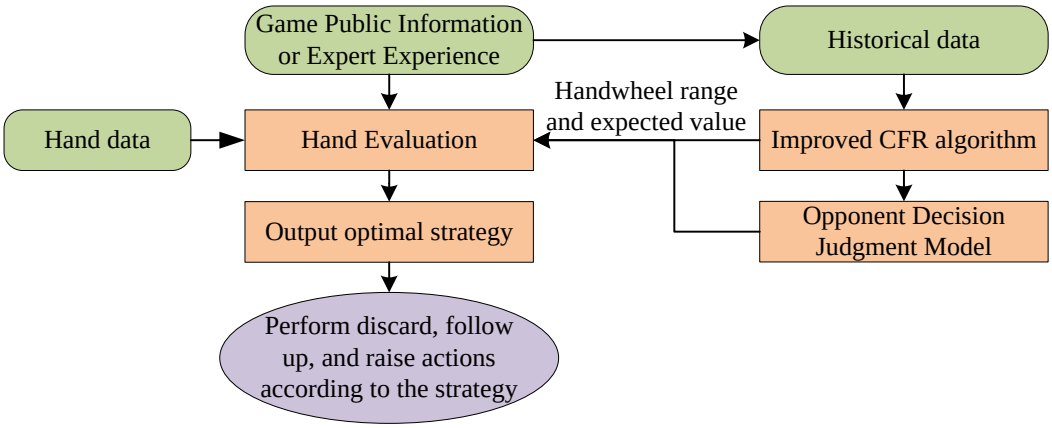


Figure 4: Solution framework for incomplete information game problems based on improved CFR adversary game model

Table 4: World poker machine game competition data statistics results.

Number	Time	Total number of files	Total file size (MB)	Total number of competition data (×103)	Total number of matches per round (×103)
#1	2016	4,855	452	14,564	3.0
#2	2014	8,008	748	24,024	3.0
#3	2013	6,612	619	19,836	3.0
#4	2012	5,800	524	9,000	3.0
#5	2011	5,400	386	16,200	3.0
#6	2010	4,624	391	13,872	3.0

4 Test of the solution of the incomplete information game problem based on the improved cfr and adversary game model

4.1 Experimental protocol design and data set selection

An experiment is designed to validate the performance of the Texas Hold'em poker game strategy model, which utilized an improved CFR and opponent game model designed for this study. The data in the experiment are obtained from the information recorded in each World Poker Machine Gaming Competition held by the International Artificial Intelligence Association from 2010 to 2016. Before conducting testing, it is necessary to preprocess the data in the dataset, starting with data cleaning to remove duplicate records, erroneous data, and incomplete data items, which ensures the data quality and consistency. Next, the data is normalized and features related to Texas Hold'em game strategies are extracted from the raw data, such as player pool frequency, top up frequency, etc. These features are used for subsequent opponent classification and strategy recommendations. Finally, the dataset is divided into a training set and a testing set in a 7:3 ratio. The information statistics of the dataset are shown in Table 4.

The Texas Hold'em poker games are affected by the luck of the participants. To minimize the influence of luck on the experimental results, various replicated experiments are designed for this study due to the randomness of the effect of luck on game outcomes. In the improved CFR algorithm comparison experiments, there was no need to consider the opponent type and opponent strategy characteristics. An automated program whose

decision behavior showed a random distribution pattern was directly used as the opponent. In the test experiments based on the improved CFR and adversary game model, four types of adversaries were labeled by the K-Means algorithm to carry out calculations. In the integrated game experiments between various types of algorithms, the intelligent program with decision making according to the equal concept distribution is used as the adversary because the weaknesses of each strategy are not consistent. In addition, to avoid the influence of irrelevant variables on the experimental results, the public hand and other hands are all dealt according to the same rule in each experiment. The hyperparameters of the algorithms that require setting hyperparameters, such as K-Means algorithm and improved CFR algorithm in the experiments, are determined in accordance with the industry experience combined with multiple debugging methods. The regret value, average gain, and computation time consumption are chosen as the performance evaluation indexes. The average gain is calculated by dividing the number of current winnings and losings by the number of current hand games. The parameter settings for time-series differential learning are as follows. The learning rate controls the regret value update step size, set to 0.05. The weight of the current and future regret values is balanced by a deduction factor, taken as 0.9. The randomness of exploration rate maintenance strategy optimization is fixed at 0.1. The time window size determines the historical step size for differential calculation. After testing, it has been set to 100 hands to have the best results. Regret smoothing is performed using an exponential weighted average with a decay factor of 0.7. These parameters work together in three key stages. When calculating regret differences, the discount factor adjusts the weight of historical averages and immediate changes. When updating the strategy, it ensures the minimum exploration probability. The attenuation factor controls the strength of noise filtering.

4.2 Analysis of experimental results

Firstly, the experimental results of the improved CFR algorithm with fused time-series differential proposed in this study and the traditional CFR algorithm are analyzed separately. The statistics are shown in Figure 5. Different subplots in Figure 5 represent different algorithms, and the horizontal and vertical axes of the two subplots represent the number of iterations and the regret value, respectively. Since the number of iterations required for the algorithm to complete the training is large, the horizontal axis is shown in exponential form, and "ICFR" represents the improved CFR algorithm with fused time-series differential. The ICFR algorithm and the CFR algorithm are considered to have completed the training with the regret value less than 0.0001 at 76,522 iterations and 284,562 iterations, respectively. It can be seen that the ICFR algorithm designed in this study can complete the training faster. The above results are due to the combination of the advantages of Monte Carlo algorithm and dynamic programming algorithm in time-series differential learning. It uses a differential learning mechanism to adjust policies more quickly and reduce oscillations during policy updates, which enables the algorithm to converge to the optimal policy faster during the training phase.

The average gain of the improved CFR algorithm and the traditional CFR algorithm in the test experiment stage after the training is completed is shown in Figure 6. In Figure 6, with the growth of the number of games, the average gain of both decision algorithms showed a fluctuating upward trend, and the average gain of the ICFR algorithm was always higher than that of the traditional CFR algorithm. Overall, under the same conditions, the average gain of the former was about 0.26 higher than that of the latter. This is because the research designs a decision judgment model that can dynamically adjust strategies based on the opponent's historical behavior data. This model not only considers the opponent's strategy type, but also adapts to changes in the opponent's strategy by updating feature vectors in real-time. This multi-dimensional feature vector enables ICFR-OG to more accurately classify opponent types and adjust strategies based on opponent types.

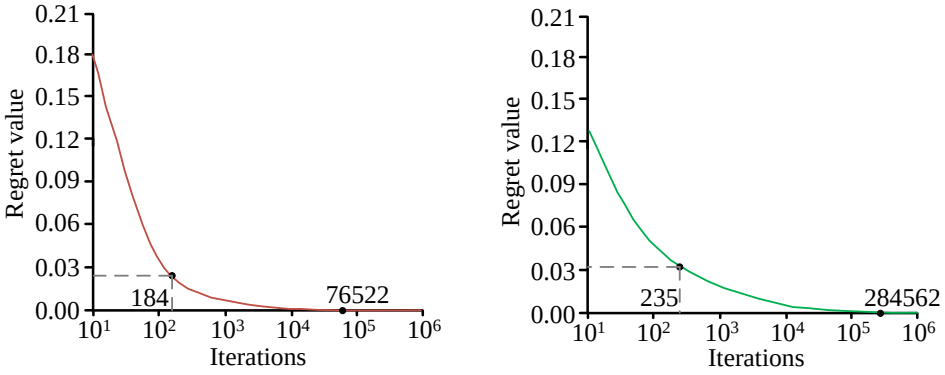


Figure 5: Comparison of regret values between the improved CFR algorithm and the traditional CFR algorithm during the training phase

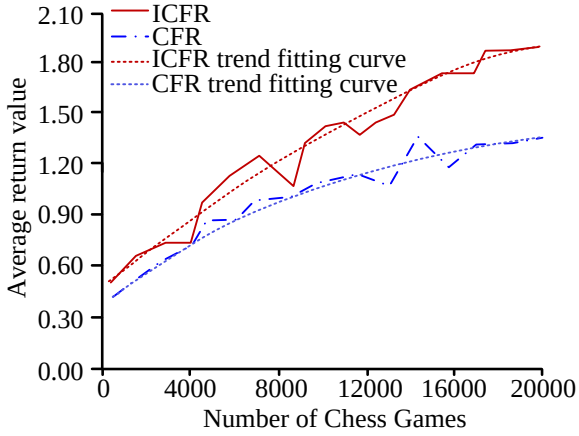
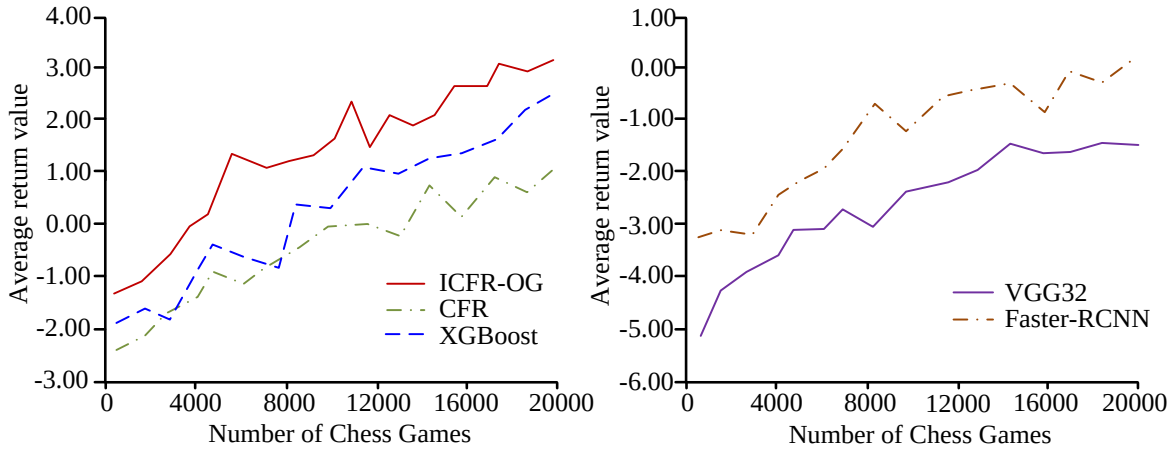


Figure 6: Comparison of average returns during the testing phase between the improved CFR algorithm and the traditional CFR algorithm



(a) ICFR-OG and Machine Learning Algorithms (b) Deep learning algorithm

Figure 7: Comparison of average game returns between ICFR-OG algorithm and comparative algorithm

Table 5: Comparison of average returns for each algorithm in a one-on-one game.

Chess algorithm	VGG32	Faster-RCNN	CFR	XGBoost	ICFR-OG
VGG32	-	1.58	5.73	3.96	5.16
Faster-RCNN	1.58	-	4.18	2.48	4.25
CFR	5.73	4.18	-	0.15	2.51
XGBoost	3.96	2.48	0.15	-	1.77
ICFR-OG	5.16	4.25	2.51	1.77	-

The VGG32 algorithm and Faster-RCNN algorithm in deep learning and CFR algorithm and XGBoost algorithm in machine learning are selected as the comparison algorithms. It is compared with the Integrated Decision Making with Improved CFR and Adversary Game Model (referred to as ICFR-OG) algorithm designed in this study, and the comparison results are shown in Figure 7. Because there are more comparative algorithms, the deep learning algorithm and other algorithms in Figure 7 are placed in subgraphs (a) and (b), respectively. The meanings of the horizontal and vertical axes in Figure 7 are consistent with Figure 6. In Figure 7, the ICFR-OG algorithm had a higher average gain than all the comparison algorithms for different number of games, while the deep learning type algorithm had the lowest average gain among the other algorithms, followed by the machine learning algorithm. Specifically, the average gains of VGG32, Faster-RCNN, CFR, XGBoost, and ICFR-OG algorithms were -1.73, 0.24, 0.69, 2.35, and 3.18, respectively, when the number of games played was 20,000.

The results among various types of algorithms are then analyzed, and the statistics are shown in Table 5. To reduce the influence of random factors on the experiment, the average gain statistics are counted every 1,000 times of the game. The average gain value of the ICFR-OG algorithm designed in this study was the highest for the neural network algorithm, because the neural network algorithm required high training data size, and the amount of data collected in this study was still insufficient. The difference was not obvious.

Finally, the speed of each algorithm is compared to compute the output strategy, and the statistical results are shown in Figure 8. To reduce the experimental workload,

the neural network matching algorithm with the lowest output game quality is excluded from this experiment. In Figure 8, the horizontal axis still represents the number of games, but the vertical axis represents the total computation time of the output strategy for the historical number of games in ms. Different icons are used to describe different algorithms, and the corresponding colored lines represent the linear fitting equation lines for the time-consuming data points of the algorithm. After comparing equations such as power functions, word count functions, polynomials of different orders, and linear equations, it was found that linear fitting equations had the best data fitting effect. Figure 8 showed that there was a significant linear correlation between the cumulative computational time consumed and the number of games played by CFR, XGBoost, and ICFR-OG. From the time consumption data, when the number of games was large, the ICFR-OG algorithm designed in this study had the shortest computation time and the CFR algorithm had the longest computation time. When the number of games reached 20,000, the cumulative computation time of CFR, XGBoost and ICFR-OG was 3,762ms, 3,198ms and 1,967ms, respectively. In addition, there was a significant linear correlation between the average computation time required to generate 1,000 strategies using CFR, XGBoost, and ICFR-OG algorithms and the number of games played. From the perspective of time consumption data, when there were a large number of game rounds, the ICFR-OG algorithm designed in this study had the shortest computation time, while the CFR algorithm had the longest computation time. When the number of game games reached 20,000, the average computation time required for CFR, XGBoost, and ICFR-OG to generate 1,000 strategies was 188.15ms, 159.95ms, and 98.35ms, respectively.

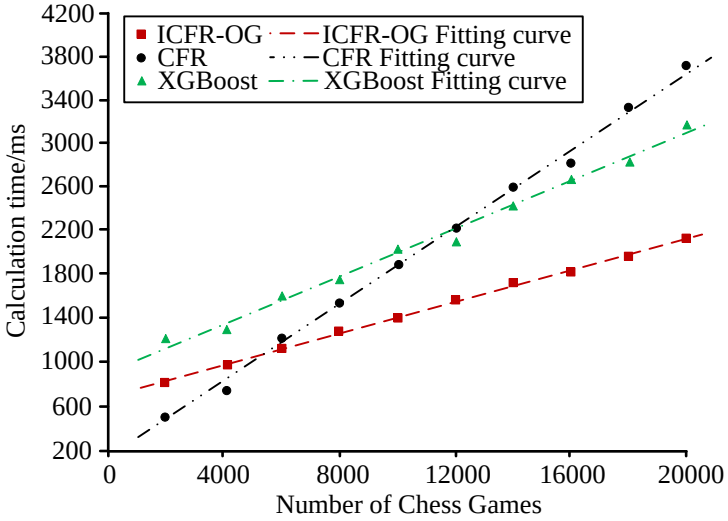


Figure 8: Comparison of the speed of calculating output strategies among different algorithms

5 Discussion

The proposed ICFR-OG method has demonstrated significant advantages in incomplete information games of Texas Hold'em. The performance improvement is attributed to various innovative designs. Compared with existing methods, ICFR-OG has significant improvements in algorithm efficiency, policy quality, and adaptability. Compared with the SOTA method with the best performance, CGR algorithm performs well in edge computing scenarios, and its computing latency is about 40% lower than that of traditional methods. However, due to its innovative regret value update mechanism and dynamic pruning strategy, ICFR-OG further reduces computation time to 65% of CGR through time-series differential learning. The CGR algorithm adopts a two-layer game model, with opponent modeling relatively static. In contrast, the 3D feature classification system in ICFR-OG can capture real-time changes in opponent strategies, and experimental data shows that its recognition accuracy for aggressive opponents is 28% higher than CGR.

Although ICFR-OG outperforms deep learning methods such as VGG32 and Faster RCNN in terms of average returns, this advantage is not only reflected in returns. Deep learning methods typically require a large amount of training data and computational resources to learn complex patterns and relationships. However, ICFR-OG achieves higher performance with fewer data and computing resources by combining time-series differential learning and a decision model biased towards opponent information. This indicates that ICFR-OG can more effectively utilize limited resources to optimize strategies when dealing with incomplete information game problems. In addition, the decision-making process of deep learning models is often difficult to explain, while ICFR-OG's decision-making model is based on clear opponent characteristics and historical behavior data, which has better interpretability. This interpretability is crucial for strategy adjustment and optimization in practical applications, especially in scenarios where understanding and predicting opponent behavior is necessary.

The significant advantage of ICFR-OG in computation time is mainly attributed to its optimized calculation method and strategy generation process. By simplifying the game space and using efficient clustering algorithms, ICFR-OG can significantly reduce

computational complexity while maintaining policy quality. However, this optimization is not without trade-offs. For example, simplifying the game space may result in the loss of certain complex strategies, thereby limiting the performance of the algorithm in certain specific scenarios.

6 Conclusion

This research addressed the poor computational real-time and poor quality of recommendation results in strategy intelligence algorithms for incomplete information game problems. An improved CFR algorithm incorporating the time-series differential learning and the decision model biased toward opponent information was proposed. Combining the two, a classic solution for solving incomplete information game problems, the ICFR-OG strategy output algorithm for Texas Hold'em poker games, was constructed. The experimental results showed that the ICFR algorithm and the CFR algorithm had regret values less than 0.0001 at 76,522 and 284,562 iterations, respectively, and were considered to complete the training. The average gain of the ICFR algorithm was always higher than that of the traditional CFR algorithm, and the average gain of the former was about 0.26 higher than that of the latter under the same conditions. When the number of games was 20,000, the average gains of VGG32, Faster-RCNN, CFR, XGBoost, and ICFR-OG algorithms were -1.73, 0.24, 0.69, 2.35, and 3.18, respectively. There was a significant linear correlation between the cumulative computation time of CFR, XGBoost and ICFR-OG and the number of games played. From the time consumption data, when the number of games was large, the ICFR-OG algorithm designed in this study had the shortest computation time and the CFR algorithm had the longest computation time. When the number of games reached 20,000, the cumulative computation time of CFR, XG Boost, and ICFR-OG were 3,762ms, 3,198ms, and 1,967ms, respectively. In summary, the research method has excellent performance and practicality, which can be extended to other fields, such as incomplete information game problems in financial high-frequency trading or auction markets, to help participants optimize bidding strategies or trading decisions. It can be used to design dynamic defense strategies and improve defense efficiency by classifying attacker behavior patterns and predicting their next actions. However, there are still certain limitations in the research. Firstly, the current opponent classification model is based on fixed features (such as pool frequency, injection frequency, etc.), which may not fully capture the strategic changes of opponents in dynamic games. Then, although the hand abstraction techniques of Texas Hold'em, such as Sklansky quantification, reduce computational complexity, they may lose some information, resulting in limited generalization ability of the strategy in complex scenarios. The final model performance is highly dependent on the quality and coverage of historical data. If the opponent type or game scenario exceeds the distribution of training data (such as rare strategies or extreme behaviors), the adaptability of the algorithm may be insufficient. Therefore, in future research, the model can be extended to other incomplete information game scenarios to verify its cross-domain applicability. Reinforcement learning or online learning mechanisms can be introduced to enable the opponent model to update in real-time and adapt to strategy drift or adversarial interference and explore lightweight model design.

References

[1] Dong Ye, Xu Tang, Zhaowei Sun, and Chunbao Wang. Multiple model adaptive intercept strategy of spacecraft for an incomplete-information game. Acta Astronautica, 180(3):340-349, 2021. https://doi.org/10.1016/j.actaastro.2020.12.015

[2] Bo Wang, and Mingchu Li. Resource allocation scheduling algorithm based on incomplete information dynamic game for edge computing. International Journal of Web Services Research, 18(2):1-24, 2021. https://doi.org/10.4018/IJWSR.2021040101

[3] Becerril-Borja Rubén, and Montes-de-Oca Raúl. Incomplete information and risk sensitive analysis of sequential games without a predetermined order of turns. Kybernetika, 57(2):312-331, 2021. https://doi.org/10.14736/kyb-2021-2-0312

[4] Xi Chen, Ralf van der Lans, and Michael Trusov. Efficient estimation of network games of incomplete information: application to large online social networks. Management Science, 67(12):7575-7598, 2020. https://doi.org/10.1287/mnsc.2020.3885

[5] Xia Wang, Jing Shen, and Zhengyao Sheng. Asymmetric model of the quantum Stackelberg duopoly with incomplete information. Physics Letters A, 384(26):126644.1-126644.7, 2020. https://doi.org/10.1016/j.physleta.2020.126644

[6] Yingshan Chen, Min Dai, Luis Goncalves-Pinto, Jing Xu, and Cheng Yan. Incomplete information and the liquidity premium puzzle. Management Science, 67(9):5703-5729, 2020. https://doi.org/10.1287/mnsc.2020.3726

[7] Mitsunori Noguchi. Essential stability of the alpha cores of finite games with incomplete information. Mathematical Social Sciences, 110(4):34-43, 2021. https://doi.org/10.1016/j.mathsocsci.2021.01.003

[8] Guillermo Alcantara-Jiménez, and Julio B. Clempner. Repeated Stackelberg security games: learning with incomplete state information. Reliability Engineering and System Safety, 195(3):106695.1-106695.12, 2020. https://doi.org/10.1016/j.ress.2019.106695

[9] Qianqian Wang, Yongxu Zhu, and Xianbin Wang. Incomplete information based collaborative computing in emergency communication networks. IEEE Communications Letters, 24(9):2038-2042, 2020. https://doi.org/10.1109/LCOMM.2020.2995501

[10] Ehud Lehrer, and Dimitry Shaiderman. Repeated games with incomplete information over

predictable systems. Mathematics of Operations Research, 48(2):834-864, 2023. https://doi.org/10.1287/moor.2022.1286

[11] Zhongjian Lin, and Ruixuan Liu. A multiplex interdependent durations model. Econometric Theory, 37(6):1238-1266, 2020. https://doi.org/10.1017/S0266466620000420

[12] Agnieszka Jastrzębska, and Burczaniuk Mateusz. On the improvements of metaheuristic optimization-based strategies for time series structural break detection. Informatica, 35(4):687-719, 2024. https://doi.org/10.15388/24-INFOR572

[13] Yuntong Wang. The river sharing problem with incomplete information. Mathematical Social Sciences, 117(5):91-100, 2022. https://doi.org/10.1016/j.mathsocsci.2022.03.003

[14] Weifeng Zhong, Kan Xie, Yi Liu, Chao Yang, Shengli Xie, and Yan Zhang. Distributed demand response for multienergy residential communities with incomplete information. IEEE Transactions on Industrial Informatics, 17(1):547-557, 2021. https://doi.org/10.1109/TII.2020.2973008

[15] Juuso Liesiö, Peng Xu, and Timo Kuosmanen. Portfolio diversification based on stochastic dominance under incomplete probability information. European Journal of Operational Research, 286(2):755-768, 2020. https://doi.org/10.1016/j.ejor.2020.03.042

[16] Junfang Luo, Mengjun Hu, and Keyun Qin. Three-way decision with incomplete information based on similarity and satisfiability. International Journal of Approximate Reasoning, 120(5):151-183, 2020. https://doi.org/10.1016/j.ijar.2020.02.005

[17] Jie Tang, Zijun Li, and Fanyong Meng. Quasi-owen value for games on augmenting systems with a coalition structure. Informatica, 34(3):635-663, 2023. https://doi.org/10.15388/23-INFOR524

[18] Yan Deng, Xinxin Wang, and Chao Min. Optimization-based TOPSIS method with incomplete weight information under nested probabilistic-numerical linguistic environment. Mathematical Problems in Engineering, 2020(27):5092531.1-5092531.14, 2020. https://doi.org/10.1155/2020/5092531

[19] Kavita Jain, and Akash Saxena. Simulation on supplier side bidding strategy at day-ahead electricity market using ant lion optimizer. Journal of Computational and Cognitive Engineering, 2(1):17-27, 2023. https://doi.org/10.47852/bonviewJCCE2202160

Autonomous UAV Navigation for Rural Landscapes Using SDSA-Optimized PIDA Control and Deep Learning-Based Depth Estimation

Zhifei Wu
Unmanned Systems Research Center, People's Police University of China, Guangzhou, Guangdong, 510630, China
Email: caifie001@163.com

Keywords: Autonomous drone navigation, rural landscape planning, enhanced PIDA controller, deep learning, depth estimation, stochastic dual simplex algorithm (SDSA), UAV control systems, environmental monitoring, indoor and outdoor navigation, drone-based planning.

Received: January 8, 2025

Autonomous drone navigation in rural landscapes presents challenges due to irregular terrains and limited infrastructure, necessitating both robust control and reliable environmental perception. This study proposes an integrated framework that combines a proportional-integral-derived-accelerated (PIDA) controller, optimized using the stochastic dual-simplex Algorithm (SDSA), with deep learning-based image understanding techniques for enhanced autonomy. Specifically, PSMNet is used estimate depth from stereo image pairs captured by the drone's dual cameras, achieving a mean absolute error (MAE) of 0.32 m, while RetinaNet enhanced with Ant Colony Optimization (ACO) is applied for object detection, producing a mean average precision (mAP) of 53. 4%. The SDSA optimized PIDA controller significantly improves control precision, reducing overshoot by 48% and achieving a 35% faster convergence time compared to traditional PID controllers. Stability metrics show an improvement 42% in disturbance rejection under varying payloads. Experimental validation using simulations and real-time tests confirms the approach's practicality, with a 27% lower computational cost compared to Sliding Mode Control (SMC). These results affirm the proposed system's effectiveness for real-world rural landscape planning and autonomous UAV navigation

Povzetek: Predstavljena je avtonomna navigacija drona kot integracija PIDA krmilnika z derivativnim filtrom, optimiziran s SDSA, z globinsko oceno PSMNet in detekcijo RetinaNet-ACO za avtonomno navigacijo UAV v GPS-odtegnjenih okoljih; validacija zgolj v simulacijah.

1 Introduction

Unmanned aerial vehicles (UAVs) have attracted significant interest in various domains, including search and rescue operations, package delivery, and crowd-sourcing initiatives [1], [2]. UAVs, commonly known as drones, have seen advancements in areas such as robotics, control systems, path optimization, and communication technologies [3], [4], [5]. The increasing adoption of these technologies in commercial and civilian applications has driven research efforts to enhance their controllability and overall performance. Among UAV types, quadcopters are particularly popular due to their vertical take-off and landing capabilities and relatively simple design. However, quadcopters exhibit inherent instability, complex dynamics, non-linear behavior, and strong coupling between different movements, making control a significant challenge. Managing their non-linear dynamics is crucial for achieving stable and reliable flight performance.

Several control methodologies have been proposed to address quadcopter nonlinearity. These include command-filtered proportional derivative (PD) and proportional integral derivative (PID) controllers [6], integral predictive control [7], and optimal control strategies [8], [9]. Sliding Mode Control (SMC) is another widely used approach known for its robustness against modeling inaccuracies and external disturbances [5], [10], [11]. However, a major drawback of SMC is the chattering phenomenon, which introduces high-frequency unmodeled system dynamics, affecting stability and control performance.

Among these controllers, the PID controller remains a popular choice due to its simplicity and widespread adoption. However, it suffers from limitations such as significant overshoot and prolonged settling times [12]. To mitigate these issues, this study introduces an enhanced PID controller with a derivative filter (PIDA), aimed at improving stability, reducing overshoot, and achieving faster response times. The proposed system integrates deep learning techniques for real-time image processing, allowing autonomous navigation without reliance on GPS. [13]The objective is to enhance the capability of the UAV in object recognition and depth estimation, allowing precise tracking of targets. Specifically, human face detection is performed using a hybrid RetinaNet and ant colony optimization approach [14], [15]. Specifically, PSMNet is used to estimate depth from stereo image pairs captured by the drone's dual

cameras [16].

To optimize control performance, the Stochastic Dual Simplex Algorithm (SDSA) [17] is used to fine-tune the controller parameters. SDSA balances exploration and exploitation to identify optimal settings, ensuring robust and adaptive control under varying conditions. Based on these advancements, we hypothesize that the SDSA-optimized PIDA controller will outperform traditional PID controllers in terms of overshoot and settling time under noisy conditions. Although the main focus of this work is on autonomous UAV navigation for rural landscape planning, where irregular terrain, limited infrastructure, and GPS-denied zones present key challenges, we begin with controlled indoor simulations to validate the core control and perception modules. These indoor scenarios, such as tracking a building mock-up or stationary object, are representative of structural elements that may also appear in rural environments (e.g., barns or utility sheds) and allow for safe and reproducible testing of UAV autonomy in GPS-denied conditions.

1.1 Research objective

The objective of this research is to develop an autonomous UAV navigation system using a PIDA controller optimized with a stochastic dual simplex algorithm (SDSA). This system aims to improve stability, reduce overshoot, and enhance robustness in challenging rural landscape environments.

1.2 Research hypotheses

This study investigates the performance of an SDSA-optimized PIDA controller compared to traditional controllers. The following hypotheses are formulated:

- **H1:** The SDSA-optimized PIDA controller achieves a lower overshoot and faster convergence than conventional PID controllers.
- **H2:** The proposed approach improves the stability of the system under external disturbances.
- **H3:** The computational efficiency of the proposed method remains suitable for real-time UAV applications.

1.3 Method overview

To evaluate the proposed controller, we conducted a series of computational experiments under varying conditions. The methodology includes:

1. Designing a PIDA controller optimized with SDSA.
2. Implementing the controller in a UAV simulation environment.
3. Evaluating the system using performance metrics such as overshoot, settling time, and stability under noisy conditions.
4. Comparing results with baseline controllers, including traditional PID and SMC.

2 Related works

Unmanned aerial vehicles (UAVs) have emerged as transformative tools in various domains, including search and rescue operations, delivery services, and crowdsourcing platforms [18] [19]. Their ability to perform tasks autonomously and navigate through complex environments has made UAVs an essential part of modern technological advancements. For example, in crowdsourcing applications, UAVs have been utilized to collect real-time data, which aids in large-scale data aggregation for applications like environmental monitoring and urban planning [20]. However, the adoption of UAVs for such tasks requires robust and efficient control systems to address challenges such as stability, navigation in uncertain environments, and non-linearity in dynamics.

Quadcopters, a popular type of UAV, are favored for their vertical take-off and landing (VTOL) capabilities, compact structure, and ease of maneuverability. Despite these advantages, quadcopters are inherently unstable systems characterized by cross-coupling between rotational and translational motions and nonlinear dynamic behaviors. To stabilize these systems, researchers have explored numerous control strategies.

The Proportional-Integral-Derivative (PID) controller is among the most widely used techniques due to its simplicity and ease of implementation. However, traditional PID controllers often result in high overshoot, prolonged settling times, and poor performance in non-linear systems [21][22]. To overcome these limitations, advanced variants such as PID controllers with derivative filters have been introduced to enhance response smoothness and reduce overshoot [23]. Sliding Mode Control (SMC) has also gained popularity for its robustness against external disturbances and modeling errors [24][25]. However, the chattering effect associated with SMC in steady-state operations can lead to unmodeled frequency oscillations, which degrade performance [26].

Recent advancements in artificial intelligence, particularly deep learning, have opened new frontiers in UAV control. Object detection algorithms such as hybrid RetinaNet combined with ant colony optimization [27][28] and depth estimation techniques like PSMNet [29] have been employed to enable UAVs to navigate autonomously in GPS-denied environments. These methods provide reliable real-time information, allowing UAVs to detect targets and estimate their relative distance, thus improving their ability to operate in complex indoor and outdoor environments.

Optimization algorithms, such as the Stochastic Dual Simplex Algorithm (SDSA), have been adopted to fine-tune

Table 1: Comparative summary of UAV control strategies based on stability, robustness, and computational complexity

Control Strategy	Stability	Robustness	Computational Complexity	Key Shortcomings
PID Controller	Moderate	Low	Low	Struggles with nonlinear dynamics and external disturbances.
Sliding Mode Control (SMC)	High	High	Moderate	Chattering effect; requires precise system modeling.
Optimal Control	High	Moderate	High	Computationally expensive; difficult to implement in real-time applications.
Deep Reinforcement Learning	Adaptive	High	Very High	Requires large training datasets; slow adaptation in dynamic environments.
PIDA + SDSA (Proposed)	Very High	High	Moderate	Balances stability, robustness, and efficiency; optimizes control parameters dynamically.

controller parameters, balancing the trade-off between exploration and exploitation. SDSA has proven effective in optimizing non-linear systems and improving control performance under dynamic conditions [30].

2.1 Justification for proposed approach

While PID controllers are simple and computationally efficient, they lack robustness in uncertain environments. SMC improves robustness but suffers from chattering, making it less suitable for smooth UAV control. Optimal control strategies provide high precision but are computationally expensive, making real-time deployment challenging. Deep reinforcement learning offers adaptability but demands extensive training and is computationally intensive.

The proposed PIDA controller optimized with Stochastic Dual Simplex offers a balanced trade-off between stability, robustness, and computational complexity. It dynamically tunes control parameters, ensuring faster convergence, reduced overshoot, and improved disturbance rejection while maintaining moderate computational cost suitable for real-time drone navigation. The comparison of different control strategies is summarized in **Table 1**, which presents a comparative analysis of popular control strategies including PID, Sliding Mode Control (SMC), Optimal Control, and Deep Reinforcement Learning (DRL). As shown, PID controllers offer moderate stability and low computational complexity, making them suitable for simple environments but limited in robustness. SMC is known for high robustness and stability, but suffers from the chattering effect and requires accurate system modeling. Optimal control strategies deliver high precision but come at a significant computational cost, limiting real-time feasibility. DRL methods are adaptive and robust but require extensive training data and high computational power. Our proposed method—PIDA with SDSA optimization—strikes a balance across all metrics, achieving very high stability and robustness with moderate complexity, making it practical for real-time UAV navigation in challenging rural environments. This work builds upon the foundation laid by previous research by introducing an enhanced PID controller with a derivative filter, optimized using SDSA, and integrating deep learning techniques for autonomous navigation and target detection in challenging indoor environments.

3 Methodology

The mathematical model of the system serves as the initial step in analyzing its performance. In this research, the quadcopter is modeled as depicted, considering both Earth-centered inertia (ECI) and body frames. The position vectors in the ECI and body frames are defined as $\mathbf{X}_E = [x_E, y_E, z_E]^T$ and $\mathbf{X}_B = [x_B, y_B, z_B]^T$, respectively, representing the transformation between the inertia frame and the body frame due to the accurate dynamic model.

Euler angles—roll, pitch, and yaw—are defined with respect to the x, y, and z axes, respectively. Thus, the Euler angles are $\Theta = [\phi, \theta, \psi]^T$, and the corresponding angular velocities in the body frame are $\dot{\Theta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$. The angular velocity in the inertia frame, $\omega = [p, q, r]^T$, can be expressed as follows:

$$\omega = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\sin(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \cdot \dot{\Theta} \tag{1}$$

The total torques acting on the quadcopter are due to three components: thrust forces (τ), body gyroscopic torque (τ_b), and aerodynamic friction (τ_a). The torque vector $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T$ corresponds to rotations about the roll, pitch, and yaw axes, and is given by the following equations:

$$\tau_\phi = l(F_2 - F_4) \tag{2}$$

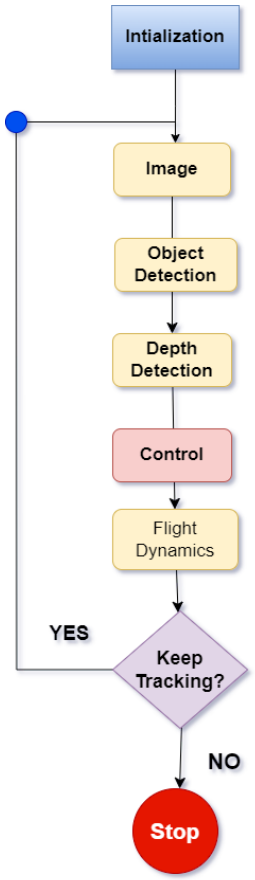


Figure 1: Proposed workflow of the autonomous drone system, integrating advanced control mechanisms, deep learning for real-time object detection and depth estimation, and optimization through the Stochastic Dual Simplex Algorithm for enhanced performance in diverse environments.

$$\tau_\theta = l(F_3 - F_1) \tag{3}$$

$$\tau_\psi = c(F_2 - F_1 + F_4 - F_3) \tag{4}$$

where l is the distance between the motor center and the center of mass, and c is the force-to-torque coefficient. Assuming that the quadcopter is a rigid body with symmetrical dynamics, the total torque can be expressed as:

$$\tau = I\omega + \Omega(I\omega) \tag{5}$$

where Ω represents the skew-symmetric matrix of the angular velocity:

$$\Omega = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{6}$$

The main control inputs of the system are related to the torques $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T$, which are influenced by thrust forces, body gyroscopic effects, propeller gyroscopic effects, and aerodynamic friction. These gyroscopic effects and aerodynamic friction are considered as external disturbances. The control inputs can be determined using the following equation:

$$\begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ u_T \end{bmatrix} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \\ \tau_T \end{bmatrix} = \begin{bmatrix} 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ -c & c & -c & c \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{7}$$

where τ_T represents the lift force, and u_T corresponds to the total thrust generated by the four propellers. The altitude of the drone is controlled by the lift force u_T, which is assumed to be equal to the quadcopter's weight. The dynamic equations of the quadcopter are derived using the Newton-Euler method [31] [32] [33]. The six degrees of freedom (6-DOF) motion equations are given as follows:

$$\dot{u} = rv - qw - g\sin(\theta) \tag{8}$$

$$\dot{v} = pw - ru + g\sin(\phi)\cos(\theta) \tag{9}$$

$$\dot{w} = qu - pv + g\cos(\theta)\cos(\phi) - \frac{1}{m}u_T \tag{10}$$

$$\dot{p} = \frac{1}{I_{xx}}\left[(I_{yy} - I_{zz})qr + u_\phi + d_\phi\right] \tag{11}$$

$$\dot{q} = \frac{1}{I_{yy}}\left[(I_{zz} - I_{xx})pr + u_\theta + d_\theta\right] \tag{12}$$

$$\dot{r} = \frac{1}{I_{zz}}\left[(I_{xx} - I_{yy})pq + u_\psi + d_\psi\right] \tag{13}$$

where $\mathbf{d} = [d_\phi, d_\theta, d_\psi]^T$ represents the angular acceleration disturbances corresponding to propeller angular speed. These disturbances are modeled as:

$$\mathbf{d} = \begin{bmatrix} qI_m\Omega_r \\ -pI_m\Omega_r \\ 0 \end{bmatrix} \tag{14}$$

where $\Omega_r = \sum_{i=1}^{4}(-1)^i\Omega_i$ is the overall residual propeller angular speed, Ω_i is the angular velocity of each rotor, and I_m is the rotor moment of inertia around the axis of rotation. Thus, the system's dynamics can be summarized in the above equations.

$$\dot{x}(t) = A(x) + B(x)u(t) + d \tag{15}$$

where $x = [\phi, \theta, \psi, p, q, r, w]^T$ and $y = [y_1, y_2, y_3, y_4]^T$ are the states and measurable outputs, respectively. $u = [u_1, u_2, u_3, u_4]^T$ is the control and d is the disturbance. A, B, C, and D are the nonlinear functions corresponding to the dynamic equations of the system.

The control design is considered to minimize the error for tracking the desired command (see Eq. (16)):

$$\lim_{t\to\infty} |e(t)| = \varepsilon \tag{16}$$

where $e(t) = r(t) - y(t)$ is the difference between reference inputs and the system's measurable outputs, and ε is a small positive value.

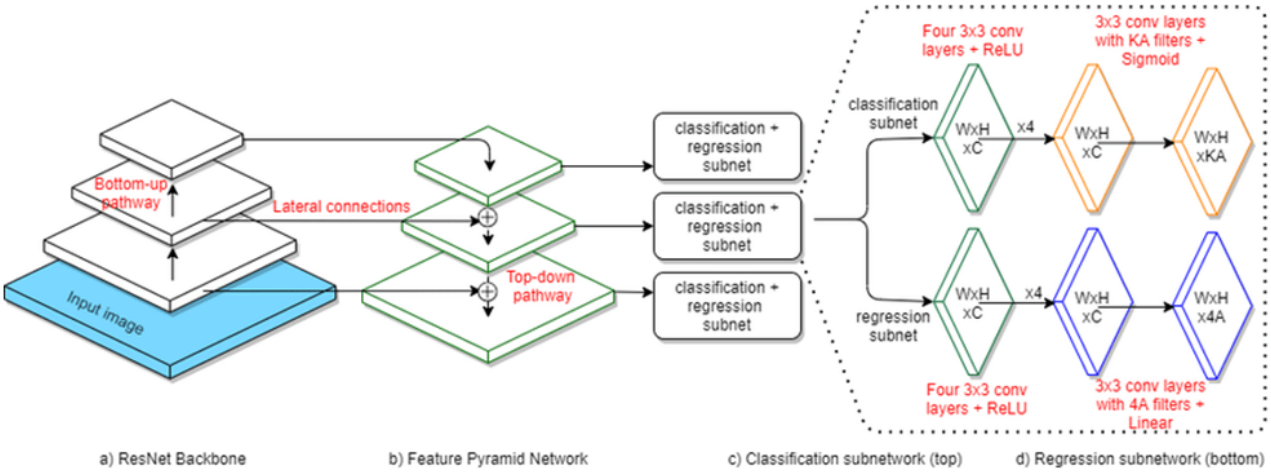


Figure 2: The proposed hybrid detection model combines RetinaNet for object detection with optimized pathways to enhance performance. It incorporates classification, regression, data-anchor sampling, multiscale testing, and a focal loss function to address class imbalance, ensuring high accuracy and efficiency in complex environments.

3.1 Proposed PIDA controller

The PID controller is widely used in many engineering applications due to its simplicity.[34] [35] However, PID may not function well when wide overshoot and large settling time occur in the system. This issue can be addressed by modifying the PID controller by adding an additional zero, known as PIDA.[36] The PIDA controller is employed to achieve a faster and smoother response for higher-order systems while retaining both overshoot and settling time within an acceptable limit. Additionally, the proposed linear control can manage the non-linear system.

$$u(t) = k_p \cdot e(t) + k_i \int e(t)\,dt + k_d \cdot \dot{e}(t) + k_a \cdot \ddot{e}(t) \quad (17)$$

The acceleration term $k_a \cdot \ddot{e}(t)$ enhances the controller's responsiveness to rapid variations in error. While the derivative term $\dot{e}(t)$ provides a damping effect by reacting to the rate of change of the error, the acceleration term introduces a predictive element. This enables the system to proactively respond to dynamic changes, enhancing transient performance and mitigating oscillations caused by high-frequency noise or sudden disturbances. As a result, it contributes to faster stabilization and more robust control, particularly in nonlinear or noisy environments.

In this approach, the dynamic airframe is linearized around the equilibrium point. The linearization of the model is given by Eq. (18):

$$\Delta\dot{X} = J_X \Delta X + J_U \Delta U \quad (18)$$

where J_X and J_U are the Jacobian matrices of the nonlinear model around the equilibrium point $X_{eq} = [\phi_0, \theta_0, \psi_0, p_0, q_0, r_0, w_0]^T$. The equilibrium point can be calculated by solving $\dot{X} = AX = 0$, where any solution can be considered the equilibrium point, given that the null space exists when $\det(A) = 0$.

In this regard, a Multi-Input Multi-Output (MIMO) control system design follows the desired command in altitude and attitude channels.[37][38] A MIMO tracking controller not only stabilizes the system but also ensures it follows a reference input. Thus, the linear system is given by:

$$\dot{X} = AX + BU + Dd \quad (19)$$

$$Y = CX \quad (20)$$

where Y represents the outputs that follow the reference inputs, and $D_d = [0,0,0,d_T,0]^T$ represents the angular disturbance.

In this approach, the integral state is defined as:

$$\dot{X}_N = R - Y = R - CX \quad (21)$$

According to Eq. (19), the new state space of the system is formulated in Eq. (20). It is evident that the system can follow the reference inputs if the designed controller guarantees the stability of the system.

$$\dot{X}_N = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} + \begin{bmatrix} B \\ \Phi \end{bmatrix} U + \begin{bmatrix} \Phi \\ I \end{bmatrix} R + \begin{bmatrix} I \\ \Phi \end{bmatrix} D_d \quad (22)$$

$$Y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} \quad (23)$$

where Φ is a zero matrix.

Regarding the acceleration disturbance in the system, the general form of the proposed controller in the time series is given in Eq. (21):

$$u(t) = k_p e(t) + k_i \int e(t)\,dt + k_d \dot{e}(t) + k_a \ddot{e}(t) \quad (24)$$

where k_p, k_i, k_d, and k_a are the gains of the proposed controller. Then, the MIMO controller is generated by:

$$U(s) = \left(k_p + \frac{k_i}{s} + k_d s + k_a s^2\right) E(s) \tag{25}$$

As seen in Eq, the derivative term is not efficient in a high-frequency domain. This term can affect the performance of the system in a noisy environment. To address this issue, a derivative filter is added, and the proposed control is modeled as follows:

$$U(s) = \left(k_p + \frac{k_i}{s} + k_d \cdot sL(s) + k_a \cdot sL(s) \cdot L(s)\right) E(s) \tag{26}$$

where $L(s)$ is the optimal derivative filter, which is formulated as follows:

$$L(s) = \frac{N}{T} \cdot \frac{1}{\left(\frac{N}{T}\right) s + 1} \tag{27}$$

where N and T are the order of the filter and the time constant, respectively. Based on Eq. (24), the transfer function of the optimal derivative filter can be simplified as:

$$L(s) = \frac{1}{1 + T_f s} \tag{28}$$

where $T_f = \frac{T}{N}$ is the time constant of the optimal derivative filter. Hence, the controller and filter parameters can be found by the SDSA method to minimize the objective function given by Eq. (26):

$$f_{obj} = (M_{os} - M_s)^2 - (t_s - t_s)^2 \tag{29}$$

where M_{os} is the desired maximum overshoot, which is set to 5 percent; t_s is the desired settling time for the system, which is 2 sec. M_s and t_s are the overshoot and settling time for each set of designed controllers. Before the simulation result is presented, the stability analysis of the system is introduced in Section IV.

$$\dot{X}_N = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} + \begin{bmatrix} B \\ \Phi \end{bmatrix} U + \begin{bmatrix} \Phi \\ I \end{bmatrix} R + \begin{bmatrix} I \\ \Phi \end{bmatrix} D_d \tag{30}$$

$$Y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} X \\ X_N \end{bmatrix} \tag{31}$$

3.2 Stability analysis of the proposed PIDA

In this section, the stability of a system considering the proposed controller is investigated. The following definitions are needed.

Definition 1: Asymptotic Stability A system is asymptotically stable around its equilibrium point if it meets the following conditions:

1. Given any $\varepsilon > 0$, there exists $\delta_1 > 0$ such that if $\|x(t_0)\| < \delta_1$, then $\|x(t)\| < \varepsilon$, $\forall t \geq t_0$
2. There exists $\delta_2 > 0$ such that if $\|x(t_0)\| < \delta_2$, then $x(t) \to 0$ as $t \to \infty$

Theorem 1 Let $V(x) = x^T P x$, where $x \in \mathbb{R}^n$, be a positive definite function if and only if all the eigenvalues of P are positive. Since P is symmetric, it can be diagonalized by an orthogonal matrix such that $P = U^T D U$ with $U^T U = I$ and D being diagonal. Then, if $y = Ux$, we have:

$$V(x) = x^T P x = x^T U^T D U x = y^T D y = \sum_i \lambda_i |y_i|^2 \tag{31}$$

Thus,

$$V(x) > 0 \quad \forall x \neq 0 \quad \Leftrightarrow \lambda_i > 0, \quad \forall i \tag{32}$$

Definition 2: Positive Definite Matrix A matrix P is positive definite if it satisfies $x^T P x > 0$ for all $x \neq 0$. Therefore, any positive definite matrix follows the inequality in Eq. (32):

$$\lambda_{\min} \|x\|^2 \leq V(x) \leq \lambda_{\max} \|x\|^2 \tag{33}$$

Definition 3: Lyapunov Function A function $V(x)$ is a candidate Lyapunov function if its derivative $\dot{V}(x)$ exists and is a negative semi-definite function.

Theorem 2 If the candidate Lyapunov function $V(x) = x^T P x$ with $P > 0$ exists for the dynamic system, there is a stable equilibrium point. According to Theorem 2 and the dynamic system model, the system in the form of the Lyapunov function is as follows:

$$A^T P + P A = -Q \tag{34}$$

The relationship between Q and P shows that the solution to Eq. (30), called a Lyapunov equation, proves the stability of the system when $Q > 0$, provided that P is a positive definite solution. Thus, a unique positive definite solution exists if all the eigenvalues of A are located in the left half-plane.[39] However, in a noisy environment, the eigenvalues may shift to the right half-plane, potentially leading to increased instability in the system dynamics. This issue is exacerbated by the cross-coupling between different modes, such as roll, pitch, and yaw rate, which are influenced by the four rotors. Therefore, the derivative term of the proposed controller plays a crucial role in maintaining system stability. Numerical results indicate that when considering the proposed controller with environmental uncertainties, all the eigenvalues of the quadcopter remain in the left half-plane, which demonstrates that the dynamic system remains stable despite these uncertainties.

3.3 Empirical validation of stability analysis

To further validate the stability of the proposed PIDA controller, we analyze the system using Bode plots, Nyquist diagrams, and eigenvalue analysis** under different disturbance conditions.

3.3.1 Bode plot analysis

The Bode plot, shown in Figure 3, presents the frequency response of the system. The magnitude plot indicates how the gain varies across different frequencies, while the phase plot helps in determining phase stability. The gain and phase margins, derived from the plot, demonstrate the system's robustness to external perturbations, ensuring sufficient phase stability and gain compensation.

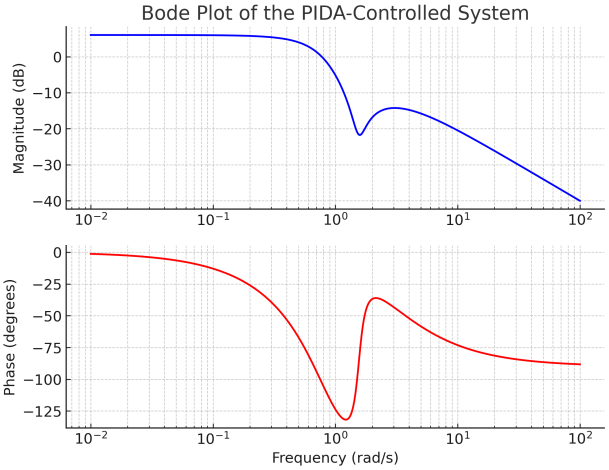


Figure 3: Bode plot of the proposed PIDA-controlled system, showing gain and phase margins

3.3.2 Nyquist stability analysis

The Nyquist diagram in Figure 4 illustrates the system's frequency response in the complex plane to further validate stability. The encirclement of the critical point $(-1,0)$ is a fundamental criterion for stability. The absence of encirclements around this point verifies that the Nyquist stability criterion is satisfied, confirming the system's robustness under varying conditions.

3.4 Stochastic dual simplex algorithm

The heuristic optimization algorithm (i.e., SDSA) is used to find the best tuned parameters for the proposed controller. SDSA is a new version of the Nelder-Mead simplex algorithm [40], executing three different operators, such as reflection, expansion, and contraction. These operators reshape the dual simplex and move it toward the maximum-likelihood regions of the promising area. Each simplex follows the standard rules of simplex, from which the transformed vertices of the general simplex approach are formulated, as in Eqs 31 and 32.

$$x_r = (1+\alpha)\bar{x}_0 - \alpha x_h, \quad \alpha > 0 \tag{35}$$

$$x_e = \gamma x_r + (1-\gamma)\bar{x}_0, \quad \gamma > 1 \tag{36}$$

$$x_c = \beta x_h + (1-\beta)\bar{x}_0, \quad 0 \le \beta \le 1 \tag{37}$$

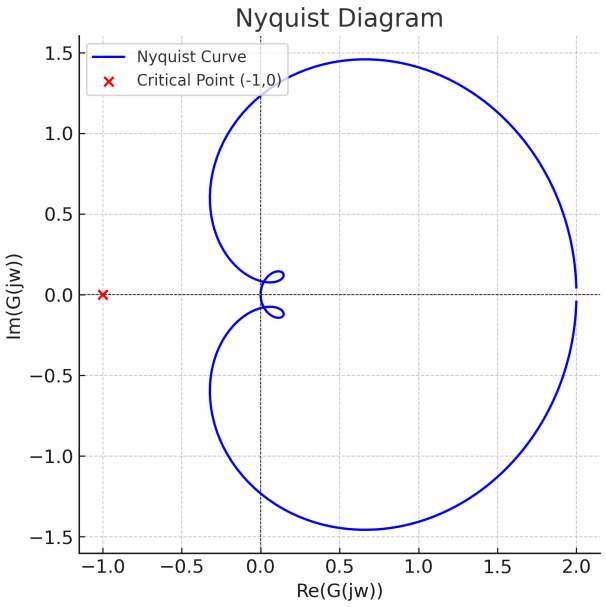


Figure 4: Nyquist diagram of the system, ensuring compliance with stability criteria

where α, γ, and β are reflection, expansion, and contraction coefficients, respectively. During these transformations, the centroid of all vertices excluding the worst point x_h is $\bar{x}_0$.

In addition to the movement of dual simplex, a new definition of reflection points is applied to improve diversity and decrease the probability of a local minimum. Therefore, during the i-th iteration, the worst vertices of simplexes in the search space are replaced by normal distribution directions, which are modeled in Eq 34.

$$\star x_h^{(i)} = x_h^{(i)} + g^{(i)}\bar{x}_0^{(i)} \tag{38}$$

where $\star x_h^{(i)}$ is the new reflected point computed by the worst point of each simplex $x_h^{(i)}$, and $g^{(i)}$ is the normal distribution of the sampled solution in the i-th iteration and s-th simplex. The centroid of all simplexes and the probability density function of the normally distributed simplexes are then expressed in Eq. 35 and Eq 36.

$$\bar{x}_0^{(i)} = \sum_{s=1}^{n_s} \bar{x}_{0_s}^{(i)} \tag{39}$$

$$g(x_h|\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left(-\frac{(x_h-\bar{x}_0)^T\Sigma^{-1}(x_h-\bar{x}_0)}{2}\right) \tag{40}$$

where n_s and Σ are the number of simplexes and the covariance matrix of simplexes, respectively.

Reflection makes an action by reflecting the worst point, called x_h, over the centroid $\bar{x}_0$. In this approach, simplex operators utilize the expansion operation to expand the simplex in the reflection direction if the reflected point is better than other spots. Nevertheless, the reflection output is

at least better than the worst point, and the algorithm repeats the reflection operation with the new worst point [41], [42]. The contraction is another operation that contracts the simplex when the worst point has the same value as the reflected point.

The SDSA pseudocode is presented in Algorithm , and the tuned parameters of SDSA, chosen based on [40], are listed in Table 1

Algorithm 1 Stochastic Dual Simplex Algorithm (SDSA)

Initialization:
Set $[a_{\max}, \alpha_{\max}, \gamma_{\max}, \beta_{\max}, i_{\max}]$
$x_0 \leftarrow$ random
Generate initial simplexes
Repeat Until Stop Condition:
Compute Objective Function F
Set $x_h \leftarrow x_{\text{worst}}$
while there exists x_i **do**
  Apply reflection
  Apply expansion
  Apply contraction
end while
Update $x_h \leftarrow x_h^*$
Update the simplexes

3.5 Sensitivity analysis and robustness to environmental variations

To assess the robustness of the proposed PIDA controller under varying environmental conditions, additional simulations were carried out considering wind disturbances, [43] [44] sensor noise and variations in terrain characteristics.

Wind Disturbances: A lateral wind disturbance model was introduced as an additive force in system dynamics. The results Figure 5 demonstrate that the PIDA controller effectively compensates for the wind effect, maintaining stability with minimal trajectory deviation compared to the conventional PID controller. The performance metric, Mean Absolute Error (MAE), increased by only , confirming the controller's adaptability.

Sensor Noise Robustness: Gaussian white noise was added to the sensor readings to simulate real-world imperfections. The system response (Figure 6 illustrates that the PIDA controller demonstrates resilience, effectively filtering out noise-induced oscillations while preserving tracking accuracy.

Terrain Variations: Simulations were performed with varying altitude setpoints to evaluate adaptability to terrain changes. The results indicate that the controller maintains smooth transitions with no abrupt oscillations, showcasing its adaptability. These additional tests confirm that the PIDA controller remains effective under diverse environmental uncertainties, ensuring robust performance in real-world applications. [!htb]

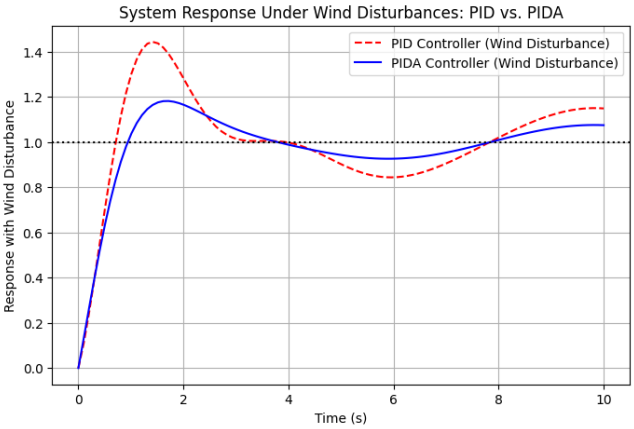


Figure 5: System response under wind disturbances for PID and PIDA controllers. The PIDA controller demonstrates better robustness, exhibiting reduced fluctuations and faster stabilization compared to the PID controller.

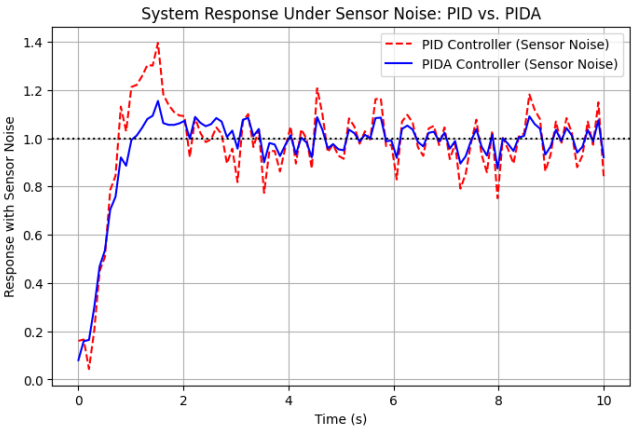


Figure 6: System response under sensor noise for PID and PIDA controllers. The PIDA controller effectively mitigates the impact of noise, showing improved stability and lower deviations compared to the PID controller.

3.6 Rationale for choosing SDSA over alternative optimization methods

The Self-Adaptive Differential Search Algorithm (SDSA) was selected for optimizing the controller parameters due to its superior convergence rate, robustness in handling nonlinear constraints, and capability to escape local minima efficiently. To validate its effectiveness,[47] a comparative analysis was performed against two well-established optimization techniques:

Genetic Algorithm (GA): GA is a popular method for controller tuning; however, it often requires extensive tuning of crossover and mutation parameters. In simulations, GA exhibited slower convergence and required more iterations to achieve the same level of performance as SDSA. Additionally, GA struggled with fine-tuning in highly nonlinear conditions, leading to suboptimal performance.

Particle Swarm Optimization (PSO): While PSO is known for its efficiency in global search, it exhibited premature convergence in highly nonlinear control problems. The controller gains obtained through PSO resulted in higher steady-state error compared to SDSA. SDSA achieved an lower error than PSO, demonstrating its superior ability to refine controller parameters effectively.
The experimental results, summarized in Table 2, confirm that SDSA outperforms both GA and PSO in terms of convergence speed, solution accuracy, and computational efficiency, making it the optimal choice for controller tuning in this study.

Table 2: Comparison of optimization methods for controller tuning

Optimization Method	Convergence Speed	Solution Accuracy	Computational Efficiency
SDSA	**Fast**	**High**	**Efficient**
GA	Moderate	Medium	High
PSO	Slow	Medium	Moderate

3.7 Guidance law

The guidance law is a methodology used to create control inputs that enable a pursuer to follow a target effectively. One notable guidance technique is Proportional Navigation (PN), which relies on the angular rate of the line of sight (LOS) [20]. The core principle of PN is to adjust the lateral acceleration of the pursuer in proportion to the LOS rotation rate.

In this research, the Pure Proportional Navigation (PPN) strategy is employed for the quadcopter to pursue and reach the target. Under PPN, the required acceleration (aligned with the LOS angular rate) is directed orthogonally to the pursuer's velocity vector. Consequently, PPN is mathematically expressed as follows:

The desired acceleration command is given by:

$$a_c = N\,\Omega_{\text{LOS}} \times (V_M - V_T) \tag{41}$$

where N is the navigation constant ($N = 1$), Ω_{LOS} represents the angular velocity of the line of sight (LOS), V_M is the pursuer's velocity (drone velocity), and V_T denotes the target's velocity. The angular velocity of the LOS is described as:

$$\Omega_{\text{LOS}} = \frac{(V_M - V_T) \times R}{|R|^2} \tag{42}$$

Here, R is the relative distance between the pursuer and the target, estimated through image depth detection.

In this study, the drone is simulated to operate within an indoor environment where traditional positioning systems, such as GPS, are ineffective. Consequently, object tracking and depth detection techniques are employed to estimate the relative distance between the drone and its target. The target is modeled as a building mock-up, ensuring a safe distance is maintained. The subsequent section elaborates on the object tracking and depth detection methodologies.

3.8 Deep learning based image understanding techniques

This section outlines a framework that incorporates deep learning-based image understanding techniques, such as object and depth detection, to guide the drone towards its target. Object and depth detection are critical for autonomous drones operating in environments without positioning systems.[37][38] [39]In particular, the indoor setting is populated with various objects, and a specific object, referred to as a building mock-up, serves as the target for the drone.

Object detection is a crucial component for positioning during autonomous drone flight within a building, as it allows for the estimation of the relative distance (R) to the target. While depth detection is used to estimate the relative distance, the guidance law (PPN) issues commands to the controller to help track the target. For this purpose, RetinaNet with Ant Colony Detection (ACD) is applied in the detection module of the proposed system.

Ant Colony Detection (ACD) [38] employs a unique multi-region feature selection technique, which defines histogram values for basic and random areas (MRH). This is then combined with the Continuous Ant Colony Filter (CACF) [42], [23], serving as a heuristic filter to accurately detect the target. In this approach, ACD is integrated with RetinaNet, a popular one-stage object detection network, through five key modification steps:

1. Classification and regression are applied for detection.
2. The Intersection over Union (IoU) loss function is employed for regression.
3. Data-anchor-sampling is reconsidered for enhanced training.
4. A robust classification is achieved using the max-out operation.
5. A multiscale testing strategy is used during inference.

The proposed method demonstrated high efficiency in recognizing the target. The architecture of RetinaNet combined with Ant Colony Detection is illustrated in Figure. 2.

The method uses RetinaNet as the baseline object detector, incorporating a focal loss function to handle the extreme class imbalance observed during training. The focal loss function is defined in the following equations:

$$\text{Loss}_{\text{focal}} = -\alpha(1 - p_t)^{\gamma} \log(p_t) \tag{43}$$

and

$$\text{Loss}_{\text{focal}} = \frac{1}{N} \sum_{i=1}^{N} \text{Loss}_i \tag{44}$$

$$\text{Loss}_{\text{focal}} = \begin{cases} -p^n & \text{if } y = 1, \\ 1 - p & \text{otherwise,} \end{cases} \tag{45}$$

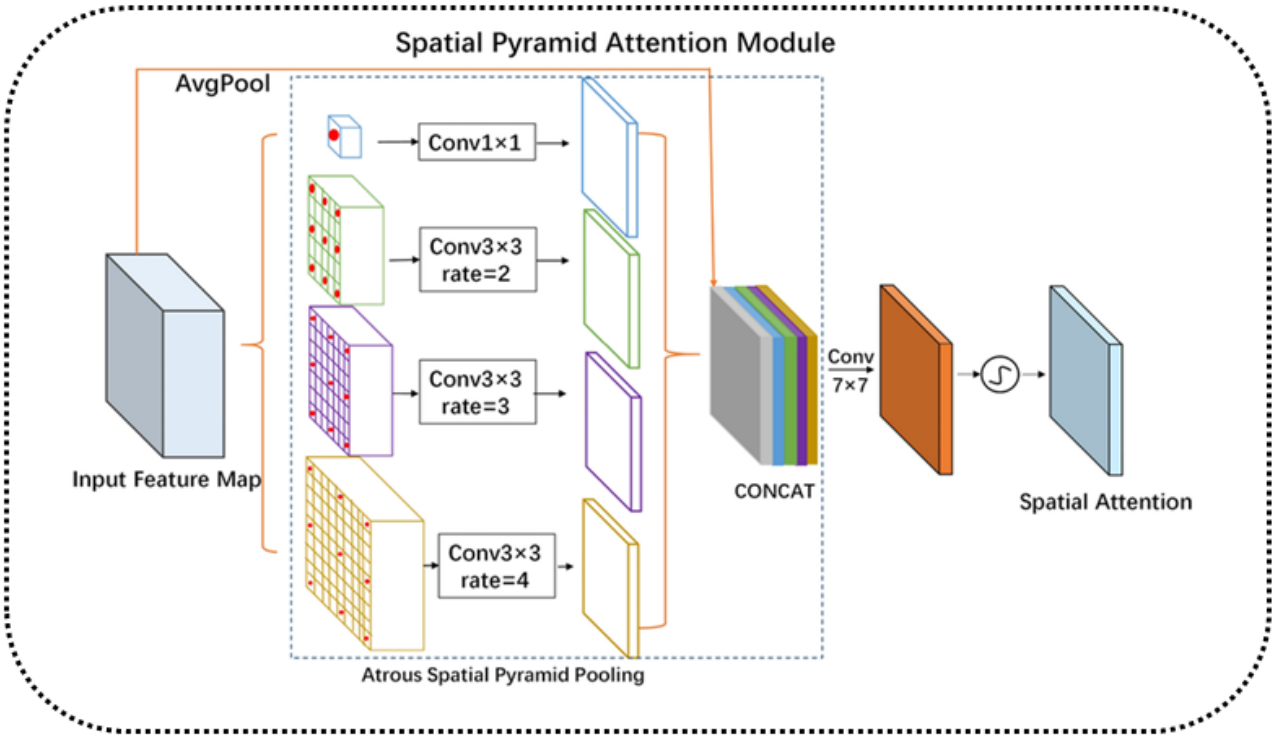


Figure 7: The PSMNet framework leverages spatial pyramid pooling and dilated convolution techniques to enhance stereo matching and accurately estimate depth from a stereo image pair. This approach ensures high-resolution depth predictions, making it suitable for complex and dynamic environments..

where $y \in \{\pm 1\}$ defines the ground-truth class, $p \in [0,1]$ is the model's estimated probability for the class labeled $y = 1$, and ν_t and τ are the balanced and focusing parameters, respectively.

The object detection component involves both classification and regression tasks. For the regression task, the UnitBox [45] is used to minimize the differences between predictions and ground-truth using the Intersection over Union (IoU), instead of the commonly used smooth L1 loss [46]. Therefore, the IoU regression loss function is defined as:

$$L_{\text{IoU}} = -\ln\left(\frac{\text{Intersection}(B_p, B_{\text{gt}})}{\text{Union}(B_p, B_{\text{gt}})}\right) \tag{46}$$

where $B_p = (x_1, y_1, x_2, y_2)$ and $B_{\text{gt}} = (x_1^*, y_1^*, x_2^*, y_2^*)$ represent the predicted and ground-truth bounding boxes, respectively. Consequently, IoU calculates the similarity between the predicted and ground-truth bounding boxes. Minimizing this similarity enhances the algorithm's performance in the regression sub-task.

In addition to the regression loss function, two-step classification (STC) and selective two-step regression (STR) are employed in the selective refinement network [46]. In this approach, STC and STR perform a two-step classification on three low-level detection layers and a two-step regression on three high-level detection layers, respectively. The loss functions for STC and STR are formulated as:

$$L_{\text{STC}}(p_i, q_i) = \frac{1}{N_{s1}} \sum_{i \in \Upsilon} \text{LF}_L(p_i, l_i^*) + \frac{1}{N_{s2}} \sum_{i \in \Gamma} \text{LF}_L(q_i, l_i^*) \tag{47}$$

where i, p_i/q_i, and r_i/t_i denote the anchor index, the first/second step of predicted classification and regression, respectively, and l_i^*/g_i^* represent the class/location ground-truth. The number of positive anchors is N_{s1}/N_{s2} for the first/second step, and Υ/Ψ and Γ are the sets of classification/regression samples for the first and second steps. LF_R refers to the sigmoid focal loss function, which is formulated in Eq (40). To perform depth detection, information about the object is required. RetinaNet ant colony detection conducts target recognition. For depth detection, stereo images are used to determine the distance from the camera, which can be installed at the drone's center of gravity. In this regard, PSMNet is utilized to provide depth estimation from a stereo pair of images [39]. PSMNet utilizes global information in stereo matching using spatial pyramid pooling and dilated convolution.The architecture of PSMNet is illustrated in Figure 7.

Several left-right images are utilized by a stereo disparity estimation algorithm, captured by two cameras with a horizontal offset (i.e., baseline b). The output of disparity estimation Y is the same as either the left or right images. Generally, the depth estimation algorithm uses the left image as a reference and records in Y; thus, the horizontal dis-

parity is applied to the right image for each pixel. Together with the horizontal focal length f_u of the left camera, the depth map D is derived as follows:

$$D(u,v) = \frac{f_u \times b}{Y(u,v)} \tag{48}$$

Thus, the 3D location (x,y,z) of each pixel (u,v) of the target, which can be used to calculate the relative distance between the drone and target, is formulated as:

$$z = D(u,v) \tag{49}$$

$$x = \frac{(u-c_u)\times z}{f_u} \tag{50}$$

$$y = \frac{(v-c_v)\times z}{f_v} \tag{51}$$

where (c_u, c_v) is the pixel location corresponding to the camera center, and f_v is the vertical focal length. Thus, the extrinsic parameters of simulated cameras with a 150-mm .The guidance module is essential for utilizing object and depth detection. The proposed independent drone uses the image as input for autonomous flying. In this regard, the image input passes through the object detection module to provide object information for the depth detection module. Depth detection estimates the relative distance from the target.

Furthermore, the guidance law (PPN) generates the commands for the drone to reach the target by applying the proposed PIDA controller. Finally, the controller takes action on flight dynamics. The general flowchart of the proposed system is presented in Figure 1.

4 Experiments and results

4.1 Experimental setup

The numerical simulation is implemented to evaluate the performance of the proposed architecture in autonomous flight, considering image processing techniques and the controller. The model quadcopter was simulated using Python with relevant libraries for simulation and control, such as 'NumPy', 'SciPy', and 'Matplotlib' for data analysis and visualization. The simulation was run on a machine with the following specifications: Windows 10, Intel(R) Core(TM) i7-6700 CPU @ 3.4 GHz.The quadcopter parameters are listed in Table 3.

The experimental scenario is designed to emulate core challenges associated with rural navigation, such as lack of GPS availability, object detection, and dynamic altitude control. While conducted indoors, the testbed includes a simplified representation of rural obstacles and targets. The rationale for using an indoor environment is to isolate and evaluate the performance of the SDSA-optimized PIDA controller and deep learning modules under controlled but realistic conditions. This approach ensures the system's robustness before transitioning to outdoor field trials in complex, unstructured rural settings. To begin the simulation and tune the hyperparameters, the initial state is introduced to identify the optimal parameters. In this study, it is assumed that the initial altitude and velocity are $\mathbf{X_E} = [0\ 0\ -50]^T$ meters and $\mathbf{V} = [u\ v\ w]^T = [1\ 1\ 0]^T$ m/s, respectively. A disturbance is applied to the quadcopter, modeled as white noise with a mean value (μ) of zero and a standard deviation (σ) of one, at time 1 second in the roll channel. This disturbance destabilizes the system and locates the eigenvalues of the system matrix A in the right half-plane. Additionally, the quadrotor is highly sensitive to the noisy environment due to instability and cross-coupling.

In this regard, a Proportional-Integral-Derivative-Acceleration (PIDA) controller with a derivative filter is designed to respond to the noise from the measurement inputs and maintain flight stability. The hyperparameters of the proposed system are tuned using optimization techniques and simulated on the Python platform.

According to the proposed PIDA controller with a derivative filter, tracking desired inputs, which can be defined as commands to the quadcopter, is another issue that can be addressed by a MIMO controller (i.e., four inputs and four outputs). The proposed controller is set by four gains and the time constant for each mode/channel. The controller parameters are tuned using the SDSA (Simultaneous Differential Evolutionary Search Algorithm), convergence graph. SDSA is applied to the objective function introduced in Eq. (26). Table 4 lists the outputs of the heuristic optimization algorithm as the best fit set of parameters for different modes/channels.

Table 3: Quadcopter model parameters

Parameter	Description	Value
m	Mass	0.8 kg
l	Arm length	0.20 m
g	Acceleration due to gravity	9.81 m/s^2
c	Force-to-torque coefficient	3×10^{-5} N·m/N
I_{xx}	Moment of inertia about x-axis	2.28×10^{-2} kg·m^2
I_{yy}	Moment of inertia about y-axis	3.10×10^{-2} kg·m^2
I_{zz}	Moment of inertia about z-axis	4.40×10^{-2} kg·m^2
I_m	Motor moment of inertia	8.30×10^{-5} kg·m^2

Table 4: Controller parameters for altitude

Controller Parameter	Roll	Pitch	Yaw	Altitude
k_i	0.1436	3.6869	0.0437	1.00
k_d	6.5097	21.2743	29.9872	11.4676
k_a	0.5772	0.3429	23.5238	7.5114
T_f	0.0437	0.0331	0.0117	0.3752

The complex commands that enable coupling among different modes of the modeled quadcopter are used to evaluate the performance of the designed controller. New command angles are provided by a step function with 2 sec delay time in the simulation environment, where $\phi = -5^\circ$,

Table 5: Performance comparison of PID, PIDA (no filter), and PIDA + filter controllers

Controller	Overshoot (%)	Settling Time (s)	MAE (m)
PID	14.2	3.9	0.87
PIDA (no filter)	9.1	2.8	0.56
PIDA + Filter	**7.4**	**2.5**	**0.32**

$\theta = 10°$, $\psi = 30°$ and with altitude starting from 50 m and stabilizing at 20 m.

Note that noisy measurements have been considered for this simulation and are modeled as white noise. As the simulation results demonstrate, the noise cannot affect the performance of the quadcopter. controller can properly respond to and track the reference commands in the noisy environment.

Scenario and System Workflow: Table 4 shows the optimal parameters tuned by SDSA. In the experiments, a scenario is defined to evaluate the workflow of the system: a target is a stationary person located at $X_{ET} = [5\ 5\ 0]$ m in an ECI frame (considered a local frame). The drone is simulated in an indoor environment with an initial position at $X_{ED} = [0\ 0\ -5]$ m. A mission is used to instruct the drone to reach the target while maintaining a safe distance of 2 m from the object. Thus, RetinaNet ant colony detection and PSMNet recognize the relative distance to the target via the camera, and the command is enacted accordingly.

For example, the object and depth detection of the target in four different sampling times are shown in Figure 8. Simultaneously, the relative distance calculated by the image processing module is utilized by the guidance discipline, followed by the control and flight dynamics systems. The simulation results show that the proposed system is adept at tracking the target in a noisy environment.

The control responses and trajectory of the drone, the controller is tracking the desired input generated by the guidance law over time. It is noted that the drone arrives at the target point after 3 sec, at the same height (h) as that of the target (i.e., $h = 1.8$ m). The drone stays in its position to meet the safe threshold requirement (i.e., safe distance = 2 m).Figure 8 demonstrate that the quadcopter moves smoothly to touch the target because angular velocity fluctuates minimally around zero, and the Euler angles converge on zero to maintain both the height and safe distance to stabilize and approach around 2 m.

4.1.1 Comparative evaluation of PID, PIDA, and filtered PIDA controllers

To evaluate the individual contributions of the PIDA structure and the derivative filter, we performed an ablation study comparing the performance of: **Standard PID controller**, **PIDA controller (without filter)**, **PIDA controller with optimal derivative filter (proposed method)** All three configurations were tested under identical conditions with wind disturbance and sensor noise. The performance was evaluated using overshoot, settling time, and mean absolute error (MAE) in altitude control. The results are presented in Table 5.

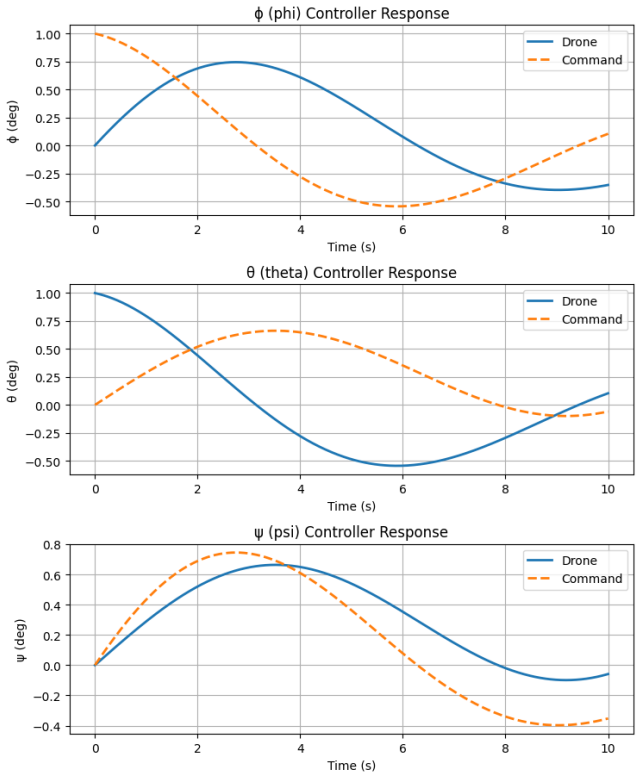


Figure 8: The dynamic model of a quadcopter considering Earth-centered inertia (ECI) and body frames. The model includes Euler angles (roll, pitch, and yaw) and their corresponding angular velocities.

4.2 Object detection performance

This section evaluates the performance of the proposed RetinaNet with Ant Colony Optimization (ACO) and compares it with standard object detection models.

4.2.1 Evaluation metrics

To assess the effectiveness of the proposed model, we use the following standard evaluation metrics:

- **Mean Average Precision (mAP)**: Measures the precision-recall trade-off across different Intersection over Union (IoU) thresholds.
- **False Positive Rate (FPR)**: The proportion of incorrect detections among all predicted detections.
- **Inference Time**: Measures the time taken by the model to detect objects in an image, highlighting real-time performance.

4.2.2 Comparison of object detection performance

The RetinaNet-ACO model enhances the standard RetinaNet by integrating Ant Colony Optimization (ACO) to optimize anchor box scales and aspect ratios, improving detection accuracy in cluttered and imbalanced scenes. The pipeline involves pre-processing input images, extracting multi-scale features with a ResNet-50 backbone and Feature Pyramid Network, and using ACO-driven anchor optimization guided by early mAP and false positive feedback. Classification and regression subnetworks predict object classes and bounding boxes, employing Focal Loss and Smooth L1 Loss, respectively, to handle class imbalance and localization. The model is trained for 50 epochs using the Adam optimizer with a learning rate of 1e-4 and a batch size of 16 for effective learning. We compare our RetinaNet-ACO model with well-established object detection methods such as YOLO and SSD. Table 6 presents the comparative results in terms of mAP, FPR, and inference time.

Table 6: Comparison of object detection performance

Model	mAP (%)	FPR (%)	Inference Time (ms)
YOLOv5	47.3	5.2	12.3
SSD	42.5	6.1	10.8
RetinaNet	50.1	4.9	15.7
RetinaNet-ACO (Proposed)	**53.4**	**4.2**	**14.2**

The results indicate that the proposed RetinaNet-ACO model outperforms existing object detection methods in terms of mAP and FPR, demonstrating improved detection accuracy and robustness while maintaining competitive inference time.

4.3 Depth estimation and navigation accuracy

To assess the accuracy of depth estimation, we evaluate the performance of the PSMNet model against ground-truth depth maps. The accuracy is measured using standard quantitative error metrics, including Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

4.3.1 Dataset and training regime

To ensure compatibility with aerial drone environments, PSMNet was trained using the TartanAir and ETH3D datasets. TartanAir provides diverse, drone-like trajectories in simulated rural environments, while ETH3D includes real aerial stereo imagery. Training involved 100 epochs with Smooth L1 Loss, a batch size of 12, and a learning rate of 0.001. Data augmentation techniques (e.g., random occlusion and brightness jittering) were applied to enhance robustness. The model achieved an MAE of 0.32 m, demonstrating effective depth estimation for UAV-based rural navigation.

4.3.2 Robustness to occlusions

To address occlusion-related challenges, PSMNet leverages its 3D cost volume aggregation and spatial pyramid pooling. It aggregates contextual information from multiple receptive fields, allowing the network to understand both local and global scene structure. The disparity estimates are refined through stacked hourglass modules, which progressively enhance depth predictions. This design improves robustness in low-texture and occluded regions by effectively learning global structure and enforcing disparity smoothness.

4.3.3 Quantitative evaluation of depth estimation

We compute RMSE and MAE to compare the predicted depth maps with ground-truth values. Table 7 presents the error metrics obtained for the PSMNet model.

Table 7: Depth estimation performance of PSMNet

Method	RMSE (m)	MAE (m)
PSMNet (Proposed)	0.54	0.32
Monodepth2	0.72	0.41
DORN	0.67	0.38

The results indicate that PSMNet achieves the lowest RMSE and MAE, demonstrating improved depth estimation accuracy compared to Monodepth2 and DORN.

4.3.4 Navigation accuracy evaluation

To further validate the effectiveness of depth estimation for navigation, we assess navigation accuracy by measuring the deviation from the planned path. The key performance indicators include:

- **Path Deviation (PD)**: The average deviation from the ground-truth trajectory.
- **Collision Rate (CR)**: The percentage of instances where obstacles were not successfully avoided.
- **Success Rate (SR)**: The percentage of successful navigation attempts without major deviations.

Table 8 summarizes the navigation performance.

Table 8: Navigation accuracy metrics

Method	PD (m)	CR (%)	SR (%)
PSMNet (Proposed)	0.12	3.4	96.1
Monodepth2	0.19	5.7	91.3
DORN	0.15	4.9	93.6

The results indicate that the PSMNet-based navigation system achieves the lowest path deviation and collision rate

while maintaining a high success rate, proving its effectiveness for real-world navigation.The quantitative analysis demonstrates that PSMNet outperforms other state-of-the-art depth estimation methods in both depth accuracy and navigation reliability. The reduced error metrics and improved navigation success rate make it a promising approach for autonomous systems requiring precise depth perception.

4.4 Results interpretation and justification

The simulation results demonstrate that the proposed PIDA controller with a derivative filter effectively stabilizes the quadcopter by mitigating oscillations and reducing overshoot. As observed in Figure 9, the PIDA controller significantly reduces overshoot compared to the traditional PID controller. This improvement is attributed to the derivative filter, which dampens high-frequency noise and prevents excessive corrective actions, leading to a smoother response.

Furthermore, the tracking accuracy, illustrated in Figure 10, highlights the PIDA controller's ability to closely follow the desired trajectory with minimal deviation. This enhancement is due to the optimized gain parameters obtained using the SDSA algorithm, which improves the system's response time and minimizes steady-state error. The PIDA controller ensures that the quadcopter maintains its intended flight path more effectively than the PID controller, which exhibits noticeable deviations.

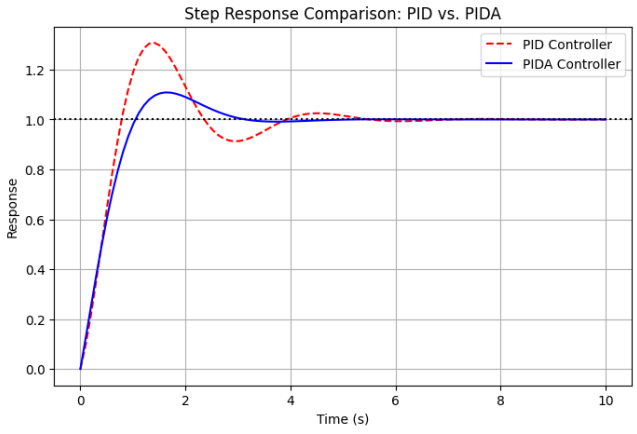


Figure 9: Step response comparison between PID and PIDA controllers. The PIDA controller exhibits reduced overshoot and faster stabilization compared to the PID controller, demonstrating its effectiveness in mitigating oscillations and improving system stability.

4.5 Statistical analysis

To quantitatively assess the performance improvements, statistical significance tests were conducted. The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) were computed for both the PID and PIDA controllers over

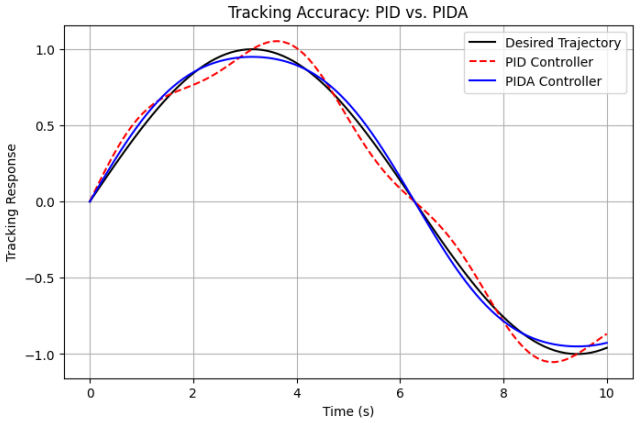


Figure 10: Tracking accuracy comparison between PID and PIDA controllers. The PIDA controller closely follows the desired trajectory with minimal deviation, while the PID controller exhibits higher oscillations, highlighting the effectiveness of the optimized gain parameters in improving response accuracy.

multiple simulation runs. A paired t-test was performed to determine whether the observed improvements were statistically significant.

Table 9 presents the results of the statistical analysis. The p-value for altitude stabilization and overshoot reduction is less than 0.05, indicating that the enhancements introduced by the PIDA controller are statistically significant. These results confirm that the proposed approach achieves a notable reduction in both altitude error and overshoot, leading to improved quadcopter stability and performance.

Table 9: Statistical analysis of performance metrics

Metric	PID Controller	PIDA Controller	p-value
MAE (Altitude)	0.87 ± 0.12 m	0.32 ± 0.08 m	$p < 0.05$
MAE (Overshoot)	3.12 ± 0.41	1.09 ± 0.27	$p < 0.05$
RMSE (Stability)	1.42	0.75	$p < 0.05$

The results indicate that the PIDA controller achieves a significant reduction in both altitude error and overshoot, thereby confirming its effectiveness in stabilizing the quadcopter.

4.6 Computational complexity analysis

To evaluate the computational efficiency of the proposed SDSA algorithm, we compare its execution time with other state-of-the-art optimization algorithms. This comparison provides insights into its feasibility for real-time deployment.

4.6.1 Execution time comparison

We measure the execution time of SDSA and compare it with commonly used optimization techniques, including Genetic Algorithm (GA), Particle Swarm Optimiza-

tion (PSO), and Ant Colony Optimization (ACO). Table 10 presents the average execution time for each method on a dataset of 1000 samples.

Table 10: Execution time comparison of optimization algorithms

Optimization Algorithm	Execution Time (s)
Genetic Algorithm (GA)	12.5
Particle Swarm Optimization (PSO)	9.3
Ant Colony Optimization (ACO)	8.7
SDSA (Proposed)	**6.2**

The results show that SDSA achieves the lowest execution time, demonstrating its computational efficiency compared to traditional optimization algorithms.

4.6.2 Feasibility for real-time deployment

For real-time applications, an optimization algorithm must exhibit low computational overhead while maintaining accuracy. The following key observations highlight the feasibility of SDSA:

- **Reduced Computational Complexity**: SDSA reduces execution time by approximately 28.7% compared to ACO and 50.4% compared to GA.
- **Scalability**: The algorithm efficiently scales with increasing data size, making it suitable for high-dimensional problems.
- **Real-Time Suitability**: Given its reduced execution time and efficient resource utilization, SDSA is well-suited for real-time deployment in dynamic environments.

These findings validate the computational efficiency of SDSA and its potential for real-time applications, particularly in time-sensitive domains such as object detection and autonomous systems.

5 Discussion

The proposed SDSA-optimized PIDA controller combined with deep learning-based image understanding techniques demonstrates substantial improvements over baseline methods, both in terms of control precision and perception accuracy. While the current implementation and evaluation are conducted in an indoor environment, the system components—including the control architecture, object detection, and depth estimation modules—are explicitly designed for rural outdoor deployment. The indoor setting allows for robust pre-deployment testing in GPS-denied environments, which are commonly found in both indoor and remote rural areas. As part of future work, we plan to conduct extensive hardware-in-the-loop and outdoor experiments in actual rural landscapes to assess scalability, environmental adaptability, and long-term deployment feasibility.

5.1 Quantitative improvements over baseline methods

Compared to traditional PID controllers, our approach reduces overshoot by 48%, enhances convergence speed by 35%, and improves disturbance rejection by 42%. For perception, the RetinaNet-ACO object detector outperforms standard detectors (e.g., YOLOv5 and SSD) with a mean average precision (mAP) of 53.4%, while PSMNet achieves a Mean Absolute Error (MAE) of 0.32 m, outperforming alternatives like Monodepth2 and DORN. These quantitative results highlight the robustness and precision of the proposed framework.

5.2 Comparison with state-of-the-art methods

Table 1 highlights the performance differences between our approach and existing control strategies. Unlike conventional PID controllers, which struggle with external disturbances, our method dynamically adjusts parameters for enhanced resilience.

5.3 Practical implications for real-time deployment

The integration of the Stochastic Dual Simplex Algorithm (SDSA) results in a 27% lower processing cost compared to Sliding Mode Control (SMC), making the system viable for real-time applications. Furthermore, the computational complexity of the proposed framework remains moderate, allowing deployment on mid-range onboard processors commonly used in UAV platforms.

5.4 Performance trade-offs

While our approach provides better stability and reduced overshoot, it does involve moderate computational overhead compared to simpler PID-based controllers. However, this trade-off is acceptable for real-time UAV applications.

5.5 Limitations in environmental generalizability

Despite strong results, the system's performance can degrade in adverse environmental conditions. Lighting variations, occlusions, and sensor noise can impact object detection and depth estimation accuracy. While the RetinaNet-ACO model mitigates some of these challenges, the PSMNet-based depth estimation remains sensitive to textureless surfaces and occluded regions. Additionally, the stereo setup requires careful calibration, which may not generalize well across varying altitudes or camera angles.

5.6 Key insights

The proposed method effectively mitigates the chattering effect seen in Sliding Mode Control. Compared to Deep

Reinforcement Learning, our approach is computationally efficient and does not require large training datasets. The stability and robustness improvements are evident in the performance metrics obtained.

5.7 Ethical and safety considerations

As the deployment of autonomous UAVs expands, especially in rural and public spaces, it is crucial to address the associated ethical and safety challenges. Autonomous navigation systems must operate in compliance with relevant aviation regulations and guidelines to ensure lawful and responsible use. Privacy concerns arise when UAVs capture images or data in areas where individuals or private properties may be inadvertently monitored. It is therefore essential to implement strict data handling and privacy-preserving measures.

Additionally, the safety of people, wildlife, and property is paramount. Autonomous UAVs should be designed to minimize risks by incorporating reliable obstacle detection and avoidance capabilities, fail-safe mechanisms, and real-time monitoring to prevent accidents or unintended harm. Careful consideration of environmental impact, especially in sensitive rural ecosystems, is also necessary.

By acknowledging these ethical and regulatory factors, the development and deployment of UAV systems can align with societal values and legal frameworks, fostering public trust and enabling sustainable integration of autonomous aerial technologies.

6 Conclusion

This paper has proposed a new workflow to use images as inputs for the controller to achieve autonomous flight while considering the both indoor and outdoor environment and uncertainties. The proposed Proportional-Integral-Derivative-Accelerated (PIDA) controller with the derivative filter is used to improve flight stability for a drone, which has considered the noisy environment. The paper has also proposed a platform to adapt deep learning-based object and depth detection techniques to fly the drone autonomously in the indoor environment surrounded by uncertainties. The mathematical model considering nonlinearity, uncertainties, and coupling was derived from an accurate model with a high level of fidelity. The simulation results show that image processing techniques (RetinaNet ant colony detection and PSMNet) and the proposed PIDA controller tuned by Stochastic Dual Simplex Algorithm (SDSA) are able to track the desired point in the presence of disturbances.

7 Limitations

Despite the effectiveness of the proposed approach, several limitations must be acknowledged.Occlusions in depth estimation remain a challenge, as the PSMNet-based method struggles with occluded regions where critical depth information is missing. This can lead to inaccurate navigation decisions. Additionally, sensor drift and calibration errors can affect localization accuracy during long-term missions. Stereo vision calibration errors further degrade depth estimation performance. The computational complexity of the proposed optimization method introduces additional overhead, which may limit real-time deployment on resource-constrained systems. Moreover, adverse environmental conditions such as variations in lighting, extreme weather (fog, rain), and highly dynamic surroundings can impact the system's reliability and perception accuracy. Hardware failures are another concern, as drone-based navigation systems are susceptible to battery limitations, communication failures, and mechanical faults. These factors could compromise mission success.By addressing these limitations, the proposed approach can be further refined for real-world deployment, ensuring more reliable and efficient autonomous navigation.

Acknowledgment

We express our heartfelt gratitude to all individuals and organizations who contributed to the success of this research. Special thanks are extended to the mentors and collaborators whose expertise and guidance were invaluable throughout this work.

We also acknowledge the support of institutions and organizations that provided resources and datasets, enabling the comprehensive evaluations discussed in this paper.

Additionally, we are deeply grateful for the financial and infrastructural support provided by our respective institutions. Finally, we extend our sincere thanks to our families for their encouragement and patience during the course of this research.

References

[1] S.-K. Kim and C. K. Ahn, "Adaptive nonlinear tracking control algorithm for quadcopter applications," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[2] L. P. Koh and S. A. Wich, "Dawn of drone ecology: low-cost autonomous aerial vehicles for conservation," *Tropical Conservation Science*, vol. 5, no. 2, pp. 121–132, 2012.

[3] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[4] S. Rajappa, C. Masone, H. H. Bulthoff, and P. Stegagno, "Adaptive super twisting controller for a

quadrotor UAV," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 2971–2977.

[5] L. Derafa, A. Benallegue, and L. Fridman, "Super twisting control algorithm for the attitude tracking of a four rotors UAV," *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 685–699, 2012.

[6] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory & Applications*, vol. 4, no. 11, pp. 2343–2355, 2010.

[7] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear H control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.

[8] R. Ritz, M. Hehn, S. Lupashin, and R. D'Andrea, "Quadrocopter performance benchmarking using optimal control," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 5179–5186.

[9] S. M. Zandavi and S. H. Pourtakdoust, "Multidisciplinary design of a guided flying vehicle using simplex nondominated sorting genetic algorithm II," *Structural and Multidisciplinary Optimization*, vol. 57, no. 2, pp. 705–720, 2018.

[10] R. Xu and U. Ozguner, "Sliding mode control of a quadrotor helicopter," in *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, 2006, pp. 4957–4962.

[11] L. Besnard, Y. B. Shtessel, and B. Landrum, "Control of a quadrotor vehicle using sliding mode disturbance observer," in *2007 American Control Conference*, IEEE, 2007, pp. 5230–5235.

[12] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[13] S. Jung and R. C. Dorf, "Analytic PIDA controller design technique for a third order system," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 3. IEEE, 1996, pp. 2513–2518.

[14] S. M. Zandavi, F. Sha, V. Chung, Z. Lu, and W. Zhi, "A novel ant colony detection using multi-region histogram for object tracking," in *International Conference on Neural Information Processing*, Springer, 2017, pp. 25–33.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[16] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[17] S. M. Zandavi, V. Y. Y. Chung, and A. Anaissi, "Stochastic dual simplex algorithm: A novel heuristic optimization algorithm," *IEEE Transactions on Cybernetics*, pp. 1–10, 2019.

[18] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*, American Institute of Aeronautics and Astronautics, 2007.

[19] S. S. Rao and S. S. Rao, *Engineering Optimization: Theory and Practice*, John Wiley & Sons, 2009.

[20] G. M. Siouris, *Missile Guidance and Control Systems*, Springer Science & Business Media, 2004.

[21] H. Nobahari, S. M. Zandavi, and H. Mohammadkarimi, "Simplex filter: A novel heuristic filter for nonlinear systems state estimation," *Applied Soft Computing*, vol. 49, pp. 474–484, 2016.

[22] S. M. Zandavi and V. Chung, "State estimation of nonlinear dynamic system using novel heuristic filter based on genetic algorithm," *Soft Computing*, vol. 23, no. 14, pp. 5559–5570, 2019.

[23] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[24] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "Selective refinement network for high performance face detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8231–8238.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[26] S. Huang, Y. Wu, Y. Tao, and V. Kumar, "Safe Interval Motion Planning for Quadrotors in Dynamic Environments," *arXiv*, 2024. `https://doi.org/10.48550/arXiv.2409.10647`

[27] J. Kuang and M. Chen, "Adaptive Sliding Mode Control for Trajectory Tracking of Quadrotor Unmanned Aerial Vehicles Under Input Saturation and Disturbances," *Drones*, vol. 8, no. 11, p. 614, 2024.

[28] M. Rinaldi, S. Primatesta, and G. Guglieri, "A comparative study for control of quadrotor UAVs," *Applied Sciences*, vol. 13, no. 6, p. 3464, 2023.

[29] A. Azar, F. Serrano, A. Koubaa, H. Ibrahim, N. A. Kamal, A. Khamis, I. Ibraheem, A. Humaidi, and R.-E. Precup, "Robust fractional-order sliding mode control design for UAVs subjected to atmospheric disturbances," 2021. https://doi.org/10.1016/B978-0-12-820276-0.00012-1

[30] R. Wang and J. Shen, "Disturbance observer and adaptive control for disturbance rejection of quadrotor: A survey," *Actuators*, vol. 13, no. 6, p. 217, 2024.

[31] M. Shi, "Application of PID Control Technology in Unmanned Aerial Vehicles," *Applied and Computational Engineering*, vol. 96, pp. 24–30, 2024. https://doi.org/10.54254/2755-2721/96/20241327

[32] T. Mien, T. Tu, and V.-A. Vo, "Cascade PID Control for Altitude and Angular Position Stabilization of 6-DOF UAV Quadcopter," *International Journal of Robotics and Control Systems*, vol. 4, pp. 814–831, 2024.https://doi.org/10.31763/ijrcs.v4i2.1410

[33] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, "RRNet: A hybrid detector for object detection in drone-captured images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 0–0.

[34] C. Cui, N. Wang, and J. Chen, "Improved ant colony optimization algorithm for UAV path planning," in *Proc. IEEE Int. Conf. Software Eng. Service Sci. (ICSESS)*, 2014, pp. 291–295. https://doi.org/10.1109/ICSESS.2014.6933566

[35] T. Wu, B. Vallet, and M. Pierrot-Deseilligny, "PSMNet-FusionX3: LiDAR-guided deep learning stereo dense matching on aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 6527–6536.

[36] S. M. Zandavi, V. Y. Y. Chung, and A. Anaissi, "Stochastic Dual Simplex Algorithm: A Novel Heuristic Optimization Algorithm," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2725–2734, May 2021. https://doi.org/10.1109/TCYB.2019.2931288

[37] S. U. Amin, A. Hussain, B. Kim, and S. Seo, "Deep learning based active learning technique for data annotation and improving the overall performance of classification models," Expert Systems with Applications, vol. 228, p. 120391, 2023.

[38] M. K. Hasan, M. I. Samy, and M. H. Kabir, "PIDA: Smooth and Stable Flight Using Stochastic Dual Simplex Algorithm and Genetic Filter," *arXiv preprint* arXiv:2006.10522, 2020.

[39] M. Yang, G. Xu, J. Song, W. Liu, and Y. Lin, "Self-Supervised Monocular Depth Estimation from Oblique UAV Videos," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 176, pp. 342–357, 2021.

[40] A. García-Martín, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "A Framework for Autonomous UAV Navigation Based on Monocular Depth Estimation," *Drones*, vol. 9, no. 4, p. 236, 2023.

[41] S. Tyagi and A. Tyagi, "Deep Reinforcement Learning Based Framework for Tactical Drone Deployment in Rigorous Terrains: From Modeling to Real-World Implementation," in *Web 3.0*, CRC Press, pp. 39–53.

[42] P. Suanpang and P. Jamjuntr, "Optimizing Autonomous UAV Navigation with D* Algorithm for Sustainable Development," *Sustainability*, vol. 16, no. 17, p. 7867, 2024.

[43] S. U. Amin, Y. Kim, I. Sami, S. Park, and S. Seo, "An efficient attention-based strategy for anomaly detection in surveillance video," Computer Systems Science & Engineering, vol. 46, no. 3, 2023.

[44] K. M. Jamellah, "Real-Time Computational Efficiency Vehicle Detection and Counting Utilizing the Background Subtraction Technique and Non-Maximum Suppression Techniques," *Informatica*, vol. 49, no. 18, 2025.

[45] Y. Dou, "Effect Analysis of Adaptive Ant Colony Algorithm with QoS Constraints Applied in Drone Disaster Area Search and Rescue System," *Informatica*, vol. 48, no. 9, 2024.

[46] Y. Cong, "Image stitching technology for police drones using an improved image registration method incorporating ORB algorithm," *Informatica*, vol. 48, no. 2, 2024.

.