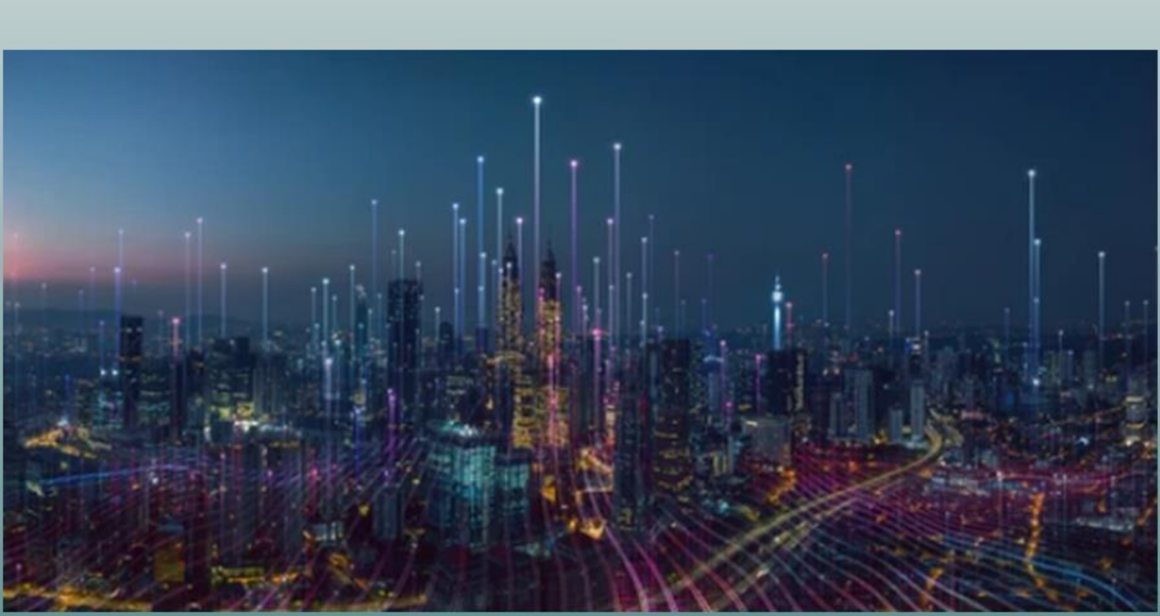
Volume 7, Issue 2 2025

Print ISSN: 2663-1938
Online ISSN: 2663-1946

JOURNAL OF COMPUTER SCIENCE AND ELECTRICAL ENGINEERING



Journal of Computer Science and Electrical Engineering

Volume 7, Issue 2, 2025

Published by Upubscience Publisher

Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938 Online ISSN: 2663-1946
Email: info@upubscience.com
Website: http://www.upubscience.com/

Table of Content

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3041

A LITERATURE SURVEY OF CRASH INJURY SEVERITY PREDICTION

YiZhi Yin, YuBin Jian, JiaYi Tan, WeiZhong Xu*
School of Civil Engineering, Southwest Jiaotong University Hope College, Sichuan 610400, Chengdu, China.
Corresponding Author: WeiZhong Xu, Email: 13608006875@163.com

Abstract: Accurate prediction of the severity of road traffic accident injuries provides a foundation for accident prevention, emergency resource allocation, and response planning. As a result, the selection of traffic accident injury severity prediction models has garnered significant attention from both authorities and researchers. This paper aims to provide a comprehensive review of the research progress on traffic accident severity prediction models. It begins by reviewing relevant datasets and features used in severity prediction. Next, it summarizes various approaches and models in traffic accident severity prediction, including traditional statistical models and machine learning techniques. A comparative analysis of these models is then conducted. Finally, the paper discusses the key challenges in current research and explores future development trends, offering theoretical guidance for both researchers and practitioners.
Keywords: Traffic accidents; Road traffic injuries; Prediction models

1 INTRODUCTION

According to the Global Status Report on Road Safety 2023 by the World Health Organization (WHO, 2023)[1], annual global road traffic fatalities have decreased to 1. 19 million, indicating initial progress in safety interventions. However, road accidents remain the leading cause of death for individuals aged 5-29 years, with vulnerable road users such as pedestrians, cyclists, and motorcyclists accounting for over half of global fatalities, particularly in low-income regions. To achieve the United Nations Sustainable Development Goal (SDG) of halving road traffic injuries and deaths by 2030, targeted strategies are urgently required. Traffic accident causation involves multi-faceted factors, including road design, environmental conditions, infrastructure quality, and driver behavior. Existing studies face critical limitations in model adaptability, data integration, and real-time responsiveness, necessitating systematic review and innovative advancements.

Traditional statistical models [2-4](eg. logistic regression, ARIMA) excel in interpretability but struggle to capture complex spatio-temporal relationships and nonlinear patterns. Machine learning methods[5-7] (eg. random forests, support vector machines) enhance prediction accuracy through data-driven approaches but rely heavily on manual feature engineering and exhibit limited generalization. Deep learning techniques[8-11] (eg. CNNs, LSTMs) demonstrate superior capabilities in automated feature extraction and spatio-temporal modeling[13-14], yet their high data dependency and poor explainability hinder practical deployment. Recent advancements in graph neural networks[12] (GNNs) and sparse spatio-temporal dynamic hypergraph learning (SSTDHL) improve predictive performance via heterogeneous data fusion, though computational complexity and real-time adaptability remain challenges. Current research predominantly focuses on isolated models or localized factors, lacking comprehensive comparisons of model performance, systematic exploration of data-driven paradigms, and strategies for multimodal fusion[15-16].

This study addresses three critical questions through a systematic review of 95 articles published between 2001 and 2024:

• Key Influencing Factors: What are the core driving factors of traffic accident severity, and how do they exhibit heterogeneity across regions and cultures?

• Model Performance Comparison: Why do prediction models (statistical, machine learning, deep learning, and graph networks) differ in accuracy, stability, and applicable scenarios?

• Optimization Pathways: How can predictive performance be enhanced through data-driven model selection, multi-modal fusion, and real-time dynamic modeling?

The methodology integrates multi-platform data (Google Scholar, Web of Science, etc.) and keyword-based searches (eg. “traffic accident prediction”, “injury severity”, “predictive models”). Comparative analyses focus on algorithmic frameworks, evaluation metrics (eg. accuracy, F1-score, AUC), and practical applications, while emphasizing model interpretability and real-time decision support.

This study presents the first comprehensive synthesis of multi-dimensional influencing factors and model evolution pathways for traffic accident severity prediction. A novel ”data-algorithm-scenario” adaptation framework is proposed to advance intelligent transportation systems. The paper is structured as follows: Section 2 analyzes severity classification criteria and key influencing factors; Section 3 evaluates model performance and limitations; Section 4 discusses optimization strategies and future directions; and Section 5 concludes with policy recommendations. By integrating emerging algorithms and multi-source data, this research provides a scientific foundation and actionable insights for reducing traffic casualties and optimizing urban planning.

2 ANALYSIS OF INFLUENCING FACTORS

2. 1 Traffic Accident Severity Classification

As per the classification criteria for human injury severity promulgated by the Ministry of Justice of the People’s Republic of China, human injuries are categorized into three tiers: minor, moderate, and severe[17]. In prior research, Kashani[18] delineates minor injury accidents and fatal or severe injury accidents based on the severity of injuries sustained by the most affected passengers, aligning with the two-tiered severity scale.

2. 2 Key Factors Affecting the Severity of the Accident

Kashani [19] has conducted an analysis to pinpoint the primary factors contributing to the severity of traffic accidents, considering a comprehensive set of 19 variables. These variables encompass demographic and behavioral aspects of drivers (such as age and gender), roadway characteristics (including pavement width, lane width, shoulder width, road type, road markings, sight distance, and safety barriers), accident-specific details (pertaining to time, day, month, crew size, number of injured individuals, cause, and type of accident), vehicular information (type of vehicle involved), and ambient conditions (lighting and atmospheric factors).

Haq[20] posits that the severity of traffic accidents is significantly influenced by a multitude of factors, which include the specific location of the accident, the state of the road surface, prevailing weather conditions, vehicular speed at the time of the incident, the adequacy of safety infrastructure, and temporal elements such as the time of day, with particular emphasis on night-time occurrences.

In research on traffic accident prediction, models usually identify the following key influencing factors:

- Road user vulnerability:Pedestrians and non-motorised drivers are more likely to be seriously injured in traffic accidents.
- Environmental factors:Such as weather conditions and road surface conditions, which have a significant impact on the occurrence of crashes and severity of injuries.
- Traffic infrastructure:Including road design, traffic signals and signs, are critical to preventing crashes.
- Time of day factors:The frequency and gravity of collisions tend to escalate during the morning and evening rush hours. Recognizing these patterns is essential for crafting strategic traffic policies and safety initiatives.
- Road conditions:These include pavement conditions, road design, lane widths, shoulder conditions, etc. , which may affect driver safety.
- Traffic flow:Traffic accidents are particularly influenced by vehicular density and circulation patterns, with a heightened effect observed during periods of peak congestion.
- Weather conditions:Rain, snow, fog, temperature and other weather factors may reduce visibility and increase roadway slippage, thereby increasing the risk of accidents.
- Light conditions:Driving at night or in low light conditions may obstruct a driver’s vision, increasing the likelihood of an accident.
- Driver characteristics:Including age, gender, driving experience, driving behaviour (eg. speeding, drink driving, disobeying traffic rules), etc.
- Vehicle characteristics:Vehicle type, size, weight, braking system and other vehicle-related factors can also affect the occurrence and severity of an accident.
- Traffic signals and signs: The setting, visibility and functional status of traffic signals guide driver behaviour.
- Road user behaviour:This includes the behaviour of pedestrians, cyclists and other road users whose actions can be unpredictable and increase the risk of accidents.
- Emergencies: Emergencies such as vehicle breakdowns and animals crossing the road may cause drivers to react suddenly, increasing the likelihood of accidents.
- Socio-economic factors:The socio-economic status of residents may affect their choice of transport mode and the likelihood of complying with traffic rules.

Incorporating these variables within traffic accident predictive models enhances the precision and dependability of forecasts. Such an analysis facilitates a deeper comprehension of the causal factors behind accidents, thereby informing the development of preventative strategies.

Variability exists in the focus of influencing factors across traffic accident prediction studies, which is contingent upon multiple factors such as research objectives, data accessibility, geographic context, and cultural disparities. Below are some specific examples, along with their representation in different studies and corresponding references:

Some Depending on the purpose of the study, Studies may focus on specific crash types or specific prediction tasks (eg. short-term prediction, long-term trend analysis, etc.). For example, some studies may focus on identifying high-risk driving behaviours, while others may be more concerned with the impact of road design on crashes[21]. Some of them are also due to the limitation of dataset characteristics. Different datasets may contain different characteristics such as weather conditions, traffic flow, road type, etc. , which may affect which factors the researcher chooses to analyse[22]. Some of them are in the consideration of the research methodology. The choice of research methodology also affects the influences considered. For example, deep learning methods may automatically extract features, whereas traditional statistical methods may require manual selection and design of features[23]. Some of this is due to different choices of geographic location. Traffic patterns and causes of accidents may vary from one area to another, and this may influence the influences that researchers focus on. For example, studies in urban areas may focus more on traffic congestion, while rural areas may focus more on road conditions and wildlife presence[24]. Some of them are due to cultural differences. Driving culture and behaviour in different regions may affect the pattern of traffic accidents, and researchers may take these factors into account to improve the accuracy of the model[25]. Others are due to factors that are unique or less commonly studied in research. Some studies may explore such unique or less studied factors as drivers psychological states, microenvironmental characteristics of the road (eg. roadside vegetation, noise levels), and so on [26].

In summary, most articles consider road user vulnerability, traffic flow, road conditions, weather conditions, lighting conditions, driver characteristics, vehicle characteristics, traffic signals and signs, time factors, socio-economic factors, urban planning, and transport policies as common influencing factors, but each article may choose to focus on specific influencing factors based on its research objectives, dataset characteristics, research methodology, and geographic location, among other factors. In addition, some studies may also explore other unique or less studied factors to provide new insights or improve the performance of predictive models.

3 MODEL EVALUATION METRICS

Numerous scholars utilize a variety of evaluative criteria and metrics to assess the performance of models. Precision[27], a pivotal metric, reflects the proportion of accurate predictive outcomes. Sensitivity[28], also known as the true positive rate, indicates the model's capability to accurately identify cases of severe injury. Specificity[29] measures the model's accuracy in distinguishing non- serious or non-injurious events. The F-score [30], serving as the harmonic mean of precision and recall, provides a comprehensive assessment of the model's overall efficacy. Lastly, the Area Under the Curve (AUC)[31], specifically the region under the Receiver Operating Characteristic (ROC) curve, quantifies the model's ability to differentiate between various outcome categories.

Castro and Kim[32] conducted an in-depth analysis of 81, 960 traffic incidents in the UK from 2010 to 2012, focusing on the predictive accuracy of accident severity. They leveraged an array of machine learning techniques, such as Bayesian networks, decision trees, and multi-layer perceptrons, for their predictive modeling. In their quest for a holistic model performance assessment, they relied on a suite of indicators beyond mere accuracy, incorporating precision, recall, and the F-measure as critical benchmarks.

In an effort to shed light on the efficacy of various methodologies and identify those with superior performance in the realm of crash injury severity prediction, we embark on a comparative analysis of 96 scholarly articles spanning from 2001 to 2024, addressing two pivotal research inquiries. Our primary objective is to juxtapose the efficacy of diverse algorithms, thereby discerning the most effective approaches for predicting crash severity. For each study under review, we delineate the spectrum of predictive models deployed, along with their respective algorithmic frameworks and distinguishing characteristics. Concluding our analysis, we synthesize a concise overview of the prevalent and impactful methodologies gleaned from the extensive examination of over 20 studies. This synthesis aims to distill the essence of the most efficacious strategies in the field, fostering a deeper comprehension of their predictive capabilities.

4 DIFFERENT TYPES OF PREDICTIVE MODELS FOR ROAD CRASH INJURY SEVERITY PREDICTION

In an effort to shed light on the efficacy of various methodologies and identify those with superior performance in the realm of crash injury severity prediction, we embark on a comparative analysis of 96 scholarly articles spanning from 2001 to 2024, addressing two pivotal research inquiries. Our primary objective is to juxtapose the efficacy of diverse algorithms, thereby discerning the most effective approaches for predicting crash severity. For each study under review, we delineate the spectrum of predictive models deployed, along with their respective algorithmic frameworks and distinguishing characteristics. Concluding our analysis, we synthesize a concise overview of the prevalent and impactful methodologies gleaned from the extensive examination of over 20 studies. This synthesis aims to distill the essence of the most efficacious strategies in the field, fostering a deeper comprehension of their predictive capabilities.

4.1 Different Types of Predictive Modelling

Among the 96 scrutinized studies, a plethora of modeling techniques for forecasting traffic accidents have been identified, categorizable into several key paradigms: traditional Statistical Models, advanced Machine Learning Techniques, cutting-edge Deep Learning Approaches, Graph Neural Networks (GNNs), Spatio-Temporal Models, Hybrid Models that integrate various methodologies, Multi-Modal Forecasting which leverages diverse data streams, Multi-Process Information Fusion that amalgamates information from multiple sources, Real-Time and Dynamic Graph Structures that adapt to instantaneous data fluctuations, and more. This diverse array of models reflects the evolving landscape of traffic accident prediction, each bringing unique insights and strategies to the forefront of research. These model types do have some crossover and overlap, but they can be categorised based on their main characteristics and application scenarios. Below are the broad categories into which these models are classified along with their corresponding data characteristics and problems addressed:

4.1.1 Traditional statistical methods

Problem Solving: prediction, classification, regression, etc.

Data Types: Usually applied to structured data, such as tabular data.

Traditional statistical methods usually include autoregressive moving average model(ARMA)[33], Linear Regression, autoregressive integral sliding average model (ARIMA)[34], seasonal autoregressive moving average model (SARIMA)[35], multivariate autoregressive moving average (VAR)[36], Logistic Regression, Negative Binomial Regression Model[37], etc.

4.1.2 Machine learning methods

Solve problems: a wide range of prediction and classification tasks, including non-linear problems. Data Types: Applicable to all types of data, including structured, semi-structured, and unstruc- tured data. Algorithmic model:Random Forest handles nonlinear relationships well but can be computationally expensive[38]. Support Vector Regression (SVR) is effective in regression tasks but may not capture temporal dynamics as effectively[39]. Knearest neighbor (KNN) is Simple to implement but may suffer from the curse of dimensionality[40].

4.1.3 Deep learning methods

CNNs is excellent at capturing spatial features but may not fully capture temporal dynamics[41]. Problem solving:CNNs are particularly adept at handling a spectrum of advanced tasks, including but not limited to image recognition, speech processing, and natural language understanding, showcasing their versatility in various domains.

Data Types:CNNs are renowned for their prowess in managing data with grid-like structures, predominantly images and videos, and to some extent, audio. This capability renders CNNs in- dispensable in domains requiring the extraction of grid-like features for predictive analytics, particularly where high-dimensional, spatially-rich data is concerned[42].

4.1.4 Hybrid prediction models

Problem solving:Combining the strengths of different models to improve performance [43].

Data types:The versatility of Hybrid Prediction Models is evident in their ability to accommodate a broad spectrum of data types, contingent upon the specific models integrated within the hybrid system. This flexibility allows them to tackle complex problems that may require the simultaneous analysis of structured data, such as tabular statistics, and unstructured data, like text and images, or even time-series data that necessitates the understanding of temporal dynamics.

Algorithmic model:LSTMCNN Combines the strengths of both LSTM and CNN but can be complex and computationally intensive.

4.1.5 Graph learning methods

Problem solving:Social network analysis, recommender systems, bioinformatics, etc [44].

Data type:graph-structured data, including nodes and edges. Algorithmic model:STGCN Captures spatiotemporal dependencies through graph convolution but may be limited by fixed graph structures.

4.1.6 Sparse Spatio Temporal Dynamic Hypergraph Learning (SSTDHL)

Problem solving: Traffic flow prediction, weather prediction, etc [45].

Data Type: Data with spatial and temporal dimensions. Algorithmic model:A novel framework that addresses the issue of sparse data and captures higherorder dependencies.

4.1.7 Fourier Enhanced Heterogeneous Graph Learning (FEHGCARN)

Algorithmic model: Integrates Fourier transformation and heterogeneous graph learning to capture complex spatiotemporal relationships [46].

Traditional statistical methods are easy to implement and understand but may lack the ability to capture complex spatiotemporal dynamics[47]. Machine learning methods are versatile and can capture nonlinear relationships but may require extensive feature engineering and parameter tuning[48]. Deep learning methods, especially CNNs and RNNs, can automatically learn complex features but can be datahungry and may not generalize well to new situations[49]. Hybrid models combine the strengths of different approaches but can be challenging to design and optimize. Graph learning methods, including SSTDHL and FEHGCARN, can capture complex interactions in data but may require careful graph construction and are computationally intensive. It's important to note that the choice of model often depends on the specific characteristics of the dataset and the prediction task at hand. Newer models like SSTDHL and FEHGCARN show promise in addressing some of the limitations of traditional models, but they may also introduce new challenges related to computational complexity and the need for largescale training data.

In this curated selection of studies, the models range from traditional statistical analyses to cutting-edge machine learning and deep learning algorithms. Each model represents a unique strategy in capturing the complexities of traffic accident data, with some focusing on spatial and temporal dynamics, while others emphasize the integration of multimodal data streams. The researchers involved have contributed significantly to the advancement of predictive modeling by applying these models to real-world traffic scenarios, thereby providing valuable insights into accident causation and prevention. In this curated selection of studies, the models range from traditional statistical analyses to cutting-edge machine learning and deep learning algorithms. Each model represents a unique strategy in capturing the complexities of traffic accident data, with some focusing on spatial and temporal dynamics, while others emphasize the integration of multimodal data streams. The researchers involved have contributed significantly to the advancement of predictive modeling by applying these models to real-world traffic scenarios, thereby providing valuable insights into accident causation and prevention (Table 1-2).

Table 1 The Road Traffic Accident Prediction Models

Model Type	Algorithms	Author
Statistical Models	1. Autoregressive Integrated Moving Average (ARIMA) 2. Negative Binomial Regression 3. Logistic Regression Models	P. B. Kumar et al. (2022)[50] Sarkar et al. (2020) [51] Al-Ghamdi et al. (2001)[52]

Machine Learning Techniques	1. Support Vector Machine (SVM) 2. Decision Trees 3. Random Forest 4. K-Nearest Neighbors (KNN) 5. Neural Networks	Li, Z.et al.(2021)[53] A, Mahdi Rezapour et al. (2013) [54] Yang et al. (2023)[56] Zhang et al. (2020) [57] J. Wang et al. (2019) [28]
Deep Learning Approaches	1. Convolutional Neural Networks (CNNs) 2. Long Short-Term Memory (LSTM) Networks 3. Hybrid Models (CNN-LSTM)	F. Milletari et al. (2024)[58] M. I. Khan et al. (2024) [59] M. M. Kunt et al. (2011) [60]
Graph Neural Networks (GNNs)	1. Graph Convolutional Networks (GCNs) 2. Graph Attention Networks (GATs)	H. Wang et al. (2022)[61] M. Shi et al. (2023)[62]
Spatio-Temporal Models	1. Spatio-Temporal Graph Convo- lutional Networks (STGCNs) 2. Dynamic Spatio-Temporal Graph CNNs 3. Hybrid Models (eg. RNN-GCN) 4. Spatial-Temporal Graph Neural Networks (STGNNs)	Y. Zhang et al. (2022)[63] X. Zhang et al. (2021) [64] H. Zhu et al. (2020)[65] Jia et al. (2024)[66]
Hybrid Models	1. HetGAT (Heterogeneous Graph Attention Network)	Jin et al. (2021)[67]
Multi-Modal Forecasting	1. GAMCN (Graph and Atten- tive Multi-Path Convolutional Net- work)	J. Qi et al. (2022)[68]
Multi-Process Information Fusion	1. Bi-GRCN (Bidirectional-Graph Recurrent Convolu- tional Network) 2. Hybrid Models (eg. RNN-GCN)	H. Zhu et al. (2022)[69] H. Zhu et al. (2020)[65]
Multi-Modal Forecasting	1. GAMCN (Graph and Atten- tive Multi-Path Convolutional Net- work)	J. Qi et al. (2022)[70]

Table 2 The Road Traffic Accident Prediction Models

Model Type	Algorithms	Author
Multi-Process In- formation Fusion	1. Bi-GRCN (Bidirectional-Graph Recurrent Convolutional Network) 2. STGC-GNNs (Spatial-Temporal Granger Causality Graph Neural Networks)	W. Jiang et al. (2022) [69] S. He et al. (2023)[71]
Real-Time and Dynamic Graph Structures	1. TL-DCRNN (Transfer Learning with Diffu- sion Convolutional Recurrent Neural Network)	Zhang et al. (2020) [72]
	2. DetectorNet	H. Li et al. (2021) [73]

Each model type has its unique advantages and limitations, and the most appropriate model is usually chosen based on the nature of the problem and the characteristics of the data. For example, a spatio- temporal model might be chosen if the problem involves complex spatial and temporal dependencies, or a graph neural network might be chosen if the data is characterised by a graph structure. Multimodal prediction and multiprocess information fusion, on the other hand, focus on integrating different types of information to improve the accuracy and robustness of predictions.

According to the latest research, better performing models usually have the following characteris- tics. Integrate multiple data sources, that high-performing predictive models in traffic accident analysis share several distinguishing traits. These models excel by amalgamating data from diverse sources, including traffic flow, velocity, and meteorological data, which allows for a more holistic and precise depiction of the traffic ecosystem. This multifaceted data integration enhances the predictive accuracy of models by providing a richer context for analysis. Advanced models adeptly account for spatio-temporal dependencies, capturing the intricate dynamics of traffic flow and the interplay between different geographic locales. They leverage deep learning techniques, with a particular emphasis on hybrid models that integrate Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Graph Neural Networks (GNNs). These models are particularly effective due to their advanced feature extraction and sequential modeling capabilities, which are crucial for understanding complex traffic patterns. Furthermore, these models are designed to be adaptive and real-time, equipping them to swiftly respond to unpredictable events such as accidents and roadworks. They utilize dynamic graph structures that facilitate the incorporation of real-time traffic data, enabling timely interventions and adjustments. Lastly, these models are not only performant on specific datasets but also possess explanatory and generalization capabilities. They offer insights into the rationale behind their predic- tions, enhancing their transparency and trustworthiness to decision-makers. Moreover, their ability to generalize across various regions makes them versatile tools for traffic management and safety initiatives globally. This convergence of data integration, deep learning, adaptability, and explanatory power positions these models at the forefront of traffic accident prediction, offering innovative solutions for urban planning and traffic governance.

4.2 Traditional Statistical Models

Traditional statistical models are suitable for processing various types of data and informa- tion, suitable for data processing with small amount of data and simple relationship between vari- ables, usually based on strict mathematical theories, the models are relatively simple and easy to under- stand and interpret. However, it is difficult to extend to high-dimensional feature space and is suitable for use in the limited environment of the theater. Traditional statistical models are widely used in data analysis and inference, and in traffic accident prediction, regression modeling is the most commonly used technique to identify risk factors associated with the severity of road traffic accidents[52, 65, 74-76]. Followed by cluster analysis models(eg. latent class cluster analysis)[77]. With the development of technology, machine learning models have emerged. In practical applications, traditional statistical mod- els are combined with machine learning models to make full use of the interpretability of traditional statistical models and the powerful predictive ability of machine learning models to adapt to different scenarios and meet the characteristics of multidimensional data(eg. latent class clus-tering analysis)[78].

4.3 Machine Learning Models

Compared to traditional statistical models, machine learning models are based on data-driven approaches that do not require strict assumptions about the distribution of the data or the relationship between the variables, and the models are complex enough to capture non-linear relationships and features of high-dimensional data. Machine learning methods are able to handle more complex functions and therefore can provide accurate predictive models [79].

Over the years, machine learning methods have been widely used in traffic accident prediction, among which Random Forest is an integrated learning method, in traffic accident prediction, Random Forest can be used to identify multiple factors that lead to accidents, such as weather conditions, traffic flow, and type of roadway. Zhang[80]concluded that Random Forest has the highest overall prediction accuracy, and serious collisions also had the highest prediction accuracy, followed by K-nearestneighbors. Singh[81] modeled injury severity of crashes using Multinomial Logit, Decision Trees, and Random Forests, and class balancing using Synthetic Minority Oversampling (SMOTE) and Stochastic Class Balancing. Among the three models, the Random Forest model has the best performance.

Support Vector Machine (SVM), can be used to categorize traffic accident data and predict whether an accident will occur or not. Li[82] developed two models, Support Vector Machine (SVM) and Ordered Probit, and the correct prediction rate of the SVM model was percent of 48. 8 and the correct prediction rate of Ordered Probit model is percent of 44. 0. Thus, the Support Vector Machine model performs better in predicting the severity of collision damage.

The K-Nearest Neighbors (KNN) algorithm is a straightforward machine learning method classified as instance-based learning. It classifies new data points based on the categories of their closest matches in the training set, using distance metrics like Euclidean or Manhattan distance to define ”closeness. ” In traffic accident analysis, KNN can predict accident severity by identifying patterns in historical data. For instance, Beshah and Hill[83] utilized KNN alongside decision trees and Naive Bayes to forecast injury severity in Ethiopian traffic accidents. Their analysis of over 18, 000 accidents demonstrated the high predictive accuracy of these algorithms, with KNN outperforming the others, as indicated by the highest AUC score, suggesting its strong discriminative power for this dataset. Bayesian networks[83, 84], which visually represent variables’ conditional probabilities, facilitate the analysis of complex factor interactions. In traffic accident analysis, these networks model accidents as outcome variables, examining factors like weather and rule adherence. Nodes and directed edges illustrate factor relationships and dependencies. Juan[85]combined Latent Class Clustering to homogenize data with Bayesian Networks to pinpoint severe accident causes. This approach helped identify key variables for predicting serious accidents, with a focus on accident types and visibility distances. In summary, Bayesian networks provide a powerful framework for modeling and analyzing the complex relationships between traffic accidents and various factors. With this model, we can not only identify and understand the key drivers of accidents but also predict and prevent future accidents, thereby improving road safety.

4.4 Deep Learning Models

In addition, there are deep learning techniques, particularly graph neural networks (GNNs), and the processing of spatio-temporal data. They utilise complex network structures to learn patterns in traffic data and make predictions. These models perform well because they are not only able to handle complex spatio-temporal data, but also adapt to the dynamics of the traffic system, while providing a certain degree of interpretability, which contributes to the development of traffic planning and intelligent transport systems. HetGAT (Heterogeneous Graph Attention Network), this model combines Heterogeneous Graph Attention Networks and Time-Expanded Convolutional Networks for modelling the multiscale temporal contextual effects of traffic flow, effectively integrating spatio-temporal factors and improving prediction accuracy[86]. STGC-GNNs leverage spatial-temporal causality to model traffic flow dependencies, ideal for forecasting up to 60 minutes ahead[71]. The Bi-GRCN merges graph convolutions with bidirectional LSTMs to enhance traffic prediction by considering past and future data[69]. ETGCN enhances speed prediction accuracy through integrated graph structures that capture spatial correlations alongside temporal dynamics[87]. SST-DHL employs hypergraph learning with cyclic units to process spatio-temporal data, enhancing model interpretability and prediction precision, especially in sparse datasets[88]. Hybrid models, such as LSTM-CNN, harness the strengths of different neural architectures to analyze both spatial and temporal aspects of traffic data[89]. However, these models tend to focus more on tasks within the sametransportation mode, with less consideration given to the interactionof traffic feature information across different modes.

4.5 Model Comparison

In the comparison of accuracy, stability and generalization ability in traffic accident prediction, the models for predicting the influencing factors of traffic accident severity have their own advantages. In many studies, in terms of accuracy, Random Forest[90-92] stands out for its high accuracy. It improves the accuracy of prediction by constructing multiple decision trees and taking their average prediction results.

Kenny Santos [93] noted that Random Forest outperformed other methods in 70 percent ctiveness. Support Vector Machine (SVM) followed closely, with a 53 percent success rate and recognized for its proficiency in handling nonlinear challenges by utilizing kernel functions to map data into higher dimensions for optimal hyperplane detection. Regarding stability, Bayesian Networks and K-Nearest Neighbors (KNN) excelled; Bayesian Networks, with their probabilistic graphical model framework, maintained robust performance under minor data fluctuations, achieving top results in 67 percent of cases. KNN, attributed to its straightforward approach and swift adaptation to new data, secured the best outcome in 40 percent of applications.

5 DISCUSSION

This paper presents a summary of the key factors influencing traffic crash severity as well as a comparative study of different analytical approaches to traffic crash injury severity prediction through a comprehensive analysis of many academic papers published between 2001 and 2024. The study reveals the strengths and limitations of prevalent models for forecasting the severity of injuries in traffic accidents, along with their suitable use cases, summarized as follows:

- Random forest, with high accuracy and stability, performs best in most studies, especially when there are complex interactions between variables. Especially in the evaluation of feature importance, it can evaluate the impact of each feature on the prediction results. But its interpretability is poor. As an integrated model, its decision-making process is not transparent enough and difficult to explain; It takes a long time and more computing resources to build and train the model. It is suitable for large-scale data sets that require high accuracy prediction.
- The support vector machine algorithm is adept at identifying the most effective hyperplane in a high-dimensional context, making it well-suited for classifying data with intricate boundary definitions. However, the kernel function and parameters need to be selected reasonably, which is sensitive to parameters. It is suitable for scenarios where there are nonlinear relationships in the data set and good generalization ability is required.
- Bayesian networks articulate the probabilistic interconnections among variables, offering valuable decision support. Nonetheless, they require substantial data to accurately estimate these probabilities within the model. They are particularly useful for predictive tasks where the expression of variable probability relationships is essential.
- K-nearest neighbors, without training, use data directly when predicting. For unbalanced datasets, performance may degrade. It is suitable for small-scale datasets or occasions requiring rapid prototyping development.
- Artificial neural networks, nonlinear modeling, can capture complex nonlinear relationships in data, and through learning, the network can adapt to changes in data. However, in order to obtain good performance, it needs a lot of data for training. It is suitable for scenes with complex patterns and a large amount of data, and requires a strong learning ability of the model.

In terms of model selection and application, this paper makes the following recommendations:

- Data-driven model selection. The most appropriate model is selected based on the characteristics of the dataset and the needs of the prediction task. For example, spatio-temporal models may be a better choice for data with significant spatio-temporal dependencies. For example, STGCN (Spatial-Temporal Graph Convolutional Network) is able to capture both spatial and temporal dependencies of traffic data [94].
- Model integration leverages the complementary advantages of various models, such as the synergy between deep learning and machine learning. This approach involves employing deep learning for feature extraction, followed by training machine learning models like SVMs or random forests on these extracted features to bolster predictive accuracy. For instance, CNNs can be utilized to identify key features, which are subsequently used to refine the performance of conventional machine learning algorithms [95].
- Adaptability&Real-time. Developing adaptable and real-time models is crucial for responding swiftly to unforeseen incidents like traffic accidents. These models, which may incorporate dynamic graph structures, are designed to adjust to fluctuations in live traffic data. For example, dynamic graph convolutional networks can modify their structure in response to real-time traffic flow and events, enhancing their predictive capabilities and responsiveness[95].
- Explanatory properties. Selection or development of models with some explanatory power to help decision makers understand the logic behind the predictions and increase the credibility and acceptance of the models. For example,

Bayesian networks, as probabilistic graphical models, can provide explanations of conditional dependencies between variables[32].

The severity of traffic accidents is shaped by numerous interrelated factors, which collectively determine the probability and extent of an incident. Investigators must take into account a range of elements such as traffic flow, road conditions, and weather when modeling accident severity. The selection of an appropriate predictive model hinges on these diverse influencing factors, necessitating a tailored approach based on their specific attributes. For the researcher, the study of influencing factors is somewhat dependent on the data set. Different datasets may contain different features and information. For example, some datasets may contain detailed weather information, while others may contain more comprehensive road design parameters, and thus the researcher needs to select and adapt the model according to the characteristics of the available datasets. At the same time, traffic patterns, driver behaviour and road infrastructure may differ significantly from one region to another due to geographical and cultural differences. For example, factors influencing traffic crashes may differ between urban and rural areas, requiring targeted analysis and modelling.

Future research in predicting the severity of traffic accident injuries could focus on advancing deep learning models. This includes investigating sophisticated network architectures like CNNs and RNNs for analyzing spatial and temporal data characteristics, crucial for deciphering traffic accident patterns. Additionally, combining ensemble learning with deep learning could enhance model precision and stability. Furthermore, developing a multi-task learning framework to forecast injury severity alongside related metrics like accident likelihood and duration may bolster the model's ability to generalize across various scenarios.

6 CONCLUSIONS

This paper identifies key factors in traffic accident severity and evaluates statistical prediction models. It analyzes model strengths and weaknesses to guide the choice of the most suitable model for predicting accident severity. Choosing the right model is essential for traffic accident prevention, as it enables precise predictions and preemptive measures by traffic management, thereby reducing accidents and injury severity. Precise forecasting of high-risk zones and times aids in optimizing traffic monitoring and enforcement, enhancing resource efficiency and road safety policy development.

In traffic accident prediction studies, the selection of influencing factors is based on data avail- ability, theoretical support, geographic location characteristics, cultural differences, model capture capa- bilities, and practical implications for traffic safety management. Key factors typically include traffic volume and density, road and infrastructure conditions, weather and lighting conditions, driver charac- teristics, vehicle characteristics, time factors, socio-economic factors, and urban planning and transport policies. The comprehensive consideration of these factors helps to construct more accurate and practical prediction models, which can provide a scientific basis for traffic planning and management and enhance road safety.

In the future, we will continue to explore new data sources and algorithms to further improve the performance of our predictive models. At the same time, we will also focus on the explanatory and real-time nature of the models to ensure that they are not only accurate, but also able to provide timely insights to decision makers. Through these efforts, we expect to be able to make a greater contribution to reducing traffic accidents and improving road safety, as well as laying a solid foundation for the development of intelligent transport systems.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] World Health Organization. World health statistics 2024: Monitoring health for the SDGs, Sustainable Development Goals. World Health Organization, Geneva, 2024. https://www.who.int/data/gho/publications/worldhealth-statistics.

[2] Abdelwahab HT, Abdel-Aty MA. Development of artificial neural network models to predict driver injury severity in traffic accidents at signalized intersections. Transportation Research Record, 2001, 1746(1): 6-13.

[3] Ahmadi A, Jahangiri A, Berardi V, et al. Crash severity analysis of rear-end crashes in California using statistical and machine learning classification methods. Journal of Transportation Safety & Security, 2020, 12(4): 522-546.

[4] Vajari MA, Aghabayk K, Sadeghian M, et al. A multinomial logit model of motorcycle crash severity at Australian intersections. Journal of Safety Research, 2024, 73: 17-24.

[5] Acevedo MA, Corrada-Bravo CJ, Corrada-Bravo H, et al. Automated classification of bird and amphibian calls using machine learning: A comparison of methods. Ecological Informatics, 2009, 4(4): 206-214.

[6] Al-Moqri T, Haijun X, Namahoro JP, et al. Exploiting machine learning algorithms for predicting crash injury severity in Yemen: Hospital case study. Applied and Computational Mathematics, 2020, 9(5): 155-164.

[7] Arhin SA, Gatiba A. Predicting crash injury severity at unsignalized intersections using support vector machines and naïve Bayes classifiers. Transportation Safety and Environment, 2020, 2(2): 120-132.

[8] Theofilatos A, Chen C, Antoniou C. Comparing machine learning and deep learning methods for real-time crash prediction. Transportation Research Record, 2019, 2673(8): 169-178.

[9] Khalesian M, Furno A, Leclercq L. Improving deep-learning methods for area-based traffic demand prediction via hierarchical reconciliation. Transportation Research Part C: Emerging Technologies, 2024, 159: 104410.

[10] Wu Z. Deep learning with improved metaheuristic optimization for traffic flow prediction. Journal of Computer Science and Technology Studies, 2024, 6(4): 47-53.

[11] Li Y, Li P, Yan D, et al. Deep knowledge distillation: A self-mutual learning framework for traffic prediction. Expert Systems with Applications, 2024, 252: 124138.

[12] Jian F, Tian L, Yuqiang Q. Traffic accident prediction method based on multi-view spatial-temporal learning. Bulletin of the Polish Academy of Sciences: Technical Sciences. 2024, 72(6).

[13] Alhaek F, Li T, Rajeh TM, et al. Encoding global semantic and localized geographic spatial-temporal relations for traffic accident risk prediction. 2024.

[14] Gao X, Haworth J, Ilyankou I, et al. Sma-hyper: Spatiotemporal multi-view fusion hypergraph learning for traffic accident prediction. 2024. arXiv preprint arXiv:2407.17642. 2024.

[15] Trirat P, Yoon S, Lee J-G. Mg-tar: Multi-view graph convolutional networks for traffic accident risk prediction. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(4): 3779-3794.

[16] Zhang D, Li J. Multi-view fusion neural network for traffic demand prediction. Information Sciences, 2023, 646: 119303.

[17] Ministry of Justice of the People's Republic of China. Standards for the Identification of the Degree of Human Injury. Tech. Rep. Announcement, Supreme People's Court Supreme People's Procuratorate Ministry of Public Security Ministry of State Security Ministry of Justice, Beijing. 2013. https://www.moj.gov.cn/pub/sfbgw/zwxxgk/fdzdgknr/fdzdgknrtzwj/201908/t20190816_207509.html.

[18] Kashani AT, Mohaymany AS. Analysis of the traffic injury severity on two-lane, two-way rural roads based on classification tree models. Safety Science, 2011, 49(10): 1314-1320.

[19] De Oña J, Mujalli RO, Calvo FJ. Analysis of traffic accident injury severity on Spanish rural highways using Bayesian networks. Accident Analysis & Prevention, 2011, 43(1): 402-411.

[20] Haq MT, Zlatkovic M, Ksaibati K. Assessment of commercial truck driver injury severity based on truck configuration along a mountainous roadway using hierarchical Bayesian random intercept approach. Accident Analysis & Prevention, 2021, 162: 106392.

[21] Wang B, Lin Y, Guo S, et al. Gsnet: Learning spatial-temporal correlations from geographical and semantic aspects for traffic accident risk forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35: 4402-4409.

[22] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. 2017. arXiv preprint arXiv:1707.01926. 2017.

[23] Guo S, Lin Y, Wan H, et al. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Transactions on Knowledge and Data Engineering, 2021, 34(11): 5415-5428.

[24] Xu Z, Lv Z, Li J, et al. A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. IEEE Intelligent Transportation Systems Magazine, 2022, 15(1): 136-159.

[25] Oeschger G, Carroll P, Caulfield B. Micromobility and public transport integration: The current state of knowledge. Transportation Research Part D: Transport and Environment, 2020, 89: 102628.

[26] Wu A, Nowozin S, Meeds E, et al. Deterministic variational inference for robust Bayesian neural networks. 2018. arXiv preprint arXiv:1810.03958. 2018.

[27] Selvaraj S, Ramani G. Classification of seating position specific patterns in road traffic accident data through data mining techniques. In: Second International Conference on Computer Applications, 2012.

[28] Wang J, Kong Y, Fu T. Expressway crash risk prediction using back propagation neural network: A brief investigation on safety resilience. Accident Analysis & Prevention, 2019, 124: 180-192.

[29] Mokhtarimousavi S, Anderson JC, Azizinamini A, et al. Improved support vector machine models for work zone crash injury severity prediction and analysis. Transportation Research Record: Journal of the Transportation Research Board, 2019, 2673(11): 680-692.

[30] Pervez A, Huang H, Lee J, et al. Factors affecting injury severity of crashes in freeway tunnel groups: A random parameter approach. Journal of Transportation Engineering, Part A: Systems, 2022.

[31] Prati G, Pietrantoni L, Fraboni F. Using data mining techniques to predict the severity of bicycle crashes. Accident Analysis and Prevention, 2017, 101: 44-54.

[32] Castro Y, Kim YJ. Data mining on road safety: Factor assessment on vehicle accidents using classification models. International Journal of Crashworthiness, 2016, 21(2): 104-111.

[33] Wei WWS. Time Series Analysis. Addison-Wesley, 2006.

[34] Box GEP, Jenkins GM, Reinsel GC. Time Series Analysis: Forecasting and Control. John Wiley & Sons, 1976.

[35] Hyndman RJ, Athanasopoulos G. Forecasting: Principles and Practice. OTexts, 2018.

[36] Lütkepohl H. New Introduction to Multiple Time Series Analysis. Springer, 2005.

[37] Lord D, Mannering F. The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives. Transportation Research Part A: Policy and Practice, 2010, 44(5): 291-305.

[38] Ospina-Mateus H, Quintana Jiménez LA, Lopez-Valdes FJ, et al. Extraction of decision rules using genetic algorithms and simulated annealing for prediction of severity of traffic accidents by motorcyclists. Journal of Ambient Intelligence and Humanized Computing, 2021, 1: 3.

[39] Dong N, Huang H, Zheng L, et al. Support vector machine in crash prediction at the level of traffic analysis zones: Assessing the spatial proximity effects. Accident Analysis & Prevention, 2015, 82: 192-198.

[40] Lv Y, Tang S, Zhao H, et al. Real-time highway traffic accident prediction based on the k-nearest neighbor method. In: 2009 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '09). IEEE, 2009.

[41] Wang W, Wang Y, Kweon Y-J. Spatial-temporal graph convolutional networks for traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(12): 7891-7903.

[42] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735-1780.

[43] Ioannou Y, Underwood J. A review of machine learning approaches to traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(1): 337-350.

[44] Zhang D, Yan J, Polat K, et al. Multimodal joint prediction of traffic spatial-temporal data with graph sparse attention mechanism and bidirectional temporal convolutional network. Advanced Engineering Informatics, 2024, 62: 102533.

[45] Cui P, Yang X, Abdel-Aty M, et al. Advancing urban traffic accident forecasting through sparse spatio-temporal dynamic learning. Accident Analysis and Prevention, 2024, 200: 107564.

[46] Zhang W, Wang H, Zhang F. Spatio-temporal Fourier enhanced heterogeneous graph learning for traffic forecasting. Expert Systems with Applications, 2024, 241: 122766.

[47] Santos K, Dias JP, Amado C. A literature review of machine learning algorithms for crash injury severity prediction. Journal of Safety Research, 2022, 80: 254-269.

[48] Pearl J. The seven tools of causal inference, with reflections on machine learning. Communications of the ACM, 2019, 62(3): 54-60.

[49] Chen C, Antoniou C. Comparing machine learning and deep learning methods for real-time crash prediction. Transportation Research Record, 2019, 2673(8): 169-178.

[50] Kumar PB, Hariharan K. Time series traffic flow prediction with hyper-parameter optimized ARIMA models for intelligent transportation system. Journal of Scientific & Industrial Research, 2022, 81(4): 408-415.

[51] Sarkar A, Sarkar S. Comparative assessment between statistical and soft computing methods for accident severity classification. Journal of The Institution of Engineers (India) Series A, 2019, 101(1).

[52] Al-Ghamdi AS. Using logistic regression to estimate the influence of accident factors on accident severity. Accident Analysis & Prevention, 2002.

[53] Li Z, Liu P, Wang W, et al. A hybrid machine learning approach for crash prediction with imbalanced data. Accident Analysis & Prevention, 2021, 159: 106256.

[54] Abellán J, López G, Oa JD. Analysis of traffic accident severity using decision rules via decision trees. Expert Systems with Applications, 2013, 40(15): 6047-6054.

[55] A M M A, B K K. Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. International Journal of Transportation Science and Technology, 2020, 9(2): 89-99.

[56] Yang J, Han S, Chen Y. Prediction of traffic accident severity based on random forest. Journal of Advanced Transportation, 2023.

[57] Zhang J, Li Z, Pu Z, et al. Comparing prediction performance for crash injury severity among various machine learning and statistical methods. IEEE Access, 2018: 1-1.

[58] Milletari F, Navab N, Ahmadi SA. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). IEEE, 2016.

[59] Khan MN, Das A, Ahmed MM. Prediction of truck-involved crash severity on a rural mountainous freeway using transfer learning with ResNet-50 deep neural network. Journal of Transportation Engineering, Part A: Systems, 2024, 150(2): 04023131.

[60] Kunt MM, Aghayan I, Noii N. Prediction for traffic accident severity: Comparing the artificial neural network, genetic algorithm, combined genetic algorithm and pattern search methods. Transport, 2011, 26(4): 353-366.

[61] Wang H, Lian D, Liu W, et al. Powerful graph of graphs neural network for structured entity analysis. World Wide Web, 2022, 25(2): 609-629.

[62] Shi M, Tang Y, Zhu X, et al. Genetic-gnn: Evolutionary architecture search for graph neural networks. Knowledge -Based Systems, 2022, 247: 108752.

[63] Zhang Y, Zhao T, Gao S, et al. Incorporating multimodal context information into traffic speed forecasting through graph deep learning. International Journal of Geographical Information Science, 2023, 37(9): 1909-1935.

[64] Zhang X, Huang C, Xu Y, et al. Traffic flow forecasting with spatial-temporal graph diffusion network. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35: 15008-15015.

[65] He S, Luo Q, Du R, et al. Stgc-gnns: A gnn-based traffic prediction framework with a spatial-temporal granger causality graph. Physica A: Statistical Mechanics and its Applications, 2023, 623: 128913.

[66] Jia H, Yu Z, Chen Y, et al. Spatial-temporal multi-factor fusion graph neural network for traffic prediction. Applied Intelligence, 2024, 19: 54.

[67] Jin C, Ruan T, Wu D, et al. HetGAT: A heterogeneous graph attention network for freeway traffic speed prediction. Journal of Ambient Intelligence and Humanized Computing, 2021: 1-12.

[68] Qi J, Zhao Z, Tanin E, et al. A graph and attentive multi-path convolutional network for traffic prediction. IEEE Transactions on Knowledge and Data Engineering, 2022, 35.

[69] Jiang W, Xiao Y, Liu Y, et al. Bi-grcn: A spatio-temporal traffic flow prediction model based on graph neural network. Journal of Advanced Transportation, 2022, 2022(1): 5221362.

[70] Qi J, Zhao Z, Tanin E, et al. A graph and attentive multi-path convolutional network for traffic prediction. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(7): 6548-6560.

[71] He S, Luo Q, Du R, et al. Stgc-gnns: A gnn-based traffic prediction framework with a spatial-temporal granger causality graph. Physica A: Statistical Mechanics and its Applications, 2023, 623: 128913.

[72] Zhang Z, Yang H, Yang X. A transfer learning-based LSTM for traffic flow prediction with missing data. Journal of Transportation Engineering, Part A: Systems, 2023, 149(10): 04023095.

[73] Li H, Zhang S, Li X, et al. Detectornet: Transformer-enhanced spatial temporal graph neural network for traffic prediction, 2021.

[74] de Dieu J, Gatesi, Bin S, et al. Analysis and modelling of the contributing factors associated with road traffic crashes in Rwanda. International Journal of Crashworthiness, 2024: 1-13.

[75] Chen Z, Li Y, Wang H, et al. Region-aware text-to-image generation via hard binding and soft refinement. 2024. arXiv preprint arXiv:2411.06558. 2024.

[76] John C, Milton, Venky N, et al. Highway accident severities and the mixed logit model: An exploratory empirical analysis. Accident Analysis & Prevention, 2008, 40(1): 260-266.

[77] Gopinath V, Prakash KP, Yallamandha C, et al. Traffic accidents analysis with respect to road users using data mining techniques.

[78] Yuan Y, Yang M, Guo Y, et al. Risk factors associated with truck-involved fatal crash severity: Analyzing their impact for different groups of truck drivers. Journal of Safety Research, 2020.

[79] Mokhtarimousavi S, Anderson JC, Azizinamini A, et al. Factors affecting injury severity in vehicle-pedestrian crashes: A day-of-week analysis using random parameter ordered response models and artificial neural networks. International Journal of Transportation Science and Technology, 2020, 9(2): 100-115. https://www.sciencedirect.com/science/article/pii/S2046043020300034.

[80] Zhang J, Li Z, Pu Z, et al. Comparing prediction performance for crash injury severity among various machine learning and statistical methods. IEEE Access, 2018, 6: 60079-60087.

[81] Singh G, Sachdeva S, Pal M. Comparison of three parametric and machine learning approaches for modeling accident severity on non-urban sections of Indian highways. Advances in Transportation Studies, 2018, 45.

[82] Li Z, Liu P, Wang W, et al. Using support vector machine models for crash injury severity analysis. Accident Analysis & Prevention, 2012, 45: 478-486.

[83] Li J, Xu S, Guo J, et al. Explanatory prediction of injury severity in traffic incidents: A hybrid approach with latent class clustering and causal Bayesian network. 2023.

[84] Zou, X., & Yue, W. L. A bayesian network approach to causation analysis of road accidents using netica. Journal of advanced transportation, 2017(1), 2525481.

[85] De Ona J, Lopez G, Mujalli R, et al. Analysis of traffic accidents on rural highways using latent class clustering and Bayesian networks. Accident Analysis & Prevention, 2013, 51: 1-10.

[86] Jin C, Ruan T, Wu D, et al. Hetgat: A heterogeneous graph attention network for freeway traffic speed prediction. Journal of Ambient Intelligence and Humanized Computing, 2021: 1-12.

[87] Zhang Z, Li Y, Song H, et al. Multiple dynamic graph based traffic speed prediction method. Neurocomputing, 2021, 461: 109-117.

[88] Cui P, Yang X, Abdel-Aty M, et al. Advancing urban traffic accident forecasting through sparse spatio-temporal dynamic learning. Accident Analysis & Prevention, 2024, 200: 107564.

[89] Li P, Abdel-Aty M, Yuan J. Real-time crash risk prediction on arterials based on lstm-cnn. Accident Analysis & Prevention, 2020, 135: 105371.

[90] Wei Z, Zhuping Z, Lei L, et al. Identifying significant injury severity risk factors in traffic accidents based on the machine learning methods. CICTP 2019, 2019.

[91] Lee J, Yoon T, Kwon S, et al. Model evaluation for forecasting traffic accident severity in rainy seasons using machine learning algorithms: Seoul city study. Applied Sciences, 2019, 10(1): 129.

[92] Al-Moqri T, Haijun X, Namahoro JP, et al. Exploiting machine learning algorithms for predicting crash injury severity in Yemen: Hospital case study. Applied and Computational Mathematics, 2020, 9(5): 155-164.

[93] Santos K, Dias JP, Amado C. A literature review of machine learning algorithms for crash injury severity prediction. Journal of Safety Research, 2022, 80: 254-269.

[94] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. 2017. arXiv preprint arXiv:1709.04875. 2017.

[95] Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, 31.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3042

ENHANCING NAMED ENTITY RECOGNITION VIA TEST-TIME SCALING MODEL

JiaYi Ning*, YiLin Cai, AiLing Hou
Faculty of Science and Technology, Beijing Normal University & Hong Kong Baptist University United International College, Zhuhai 519088, Guangdong, China.
Corresponding Author: JiaYi Ning, Email: njyggs@qq.com

Abstract: This paper addresses the challenge of Named Entity Recognition (NER) using large language models (LLMs) in zero-shot and few-shot settings. While LLMs demonstrate promising capabilities, they often generate hallucinations—spurious or inaccurate outputs—that hinder reliable performance. To overcome this limitation, we propose use chain-of-thought scaling approach in which the model explicitly reasons through an inferred thought process prior to outputting final entity labels. We evaluate our method on the CoNLL-2003 and FewNERD benchmarks, demonstrating consistent performance gains over strong baseline models and attaining an F1 improvement in FewNERD from 0.45 to 0.55 in zero-shot NER. Our findings suggest that explicitly structured reasoning significantly mitigates hallucinations and enhances label precision, even without extensive task-specific fine-tuning. This work provides a blueprint for scaling and refining NER in resource-constrained scenarios, and paves the way for broader applications of reasoning-based LLM strategies to complex information extraction tasks.

Keywords: Named entity recognition; Test-time scaling; Large language model; Zero-shot

1 INTRODUCTION

The exponential growth of data from diverse sources offers both remarkable opportunities and substantial challenges for knowledge extraction from unstructured text[1]. Named Entity Recognition (NER)—a cornerstone of Information Extraction (IE)—plays a pivotal role in extracting structured entities (e.g., people, locations, organizations) from unstructured data[2]. Accurate and efficient NER is crucial for a wide range of downstream applications, including knowledge base construction, question answering, and information retrieval[3]. However, achieving high performance in NER typically relies on sophisticated models with extensive training data and domain-specific annotations, which can be both costly and time-consuming to acquire[3].

Recent advances in large-scale language models (LLMs) have significantly pushed state-of-the-art across various natural language processing (NLP) tasks. The latest generations of such models, exemplified by GPT-3.5 and GPT-4, exhibit remarkable generalization capabilities, often performing well in zero-shot or few-shot settings. Despite these promising results, a persistent challenge is the tendency of LLMs to produce hallucinations—outputs that contain information unsupported by or contradictory to the provided input[4]. Within the NER domain, these hallucinations can manifest as incorrect entity boundaries or misclassified entities, undermining the reliability of LLM-based systems.

The hallucinatory behavior of LLMs arises from their inherent complexity and the uncertainty introduced by incomplete context or ambiguous input. While in-context learning (ICL) has proven effective in some tasks, the inability to easily manipulate or optimize the underlying model during inference poses a significant challenge. Attempts to control LLM predictions using few-shot demonstrations can be unpredictable[5,6].

In this work, we propose the use of models based on reason using chain-of-thought scale. This class of models scales on complex reasoning problems, mitigating the bias and inaccuracy of these models in performing reasoning. Specifically, we let this powerful LLM first elicit the explanatory inference step before proceeding to the final entity prediction. We evaluate our approach on two established datasets: CoNLL-2003[7], representing a well-studied and standard benchmark, and FewNERD[8], representing a more granular, few-shot scenario. Our experimental results demonstrate a notable improvement in zero-shot F1 score—from 0.45 to 0.55—highlighting the efficacy of this structured reasoning paradigm in mitigating hallucinations and boosting extraction accuracy.

In summary, our study provides both empirical and conceptual insights into the use of chain-of-thought-based reasoning for NER. We illustrate how scaling and structuring the model's thought process can reduce hallucinations and enhance precision without requiring extensive task-specific training. Our main contributions are threefold:

1. We identify one factor that lead to LLM hallucinations in NER and demonstrate how chain-of-thought reasoning can alleviate these issues.
2. We propose a reason using chain-of-thought scale approach that systematically improves NER performance in zero-shot settings.
3. We validate our method on the CoNLL-2003 and FewNERD benchmarks, showing notable gains in zero-shot NER performance, thus reinforcing the promise of structured reasoning for robust information extraction.

Through these contributions, we aim to advance the ongoing dialogue on the potential of LLMs for reliable information extraction and pave the way for further research on interpretive, transparent model reasoning in complex NLP tasks.

2 RELATED WORK

2.1 Named Entity Recognition

Named Entity Recognition (NER) is a task that aims to identify key information in text and classify it into predefined categories. NER is commonly defined as a sequence segmentation task, which can be solved through sequence labeling methods. In the past, most approaches relied on supervised learning to segment and construct undirected graph features. Typical methods include LSTM, CDF[9–11].

In recent years, there has been a growing trend towards fine-tuning pre-trained language models by incorporating additional semantic information and designing task-specific architectures. This shift from traditional symbolic approaches to neuroconnectionism has been gradual[12–14].

In recent years, the rapid development of efficient architectures for training decoders in large-scale language models has led to a gradual realization of the powerful capabilities embodied in ultra-large language models for MRC-like tasks, resulting in a growing number of studies biased towards this direction. Representative methods include prompt, which models NER as a question-and-answer problem. However, readout based on large models often suffers from overconfidence[13,15,16]. This paper will address the discrepancy between the matching form in the prompt and the original task by introducing a decoding architecture, which is expected to yield more accurate results.

2.2 Test-Time Scaling

Recent advances in language models (LMs) have been predominantly driven by scaling self-supervised pretraining using massive computational resources[17,18]. This paradigm has enabled the creation of powerful models that, in turn, have sparked interest in new test-time expansion strategies, wherein additional computational steps during inference are employed to enhance performance.

Building on the success of self-supervised pretraining, various studies have investigated the feasibility of expanding computation at test time to boost model accuracy and robustness[19,20]. These works suggest that conventional one-shot inference may not fully exploit a model's latent potential, and that iterative or otherwise extended inference steps—whether by re-ranking outputs or refining intermediate representations—can yield sizable gains.

A pivotal milestone in test-time scaling has been demonstrated by OpenAI's O1 system[21]. O1 is notable for its strong reasoning abilities and the consistent performance improvements it achieves through extended test-time computation. According to OpenAI, this approach relies on large-scale reinforcement learning (RL), which necessitates vast datasets and high-capacity models. By systematically increasing the complexity of inference, O1 showcases how further gains can be realized even after substantial pretraining.

Multiple efforts have emerged to replicate O1-level performance using different RL algorithms. Notably, Monte Carlo Tree Search has been widely adopted for guiding the search process in language model inference[22,23]. Through iterative evaluation and branching, these methods adaptively refine the generation path, leveraging additional computations to improve the final output. While effective, these approaches demand considerable computational overhead, which can limit their practicality in real-world scenarios.

Another branch of work employs multi-agent systems to distribute the inference task across several specialized models[24–26]. By allowing agents to cooperate—or compete—through communication protocols, these frameworks can reduce the burden on any single model. They thereby enable more targeted exploitation of test-time computation, although careful orchestration is required to avoid redundant or conflicting information exchange.

Among these various methodologies, DeepSeek R1[27] stands out for its success in replicating O1-level results by employing multiple RL training stages over millions of samples. This multi-phase RL procedure strategically refines the model's parameter space, pushing it closer to the performance envelope demonstrated by O1. The DeepSeek R1 approach underscores how test-time expansion, when combined with massive training data and robust optimization, can yield powerful inference capabilities that rival or match leading systems (Figure 1).

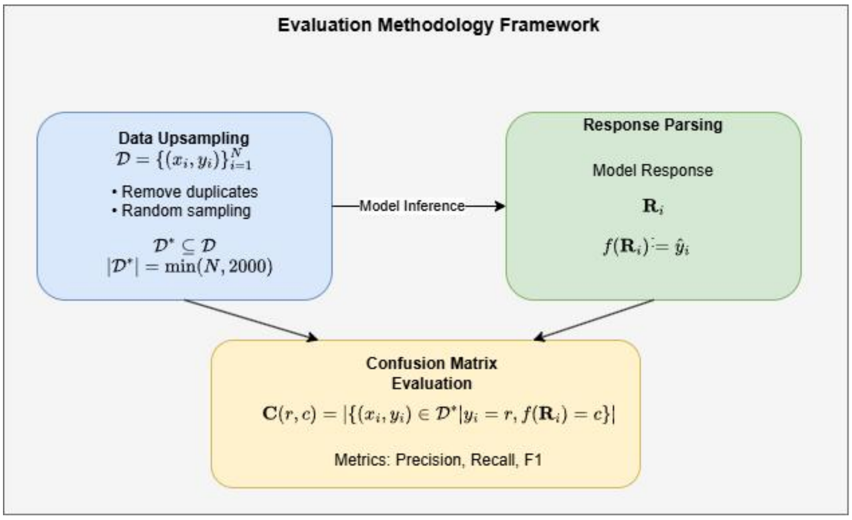

Evaluation Methodology Framework
Data Upsampling
$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
• Remove duplicates
• Random sampling
$\mathcal{D}^* \subseteq \mathcal{D}$
$|\mathcal{D}^*| = \min(N, 2000)$
Model Inference
Response Parsing
Model Response
$\mathbf{R}_i$
$f(\mathbf{R}_i) = \hat{y}_i$
Confusion Matrix
Evaluation
$\mathbf{C}(r, c) = |\{(x_i, y_i) \in \mathcal{D}^* | y_i = r, f(\mathbf{R}_i) = c\}|$
Metrics: Precision, Recall, F1

Figure 1 Methodology Framewor

3 METHODOLOGY

3.1 Data Upsampling

When evaluating our model's performance, we face a trade-off between using too few test samples—leading to poor coverage and potentially high variance—and using too many samples, which becomes expensive and rate-limited by the OpenAI API. Many of our datasets have fewer than 2,000 test examples after deduplication; hence, our strategy is to set 2,000 as the maximum number of test instances.

Formally, we define

$$\mathcal{D}^* \subseteq \mathcal{D} \text{ such that } |\mathcal{D}^*| = \min(N, 2000) \tag{1}$$

We first remove near-duplicate or exact-duplicate entries from D to avoid skewed distributions and inflated performance metrics.

If N>2000, we randomly sample 2,000 records for evaluation; otherwise, we use all N records.

This approach balances computational cost, API rate limits, and coverage of diverse test examples. It ensures that our evaluations remain reliable while avoiding excessive resource consumption.

3.2 Formalized Model Response Parsing and Confusion Matrix-Based Evaluation

We need a standardized procedure to extract final predictions from the model's JSON-formatted responses, which contain both a "content" field (the prediction) and a "reasoning_content" field. In our evaluation, we focus exclusively on the "content" field as the source of the model's predicted label. We then compare these predictions against gold-standard labels to compute a confusion matrix and derive standard performance metrics.

Let $\mathbf{R}_i$ be the JSON response produced by the model for test example (x_i, y_i) in $\mathcal{D}^*$.

where $\hat{y}_i$ is the predicted label and r_i is any intermediate reasoning. We define the function

$$f(\mathbf{R}_i) = \hat{y}_i \tag{2}$$

indicating that the final prediction $\hat{y}_i$ is extracted only from the "content" field while the "reasoning_content" field r_i is ignored.

To quantify performance, we construct a confusion matrix $\mathbf{C}$ where the entry $\mathbf{C}(r, c)$ represents the number of times the gold label is r while the predicted label is c :

$$\mathbf{C}(r, c) = |\{(x_i, y_i) \in \mathcal{D}^* | y_i = r, f(\mathbf{R}_i) = c\}| \tag{3}$$

For each $\mathbf{R}_i$, extract only the "content" field, treating it as $\hat{y}_i$.Compare $\hat{y}_i$ with the gold label y_i. For label pairs ($y_i, \hat{y}_i$), increment the corresponding entry $\mathbf{C}(y_i, \hat{y}_i)$ in the confusion matrix.

4 EXPERIMENT

4.1 DataSets

To evaluate the proposed architecture, we experiment on the publicly available dataset CoNLL-2003 and FewNERD from the named entity recognition task.

CoNLL03: In this dataset, there are four types of entities: Locations (LOC), Miscellaneous Entities (MISC), Organizations (ORG), and Persons (PER). In this experiment, we choose the development set of English data as our development set and the test set of English data as the test set.

FewNERD: Ding et al. [8]propose a large scale dataset Few-NERD for few-shot NER, which contains 66 fine-grained entity types across 8 coarsegrained entity types. It contains intra and inter tasks where the train/dev/test sets are divided according to the coarse-grained and fine-grained types, respectively.

Table 1 shows the statistics of each dataset.

Table 1 The Statistics of Datasets

Dataset	Sents	Ents(types)
CoNLL03	22.1k	35.1k(4)
FewNERD	118.2k	491.7k(66)

4.2 Task Setting

4.2.1 prompt

The prompts designed in this paper all consist of five main elements: task instruction, candidate target labels, output format description, demonstration examples, and input text. The task instruction describes the specific IE subtask, candidate target labels are the types of target information, such as entity types, relation types, etc. The output format description specifies the format of outputs to facilitate the distinguishing of target information

4.2.2 Methods to Compare

Table 2 summarizes the comparative results of various large language models (LLMs) on two benchmark datasets, CoNLL-2003 and FewNERD. We evaluate four models—GPT-3.5, GPT-4, Qwen-32B, and Qwen-32B-Reason—and report the F1 score along with the relative performance ΔF1, where ΔF1 indicates the F1 difference compared to the Qwen-32B-Reason.

4.3 Performance

On CoNLL-2003, the best-performing model is Qwen-32B-Reason[28] with an F1 score of 76.16. In comparison, GPT-3.5 and GPT-4 achieve F1 scores of 60.10 and 72.30, respectively; both lag behind Qwen-32B-Reason by 16.06 and 3.86 points. Qwen-32B without the reasoning mechanism obtains 70.85 F1, a 5.31-point deficit relative to Qwen-32B-Reason.

A similar pattern emerges on FewNERD, where Qwen-32B-Reason remains the reference baseline (F1 = 55.30, not shown directly in the table), and GPT-3.5 and GPT-4 reach F1 scores of 31.56 and 47.84 respectively, yielding negative margins (−23.74 and −7.46). Meanwhile, Qwen-32B attains an F1 of 45.72, which is 9.58 points below Qwen-32B-Reason.

These findings suggest that incorporating a reasoning mechanism into Qwen-32B[29] consistently enhances entity recognition performance across both standard (CoNLL-2003) and fine-grained (FewNERD) benchmarks. GPT-3.5 and GPT-4, while demonstrating competitive results, do not surpass Qwen-32B-Reason, indicating the potential effectiveness of specialized chain-of-thought or reasoning components in improving zero-shot and few-shot NER tasks (Figure 2).

Table 2 The Main Result

Dataset	Model	F1	$\Delta F1$
CoNLL03	GPT-3.5	60.10	-16.06
	GPT-4	72.30	-3.86
	Qwen-32B	70.85	-5.31
	Qwen-32B-Reason	76.16	-
FewNERD	GPT-3.5	31.56	-23.74
	GPT-4	47.84	-7.46
	Qwen-32B	45.72	-9.58
	Qwen-32B-Reason	55.30	-

Note: ΔF1 mean F1 compare to Qwen-32B-Reasoning

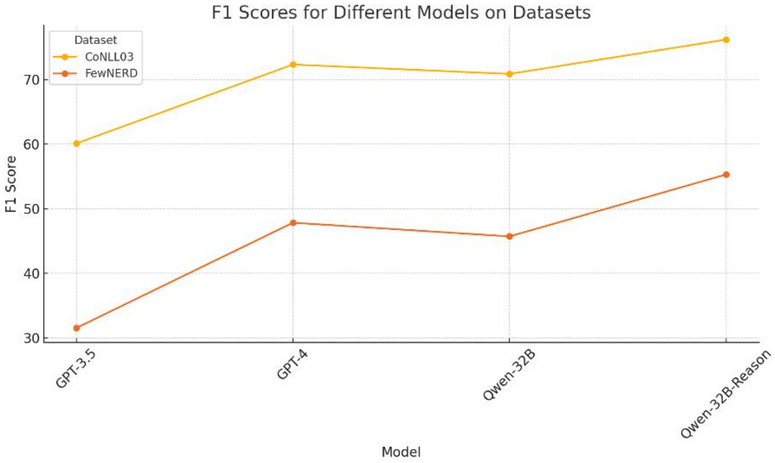


Figure 2 F1 Scores for Different Models on Datasets

5 CONCLUSION

In summary, our study investigates the strengths and weaknesses of LLMs for entity recognition, proposing a chain-of-thought scaling mechanism to elevate their reliability and accuracy. Experimental results confirm the efficacy of this structured reasoning paradigm, reinforcing its potential as a robust solution to LLM hallucinations. Future work may explore extending this framework to other domains (e.g., relation extraction or event detection) and integrating additional interpretability components that offer finer control over the reasoning process.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

This paper was supported by Guangdong Science and Technology Innovation Strategy Special Fund (pdjh2023b0592).

REFERENCES

[1] Zhou ZH, Chawla NV, Jin Y, et al. Big Data Opportunities and Challenges: Discussions from Data Analytics Perspectives [Discussion Forum]. IEEE Computational Intelligence Magazine, 2014, 9(4): 62–74. DOI: 10.1109/MCI.2014.2350953.

[2] Jurafsky D, Martin JH. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 2000.

[3] Chang Y, Wang X, Wang J, et al. A Survey on Evaluation of Large Language Models. 2023. DOI: 10.48550/arXiv.2307.03109.

[4] Li B, Fang G, Yang Y, et al. Evaluating ChatGPT's Information Extraction Capabilities: An Assessment of Performance, Explainability, Calibration, and Faithfulness. 2023. DOI: 10.48550/arXiv.2304.11633.

[5] Ma Y, Cao Y, Hong Y, et al. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples! 2023. DOI: 10.48550/arXiv.2303.08559.

[6] Wan Z, Cheng F, Mao Z, et al. GPT-RE: In-context Learning for Relation Extraction using Large Language Models, 2023. DOI: 10.48550/arXiv.2305.02105.

[7] Sang EF, De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050, 2003.

[8] Ding N, Xu G, Chen Y, et al. Few-NERD: A Few-shot Named Entity Recognition Dataset. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. 2021(1): 3198–213, DOI: 10.18653/v1/2021.acl-long.248.

[9] Chiu JP, Nichols E. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the association for computational linguistics, 2016, 4: 357–70.

[10] Collobert R, Weston J, Bottou L, et al. Natural Language Processing (Almost) from Scratch. Natural Language Processing, 2011, 45.

[11] Hammerton J. Named entity recognition with long short-term memory. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, 2003: 172–5.

[12] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019.

[13] Li X, Feng J, Meng Y, et al. A unified MRC framework for named entity recognition. arXiv preprint arXiv:1910.11476, 2019.

[14] Sarzynska-Wawer J, Wawer A, Pawlak A, et al. Detecting formal thought disorder by deep contextualized word representations. Psychiatry Research, 2021, 304: 114135.

[15] Liu AT, Xiao W, Zhu H, et al. QaNER: Prompting Question Answering Models for Few-shot Named Entity Recognition. 2022. DOI: 10.48550/arXiv.2203.01543.

[16] Yan H, Gui T, Dai J, et al. A Unified Generative Framework for Various NER Subtasks. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021(1): 5808–22, DOI: 10.18653/v1/2021.acl-long.451.

[17] Hoffmann J, Borgeaud S, Mensch A, et al. Training Compute-Optimal Large Language Models. 2022.

[18] Kaplan J, McCandlish S, Henighan T, et al. Scaling Laws for Neural Language Models. 2020.

[19] Snell C, Lee J, Xu K, et al. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. 2024.

[20] Welleck S, Bertsch A, Finlayson M, et al. From Decoding to Meta-Generation: Inference-time Algorithms for Large Language Models. 2024.

[21] OpenAI. Learning to Reason with LLMs. 2024.

[22] Gao Z, Niu B, He X, et al. Interpretable Contrastive Monte Carlo Tree Search Reasoning. 2024.

[23] Zhang Y, Yang J, Yuan Y, et al. Cumulative Reasoning with Large Language Models. 2024.

[24] Huang Z, Zou H, Li X, et al. O1 Replication Journey – Part 2: Surpassing O1-preview through Simple Distillation, Big Progress or Bitter Lesson? 2024.

[25] Qin Y, Li X, Zou H, et al. O1 Replication Journey: A Strategic Progress Report – Part 1. 2024.

[26] Wang P, Li L, Shao Z, et al. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. 2024.

[27] DeepSeek-AI, Guo D, Yang D, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. 2025.

[28] Li C, Xue M, Zhang Z, et al. START: Self-taught Reasoner with Tools. 2025, DOI: 10.48550/arXiv.2503.04625.

[29] Bai J, Bai S, Chu Y, et al. Qwen Technical Report. 2023.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3043

MODELING AND FREQUENCY ESTIMATION OF FLIGHT CYCLE SIGNALS IN COMPLEX ENVIRONMENTS BASED ON THE FAST FOURIER TRANSFORM

QiuLin Yao[1*], ShaoJie Guo[1], FanZhi Qu[1], QiSheng Liu[1], KuiSong Wang[1], JinZhu Cai[2], ChengCai Liang[1]
[1]School of Mechanical Engineering, Jiamusi University, Jiamusi, 154007, Heilongjiang, China.
[2]School of Materials Science and Engineering, Jiamusi University, Jiamusi, 154007, Heilongjiang, China.
Corresponding Author: QiuLin Yao, Email: wy524887227@163.com

Abstract: Airplanes have now become an important travel tool, and airspeed is the key data to be observed in the process of aircraft flight, therefore, in order to reduce the probability of aircraft stalls and other accidents due to abnormal airspeed data, it is of great significance to accurately measure the airspeed. This study focuses on the frequency estimation of the received signal in flight cycle, combines with the signal processing requirements of laser speed measurement system, constructs a variety of mathematical models and designs the corresponding solution algorithms, and systematically explores the noise characteristics, frequency estimation and intermittent signal processing and other key issues. In this paper, the fast Fourier transform (FFT) algorithm and descriptive statistical analysis of z(t) noise type belongs to Gaussian white noise. The reasonableness and estimation accuracy of the main frequency are verified by comparing the filtered signal with the fitted non-noise signal. This study proposes a frequency estimation method that combines Fast Fourier Transform (FFT) with descriptive statistical analysis. The method effectively suppresses environmental noise interference using an adaptive threshold adjustment based on the Constant False Alarm Rate (CFAR) algorithm. A band-pass filter is employed to eliminate noise, and residual analysis is used to validate the accuracy of frequency estimation. Additionally, an innovative signal reconstruction method under intermittent sampling is proposed, providing theoretical support for frequency estimation and signal processing in complex environments.

Keywords: Fast Fourier transform; Residual analysis; Bandpass filtering; Gaussian white noise; Doppler shifted letter

1 INTRODUCTION

Laser velocimetry is a viable method of measuring air speed. Laser velocimetry systems capture the spectrum of all scattered signals within the sampling bandwidth of the electronic system at a given moment in time through a technique called coherent detection frequency discrimination. These signals include airspeed Doppler-shifted signals, DC noise, relative intensity and phase noise of the laser, and Doppler noise generated by interfering objects [1]. Due to the different frequency components of the interfering signals at different time points, the background noise cannot be simply suppressed. If a fixed detection threshold is used, it is easy to incorrectly detect interfering signals [2]. In particular, the intensity noise and phase noise of lasers, which have a large amount of energy, can easily exceed the detection threshold, thus interfering with the detection results [3]. In order to accurately solve the atmospheric vector signal and accurately detect the airspeed Doppler shifted signal from the signal spectrum, the current particle image velocimetry technique can accurately measure the speed of sound, but it still has defects in filtering the interference signal [4]. In this paper, we design an algorithm to adaptively adjust the detection threshold for laser velocimetry based on the principle of Constant False Alarm (CFAR) detection, which adaptively adjusts the number of reference and protection units and selects the CFAR processing method through the change of the exponential angular coefficient of the detection unit, $S_k = x_D - x_{D-1}$ [5]. In this study, we model and analyze the received signal frequency estimation problem, and the main goal is to design suitable frequency estimation algorithms according to the characteristics of the received data in different flight cycles. The marginal contributions of this study are primarily reflected in four aspects. First, an innovative high-precision frequency estimation method is proposed by combining Fast Fourier Transform (FFT) with descriptive statistical analysis, enabling accurate frequency identification in complex environments. Second, the adaptive Constant False Alarm Rate (CFAR) algorithm is introduced into the laser velocimetry system, effectively addressing the variability of environmental noise and improving the reliability of signal detection. Third, a method combining band-pass filtering and residual analysis is applied to accurately extract valid signals and validate the accuracy of frequency estimation. Finally, a signal reconstruction method under intermittent sampling conditions is proposed, providing a new theoretical basis for signal processing in future complex environments. In conclusion, this study offers innovative theoretical support for laser velocimetry technology and signal processing in dynamic, complex environments.

2 NOISE ANALYSIS BASED ON FOURIER TRANSFORM AND AUTOCORRELATION ANALYSIS

Some basic assumptions are established for the convenience of this study: the noise z(t) is independent in each flight cycle and obeys a specific distribution. The noise characteristics are stable over a flight cycle, i.e., the mean and

variance are constant. The received signal satisfies the expression: $x(t) = Asin(2\pi ft + \varphi) + z(t)$ where A, f, φ denote amplitude, frequency, and phase. The non-noise signal portion (amplitude, frequency, phase) remains stable from flight cycle to flight cycle, but may vary from flight cycle to flight cycle. The sampling interval $T = 2 \times 10^{-9}$ seconds is fixed to satisfy the Nyquist sampling theorem, which states that the signal frequency is less than one-half of the sampling frequency. The data were sampled long enough to support frequency analysis.
In this paper, a fast Fourie transform r (FFT) algorithm is used
Let the sequence of sinusoidal signals y(n) be

$$y(n) = Acos\left(\frac{2\pi f_0 n}{f_s} + \emptyset\right) \tag{1}$$

The rectangular window function w(n) is 1 from n=0 to n=N-1 and 0 at other times. its discrete Fourier transform W(k) is N at k=0 and 0 at other values of k. The discrete Fourier transform W(k) of the rectangular window function is

$$W(k) = \frac{sin\,(\pi k)}{sin\,(\pi k/N)} e^{-j\pi k(N-1)/N} \tag{2}$$

Doing N-point DFT on y(n) yields the discrete spectrum Y(k), due to the symmetry of the real sequence DFT, the effect of negative frequency components can be ignored, and only the first N/2 points of the discrete spectrum are considered, and let Re(x) denote to take the real part of x, then the real part of FFT coefficients Re(Y(k)) is

$$Re\big(Y(k)\big) = \frac{A}{2}\left[\frac{sin\,(\pi(k-k_0))}{sin\,(\pi(k-k_0)} \cdot cos\,(\emptyset - \pi(k-k_0)(N-1)/N)\right] \tag{3}$$

Due to the fence effect of the Discrete Fourier Transform (DFT), if the frequency of the input signal does not correspond exactly to the discrete frequency points of the DFT, the true frequency of the signal may fall between two neighboring DFT frequency lines. In this case, an error occurs when the peak spectral line positions obtained from the Fast Fourier Transform (FFT) are used directly to estimate the signal frequency. The range of this error $[-f_s/(2N), f_s/(2N)]$ is between, where f_s is the sampling frequency and N is the number of samples. Remember that the peak spectral line number in the signal spectrum is m. The peak spectral line Re(Y(k)) and its corresponding phase $\emptyset(m)$ are respectively

$$Re\big(Y(m-1)\big) = \frac{Asin(\pi\delta)}{2\pi(1+\delta)/N} cos\,(\emptyset(m)) \tag{4}$$

$$Re\big(Y(m+1)\big) =- \frac{Asin(\pi\delta)}{2\pi(1-\delta)/N} cos\,(\emptyset(m)) \tag{5}$$

An interpolating polynomial is constructed using the ratio of the real part of the coefficients of the peak spectral line to its neighboring spectral line of the next largest value, and the frequency bias δ estimated by this method is

$$\delta = \begin{cases} \delta_1 =- \frac{Re(Y((m+1))}{\left(Re(Y(m))-Re(Y(m+1))\right)}, [\delta_1 > 0, \delta_2 > 0] \\ \delta_2 =- \frac{Re(Y((m-1))}{\left(Re(Y(m))-Re(Y(m-1))\right)}, ot\hbar\ er \end{cases} \tag{6}$$

The generalized frequency correction formula is

$$f_0 = (m+\delta)f_s/N \tag{7}$$

The above analysis can lead to a graph of the noise signal (time domain) and spectral analysis about Z(t) as shown in Figure. 1 and Figure. 2.

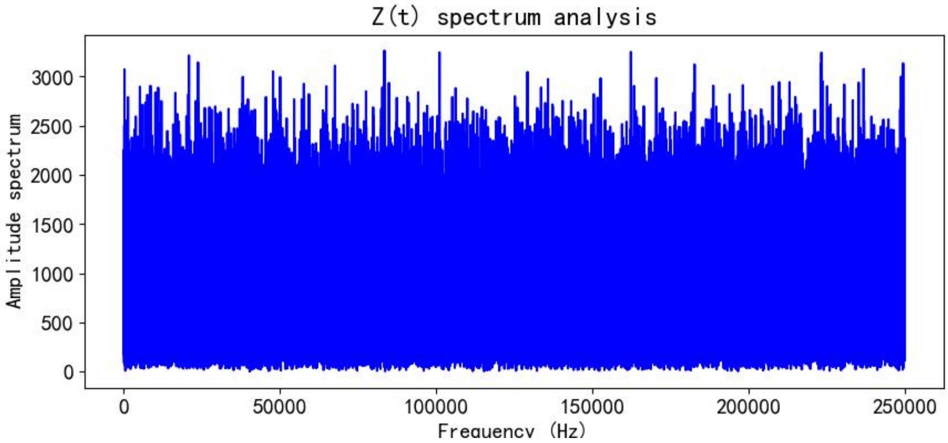


Figure 1 Spectral analysis of Z(t)

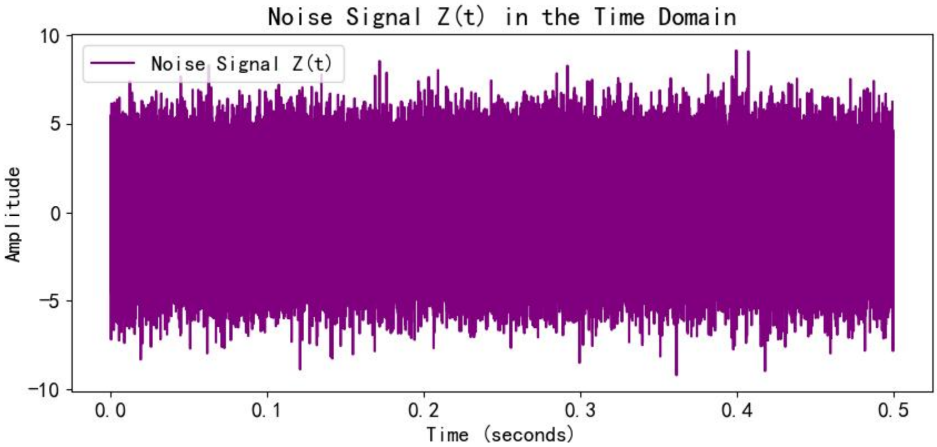


Figure 2 Noise Signal of Z(t) (Time Domain)

In order to obtain the noise characteristics, descriptive statistical analysis is used in this paper to evaluate the distributional characteristics of the noise by calculating the mean, variance and standard deviation respectively.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{8}$$

where μ denotes the mean, n is the number of data, and x_i is the first data value

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2 \tag{9}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2} \tag{10}$$

where σ denotes the overall variance, N is the number of overall data, x_i is the first data value, μ is the overall mean.

Finally, the mean estimate of z(t) was calculated to be 0.0058; the variance estimate of z(t) was 3.9819; and the standard deviation estimate of Z(t) was 1.9955.

Then the Q-Q plot is used to further verify the normality, if the points on the Q-Q plot of z(t) coincide with the reference line, it indicates that the distribution may obey the normal distribution. As shown in Figure 3, the paper can see that the blue points are almost completely distributed along the red line, which indicates that the distribution is very close to the normal distribution.

Points on the Q-Q graph =($Q_N(P_i), Q_X(P_i)$), where $P_i = \frac{i}{n+1}, i = 1,2, \ldots.., n$, is the sample quantile, $Q_X(P_i)$ is the normal distribution quantile.

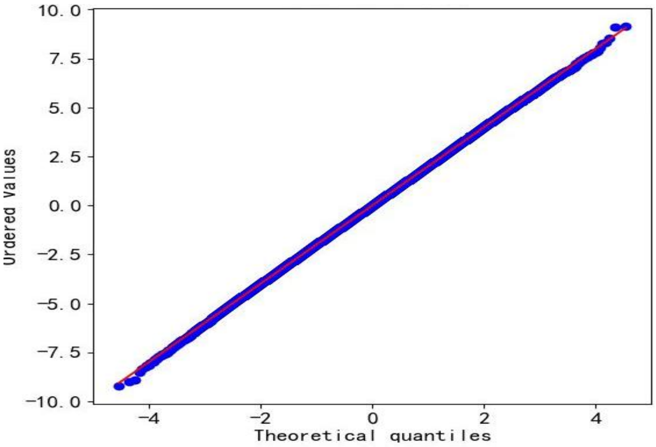


Figure 3 Q-Q Diagram for z(t)

Autocorrelation analysis is an important tool to assist in determining white noise, and for Gaussian white noise, the correlation coefficient $r_X(\tau)$ is defined as

$$r_X(\tau) = \frac{R_X(\tau) - \mu_X^2}{\sigma_X^2} \tag{11}$$

where μ_X is the mean and σ_X^2 is the variance. The autocorrelation is shown in Figure. 4, where the autocorrelation coefficient of z(t) is 1 (perfectly correlated) at lag 0, and rapidly decays to close to 0 at the other lags.This is a typical white noise autocorrelation feature, which indicates that the noise component of the signal is random and time uncorrelated.

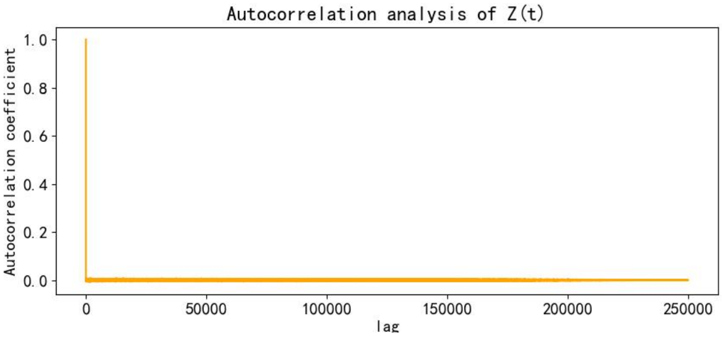


Figure 4 Correlation Plot of Noise over Time

Using probability distribution analysis models, especially the normal distribution model (Gaussian Model), as shown in Figure 5, the final judgment is Gaussian noise.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{12}$$

where μ is the mean of the distribution, σ is the standard deviation of the distribution, and π is the circumference of the circle.

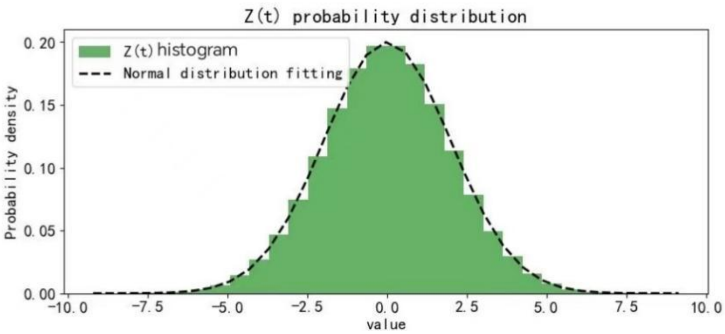


Figure 5 Fitted Plot of Z(t) Normal Distribution

A quantitative measure of the difference between a sample distribution and a reference distribution (e.g., normal distribution) using the K-S test statistic. It is calculated using the formula:

$$D = \overset{sup}{x} \left| F_{empirical}(x) - F_{theoretical}(x) \right| \tag{13}$$

where $F_{empirical}(x)$ is the empirical distribution function of the sample, $F_{theoretical}(x)$ is the Cumulative Distribution Function (CDF) of the reference distribution, and $\overset{sup}{x}$ denotes the maximization of the value for all x. Finally, the Kolmogorov-Smirnov test statistic was found to be 0.0015 with a p-value of 0.6575, and since p>0.05, the null hypothesis could not be rejected, and it was concluded that the sample distribution could be approximated as normal.

In summary, the fast Fourier transform (FFT) algorithm is used to avoid the complex modulo operation of signal amplitude spectrum and the complex division operation of Quinn's algorithm, which greatly reduces the amount of computation and is easy to be implemented in hardware, and the descriptive statistical analysis is used to calculate the mean estimate of z(t) to be 0.0058, the variance estimate of z(t) to be 3.9819, and the standard deviation estimate of Z(t) to be 1.9955, respectively. 1.9955. Then the white noise is judged to be Gaussian white noise by using Q-Q plots for normality verification and autocorrelation analysis to assist in judging the white noise, so as to better further analyze the noise characteristics of flight cycle one.

3 NOISE FILTERING BASED ON BAND-PASS FILTERING

In this paper, the received signal data is first read from the flight cycle and the average value of the signal is calculated, which is subtracted from the original signal to remove the 0Hz component (DC component) to ensure that the spectrum is focused on the effective AC component, thus avoiding interference at 0Hz. First a bandpass filter can be constructed by cascading a low-pass filter and a high-pass filter. Such filters allow signals in a specific frequency range to pass while blocking or attenuating signals at lower and higher frequencies[6], setting the upper and lower limits of the bandpass filters to ±20% of the initial estimated frequency (i.e., 0.8 × the initial estimated frequency and 1.2 × the initial estimated frequency), and retaining only signal components near the dominant frequency. The results are shown in Figure. 6, where the high-pass and low-pass partial cutoff frequencies can be obtained separately.

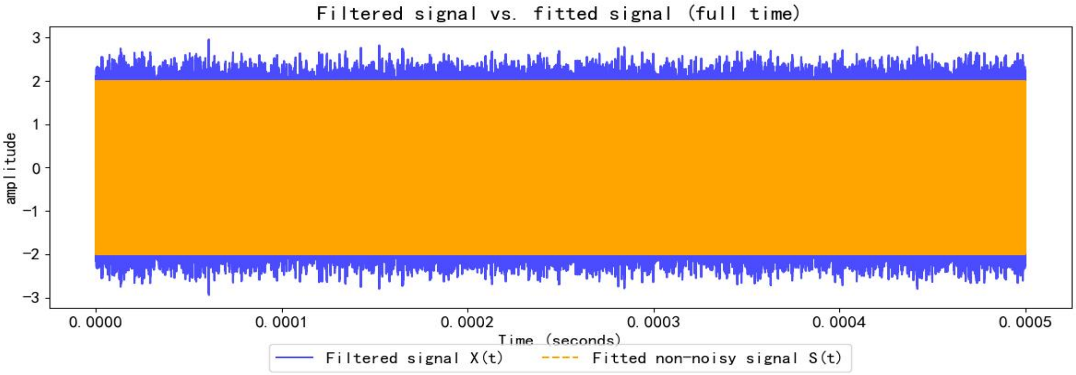


Figure 6 Filtered Signal vs. Fitted Signal (Full Time Period)

When analyzing signals to take spectral data, the most commonly used method is the Fourier transform, of which the most mature is the fast Fourier transform. There is a sampled signal of length N , where n = 0, 1, ..., N.1 refers to the time domain sampling point number. Then the signal is segmented, i.e., divided into frames, the original sampled signal is set to $X_m(n)$, the same n = 0, 1, ..., N-1, where n is the frame synchronization of the time serial number, m is the frame of the serial number, N refers to the number of sampling points in a frame. Fourier transform (DTFT) of the signal $X_m(n)$:

$$X(m, e^{jw}) = \sum_{n=0}^{N-1} w_m(n) \cdot X_m(n) \cdot e^{-jwn} \tag{14}$$

To be able to perform better discrete calculations, the discrete Fourier transform (DFT) by $w_m(n) \cdot x_m(n)$ is shown in Eq:

$$X(m, k) = \sum_{n=0}^{N-1} w_m(n) \cdot x_m(n) \cdot e^{\frac{-j2\pi nk}{2N}}, k = 0,1\ldots\ldots, N-1 \tag{15}$$

where $|X(m, k)|$ refers to the short-time magnitude spectrum estimate of $X_m(n)$. where m refers to the time variable and k refers to the frequency variable, so $|X(m, k)|$ represents the spectrum of the dynamics. To improve the computation rate of $|\mathrm{X}(\mathrm{m}, \mathrm{I})|$, a fast Fourier transform algorithm, i.e., FFT algorithm, is used [7][8]. The resulting spectral analysis of the received signal after bandpass filtering is shown in Figure. 7, where the green curve is the spectral amplitude of the filtered signal, showing that there is only one significant peak of the dominant frequency in the spectrum. The red dashed line marks the position of the dominant frequency (frequency 40.999836 MHz), which corresponds to the dominant frequency retained after bandpass filtering. The estimated value of the dominant frequency (about 40MHz) agrees well with the theoretical value, further verifying the accuracy of the filtering and signal analysis.

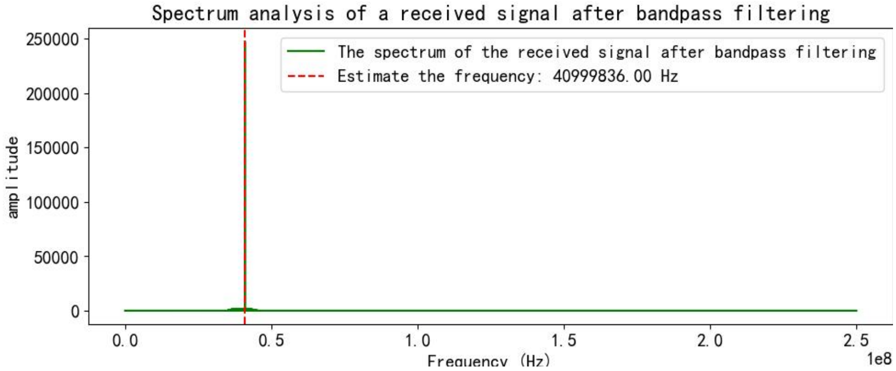


Figure 7 Spectral Analysis of the Received Signal after Bandpass Filtering

$$R(t) = X_{filtered}(t) - S_{fitted(t)} \tag{16}$$

Where $X_{filtered}(t)$ is the signal after band-pass filtering and $S_{fitted(t)}$ is the non-noise signal generated by the main frequency fitting, the residual mean is 1.1102×10^{-5} and the residual standard deviation is 4.8338×10^{-1}, so that the residual's mean is approximated to be 0, which indicates that the main signal after filtering has been well fitted. Finally, after Eq. 14 and Eq. 15, the time-domain characteristic diagram of the residual signal R(t) is shown in Figure. 8 and the spectrogram of the residual signal R(t) is shown in Figure 9.

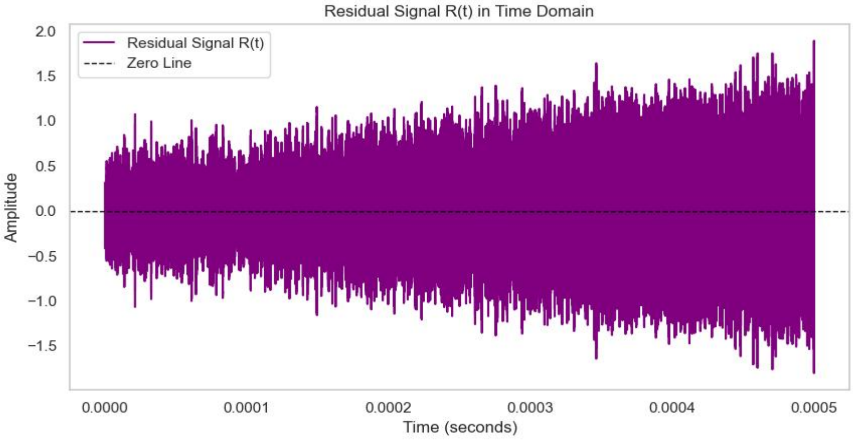


Figure 8 Time-domain Characteristics of the Residual Signal R(t)

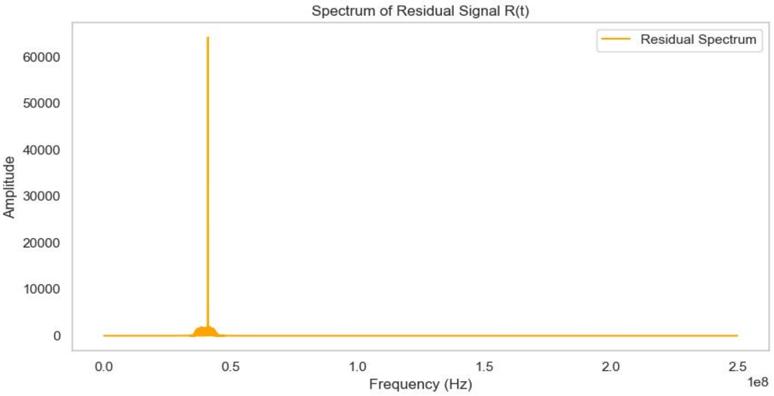


Figure 9 Spectrogram of the Residual Signal R(t)

Finally, the estimated main frequency is used to generate a pure sinusoidal signal $s(t) = Asin(2\pi ft + \varphi)$, where A is the known amplitude, f is the estimated main frequency after bandpass filtering, and φ is the known phase. By comparing the filtered signal $X_{filtered}(t)$ with the fitted non-noise signal $s(t)$, the results are shown in Figure. 10, which can verify the reasonableness and estimation accuracy of the main frequency.

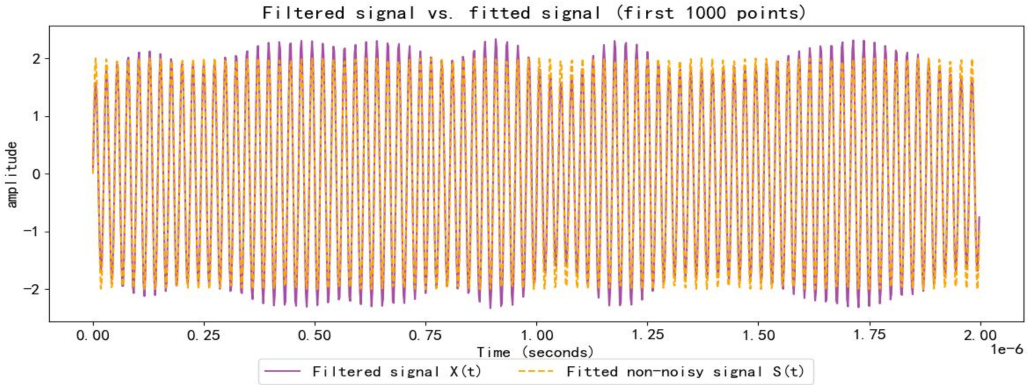


Figure 10 Plot of the Filtered Signal and the Fitted Non-Noise Signal

In summary, firstly, the band-pass filter is used to retain the components of the signal close to the frequency and exclude the noise at other frequencies to obtain the filtered signal$X_{filtered}(t)$. Then the FFT algorithm is performed again on the filtered signal $X_{filtered}(t)$ to calculate its spectrum and amplitude spectrum to obtain the spectrum analysis graph of the received signal after bandpass filtering. Next, the frequency with the largest amplitude is found in the filtered spectrum, which is used as the final estimate of the principal frequency (i.e., the frequency of the non-noise portion of the signal) Since the band-pass filtering has removed most of the noise components, this estimate will be more accurate than the preliminary frequency. The estimated main frequency is then used to generate a pure sinusoidal signal $Asin(2\pi ft + \varphi)$ for residual analysis.The mean value of the calculated residuals is approximately 0, indicating that the filtered main signal has been well fitted. Finally, the reasonableness and estimation accuracy of the main frequency are verified by comparing the filtered signal $X_{filtered}(t)$ with the fitted non-noise signal.

4 CONCLUSION

In this study, this paper presents an in-depth analysis and modeling of the frequency estimation problem of flight cycle signals, designs solution methods applicable to different scenarios, and systematically and comprehensively explores the noise characteristics, signal frequency estimation, and model performance. Noise characterization: Evaluate the noise characteristics of the flight cycle to provide a basis for frequency estimation. Frequency estimation with known partial parameters: design frequency estimation algorithms adapted to changing noise backgrounds under the condition of known amplitude and phase in the flight cycle.

In this study, the frequency estimation of complex flight cycle signals is successfully realized through theoretical modeling and algorithm design, which provides a strong support for the laser velocimetry technology of aircraft. The results not only solve the frequency estimation problem under the condition of known and unknown parameters, but also explore the signal reconstruction method of intermittent sampling, which provides a theoretical reference for future signal processing in complex environments.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] MARIANIMJ, DIRUSCIOA, NOTAROV, et al. Improvement of BepiColombo's Radio Science Experiment Through an Innovative Doppler Noise Reduction Technique. Journal of Deep Space Exploration, 2018, 5(2): 115-123. DOI: 10.15982/j.issn.2095-7777.2018.02.002.
[2] Li Bin, Lei Hongjie. Analysis of data inversion accuracy of airborne optical atmospheric data system. Infrared and Laser Engineering, 2021, 50(08): 215-220.
[3] Wang Xiaowei, Liang Yingjian, Li Xiang, et al. Research on low airspeed measurement method based on optical Doppler shift. Laser Technology, 2016, 40(05): 629-632.
[4] Chestnut J W, Luo K, Shang A H, et al. Progress of laser velocimetry in hypersonic flow field. Journal of Mechanics, 2024, 56(04): 890-914.
[5] Huang Liping, Tang Shi, Liu Xingtao. An adaptive CFAR algorithm for laser speed measurement system. Electronic Fabrication, 2024, 32(06): 57-59+40.
[6] Ma Luowen, Liu Ning, Hu Xinyue, et al. FFT amplitude-phase joint method for fast and high-precision frequency estimation. Measurement Technology, 2022, 42(06): 34-39.
[7] Yu Maoquan, Zhang Lei, Zhao Lianyi. Harmonic spectrum analysis of power system based on improved FFT. Journal of Anhui Institute of Electrical Engineering Technology, 2023, 28(02): 43-47.
[8] Luo Dongjia, Zhang Libiao. Research on spectrum analysis based on FFT. Electronic Fabrication, 2015, (9): 34-34.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3044

OPTIMIZATION OF STUDENT CLASSROOM BEHAVIOR RECOGNITION ALGORITHM BASED ON DEEP LEARNING

YunJiao Duan, HaiJun Zhang*
College of Computer Science and Technology, Xinjiang Normal University, Urumqi 830054, Xinjiang, China.
Corresponding Author: HaiJun Zhang, Email: zhjlp@163.com

Abstract: The identification and analysis of student behaviors in the classroom are beneficial for educators to understand and monitor students' learning dynamics and outcomes. Currently, existing deep learning-based classroom behavior recognition models face issues such as low recognition accuracy, limited generalization capabilities, and a narrow dataset coverage, which adversely affect the effectiveness of educational assessments. To address these challenges, this study collected a classroom behavior dataset comprising data from elementary, middle, and high school students and proposed a DCB-YOLOv11 model for student behavior recognition. This model incorporates deformable convolution (DCNv4) in the backbone and detection head of YOLOv8, along with a redesigned CBAM attention module in the pooling layer. The proposed model achieved an average precision of 92.43%, representing a 2.1% improvement over the baseline model, while also significantly reducing computational overhead. This research, combining mobile networks and educational big data, facilitates the personalized development of intelligent learning environments and enhances the effectiveness of the educational process.
Keywords: Deep learning; YOLO; Behavior recognition; Classroom behavior

1 INTRODUCTION

Student behaviors exhibited in the classroom reflect individual learning styles and engagement levels, and they can also indicate the degree of understanding of the knowledge being imparted. Educators can adjust their teaching methods and pacing by understanding students' states. These behaviors are crucial indicators that assessment experts consider during classroom evaluations. In the field of education, commonly used classroom evaluation methods include the Flanders Interaction Analysis System (FIAS), Student-Teacher (S-T) classroom teaching analysis, and Information Technology-based Interaction Analysis Systems (ITIAS). Traditional classroom evaluation methods often require manual observations and detailed on-site recordings. Although manual recording can ensure high-quality qualitative data, it inherently limits the scope for comprehensive quantitative assessments.

With the increasing integration of intelligent technologies in education, the application of deep learning techniques to embed object detection and recognition algorithms into relevant school equipment enables the automatic identification of students' classroom states and behaviors. This approach not only enhances processing speed and recognition accuracy but also broadens the scope of application, which can promote the digital and intelligent development of the education evaluation. In student behavior recognition, object detection methods are primarily categorized into two-stage and single-stage approaches [1].

The two-stage approach is represented by the R-CNN (Regions with Convolutional Neural Networks) series, such as Fast R-CNN and Faster R-CNN. These methods first generate candidate regions and then classify and regress each candidate region. Their advantages are high precision, making them suitable for fine recognition in complex scenarios. However, they have a high computational complexity and slower processing speed, making them unsuitable for applications with stringent real-time requirements. In contrast, the single-stage approach is exemplified by YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector). YOLO models perform detection and classification directly on the entire image in a single step, significantly increasing detection speed. These methods are computationally efficient and well-suited for real-time detection, but their performance in terms of detail and complexity may not match that of the two-stage approach.

Despite significant advancements in both methods for object detection and behavior recognition, there are still many challenges for recgnitions of classroom behaviors. The two-stage approach, while precise, consumes substantial computational resources, making it difficult to implement on resource-constrained devices. The single-stage approach, although fast, still requires improvement in handling complex scenes and fine detail recognition. Moreover, the generalization ability of existing models across different age groups and classroom environments needs further enhancement to adapt to diverse educational contexts. To address these issues, this paper proposes the DCB-YOLOv11 model (DCNv4+CBAM+YOLOv11), aiming to combine the strengths of both approaches by introducing deformable convolutions and attention mechanisms to enhance the model's recognition accuracy and generalization capability, thereby achieving efficient and precise recognition of student behaviors in the classroom. The main contributions of this study are as follows:

(1) Construction of a Student Classroom Behavior Dataset: This dataset is collected from real classroom scenarios involving primary, secondary, and university students. A total of 5,850 frames and 14,200 behavior annotations were manually extracted from 13 classroom sessions. Student behaviors are categorized into five types: attentive listening, reading and writing, raising hands, standing, and whispering.

(2) Replacement of Key Convolutional Layers: Some convolutional layers in the backbone network of YOLOv11 are replaced with deformable convolution (DCNv4), effectively controlling the model's computational complexity. By introducing dynamic spatial transformations and adaptive weighting, computational cost is reduced while maintaining recognition performance.

(3) Integration of an Attention Module: A redesigned CBAM attention module is incorporated into the SPPF pooling layer to effectively extract and utilize critical information from feature maps, enhancing the model's ability to focus on key behavioral features.

2 RELATED WORK

Deep learning algorithms in the target domain can be broadly categorized into two approaches: two-stage and one-stage methods. The two-stage method, represented by R-CNN, first generates candidate regions and then classifies the extracted samples using a convolutional neural network. In contrast, the one-stage method, exemplified by SSD and the YOLO series, directly extracts features within the network to simultaneously predict the location and category of samples. Similarly, algorithms for student behavior detection can also be classified into these two categories based on this conceptual framework [2].

2.1 Student Behavior Detection Based on Two-Stage Approachs

The two-stage method, also known as region-based object detection, consists of two sequential steps: first, extracting candidate regions of the target objects, and then classifying these regions using a CNN network. The following is a review of representative studies that have applied this method to student classroom behavior detection.

Zaletelj et al. [3] utilized Microsoft Kinect software to collect images and categorized student attention into three levels—low, medium, and high—by integrating facial expressions, eye movements, and body postures. By employing decision trees and the k-nearest neighbors (KNN) algorithm, they achieved an accuracy of 0.753. Zheng et al. [4] proposed an intelligent system for analyzing students' classroom behaviors, capable of detecting three behaviors: raising hands, standing, and sleeping. Their algorithm improved upon the R-CNN-based model by incorporating a scale-aware detection head, enabling the detection of students' postures at varying sizes within the frame.Huang Yongkang et al. [5] introduced an algorithm based on a deep spatiotemporal residual convolutional neural network for real-time student behavior recognition in the classroom. By combining object detection and tracking techniques to extract student images, the model learns each target's spatiotemporal behavioral features through the deep spatiotemporal residual CNN, achieving real-time recognition of multiple student behaviors in classroom scenarios.Lin et al. [6] proposed a network structure based on Feature Pyramid Networks (FPN), leveraging multi-scale feature extraction through deep convolutional networks and the hierarchical structure of FPN to efficiently detect objects of different scales. Their method achieved state-of-the-art single-model performance on the COCO detection benchmark within a standard Faster R-CNN system, offering a practical and accurate multi-scale object detection solution.Zhao et al. [7] further improved the FPN network by enhancing the Pyramid mechanism through effective fusion of multi-level features extracted by the backbone. By integrating the Multi-Level Feature Pyramid Network (MLFPN) into the end-to-end single-stage object detector M2Det, their approach surpassed the performance of state-of-the-art single-stage detectors on the MS-COCO benchmark.

Although the two-stage method offers a significant advantage in accuracy, its application in evaluating students' classroom behavior in educational settings is hindered by its slower processing speed and complex network structure. The limitations in real-time performance and computational complexity restrict the practical implementation of two-stage algorithms in the education domain.

2.2 Student Behavior Detection Based on Single-Stage Approachs

Unlike the two-stage method, the one-stage algorithm performs object localization and classification in a single step. It typically employs a dense prediction grid, where each grid cell predicts multiple bounding boxes along with their corresponding class probabilities, enabling rapid object detection within an image. Due to its high speed and real-time performance, the one-stage algorithm, represented by the YOLO series, has made significant advancements in student behavior recognition.

Wang et al. [8] utilized the backbone convolutional layers of YOLOv5 to extract features from input images and incorporated a Squeeze-and-Excitation (SE) attention mechanism to reduce excessive focus on irrelevant information, thereby mitigating background interference.Guo Junqi et al. [9] proposed an improved network structure and loss function for the YOLOv5 model based on the characteristics of classroom scenarios. The model, designed for multi-object detection, was specifically applied to student behavior recognition, and its effectiveness was validated through comparative experiments.Z. Zheng et al. [10] retained the FPN+PAN feature extraction framework while optimizing the CBL modules in YOLOv5 by replacing the default LeakyReLU activation function with the GELU activation function. Additionally, the SIoU loss function was introduced to accelerate convergence. Experimental results demonstrated improvements in both accuracy and detection speed.W. Niu et al. [11] integrated a coordinate attention mechanism into each CSP module within the YOLOv5 architecture to address the issue of missed detections. By decomposing channel attention into one-dimensional features for encoding, the mechanism enhanced output precision.Z. Zhang et al. [12] introduced the CloU loss function to replace the default loss function, addressing the issue of distance miscalculation when two predicted bounding boxes do not intersect. Additionally, the default activation function was replaced, making the improved YOLOv5-based model better suited for capturing student behavior data features.Yang et al. [13,14] integrated the dual-attention module and Wise-IoU into the YOLOv7 network for classroom behavior detection. Through experiments on their own dataset, they achieved a 1.8-point increase in average precision. Later, they further improved the detection accuracy by incorporating results from YOLOv7 CrowdHuman, slow-fast, and deep sorting models.YOLOv8 [22], a relatively newer version in the YOLO series, introduced further optimizations over YOLOv7 in terms of accuracy, speed, model scalability, and flexibility, enhancing its performance across various application scenarios. It demonstrates particularly improved performance in handling complex scenes and multi-scale objects. YOLOv8 introduces a more robust network architecture that combines the advantages of deep convolutional neural networks and lightweight models, increasing accuracy while ensuring efficiency. The inclusion of an adaptive anchor box generation algorithm allows the model to dynamically adjust the size and position of anchor boxes, further enhancing its performance in small object detection and high-density scenes.

The improvements made to YOLO based on one-stage algorithms are more aligned with the requirements for student behavior recognition. Researchers have continuously refined the algorithm to enhance recognition accuracy while ensuring real-time performance. YOLOv11 [23] is the latest version in the YOLO series. Compared to YOLOv8, the previous CF2 module has been upgraded to C3K2, and a new module, C2PSA, has been added after the SPPF module. Additionally, the YOLOv10 head concept has been incorporated, employing depthwise separable convolutions to improve computational efficiency and reduce redundancy.

In summary, there are several key challenges currently faced in the field of student behavior recognition: first, in classroom environments, students' actions continuously change in response to the teacher's lecture content, and improvements in behavior action accuracy are still significantly limited; second, due to the large number of students in the classroom, the ability to simultaneously recognize multiple behaviors is insufficient; third, the existing datasets lack diversity and coverage, which limits the model's generalization ability and its capacity to adapt to various educational scenarios. These issues collectively affect the reliability and effectiveness of algorithms in practical teaching evaluation.

This paper proposes replacing key convolutional layers in YOLOv11 with DCNv4 and introducing an improved CBAM attention module to enhance the model's ability to represent and accurately detect student behavior features. This combination aims to maintain high efficiency while improving the model's recognition performance in complex scenarios, addressing the dual requirements of real-time performance and accuracy. Additionally, by expanding and optimizing the existing dataset, a more comprehensive coverage of various student behaviors and classroom environments will be achieved, further enhancing the model's generalization ability and adaptability.

3 METHOD AND MODEL DESIGN

3.1 Model Overview

In classroom environments, there are often a large number of students, and the relative sizes of individuals in the same frame vary, which raises the demands on object detection algorithms. This paper uses YOLOv11 as the backbone network and replaces the convolutional layers in both the Head and Backbone sections with DCNv4 convolutions. In this process, the kernel size, stride, and padding method of the new convolutions are kept consistent with the original layers to ensure that the feature map dimensions remain unchanged. Additionally, an improved CBAM attention module is integrated into the pooling layer. The overall model structure is shown in Figure 1.

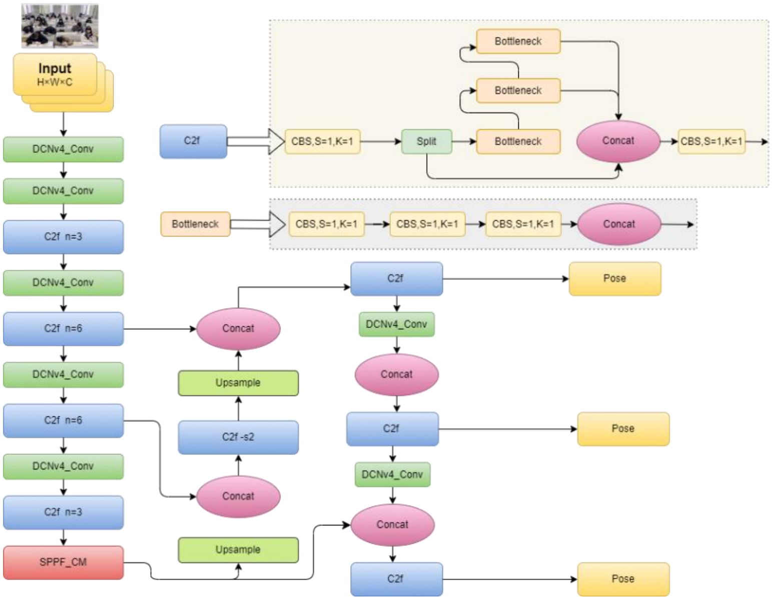


Figure 1 Structure of the DCB-YOLOv8 Model

DCNv4 is an enhanced convolutional network that adaptively adjusts the position of convolutional kernels, allowing for better capture of the deformation features of objects. This adaptability results in superior performance when handling targets of varying sizes. Additionally, the incorporation of the CBAM attention mechanism based on multi-scale feature fusion in the pooling layer enhances the expressive power of the features. This approach not only utilizes multi-scale information but also further optimizes the representation of feature maps through the CBAM mechanism, which improves model performance across various tasks. By integrating DCNv4 and the improved CBAM attention module, the model achieves further optimization in feature extraction and representation which can enhance the robustness and accuracy of object detection, particularly in complex classroom scenarios with many students.

3.2 Replacement of Standard Convolutions with DCNv4 Deformable Convolutions

Although YOLOv11 incorporates a refined architecture design and an optimized training process, achieving faster processing speed with an effective balance between accuracy and performance, there is still room for improvement in recognizing specific scenarios in this study. The default convolutions in YOLOv11 tend to overlook important details when recognizing student behaviors of varying sizes in the classroom, which may lead to misclassifications. Replacing certain convolutional layers in YOLOv11 is thus essential for the accurate behavior recognition in the specific classroom scenarios addressed in this paper.

In this paper, the YOLOv11 model is improved by replacing the conventional convolutional layers in its Head and Backbone with a Deformable Convolutional Network (DCNv4) [15]. DCN is a deformable convolutional neural network module used for object detection and image segmentation. The DCN series significantly enhances convolution adaptability by adding learnable offsets to the convolutional kernels. Unlike standard convolutions, attention mechanisms, due to their ability to model long-range dependencies, have been successfully applied to various computer vision tasks. Window attention limits the attention operation to a fixed-size window, thereby reducing the computational complexity of regular attention. To further reduce the high computational complexity introduced by standard attention, deformable attention assigns dynamic positions and weights, allowing each query to focus on a specific number of key sampling points.

DCNv4 introduces deformable convolutional kernels that can adaptively adjust their shapes, enabling a more effective capture of spatial deformation features. The standard convolution operation can be expressed as follows:

$$y(p_0) = \sum_{p_n \varepsilon R} \omega(p_n) \cdot x(p_0 + p_n) \quad (1)$$

In this context, $y(p_0)$ represents the value of the output feature map at position p_0, and x denotes the input feature map, and ω refers to the convolutional kernel weights, and p_n is the set of sampling positions.

In DCNv4, deformable Δp_n offsets are introduced, allowing the convolution operation to adaptively select sampling positions. The specific formula can be changed as follows:

$$y(p_0) = \sum_{p_n \varepsilon R} \omega(p_n) \cdot x(p_0 + p_n + \Delta p_n) \quad (2)$$

In the initial module design of DCNv3, the calculation of offsets and dynamic weight allocation was performed by a complex sub-network comprising deep 3×3 convolutions and layer normalization and GELU activation, and linear layers, as shown in Figure 2. In contrast, DCNv4 follows the design principles of Xception by eliminating the normalization and GELU layers while incorporating the original separable convolution structure. This architectural change significantly reduces the computation time required for operations [16].

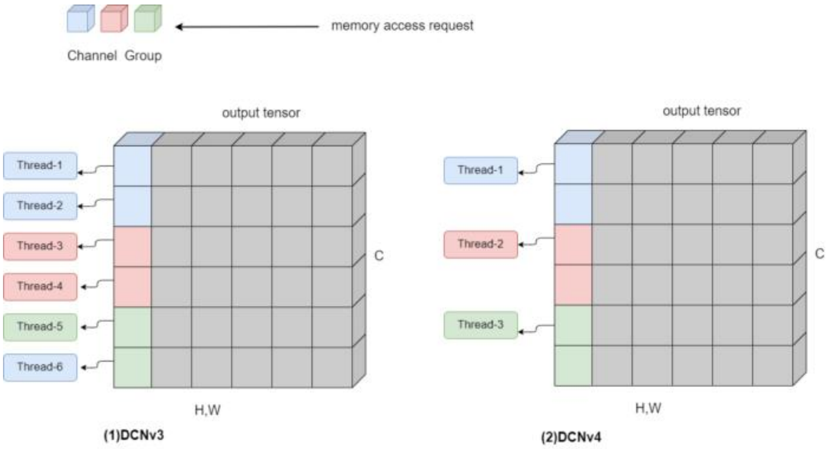


Figure 2 Structure of the DCNv4 Model

In the Backbone of YOLOv11, the convolutional layers are responsible for feature extraction. However, standard convolutional layers are not effective in capturing complex spatial deformations or adaptively adjusting the sampling locations. To address these limitations, the following improvements have been made in this paper.

The first step involves locating all the standard convolutional layers and introducing an offset learning module within these layers. This module is a small convolutional network that takes the feature map as input and outputs the offsets. By replacing the standard convolution operations with deformable convolutions, the Backbone can adaptively sample the input features, better capturing complex spatial deformations and thereby enhancing the feature extraction capabilities.

In the Head section, where convolutional layers are responsible for object classification and bounding box regression, the second step similarly locates all standard convolutional layers and introduces the offset learning module. This allows the convolution operations in the Head to adaptively adjust the sampling positions, improving the accuracy of classification and bounding box regression.

This replacement strategy enables YOLOv11 to adaptively adjust the shape of the convolutional kernels in both feature extraction and object detection processes, effectively handling complex situations such as deformations, rotations, and scale variations. Experimental results show that this improvement significantly enhances the model's average precision while reducing computational time.

3.3 The Improved SPPF Layer

The original SPPF module in YOLOv11 primarily relies on multi-scale pooling for feature fusion but lacks a refined attention mechanism for feature selection. Additionally, it does not fully utilize global contextual information, which particularly limits its performance in fine-grained object detection and recognition tasks. This is detrimental to behavior recognition in complex classroom environments. In this study, we introduce a CBAM attention mechanism based on multi-scale feature fusion within the SPPF, which enhances the model's understanding and detection capabilities and maximizes the utilization of multi-scale features, and improves overall performance.

CBAM (Convolutional Block Attention Module) [17] is a lightweight attention mechanism for Convolutional Neural Networks (CNNs), primarily consisting of two parts: the Channel Attention Module and the Spatial Attention Module. These two modules separately apply attention to the channel and spatial dimensions. However, this structure overlooks the interaction between the channel and spatial dimensions. Additionally, CBAM lacks a comprehensive consideration of global contextual information, focusing primarily on local features. In the context of classroom environments, global information—such as student distribution and overall classroom atmosphere—is crucial for accurately understanding student behavior. The absence of a global perspective can impact the model's overall judgment ability [18].

Multi-Scale Feature Fusion is a method that enhances feature representation by combining information from different scales. It helps the model capture details and contextual information at various levels, improving overall performance. In this paper, the model's overall performance is enhanced by enriching feature representations and strengthening the interrelationship between features. The specific approach is to implement multi-scale feature fusion, which aids in capturing information about students at different scales. The modified module is named m_CBAM, and the detailed approach is outlined as follows.

Multi-scale features are extracted from the input feature map, denoted as $F_1 \in R^{C\times H\times W}$, where C is the number of channels, and H and W are the height and width, respectively. By applying convolutional kernels of different sizes, three distinct scale feature maps can be obtained:

$$F_1 = \mathrm{Conv}_{1\times 1}(F) \tag{3}$$

$$F_3 = \mathrm{Conv}_{3\times 3}(F) \tag{4}$$

$$F_5 = \mathrm{Conv}_{5\times 5}(F) \tag{5}$$

Here, the $\mathrm{Conv}_{k\times k}$ convolutional kernel sizes are represented as $k \times k$, and next, the multi-scale feature maps are fused by concatenation to obtain a feature fusion map F_{fused}. The symbol "Concat" refers to the concatenation operation along the channel dimension.

$$\mathrm{Ffused} = \mathrm{Concat}(F_1 F_3, F_5) \tag{6}$$

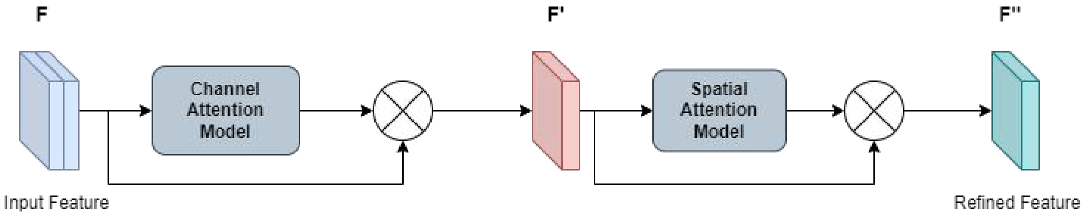


Figure 3 CBAM Model Architectur

As shown in Figure 3, CBAM takes the intermediate feature map F as input and first applies a one-dimensional convolution within the Channel Attention Module. The convolution result is then multiplied by the original feature map to generate the input for the CAM module. Subsequently, the Spatial Attention Module applies a two-dimensional convolution, and the output is multiplied with the original feature map. This process allows the model to focus on the most relevant features, enhancing its ability to interpret and analyze the input effectively [19].

The fused feature map from the Channel Attention Module undergoes global average pooling and global maximum pooling operations, yielding two feature vectors F_{avg} and F_{max}, the two vectors are processed through a shared multi-layer perceptron (MLP) and then summed together. This combined output is passed through a sigmoid function to obtain the channel attention map:

$$F_{avg} = \mathrm{GlobalAvgPool}(F_{fused}) \tag{7}$$

$$\mathrm{F_{max}} = \mathrm{GlobalMaxPool(F_{fused})} \tag{8}$$

$$\mathrm{M_c} = \sigma(\mathrm{MLP(F_{avg}) + MLP(F_{max})}) \tag{9}$$

The Spatial Attention Module performs a similar operation on the fused feature map by applying average pooling and maximum pooling, resulting in two single-channel feature maps $\mathrm{F_{avg\_s}}$ and $\mathrm{F_{max\_s}}$, The two single-channel feature maps are concatenated and passed through a 7×7times convolutional layer. The output is then processed through a sigmoid function to obtain the spatial attention map $\mathrm{M_s}$.

$$\mathrm{F_{avg\_s}} = \mathrm{Mean(F_{fused}, dim = 1)} \tag{10}$$

$$\mathrm{F_{max\_s}} = \mathrm{Max(F_{fused}, dim = 1)} \tag{11}$$

$$\mathrm{M_s} = \sigma(\mathrm{Conv_{7\times7}(Concat(F_{avg\_s}, F_{max\_s}))}) \tag{12}$$

Finally, the channel attention map and the spatial attention map are applied to the fused feature map through element-wise multiplication, represented as:

$$\mathrm{F}_{\mathrm{en}h\mathrm{anced}} = \mathrm{F_{fused} \cdot M_c \cdot M_s} \tag{13}$$

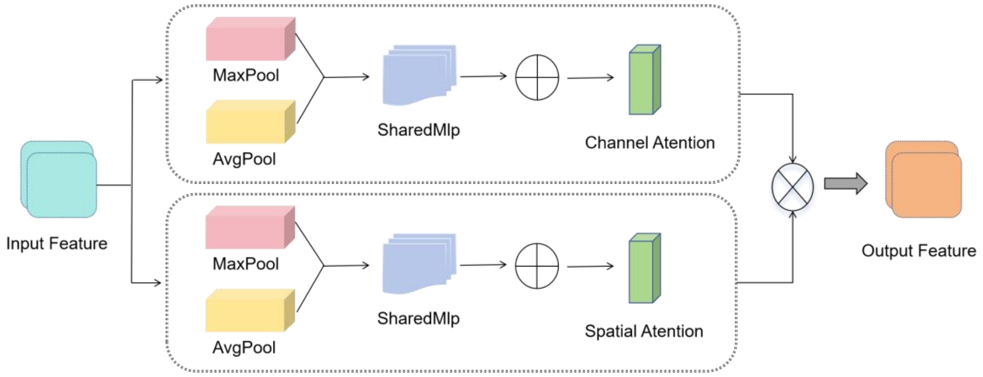


Figure 4 m_CBAM Model Architectur

Through the steps outlined above in Figure 4, the multi-scale feature fusion-based m_CBAM attention mechanism is implemented, enhancing the feature representation capability. This approach effectively leverages multi-scale information and further optimizes the feature map representation through the CBAM attention mechanism, resulting in improved performance of the model across various tasks.

4 EXPERIMENTS DESIGN AND RESULTS ANALYSIS

4.1 Dataset Preparation and Parameter Settings

The study utilized our constructed dataset comprising real classroom scenes from primary, middle, and high school students, ensuring diversity and authenticity in the data. A total of 5,850 video clips and 14,200 behavior annotations were manually extracted from 13 classes. The observed student behaviors were categorized into five types: attentive listening, reading and writing, raising hands, leaning on the desk, and whispering [20]. The dataset was then divided into training and testing sets in a 4:1 ratio to facilitate model evaluation. An illustration of the sample data collection is provided in the figure 5 and 6 below.

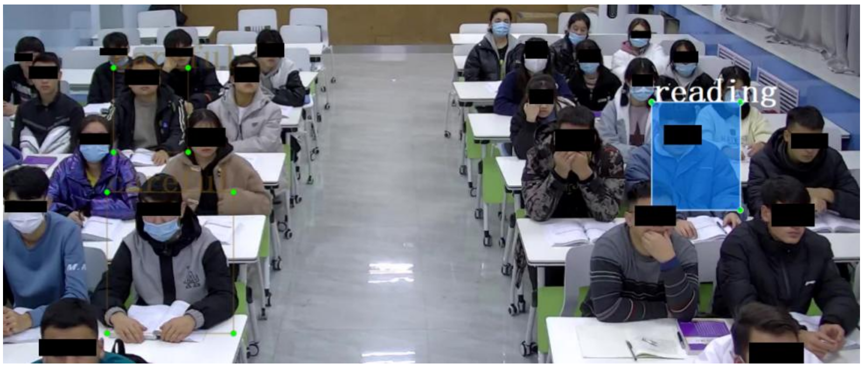


Figure 5 Sample Data Collection from the Dataset (University Classroom)

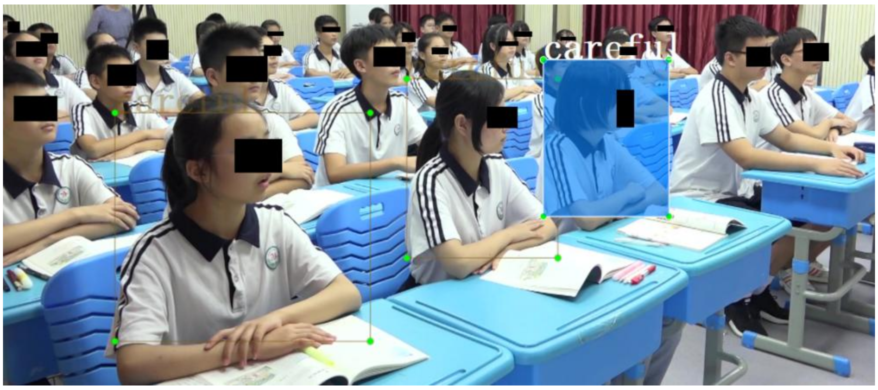


Figure 6 Sample Data Collection from the Dataset (Middle School Classroom)

The analysis of the dataset revealed a significant class imbalance in student behavior categorization. Behaviors such as attentive listening and reading and writing were predominant, while the other three behaviors were relatively infrequent,which can lead to biased model training . To tackle this class imbalance, the study introduced a weighted cross-entropy loss function, where the weight for each class was determined based on its sample proportion relative to the total number of samples in the training set. Additionally, to prevent the underrepresentation of minority classes during training, a customized sampling strategy was implemented. This strategy ensured that samples from all classes were included in the mini-batches during training with Stochastic Gradient Descent (SGD), thus improving the model's robustness.

For experiments, the input image size was set to 640×640 pixels, with a batch size of 16 and an initial learning rate of 0.01, utilizing a cosine annealing learning rate scheduler over 200 batches to achieve smooth learning rate decay. The AdamW optimizer was selected for its effective balance between convergence speed and stability, with a weight decay of 0.005 and a momentum of 0.9. The number of deformable groups in DCNv4 was set to 1, with the modulation mechanism enabled to enhance flexibility in feature extraction.

The data augmentation techniques, including random cropping, flipping, and color jittering, were applied to increase the diversity of the training data and to improve the model's generalization capability. The loss function used in the model comprised localization loss, classification loss, and confidence loss to optimize different aspects of detection performance.

The model training was conducted on a hardware environment featuring a 2.50 GHz, 11th Gen Intel(R) Core(TM) i7-12700H CPU (20 cores) and an NVIDIA GeForce RTX 3090 GPU with 12 GB of VRAM, providing the computational power necessary for efficient training.

4.2 Testing Metrics

In experiments, Recall and Mean Average Precision (mAP) were employed as key metrics for evaluating algorithm performance, alongside considerations of parameter count and Floating Point Operations per Second (FLOP). Recall measures the model's ability to identify all relevant instances, while mAP evaluates detection accuracy across a range of thresholds, providing a comprehensive assessment of the model's precision [21].

Recall evaluates the model’s ability to identify all positive samples by calculating the proportion of correctly detected positive samples out of all actual positive samples. Here, TP (True Positive) represents the number of correctly detected positive samples, and FN (False Negative) is the number of positive samples that were missed.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{14}$$

Mean Average Precision (mAP) is calculated as the mean of average precisions (AP) for all categories. Here, N denotes the total number of categories, and APi represents the average precision for the i-th category. The average precision for each category is computed by calculating the area under the precision-recall curve at various thresholds, offering insights into the model's performance across different levels of confidence.

$$\mathrm{mPA} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{AP_i} \tag{15}$$

The number of parameters reflects the complexity and computational demands of the model, representing the total count of all trainable parameters. Meanwhile, the number of floating point operations per second (FLOP) measures the computational complexity of the model, indicating the number of floating point operations required for a single forward pass. This includes the count of Multiply-Accumulate Operations (MAC). The specific calculation depends on the operations involved in each layer of the model.

$$\mathrm{FLOP} \ = \ 2 \times (\mathrm{MAC}) \tag{16}$$

By combining these metrics, we can comprehensively evaluate the model's performance, complexity, and computational requirements, facilitating informed decisions and regarding optimization and selection.

4.3 Comparative Experiments

To validate the accuracy of the improved algorithm, experiments were conducted comparing it with several advanced algorithms currently used in behavior recognition: YOLOv5, YOLOv8, YOLOv10, and YOLOv11.

Table 1 Recognition Results of Student Classroom Behaviors by Different Algorithm Models

Classroom Behaviors	YOLOv5	YOLOv8	YOLOv10	YOLOv11	**DCB-YOLOv11**
Listening Attentively	0.80	0.72	0.79	0.81	**0.83**
Raising Hands	0.85	0.85	0.94	0.88	**0.94**
Leaning on Desk	0.78	0.82	0.86	0.83	**0.83**
Reading/ Writing	0.81	0.77	0.82	0.81	**0.84**
Whispering	0.74	0.70	0.76	0.77	**0.78**

The four models mentioned above represent the state-of-the-art models in the field of object detection, each with its own strengths and weaknesses, and are commonly chosen for comparison in many studies and experiments. According to the experimental results presented in this paper, the average accuracy of the five algorithms on the same dataset is shown in Table 1 The DCB-YOLOv11 model proposed in this paper outperforms the other models in terms of recognizing most classroom behaviors, except for the behavior of "leaning on the desk," where it does not show a clear advantage over other models.

Upon analysis, it is noted that the "raising hand" gesture exhibits a larger range of motion compared to other classroom behaviors, leading to higher recognition accuracy across all algorithms, with little difference in their performance. For the "whispering" behavior, the recognition accuracy is generally lower due to the camera's positioning, which often misidentifies students sitting at the edge as engaging in whispering, even when they are simply lifting their heads to listen. This issue leads to a lower overall recognition accuracy for this category.

Table 2 Presents the comparative results of the experiments

Algorithm Model	Number of Parameters	Inference Time (ms)	mPA (%)	FLOP(GB)
YOlOv5	1,756729	46	83.97	15.4
YOLOv8	2,946151	65	80.99	13.8
YOLOv10	3,006623	59	84.02	14.2
YOLOv11	2,533114	54	89.50	16.7
DCB-YOLOv11	**2,736232**	**57**	**90.63**	**15.6**

As shown in Table 2, a comprehensive comparison experiment was conducted on advanced models such as YOLOv5, YOLOv8, YOLOv10, and others, evaluating their performance in four aspects: parameter count, inference time, mean average precision (mAP), and floating-point operations (FLOP). The results show that YOLOv5 has the fewest parameters and the shortest inference time, with an mAP value ranking in the middle. This may be due to YOLOv5's design philosophy, which is known for being lightweight and real-time, and its design helps it better integrate into various applications. YOLOv8 has an advantage only in terms of floating-point operations, while YOLOv10 achieved improved accuracy with an increase in parameter count. YOLOv11, due to its own significant improvements, achieved a substantial increase in accuracy while reducing the parameter count.

However, the improvement proposed in this paper not only results in a slight increase in inference time and parameter count but also achieves a higher accuracy while ensuring a reduction in floating-point operations. Considering the comparative dimensions and practical application scenarios, DCB-YOLOv11 is more suitable for use in classroom student behavior recognition.

5 CONCLUSIONS AND FUTURE WORKS

The combination of scalable education and personalized training is a key goal of intelligent education. This paper proposes a student behavior recognition model based on YOLOv11 (DCB-YOLOv11), which improves classroom student behavior recognition accuracy by incorporating DCNv4 deformable convolutions in the backbone and detection head, and adding the CBAM attention mechanism in the SPPF pooling layer. Experimental results show that the model achieves an average precision improvement of 2.1% over the original YOLOv11 on a self-constructed dataset, demonstrating superior performance. The improved model can better capture target deformation and complex features, enhance feature dependencies, and support behavior recognition of multiple students. This provides more convenient and accurate information for teaching evaluation, promoting the integration of artificial intelligence and educational big data to improve classroom teaching.

In the future, by expanding the dataset, optimizing the model structure, achieving real-time applications, and extending to other fields, an intelligent education system can be built to further enhance the technical level and application breadth. By combining behavior analysis with multi-dimensional student data and customizing personalized teaching strategies, this will promote more intelligent and efficient education, significantly improving educational quality and effectiveness.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

This study was supported by the:
(1) Natural Science Foundation of Xinjiang Uygur Autonomous Region (Grant No. 2022D01A226);
(2) Regional Innovation Cooperation Project of Sichuan Province (Grant No. 2023YFQ0066).

REFERENCES

[1] Liu GL, Chen ZZ, Chen R. Real-time S-T analysis method based on speaker recognition[C]// In: 2020 2nd international conference on advanced control automation and artificial intelligence(ACAAI 2020), Wuhan, Hubei, China. 2020, 147-151 .

[2] 2024 Global Smart Education Conference. Research on Educational Technology, 2024, 45(08): 2.

[3] Zaletelj J, Košir A. Predicting students’ attention in the classroom from Kinect facial and body features. EURASIP J. Image Video Process, 2017, 1-12.

[4] Zheng R, Jiang F, Shen R. GestureDet. Real-time student gesture analysis with multi-dimensional attention-based detector[C]// In Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020). Yokohama, Japan. 2020, 680-686.

[5] Huang Y, Liang M, Wang X, et al. Multi-person classroom behavior recognition in teaching videos based on deep spatiotemporal residual convolutional neural networks. Journal of Computer Applications, 2022, 42(3): 736-742.

[6] Lin T Y, Dollar P, Girshick S, et al. Feature Pyramid Networks for Object Detection[C]// In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, 936-944.

[7] Zhao Q, Sheng T, Wang Y, et al. M2det: A single-shot object detector based on multi-level feature pyramid network[C]// In Proceedings of the AAAI conference on artificial intelligence, 2019, 9259-9266.

[8] Wang Z, Jiang F, Shen R. An effective yawn behavior detection method in classroom[C]// In Proceedings of the 26th International Conference on Neural Information Processing (ICONIP2019). 2019, 430-441.

[9] Guo J, Lv J, Wang R, et al. Classroom behavior recognition of teachers and students driven by deep learning models. Journal of Beijing Normal University (Natural Science Edition), 2021, 57(6): 905-912.

[10] Zheng Z, Liang G, Luo H, et al. Attention assessment based on multi-view classroom behaviour recognition. IET Comput. Vis, 202.

[11] Niu W, Sun X, Yi K. Improved YOLOv5 for skeleton-based classroom behavior recognition[C]// In: Proc of the third International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI 2022), 2023, 107-112.

[12] Zhang Z, Ao D, Zhou L, et al. Laboratory Behavior Detection Method Based on Improved Yolov5 Model[C]// In: Proc. of 2021 Int. Conf. Cyber-Physical Soc. Intell, 2021, 1-6.

[13] Yang Fan. Student Classroom Behavior Detection based on Improved YOLOv7. 2023. DOI: 10.48550/arXiv.2306.03318.

[14] Yang Fan, Tao Wang, Wang Aofei. Student Classroom Behavior Detection Based on YOLOv7+ BRA and Multi-model Fusion[C]// International Conference on Image and Graphics. Cham: Springer Nature Switzerland. 2023, 41-52.

[15] Yuwen ong, et al. Efficient deformable convnets: Rethinking dynamic and sparse operator for vision applications[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 5652-5661.

[16] Qi Han, Zejia Fan, Qi Dai, et al. On the connection between local attention and dynamic depth-wise convolution. 2021. DOI: https://doi.org/10.48550/arXiv.2106.04263.

[17] Sanghyun W, Jongchan P. CBAM: convolutional block attention module proceedings of the European Conference on Computer Vision (ECCV). 2018, 3-19.

[18] Xizhou Zhu, Weijie Su, Lewei Lu, et al. Deformable detr: Deformable transformers for end-to-end object detection. 2020. DOI: https://doi.org/10.48550/arXiv.2010.04159.

[19] Kim J H, Kim N, Yong Woon Park, et al. Object detection and classification based on YOLO-V5 with improved maritime dataset. Journal of Marine Science and Engineering, 2022, 10(3): 377.

[20] Wang Chien-Yao, Alexey Bochkovskiy, Hong-Yuan Mark Liao, et al. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[D]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, 7464-7475.

[21] Wang X. Focus on Development, Emphasize Process: Exploration of Developmental Comprehensive Quality Evaluation for High School Students. Primary and Secondary School Management, 2020, (10): 21-23.

[22] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, et al. Flexible and deformable convolution for point clouds[D]// In Proceedings of the IEEE/CVF international conference on computer vision. 2019, 6411-6420.

[23] Fagad Rasheed A, Zarkoosh M. YOLOv11 Optimization for Efficient Resource Utilization. 2024. DOI: https://doi.org/10.48550/arXiv.2412.14790.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3045

BIDIRECTIONAL SEMANTIC AND HIERARCHICAL SYNTACTIC SENTIMENT CLASSIFICATION BASED ON GCN

Can Jia, Azragu*
College of Computer Science and Technology, Xinjiang Normal University, Urumqi 830054, China.
Corresponding Author: Azragu, Email: Azragul2010@126.com

Abstract: In order to solve the problems of insufficient coordination between global semantics and local syntactic features and noise interference of dependency parsing in aspect sentiment classification tasks, this paper proposes a bidirectional semantic enhancement and hierarchical syntactic analysis model based on graph convolutional network (GCN). The model effectively integrates semantic enhancement GCN and syntactic enhancement GCN for feature interaction to accurately model the complex hierarchical relationship between aspect words and sentiment words. In semantic modeling, self-attention and perceptual aspect attention (ASA) are integrated to extract deep semantic information through the attention fusion mechanism (GAFM). In terms of syntactic feature extraction, the syntactic distance mask matrix is introduced to measure semantic association, and the syntactic modeling is optimized in combination with the dependency relationship. In terms of structural optimization, the hierarchical phrase structure is adopted to fuse the syntactic dependency matrix with the phrase matrix to significantly reduce the noise of dependency tree parsing. Experimental results show that the model performs well on multiple datasets, with consistently improved accuracy and stability. Ablation experiments and visualization analysis further verify the effectiveness of each module, proving that the combination of bidirectional semantic enhancement and hierarchical syntactic analysis helps substantially improve the performance of aspect sentiment classification.

Keywords: Graph convolutional network; Semantic enhancement; Syntactic enhancement; Feature interaction; Aspect sentiment classification

1 INTRODUCTION

Aspect-based Sentiment Analysis (ABSA) aims to identify the sentiment tendencies of different aspects in text and is a key task in sentiment analysis. Traditional methods mainly focus on overall sentiment analysis, which is difficult to meet the needs of accurate identification. Therefore, ABSA improves the precision of sentiment analysis by modeling the association between sentiment polarity and specific aspects.

Current research mainly focuses on the self-attention mechanism and the graph convolutional network (GCN) based on the dependency tree to model the dependency relationship between aspect words and context. However, the self-attention mechanism may cause semantic feature loss when capturing the association between aspect words and context, while the dependency tree may introduce noise due to parsing errors or irregular text structure, weakening the model's discriminative ability.

In response to the above challenges, this chapter proposes a bidirectional semantic and hierarchical syntactic aspect sentiment classification model (SEAFM-GCN) based on GCN. The model improves the feature representation ability and enhances the accuracy and robustness of sentiment classification by integrating semantic information and syntactic information. In terms of semantic enhancement, we first use BiLSTM to extract context representation, and then combine self-attention and perceptual attention to calculate attention weights, which are then fused with the syntactic mask matrix to generate the adjacency matrix of GCN input to extract global semantics and local syntactic features. In terms of syntactic enhancement, we introduce a hierarchical phrase structure to explore the phrase collocation relationship within the sentence, and optimize the syntactic feature representation by screening out irrelevant dependency edges. Finally, we interactively fuse the extracted features to improve the classification ability of sentiment polarity.

2 RELATED WORK

GCN has outstanding performance in text syntactic modeling. Zhou et al[1]. introduced common sense knowledge to optimize the representation of aspect terms. Qi et al[2]. used weight matrices to enhance syntactic relations. Chen et al[3]. proposed D-GCN, combining dependency encoding to optimize aspect word extraction. Zhang et al[4]. first introduced syntactic dependency in GCN to improve the sentiment classification effect. Nguyen et al[5]. strengthened the connection between aspect words and sentiment words through dependency trees. Gu et al[6]. proposed MFSGC to optimize the adjacency matrix, remove irrelevant information, and strengthen dependencies.

The current study combines self-attention with GCN, integrates semantic and syntactic features, promotes the development of aspect sentiment classification, and provides theoretical support for this study.

3 MODLE CONSTRUCTION

The SEAFM-GCN model framework is shown in Figure 1, which consists of five parts: input module, attention fusion module, semantic enhanced graph convolution module, syntactic enhanced graph convolution module, feature interaction module and classification module.

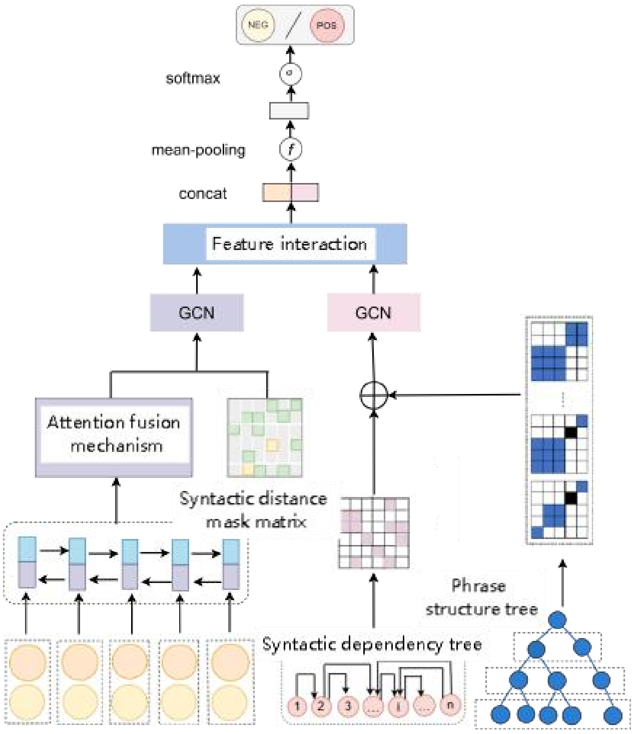


Figure 1 SEAFM-GCN Model Framework

3.1 Input Module

The input module consists of two parts: the embedding layer and the encoding layer.

(1) Embedding layer

Currently, models with relatively good results widely use pre-trained language models as the basis. For the sake of comparison, this model uses the BERT pre-trained language model as part of the word embedding. Considering that the task of performing aspect-level sentiment analysis on the course review text is to adjust the sentence format to [CLS]+sentence+[SEP] +aspect+[SEP] before sending it to BERT.

(2) Encoding layer

In the encoding layer, BiLSTM is used to encode the word vector sequence X to generate a hidden vector that integrates context information. The word vector sequence generated by the embedding layer is input into the BiLSTM, and is processed according to the encoding order from front to back and from back to front, so that two different hidden vectors are generated at the output of the two LSTMs, which can be expressed as and. Finally, the final hidden vector is obtained by concatenating the vector and the vector. This paper uses the vector and the vector as the output of the encoding layer, mainly because these two vectors can capture richer feature information. The specific formulas are shown in Equations 1 to 3.

$$\overrightarrow{h}_t = \text{LSTM}\left(x_t, \overrightarrow{h}_{t-1}\right) \tag{1}$$

$$\overleftarrow{h}\text{t} = \text{LSTM}\left(x_t, \overleftarrow{h}_{t-1}\right) \tag{2}$$

$$h_t = \left[\overrightarrow{h}_t, \overleftarrow{t}_t\right] \tag{3}$$

Among them, represents the word vector at time t, represents the output of the LSTM from left to right at time t, represents the output of the LSTM from right to left at time t, and represents the output from the BiLSTM at time.

3.2 Attention Fusion Module

This chapter proposes a Global-Aware Fusion Mechanism (GAFM) to extract global and aspect-related semantic features efficiently. GAFM combines a Self-Attention Mechanism (SA) and an Aspect-Sensitive Attention Mechanism (ASA). Additionally, a syntactic distance mask matrix serves as an adjacency matrix, enhancing the SEAFM-GCN model's ability to capture semantic and syntactic distance features.

3.2.1 Self-Attention Mechanism (SA)

In this study, the hidden state output of the encoding layer is represented as , and the self-attention mechanism is employed to extract global semantic feature information. After three linear transformation operations, the Q, K, and V matrices are obtained. The self-attention score matrix for the text data $A_{\text{Self}} = \{h_1^S, h_2^S, \cdots, h_n^S\}$ can be computed using Equation 4, as shown below.

$$\mathrm{A_{Self}} = \frac{h^{\mathrm{i}}\mathrm{W^{q}}\times\left(h^{\mathrm{i}}\mathrm{W^{k}}\right)^{\mathrm{T}}}{\sqrt{\mathrm{d^{i}}}}h^{\mathrm{i}}\mathrm{W^{v}} \tag{4}$$

Among them,$\mathrm{W^q}$、$\mathrm{W^k}$and$\mathrm{W^v}$are parameter matrices that can be adaptively optimized during the training process of the SEAFM-GCN model to improve prediction accuracy. represents the feature dimension of the input vector, playing a crucial role in the model's performance.

3.2.2 Aspect-Sensitive Attention Mechanism(ASA)

In the Aspect-Sensitive Attention Mechanism, the output vectors from the encoding layer are multiplied by the aspect mask matrix to extract the vectors corresponding to the aspect terms,represented by a∈R1×d,where d denotes the dimension of the hidden layer. The detailed calculation process is shown in Equation 5.

$$a = mean\ (mask \times h^{i}) \tag{5}$$

In the formula, the mean function represents average pooling. The final feature representation is generated by replicating a n times.

The obtained representation is fed into the ASA layer to capture aspect-related semantic features. The aspect-sensitive attention score matrix is computed using Equation 6.

$$A_{ASA} = tanh\ \left((h^{a}W^{a}) \times \left(h^{i}W^{k}\right)^{T} + b^{a}\right) \tag{6}$$

$tanh$ represents the activation function, h^{i} denotes the output of the encoding layer, W^{a} and W^{k} are the weight matrices, and b^{a} is the bias term.

Finally, according to Equation 7, the global semantic information is fused with the aspect-specific semantic information to generate the final attention score matrix. In the equation, α is the scaling factor.

$$A^{Att} = softmax\ (A_{Self} + \alpha A_{ASA}) \tag{7}$$

3.2.3 Syntactic Mask Matrix

First, the syntactic mask matrix is applied to mask each fully connected graph. The syntactic dependency tree is treated as an undirected graph, where nodes represent tokens and the distance between nodes is defined by a function. The shortest path distance between nodes is denoted as DD, as shown in Equation 8.

$$D(i,j) = mind\left(v_{i}, v_{j}\right) \tag{8}$$

3.3 Semantic Enhanced Graph Convolutional Module

As shown in Equation 9, The model performs graph convolution on the fused attention mask matrix.

$$h_{i}^{l} = \sigma\left(\sum_{j=1}^{n} A_{ij} W^{l} h_{j}^{l-1} + b^{l}\right) \tag{9}$$

Here, σ is the nonlinear function,$\mathrm{W^l}$ is the weight matrix, $\mathrm{b^l}$ is the bias term.

3.4 Syntactically Enhanced Graph Convolutional Module

As shown in Equation 10.This paper uses the Stanford Dependency Parser developed by Stanford University to automatically parse syntactic structures. This tool extracts part-of-speech information and their dependency relations, providing detailed syntactic analysis support for text understanding. In contrast, phrase structure trees focus on representing the hierarchical structure and phrase-level relationships of the text. Their construction relies on phrase structure analysis, where each node corresponds to a specific phrase, and the phrases at each level cover all the words in the sentence, capturing richer syntactic features. The proposed method uses the Stanford Constituency Parser to generate phrase structure trees, revealing the hierarchical structure and lexical organization within a sentence. By segmenting the hierarchical nodes in the phrase structure tree, each node corresponds to a phrase, providing detailed syntactic information that supports subsequent sentiment analysis and enhances the model's understanding of sentence semantics.

$$P_{i,j}^{l} = \begin{cases} 1 & if\ w_{i}, w_{j} in\ same\ phrase\ of\ \left\{h_{p}^{l}\right\} \\ 0 & otherwise \end{cases} \tag{10}$$

Here, $h_{\mathrm{p}}^{\mathrm{l}}$ represents the vector representation of each phrase at layer.L denotes the number of layers in the phrase set, and p refers to the number of phrases at that layer.

Relying only on syntactic dependency trees may result in incomplete word relationship modeling, affecting sentiment classification. To address this, we combine phrase structure tree information to improve contextual modeling and enhance sentiment classification accuracy, as shown in Equation 11.

$$\mathrm{D_{i,j}} = \begin{cases} 1 & \mathrm{if\ i{=}j\ o\ rw_{i}, w_{j}\ dependency\ tree} \\ 0 & \mathrm{otherwise} \end{cases} \tag{11}$$

As shown in Equation 12.By merging the phrase and syntactic dependency matrices, the adjacency matrix captures both phrase-level and sentence-level syntactic features{$\mathrm{PD^1}$，$\mathrm{PD^2}$，…, $\mathrm{PD^l}$},and is used as the input to the GCN.

$$h_{Di}^{(l)} = \rho\left(\sum_{j=1}^{n} PD_{ij} W^{(l)} h_{j}^{(l-1)} + b^{(l)}\right) \tag{12}$$

3.5 Feature Interaction Module and Classification Module

3.5.1 Feature fusion layer

The feature fusion extraction module performs deep interaction and fusion of the semantic feature matrix and syntactic feature matrix, fully integrating information to overcome the limitations of a single feature. This reduces the impact of information redundancy on model performance and improves the accuracy of sentiment polarity classification. After the interaction, the module concatenates the two types of features to construct the final feature representation, as shown in equation (13).

$$\mathrm{H}^{\mathrm{AB}} = \mathrm{concat}(\mathrm{H}^{\mathrm{Sem}},\ \mathrm{H}^{\mathrm{Syn}}) \tag{13}$$

3.5.2 Sentiment output layer

The final feature vector is obtained through average pooling, and then the input is passed to the sentiment classification module, which uses a function for sentiment polarity classification. The calculation formulas are shown in equations 14 and 15.

$$\mathrm{H} = \mathrm{Average\ Pooling}(\mathrm{H}^{\mathrm{AB}}) \tag{14}$$

$$\mathrm{Y} = \mathrm{softmax}(\mathrm{W}^{\mathrm{T}}\mathrm{H} + \mathrm{b}) \tag{15}$$

4 EXPERIMENTS AND RESULTS ANALYSIS

4.1 Experimental data and evaluation indicators

The experiments in this chapter are conducted based on the dataset constructed in Chapter 3. The relevant data and statistical information have been thoroughly explained in the previous section and will not be repeated here.

For evaluation metrics, this study follows the evaluation methods used in previous related works, employing Precision (P) and Recall (R) to calculate the F1-score, which measures the model's classification performance (as detailed in Section 6 of Chapter 2). The F1-score, as the harmonic mean of precision and recall, approaches 1 when the model achieves better classification performance.

4.2 Experimental parameter settings

The SEAFM-GCN model utilizes the bert-base-chinese version of the BERT model, with a word embedding dimension of 768 and a part-of-speech (POS) embedding dimension of 100. The specific experimental hyperparameter settings are shown in Table 1. The loss function employed is the cross-entropy loss function.

Table 1 Experimental Parameter Settings

Parameter	Value
BERT Embedding Dimension	768
POS Embedding Dimension	100
Optimizer	BertAdam
Batch Size	8
Number of GCN Layers	2
GCN Dropout	0.1
Dropout Rate	0.3
Learning Rate	2×10^{-5}
Number of Attention Heads	5
Epochs	20

4.3 Comparative Experiments on Different Models

To evaluate the performance of the SEAFM-GCN model on aspect-level sentiment analysis of MOOC course reviews, this study selected the following baseline models for comparative experiments on the constructed aspect-level MOOC review dataset. The experimental results are shown in Table 2.

AOA: Utilizes multiple attention layers to model the interaction between aspects and sentences, effectively preserving aspect-specific sentiment features.

MGAN[7]: Introduces multi-granularity attention to better capture word-level interactions between aspects and sentences.

ASGCN: Combines syntactic dependency trees and graph convolutional networks (GCN), leveraging attention mechanisms to adjust feature weights for classification.

BERT-SPC[8]: Leverages the BERT pre-trained model to extract contextual information, followed by pooling word vectors to optimize classification performance.

AEN-BERT[9]: Employs the BERT model and an attention-based encoder to model the relationship between context and aspect terms.

R-GAT+BERT[10]: Transforms the dependency tree into a star graph with edges defined by minimal distance and dependency types, incorporating relational GAT with attention-based aggregation.

SSEGCN+BERT[11]: Uses a syntactic mask matrix obtained via a distance-based mask mechanism and incorporates attention mechanisms with GCN to enhance the representation of nodes related to aspect-specific sentiment features.

SEAFM-GCN: Proposes a model that integrates bidirectional semantics with hierarchical syntax through a graph convolutional network.

Table 2 Comparison of Results on MOOC Aspect-Level Datasets (%)

	MOOC1		MOOC2	
Method	Accuracy	F1	Accuracy	F1
AOA	83.25	69.15	82.62	68.82
MGAN	82.64	74.32	82.45	74.17
ASGCN	83.76	75.48	83.64	75.26
BERT-SPC	92.38	83.05	92.02	82.43
AEN-BERT	94.47	85.73	93.73	84.68
R-CAT+BERT	94.61	85.91	94.24	84.88
SSEGCN+BERT	95.05	86.62	94.65	85.74
SEAFM-GCN	**96.47**	**87.89**	**95.22**	**86.48**

Table 2 presents the experimental results of the proposed SEAFM-GCN model compared with baseline models, showing superior performance on both MOOC1 and MOOC2 datasets. The first three models, lacking BERT embeddings, demonstrated limited performance. However, ASGCN, incorporating GCN with syntactic information, achieved relatively good results. Models utilizing BERT significantly improved performance due to pre-trained contextual knowledge. Compared with R-CAT+BERT, the proposed model improved F1 scores by 1.98 and 1.6 points on MOOC1 and MOOC2, respectively, attributed to the integration of hybrid attention and syntactic information, enhancing context-aspect interaction. While SSEGCN+BERT considered syntactic distances in attention computation, it underutilized local context relevant to aspect terms, resulting in lower F1 scores than SEAFM-GCN.

Overall, SEAFM-GCN effectively integrates structural and semantic information from local to global levels, outperforming other models and validating the fusion of syntax and semantics in aspect-level sentiment classification.

4.4 Feature Ablation Experiment

This section analyzes the impact of key modules within the SEAFM-GCN model by conducting extensive ablation experiments. The experimental results were recorded by averaging multiple runs, as shown in Table 3. In the table, “w/o” indicates the removal of a specific module, with the best results highlighted in bold.

Without Phrase Structure Tree (SEAFM-GCN w/o phrase): Retains only the traditional syntactic dependency matrix.

Without Syntactic Distance Mask Matrix (SEAFM-GCN w/o SDMM): Uses only the attention score matrix as the semantic feature map.

Without Aspect-Sensitive Attention (SEAFM-GCN w/o ASA): Replaces the aspect-sensitive attention matrix with a self-attention score matrix as the adjacency matrix, combined with the syntactic distance mask matrix.

Without Self-Attention (SEAFM-GCN w/o SA): Uses only the aspect-sensitive attention score matrix as the adjacency matrix, combined with the syntactic distance mask matrix.

Table 3 Ablation Study Results on MOOC Review Datasets (%)

	MOOC1		MOOC2	
Method	Accuracy	F1	Accuracy	F1
SEAFM-GCN	96.47	87.89	95.22	86.48
w/o phrase	92.18	86.77	91.23	86.12
w/o SDMM	94.43	86.48	92.46	85.89
w/o ASA	95.45	85.64	94.57	84.53
w/o SA	93.42	84.71	93.18	83.22

Conclusions from Ablation Study Results:

(1) Removing the phrase structure tree module led to performance degradation, highlighting the importance of phrase adjacency matrix integration for reducing parsing errors.

(2) The ASA layer improves feature extraction by optimizing noise in the self-attention mechanism. Removing the syntactic distance mask matrix caused a drop in F1 scores by 1.41% and 0.59%, showing its role in enhancing structural information extraction.

(3) Deleting the ASA layer reduced classification performance, demonstrating its contribution to aspect-specific feature extraction. Without ASA, self-attention alone loses aspect-related information, especially with multiple aspects of different sentiments.

(4) Removing the self-attention layer decreased classification accuracy, confirming its importance in extracting global semantic features. The impact was more significant than removing the ASA module, indicating that self-attention benefits final sentiment classification.

4.5 Visualization Analysis

This experiment first verifies the impact of the number of GCN layers on model performance to optimize the model structure and guide subsequent experimental design. Specifically, experiments were conducted with GCN layers ranging from 1 to 5, and the performance at each layer count was evaluated. The results are shown in Table 4, presenting the F1 scores of the SEAFM-GCN model for different GCN layer counts. Additionally, Figure 2 further visualizes the trend of how varying the number of GCN layers affects the model's F1 score on the two datasets. The experiment shows that increasing the number of GCN layers moderately improves the model's performance, but too many layers may lead to a plateau or overfitting.

Table 4 F1 Values of SEAFM-GCN Model at Different GCN Layer Numbers

Number of GCN Layers	1	2	3	4	5
MOOC1	86.23	**87.89**	86.42	87.28	86.16
MOOC2	85.51	**86.48**	85.79	86.35	85.34

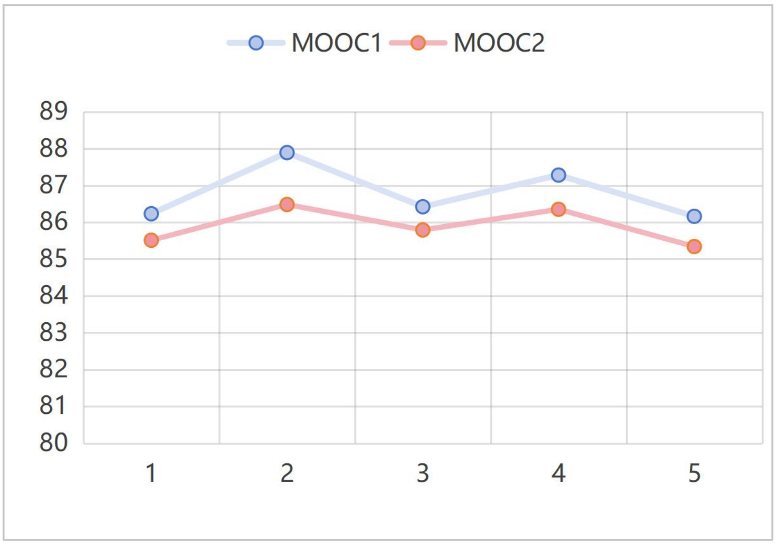


Figure 2 The Impact of GCN Layers on F1 Values

From Figure 2, it can be observed that the number of GCN layers significantly impacts the model's performance on both the MOOC1 and MOOC2 datasets. On the MOOC1 dataset, the best performance (87.89) was achieved with 2 GCN layers, while performance slightly declined with other layer configurations, suggesting that 2 layers of GCN are most effective in balancing information extraction and noise control. The results for the MOOC2 dataset were similar, with 2 GCN layers also yielding the best performance (86.48), although the performance fluctuation was smaller, likely due to the simpler or more robust nature of the dataset. In both datasets, the model's F1 value reached its maximum with 2 layers, indicating that 2 layers should be considered the optimal choice for the number of GCN layers.

5 CONCLUSION

This chapter presents a bidirectional semantic and hierarchical syntactic aspect sentiment classification model (SEAFM-GCN) based on Graph Convolutional Networks (GCN). The model first designs a perception-based aspect attention mechanism to learn aspect-related semantic information, which is then combined with a self-attention mechanism to form a fused attention mechanism. Next, by integrating phrase structure and syntactic dependency structures, it enhances syntactic feature representation, addressing the noise issue in traditional syntactic dependency analysis. Finally, Graph Convolutional Networks are used to further optimize the syntactic features. Experimental results demonstrate that the model achieves significant performance improvement on the MOOC review dataset, validating the effectiveness of the attention mechanism and syntactic enhancement strategy.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Zhou J, Huang J X, Hu Q V, et al. Sk-gcn: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. Knowledge-Based Systems, 2020, 205: 106292.

[2] Gu Yuying, Gao Meifeng. Aspect-level sentiment analysis based on the fusion of part-of-speech and external knowledge . Computer Science and Exploration, 2022(06):1-13.

[3] Chen G, Tian Y, Song Y. Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks//Proceedings of the 28th International Conference on Computational Linguistics. 2020: 272-279.

[4] Zhang Chen, Li Qiuchi, Song Dawei. Aspect-based sentiment classification with aspect-specific graph con-volutional networks / /Proceedings of the 2019 Con-ference on Empirical. Methods in Natural Language Pro-

cessing and the 9th International Joint Conference on Natural Language Processing(EMNLP-IJCNLP), Hong Kong, China, November 3-7, 2019, New York: ACM, 2019: 4567-4577.

[5] Nguyen T H, Shirai K. Aspect-Based Sentiment Analysis Using Tree Kernel Based Relation Extraction//The 16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Cairo, Apr 14-20, 2015. Berlin: Springer, 2015: 114-125.

[6] Gu Yuying, Gao Meifeng. Aspect-level sentiment analysis based on the fusion of part-of-speech and external knowledge. Computer Science and Exploration, 2022(06): 1-13.

[7] Fan F, Feng Y, Zhao D. Multi-grained attention network for aspect-level sentiment classification//Proceedings of the 2018 conference on empirical methods in natural language processing. 2018: 3433-3442.

[8] Yang H, Zeng B, Yang J,et al. A multi-task learning model for Chinese-oriented aspect polarity classification and aspect term extraction.Neurocomputing, 2020, 419:344-356.

[9] Song Y, Wang J, Jiang T, et al. Attentional encoder network for targeted sentiment classification. arXiv preprint arXiv:1902.09314, 2019.

[10] Wang K, Shen W Z, Yang Y Y, et al. Relational Graph Attention Network for Aspect-based Sentiment Analysis//Pro-ceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020: 3229-3238.

[11] Zhang Z, Zhou Z, Wang Y. SSEGCN: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis//Proceedings of the 2022 conference of the North American chapter of the association for computational Linguistics: human language technologies. 2022: 4916-4925.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3046

UNDERWATER IMAGE ENHANCEMENT TECHNIQUE BASED ON CYCLEGAN AND FREQUENCY DECOMPOSITION CORRECTION MODEL

LuHeng Wang, HaoLong Qi, LiYe Zhang*
School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, Shandong, China.
Corresponding Author: LiYe Zhang, *Email: zhangliye@sdut.edu.cn*

Abstract: Enhancing underwater images is a challenging task due to degradation from light scattering, absorption, and low contrast, which obscure important details. This paper proposes a novel method that combines Cycle-Consistent Generative Adversarial Network (CycleGAN) with Frequency Decomposition Correction to improve underwater image quality. Initially, CycleGAN was used to generate clearer images from blurred underwater photos, addressing distortions caused by the underwater environment. Next, Frequency Decomposition Correction separates the image into low-frequency (smooth areas) and high-frequency (details and edges) components. The low-frequency component is enhanced for better clarity, while the high-frequency component is sharpened to improve fine details and edges. The enhanced images are evaluated using PSNR, UCIQE, and UIQM metrics. Results show that CycleGAN significantly improves image quality, and the additional Frequency Decomposition Correction boosts clarity, contrast, and detail further. This combined method presents a promising solution for underwater image enhancement, with potential applications in marine research, underwater navigation, and environmental monitoring.

Keywords: CycleGAN; Frequency decomposition; Wavelet transform; CLAHE

1 INTRODUCTION

Underwater images are frequently degraded due to light scattering and absorption, leading to issues such as low contrast, color distortion, and blurriness [1]. These problems diminish visibility and complicate underwater analysis, making tasks like marine research, navigation, and environmental monitoring more challenging. The degradation results in distorted colors reduced clarity, and blurriness, which hampers important tasks like object detection and scene reconstruction in underwater environments [2]. Several techniques have been proposed to address these challenges in underwater image enhancement. One approach combines Contrast Limited Adaptive Histogram Equalization (CLAHE) with Total Generalized Variation (TGV) to improve contrast while preserving image details. Although this method improves UCIQE and PSNR metrics, it falls short in recovering color accuracy and fine details, especially in highly degraded images [3]. UIE-Net, a CNN-based approach, attempts to perform both color correction and haze removal. However, it struggles with artifacts and often fails to restore fine details in complex underwater conditions [4]. GAN-based methods like GAN-RS utilize multi-branch discriminators to preserve image content and remove noise. While they show effectiveness in real-time seabed experiments, these methods still do not adequately address the low contrast problem or recover fine details, which are essential for high-quality underwater image restoration [5]. UW-CycleGAN enhances unpaired underwater images by using content loss regularization and adversarial loss. Although it provides better visual quality, it does not fully address contrast enhancement or fine detail sharpening in severely distorted underwater conditions [6].

The paper addresses these limitations by proposing a novel method that combines CycleGAN with frequency-domain decomposition using wavelet transforms. Unlike existing approaches that enhance either color or details separately, the proposed method decomposes the image into low and high-frequency components [7]. CLAHE is applied to enhance the low-frequency components, improving overall clarity, while high-frequency sharpening is used to recover fine details [8]. This dual enhancement strategy not only enhances image contrast and clarity but also significantly improves detail preservation, overcoming the shortcomings of both traditional and GAN-based methods. The innovation of this paper lies in the integration of CycleGAN with frequency-domain correction, which allows for simultaneous enhancement of contrast and detail recovery. Experimental results demonstrate that the proposed method significantly improves image quality, as evaluated through PSNR, UCIQE, and UIQM metrics. By addressing the limitations of previous methods in recovering fine details and enhancing contrast, the paper presents a more effective and practical solution for underwater image enhancement, offering a robust approach to the challenges faced with underwater imaging.

2 MODEL

2.1 Cycle GAN model

2.1.1 The structure of generator and discriminator

In CycleGAN, the generator transforms blurry underwater images (source domain) into clearer, enhanced images (target domain). Generator G produces enhanced images, while Generator F converts them back to their blurry versions. The generator utilizes a CNN with an encoder-decoder structure, where the encoder maps the input image to a latent space, and the decoder reconstructs it into the target domain.

The discriminator in CycleGAN differentiates between real and generated images. Discriminator D_X determines if an image belongs to the source domain (blurry underwater images) or is produced by Generator F, while Discriminator D_Y does the same for enhanced images generated by G. Both discriminators act as binary classifiers, aiming to reduce the likelihood of misclassifying generated images as real [9]. The generator and discriminator structures are illustrated in Figure 1:

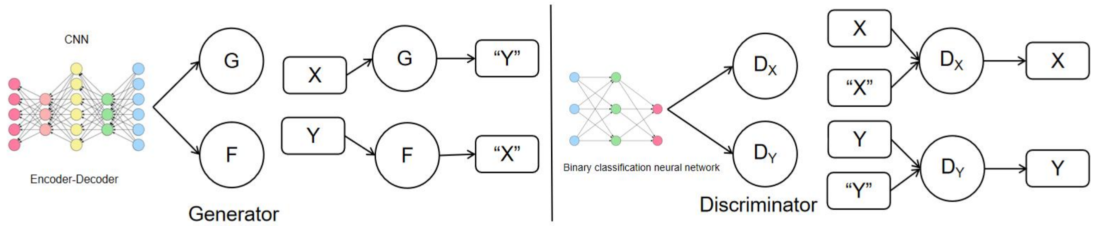


Figure 1 The Structure of the Generator and the Discriminator

2.1.2 CycleGAN loss functions

1. Cycle Consistency Loss and Identity Loss

The specific formula for this loss is as follows:

$$\mathcal{L}_{\text{cycle}}(\mathrm{G},\mathrm{F}) = \left[\mathrm{E}_{\mathrm{x}\sim \mathrm{P}_{\text{data}}(\mathrm{x})}\left[\| \mathrm{F}\big(\mathrm{G}(\mathrm{x})\big) - \mathrm{x}\|_1\right] + \mathrm{E}_{\mathrm{y}\sim \mathrm{P}_{\text{data}}(\mathrm{y})}\left[\| \mathrm{G}\big(\mathrm{F}(\mathrm{y})\big) - \mathrm{y}\|_1\right]\right]_{\min} \quad (1)$$

where $\mathrm{E}_{\mathrm{x}\sim \mathrm{P}_{\text{data}}(\mathrm{x})}\left[\| \mathrm{F}\big(\mathrm{G}(\mathrm{x})\big) - \mathrm{x}\|_1\right]$: This term calculates the difference between the source domain image x and the image $\mathrm{F}\big(\mathrm{G}(\mathrm{x})\big)$, where G converts x to the target domain, and F maps it back to the source domain. $\mathrm{E}_{\mathrm{y}\sim \mathrm{P}_{\text{data}}(\mathrm{y})}\left[\| \mathrm{G}\big(\mathrm{F}(\mathrm{y})\big) - \mathrm{y}\|_1\right]$: This term calculates the difference between the target domain image y and the image $\mathrm{G}\big(\mathrm{F}(\mathrm{y})\big)$, where F converts y to the source domain, and G transforms it back to the target domain. The goal is to minimize this difference, ensuring that the content of the original image y is preserved after the transformation [10].

The formula for identity loss is as follows:

$$L_{idt}(G,F) = E_{y\sim P_{data}(y)}[\| G(y) - y\|_1] + E_{x\sim P_{data}(x)}[\| F(x) - x\|_1] \quad (2)$$

where $\mathrm{E}_{\mathrm{y}\sim \mathrm{P}_{\text{data}}(\mathrm{y})}[\| \mathrm{G}(\mathrm{y}) - \mathrm{y}\|_1]$: The term measures the difference between the target domain image y and the output generated by G when y is input into the generator. $E_{x\sim P_{data}(x)}[\| F(x) - x\|_1]$: The term calculates the difference between the source domain image x and the output generated by F when x is input into the generator [11].

2. Total Loss Function and Training Process

The overall loss function combines adversarial loss, cycle consistency loss, and identity loss. Total Loss Function:

$$\mathcal{L}_{CycleGAN}(G,F,D_A,D_B) = \mathcal{L}_{adv}(G,D_Y) + \mathcal{L}_{adv}(F,D_X) + \lambda_1\mathcal{L}_{cycle}(G,F) + \lambda_2\mathcal{L}_{idt}(G,F) \quad (3)$$

where λ_1 and λ_2 are hyperparameters that control the relative contributions of the different loss terms. CycleGAN training includes Generator Training and Discriminator Training, which alternate optimization to gradually converge the model. Figure 2 shows the overall training process.

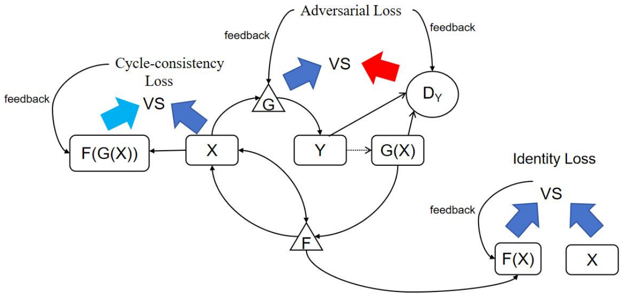


Figure 2 Implementation Process of the Loss Function

CycleGAN training alternates between updating the generator and discriminator. The generator transforms a blurred image x into G(x) and a clear image y into F(y). Adversarial loss ensures G(x) closely resembles y, deceiving discriminator D_Y, while cycle consistency loss ensures F(G(x)) ≈ x and G(F(y)) ≈ y. Identity loss minimizes changes to clear images fed into G and blurred images fed into F. The generator is trained to produce more realistic images. In discriminator training, real images y and generated images G(x) are used to calculate real and fake losses, refining the discriminator's ability to distinguish between them [12].

2.2 Frequency Decomposition Model

Frequency decomposition enhances images by separating them into low-frequency and high-frequency components. Wavelet Transform, a signal processing method, converts an image from the spatial domain to the frequency domain, with the low-frequency part capturing the main structure and the high-frequency part highlighting details like edges and textures [13]. Figure 3 shows its process.

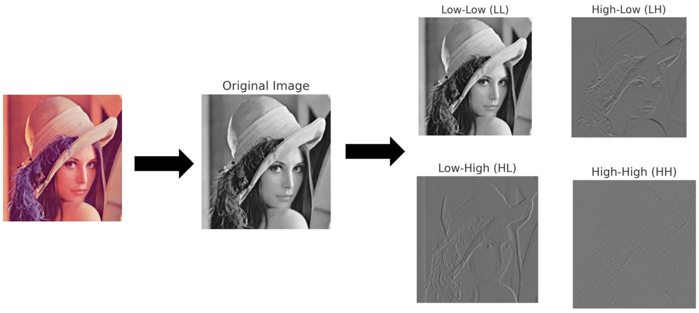


Figure 3 The Process of Frequency Decomposition

The low-frequency part cA and high-frequency parts cH,cV,cD can be expressed by the wavelet transform as:

$$f(x,y) = cA + cH + cV + cD \tag{4}$$

where cA is the low-frequency part, and cH, cV, cD represent the horizontal, vertical, and diagonal detail coefficients, respectively.

The low-frequency component typically represents the general structure of an image. Since underwater images are often blurry due to light attenuation, enhancing the low-frequency component helps improve the overall clarity and brightness. Figure 4 demonstrates the effectiveness of the CLAHE algorithm.

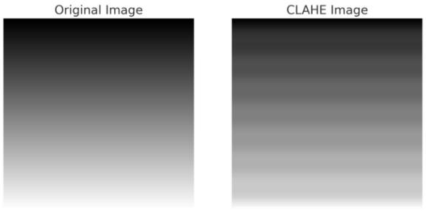


Figure 4 CLAHE Image

In this process, the low-frequency part is primarily enhanced by applying CLAHE to the image's luminance channel (L channel), further boosting the image's contrast. The CLAHE formula is:

$$CLAHE(I) = \frac{(I-\mu)}{\sigma} \cdot clip(I) + \mu \tag{5}$$

where I is the pixel value of the image, μ is the local mean, σ is the local standard deviation, and clip () is the contrast limiting function.

The high-frequency part contains edges and details, which often become unclear in underwater images due to turbidity and light refraction. Sharpening the high-frequency components with filters like high-pass filters enhances image details and edges, improving clarity and realism.

In image synthesis, the low-frequency and high-frequency parts are enhanced separately and then combined with a specific weighting ratio to produce a balanced and realistic final image. The synthesis formula is as follows:

$$f_{enhanced}(x,y) = \alpha \cdot f_{low}(x,y) + \beta \cdot f_{high}(x,y) \tag{6}$$

2.3 Image Quality Indicators

To evaluate image quality, metrics like PSNR, UIQM, and UCIQE are commonly used.

PSNR measures the similarity between the original and enhanced images, with higher values indicating better image quality. It is calculated based on the mean squared error (MSE). The PSNR is defined as:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{(255)^2}{\mathrm{MSE}}\right) \tag{7}$$

UCIQE evaluates image quality by focusing on color aspects such as saturation, contrast, and consistency, which influence perceived quality. It assesses the image using three parameters. The formula for UCIQE is as follows:

$$UCIQE = C_1 \cdot S_a + C_2 \cdot C_a + C_3 \cdot Q_a \tag{8}$$

C_1=0.4680 C_2=0.2745 C_3=0.2576 [14].

UIQM combines various quality metrics, including color contrast, consistency, saturation, and structural similarity index, to assess the perceptual quality of an image by considering both its color and structural elements. The formula for UIQM is as follows:

$$UIQM = C_1 \cdot C_b + C_2 \cdot Q_b + C_3 \cdot S_b \tag{9}$$

C_1 = 0.2953 C_2= 0.2953 C_3 =0.1583 C_4 =0.2471 [15].

3 3 Result and Analyze

In this section, we evaluate the effectiveness of the underwater image enhancement method based on Cycle GAN and frequency decomposition correction.

3.1 Training Part

3.1.1 Dataset Description and preprocessing

In this experiment, this paper uses an underwater image dataset with clear images (target domain Y) and blurry images (source domain X), sourced from publicly available datasets covering environments like coral reefs and marine life. The images are preprocessed by standardizing their size to 256×256 for CycleGAN training. Preprocessing steps include normalizing pixel values to the range [-1, 1] and applying data augmentation, such as random horizontal flipping, to improve model generalization.

3.1.2 Weight setting of the loss function

Cycle Consistency Loss and Identity Loss are used to maintain image consistency and enhance model stability. The weights for each loss are shown in Table 1:

Table 1 The Weights for Each Loss

Loss Type	Parameter	Value	Purpose
Cycle Consistency Loss	λcycle	10	Ensures the generator recovers the original input by transforming between domains.
Identity Loss	λidt	0.5	Maintains image identity, ensuring no drastic changes to target domain images

3.1.3 Training strategy

During training, the generators G and F, along with discriminators D_Y and D_X, are alternately optimized to ensure stable learning. The learning rate begins at 0.0002 for the first 100 epochs and then decays linearly for the following 100 epochs to encourage stable convergence.

3.1.4 Training results

To monitor the training process, this paper uses Visdom for real-time visualization, recording the changes in the model during the training process, in this case, domain A refers to the underwater blurry images, and domain B refers to the underwater clear images. Table 2 is an explanation of the coordinates in Figure 5 which shows the training iteration process:

Table 2 Image Coordinate Interpretation

Name	Color	Description
cycle_A	green curve	The cycle consistency loss for underwater blurry images.
idt_A	red curve	The identity loss for underwater blurry images.

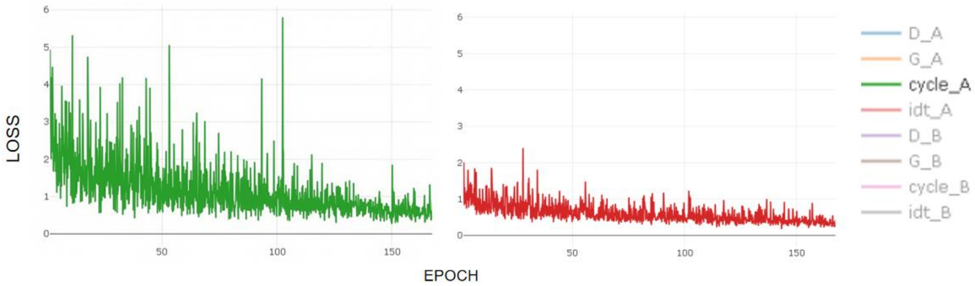


Figure 5 Training Iteration Process

From the graph, it is evident that as training progresses, all losses stabilize, indicating that the generator and discriminator are progressively learning better generation and discrimination capabilities, and the stability of training is improving. Figure 6 shows these changes during training:

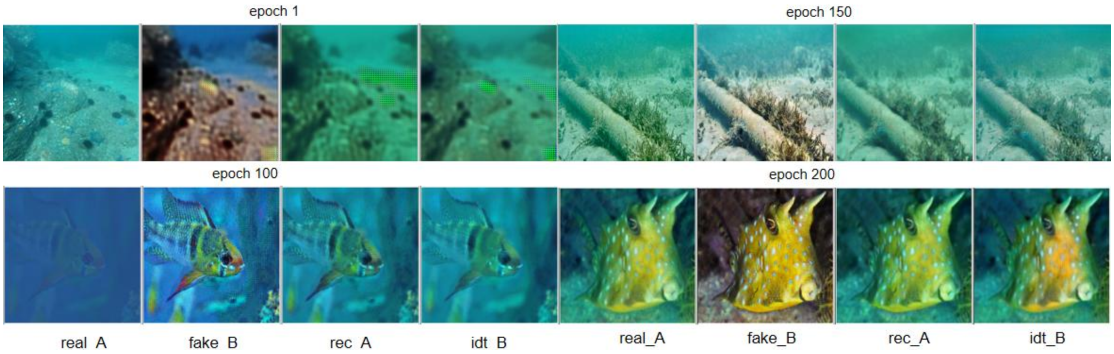


Figure 6 Image Changes During Iteration

At Epoch 1, the generated underwater images are blurry with unrealistic colors. By Epoch 100, the images are sharper with more natural colors and enhanced details. By Epoch 150, the quality improves further with reduced noise and clearer contours. By Epoch 200, the generator consistently produces high-quality, realistic enhanced images. The experiment also shows that the training for both the cycle consistency loss function and the identity function gradually improves.

According to the content of the image, the four images correspond to the following parameters: real_A is the real blurry underwater image, used as input. fake_B is the generated clear image, produced by the generator. rec_A: The underwater blurry image was restored by converting a generated clear image back to a blurry version. idt_B: The identity function image, where a clear image is input into the generator and expected to remain unchanged.

As training progresses, CycleGAN improves its ability to transform blurry images into clearer ones, better matching the real underwater environment. By the end of training, the model weights reach an optimal state, ensuring stable and high-quality image generation.

3.2 Experimental Results

The article focuses on the bias in the convolutional neural network of the weight model, as shown in Figure 7:

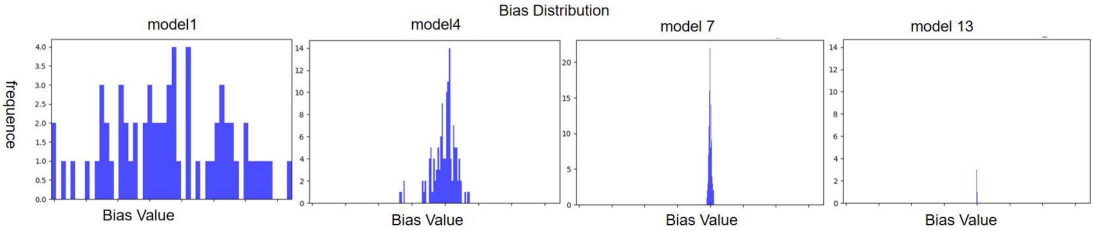


Figure 7 Bias of Convolutional Neural Networks

As the depth of the convolutional network increases, the bias tends to stabilize, which indirectly proves the accuracy of the weight model.

The experiment involves loading the pre-trained generator (G) weights. During the training process, the generator G's weights are continuously optimized by minimizing the loss function of the Generative Adversarial Network (GAN). In this experiment, we loaded the generator weights saved during training and applied them to the underwater image enhancement task. The image processed by CycleGAN is shown in Figure 8:

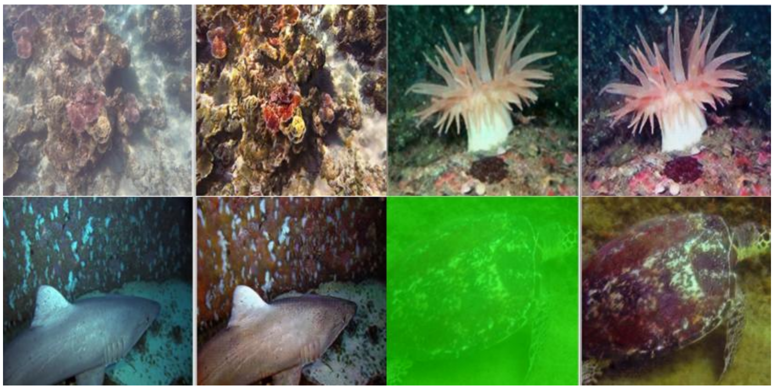


Figure 8 Comparison of underwater enhancement effects

By loading the pre-trained generator weights, we successfully enhanced underwater blurry images, generating clear underwater images.

3.3 Frequency Decomposition Correction

After training the CycleGAN model, this paper enhanced the generated images using wavelet transform for frequency domain decomposition. The image is divided into four parts shown in Figure 9:

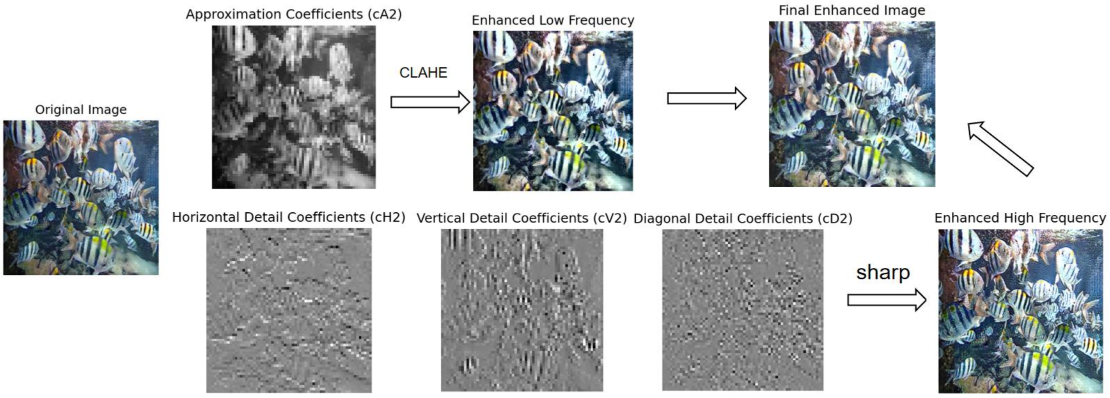

Original Image
Approximation Coefficients (cA2)
CLAHE
Enhanced Low Frequency
Final Enhanced Image
Horizontal Detail Coefficients (cH2)
Vertical Detail Coefficients (cV2)
Diagonal Detail Coefficients (cD2)
sharp
Enhanced High Frequency

Figure 9 Frequency Division Process

The enhanced low and high-frequency components are combined to produce the final image, improving its visual quality.

3.4 Analysis of experimental results

We conducted experiments comparing the image enhancement effects based on the frequency domain method and the frequency domain method after calibration. The experimental results are shown in Figure 10:

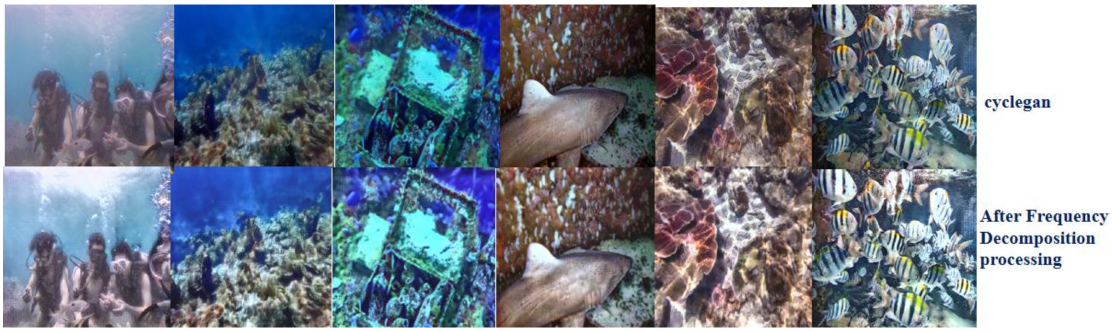


Figure 10 Comparison of Frequency Division Processing Results

The following images show the original underwater image, the image enhanced using CycleGAN, and the image further enhanced using the frequency domain method. We can observe that the CycleGAN-enhanced image is clearer than the original image, while the image enhanced with the frequency domain method excels in sharpness and detail restoration. Figure 11 shows the difference between before and after image enhancement:

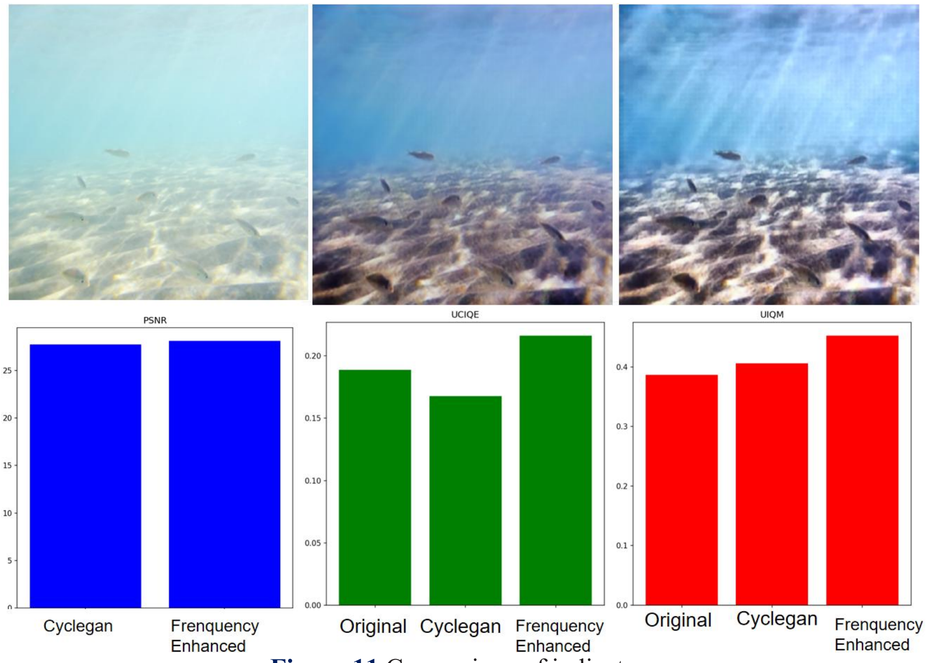


Figure 11 Comparison of indicators

The following table 3 shows the average PSNR, UCIQE, and UIQM scores for the dataset images, including the original images, CycleGAN-enhanced images, and frequency division-enhanced images. The average score of datasets is shown in Table 3:

Table 3 Average score of datasets

image type	PSNR (dB)	UCIQE	UIQM
original image		0.22	0.39
CycleGAN enhanced images	26.43	0.19	0.41
Frequency division method enhancement	28.75	0.24	0.43

PSNR shows that the frequency-enhanced image has a significant quality improvement, with a higher PSNR value, indicating that the enhanced image has less noise and better image quality. UCIQE decreases after CycleGAN processing, but the frequency method restores and further improves the UCIQE value of the image. UIQM shows that after the image is enhanced by CycleGAN, the value increases after correction with the frequency method. In the end, the image has significantly improved in clarity, details, and color.

4 CONCLUSION

The article introduces a new underwater image enhancement method that combines CycleGAN with frequency-domain correction to improve clarity, contrast, and sharpness. Experimental results show that this method outperforms others in detail restoration and contrast enhancement. While CycleGAN alone enhances fine details, it reduces UCIQE values, affecting color consistency. By incorporating frequency-domain decomposition, the approach effectively addresses this issue. CLAHE enhances low-frequency components, improving image clarity, while high-frequency sharpening refines details, resulting in a more visually pleasing outcome. This combined method surpasses traditional pixel-to-pixel models in terms of generalization and robustness across various underwater conditions.

Future research will focus on optimizing wavelet transformations and exploring hybrid models to further enhance the robustness and efficiency of underwater image enhancement techniques.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Lepcha D C, Goyal B, Dogra A, et al. A deep journey into image enhancement: A survey of current and emerging trends. Information Fusion, 2023, 93:36-76.

[2] Moghimi M K, Mohanna F. Real-time underwater image enhancement: a systematic review. Journal of Real-Time Image Processing, 2021, 18(5):1509-1525.

[3] Kavitha S T, Vamsidhar A, Kumar S G, et al. Underwater Image Enhancement using Fusion of CLAHE and Total Generalized Variation. Engineering Letters, 2023, 31(4).

[4] Wang Y, Zhang J, Cao Y, et al. A deep CNN method for underwater image enhancement. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017:1382-1386.

[5] Chen X, Yu J, Kong S, et al. Towards real-time advancement of underwater visual quality with GAN. IEEE Transactions on Industrial Electronics, 2019, 66(12):9350-9359.

[6] Du R, Li W, Chen S, Li C, Zhang Y. Unpaired Underwater Image Enhancement Based on CycleGAN. Information, 2022, 13:1.

[7] Niu Yuzhen, Zhang Lingxin, Lan Jie, et al. Non-paired underwater image enhancement based on frequency division generative adversarial network. Acta Sinica, 2020, 1-18.

[8] Shah R, Sheikh A, Shukla S, et al. Comparing effectiveness of gan and clahe for enhancing underwater images. In: 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2023:1499-1503.

[9] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. Advances in Neural Information Processing Systems, 2014, 27.

[10] Wu J, Liu X, Lu Q, et al. FW-GAN: Underwater image enhancement using generative adversarial network with multi-scale fusion. Signal Processing: Image Communication, 2022, 109:116855.

[11] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. 2017:2223-2232.

[12] Du R, Li W, Chen S, et al. Unpaired underwater image enhancement based on CycleGAN. Information, 2021, 13(1):1.

[13] Saifullah S, Suryotomo A P, Dreżewski R, et al. Optimizing brain tumor segmentation through CNN U-Net with CLAHE-HE image enhancement. In: 2023 1st International Conference on Advanced Informatics and Intelligent Information Systems (ICAI3S 2023). Atlantis Press, 2024:90-101.

[14] Kurinjimalar R, Pradeep J, Harikrishnan M. Underwater Image Enhancement Using Gaussian Pyramid, Laplacian Pyramid and Contrast Limited Adaptive Histogram Equalization. In: 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC). IEEE, 2024:729-734.

[15] Tang X, Wu Y. Single Underwater Image Enhancement Based on Transmission Map Weighted Fusion and Adaptive Color Correction. In: 2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML). IEEE, 2023:25-28.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3047

A FIRE STATION LAYOUT OPTIMIZATION MODEL BASED ON SIMULATED ANNEALING

HaoRan Xiong
International College, Jiangxi University of Finance and Economics, Nanchang 330000, Jiangxi, China.
Corresponding Email: 206109465@qq.com

Abstract: Fires pose a serious threat to life and property. A rational layout of fire stations is crucial for enhancing the fire - fighting emergency response capacity. This paper comprehensively applies the TOPSIS comprehensive evaluation model, the non - integer rank - sum ratio (RSR) model, the simulated annealing algorithm (SA), and the analytic hierarchy process (AHP) to study the optimization of urban fire station layouts. By using the first two models combined with the entropy weight method to evaluate the existing layout, problems such as insufficient coverage and long response times were identified. The SA was used to optimize the layout, which reduced the average response time by 4.7 minutes and balanced the jurisdiction scope. The AHP was used to clarify that factors such as fire risk levels are key factors affecting the layout. However, the simulated annealing algorithm has limitations such as slow convergence. Future research can introduce adaptive simulated annealing algorithms and strengthen the study of dynamic factors for improvement. This research provides a scientific basis and a feasible solution for the optimization of urban fire station layouts.

Keywords: Layout Optimization; TOPSIS; NIRSR model; SA algorithm; AHP

1 INTRODUCTION

Fires have always been a major threat to human life and property. In recent years, the fire situation in China has been severe, with approximately 100,000 fires occurring each year, causing huge losses to the country and the people [1]. The ability of firefighters to quickly reach the scene when a fire breaks out plays a decisive role in controlling the fire and reducing casualties, which largely depends on the rational layout of fire stations. A reasonable fire station layout can not only shorten the fire - fighting response time but also improve the utilization efficiency of fire - fighting resources and minimize fire losses. However, currently, urban fire station layouts generally suffer from problems such as irrational planning and unbalanced coverage, which seriously affect the fire - fighting emergency response capacity. Therefore, optimizing the fire station layout has become an urgent issue to be addressed.

In the field of urban fire station layout optimization, many scholars have conducted in - depth research and achieved a series of results. Wang Yarong et al.[2] used cluster analysis and genetic algorithms to optimize the location of fire stations. By dividing urban areas into different clusters and combining the global search ability of genetic algorithms, they found the best locations for fire stations, effectively improving the coverage efficiency of fire - fighting resources. Yao Yongfeng et al. [3] optimized the fire station layout based on the geographic information system (GIS) and multi - objective planning model, comprehensively considering factors such as population distribution, road networks, and fire risks. This not only improved the response speed but also reduced construction and operation costs. These studies have provided important theoretical and practical experience for urban fire station layout optimization[4-5]. However, existing research mostly focuses on the analysis of single or a few factors when considering the factors affecting the fire station layout, lacking a systematic and comprehensive consideration of multiple factors. At the same time, in the selection of optimization algorithms, some algorithms have problems such as slow convergence and being prone to getting trapped in local optima, which limit the effect of layout optimization[6].

In summary, this paper aims to comprehensively consider multiple factors, use the analytic hierarchy process (AHP) to assign weights to each factor, and construct a more scientific and reasonable fire station layout optimization model. At the same time, the simulated annealing algorithm (SA) is used to optimize the layout to overcome the limitations of traditional algorithms and improve the quality of layout optimization. This paper will evaluate the existing and optimized layouts through the TOPSIS comprehensive evaluation model and the non - integer rank - sum ratio (RSR) model, conduct a comparative analysis of their rationality, and provide more targeted and practical solutions for urban fire station layout optimization, thereby effectively enhancing the urban fire - fighting emergency response capacity and ensuring the safety of people's lives and property [7,8].

2 OPTIMIZATION AND ANALYSIS OF FIRE STATION LAYOUT BASED ON MULTIPLE MODELS

2.1 Construction of the Rationality Evaluation Model for the Existing Layout and Model Solution

In the research on urban fire station layout optimization, the TOPSIS comprehensive evaluation model can integrate multi - index information and evaluate the advantages and disadvantages of objects by comparing them with the ideal solution; the non - integer rank - sum ratio (RSR) model has a wide range of applications and strong anti - interference ability, which can stably and reliably handle various types of data; the simulated annealing algorithm (SA) can jump out of local optimal solutions and effectively solve large - scale combinatorial optimization problems; the analytic hierarchy process (AHP) can systematically analyze multiple factors, combine qualitative and quantitative factors, accurately determine the weights of factors, and provide quantitative basis for decision - making. First, in the TOPSIS - based multi - index evaluation model, the entropy weight method is used to obtain the weights of the number and distance of jurisdiction nodes and the average monthly number of cases, as shown in Table 1.

Table 1 Weights of Each Index Before Optimization Obtained by the Entropy Weight Method

Index	Number and Distance of Jurisdiction Nodes	Average Monthly Number of Cases
Weight	0.41249763	0.58750237

According to the TOPSIS - based multi - index evaluation model, the ranking results of fire stations according to the number and distance of jurisdiction nodes and the average monthly number of cases are shown in Table 2.

Table 2 Ranking under the TOPSIS Model

Fire Station Number	Score of Number and Distance of Jurisdiction Nodes	Rank of Number and Distance of Jurisdiction Nodes	Score of Average Monthly Number of Cases	Rank of Average Monthly Number of Cases
1	0.000	4	0.000	4
2	1.000	1	1.000	1
3	0.489	3	0.240	3
4	0.527	2	0.557	2

According to the RSR evaluation model, the ranking results of fire stations according to the number and distance of jurisdiction nodes and the average monthly number of cases are shown in Table 3.

Table 3 Ranking under the RSR Model

Fire Station Number	Score of Number and Distance of Jurisdiction Nodes	Rank of Number and Distance of Jurisdiction Nodes	Score of Average Monthly Number of Cases	Rank of Average Monthly Number of Cases
1	0.250	4	0.250	4
2	1.000	1	1.000	1
3	0.725	2	0.500	3
4	0.525	3	0.750	2

Combining the above two tables, it can be found that the scores and rankings of fire stations based on the number and distance of jurisdiction nodes are different under the two models, further indicating the irrationality of the fire station layout in City A.

Therefore, the simulated annealing algorithm (SA) model is used to optimize the fire station layout in City A.

First, the entropy values and weights of each index are obtained by the entropy weight method, as shown in Table 4.

Table 4 Entropy Values and Weights of Each Index in the TOPSIS Multi - Index Evaluation Model

Evaluation Index	Entropy Value	Weight
Total Distance	-1908782000000	0.528883
Average Monthly Number of Cases	-1700302000000	0.471117

After optimizing with the simulated annealing algorithm (SA) model, the jurisdiction scope of each fire station under the new layout is shown in Table 5, and the jurisdiction scope diagram is drawn as shown in Figure 1.

Table 5 Jurisdiction Scope of Each Fire Station in City A after Optimization

Node Number	Fire Station Number	Required Time/min	Distance/hm	Node Number	Fire Station Number	Required Time/min	Distance/hm
1	22	3.27	32.70	47	30	1.73	17.27
2	30	0.58	5.83	48	30	0.71	7.07
3	67	2.26	22.64	49	30	3.68	36.83
...	...	...	...	...	...	...	...
44	67	1.48	14.76	90	76	5.09	50.85

45	30	4.59	45.87	91	76	5.56	55.60
46	30	3.99	39.91	92	76	7.56	75.62

Note: Time and distance are uniformly rounded to two decimal places

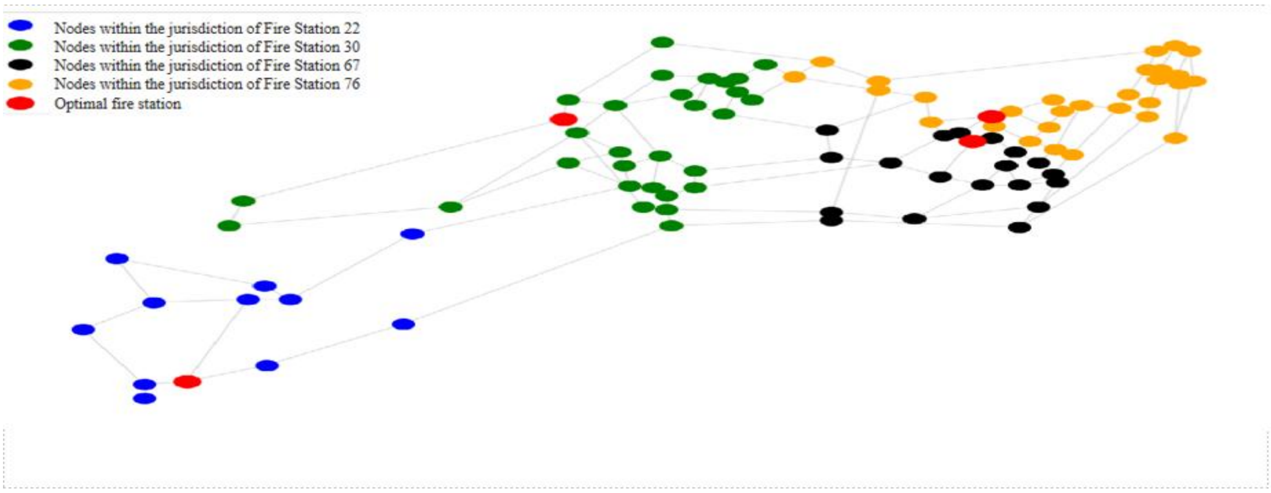

Nodes within the jurisdiction of Fire Station 22
Nodes within the jurisdiction of Fire Station 30
Nodes within the jurisdiction of Fire Station 67
Nodes within the jurisdiction of Fire Station 76
Optimal fire station

Figure 1 Layout and Jurisdiction Scope of Each Fire Station after Optimization

Analyzing Table 5 shows that after the fire station layout is optimized, firefighters can reach all nodes except Node 28 within 8 minutes after the fire occurs, which meets the time requirements of the fire station layout. At the same time, the number of nodes, distance, time, and case - occurrence frequency within the jurisdiction scope of each fire station after optimization are summarized as shown in Table 6. The number of jurisdiction nodes of Fire Stations 22, 30, 67, and 76 is (12, 30, 20, 30) respectively, and the jurisdiction scope distribution and fire - fighting resource allocation are more balanced than before optimization.

Table 6 Jurisdiction Scope, Distance, Time, and Case - Occurrence Frequency of Each Fire Station after Optimization

Fire Station Number	Number of Jurisdiction Nodes	Total Distance/hm	Total Time/min	Average Monthly Total Number of Cases
22	12	445.0862213	44.50862213	21.2
30	30	1114.030147	111.4030147	43.6
67	20	543.7649218	54.37649218	25.7
76	30	1050.610701	105.0610701	34

According to the TOPSIS -based multi-index evaluation model, the ranking results of fire stations after optimization according to the number and distance of jurisdiction nodes and the average monthly number of cases are shown in Table 7.

Table 7 Ranking of the Optimized Layout under the TOPSIS Model

Fire Station Number	Score of Number and Distance of Jurisdiction Nodes	Rank of Number and Distance of Jurisdiction Nodes	Score of Average Monthly Number of Cases	Rank of Average Monthly Number of Cases
22	0.000000	4	0.000000	4
30	1.000000	1	1.000000	1
67	0.270593	3	0.200893	3
76	0.923881	2	0.571429	2

According to the RSR evaluation model, the ranking results of fire stations after optimization according to the number and distance of jurisdiction nodes and the average monthly number of cases are shown in Table 8.

Table 8 Ranking of the Optimized Layout under the RSR Model

Fire Station Number	Score of Number and Distance of Jurisdiction Nodes	Rank of Number and Distance of Jurisdiction Nodes	Score of Average Monthly Number of Cases	Rank of Average Monthly Number of Cases
22	0.250	4	0.250	4
30	0.950	1	1.000	1

67	0.500	3	0.500	3
76	0.800	2	0.750	2

Combining the above two tables, it can be found that the rankings of fire stations based on the number and distance of jurisdiction nodes and the average monthly number of cases are the same under the two models, indicating that the optimized layout is more reasonable than the original layout.

2.2 Analysis of the Influencing Factors of Fire Station Layout Based on the Analytic Hierarchy Process

2.2.1 Construction of the evaluation index system

For the location selection of fire stations, each location selection plan of fire stations is regarded as a decision - making unit. Through investigations of fire protection planning and fire station construction, based on the "Regulations on the Construction and Management of Urban Fire Protection Planning" and the "Construction Standards for Urban Fire Stations", and combined with the influencing factors of fire station location selection, a fire station location selection evaluation index system is established (as shown in Table 9). It should be noted that the system is divided into two categories of qualitative and quantitative indicators as secondary indicators. The tertiary indicators are 5 indicators including the aforementioned hot - spot factors, which reflect the advantages and disadvantages of fire station location selection from aspects such as hot - spot factors, society, geographical environment, rescue time, and fire - fighting economic construction. The fire risk level within the jurisdiction can reflect the importance of the fire station. The higher the risk, the more important the fire station is, the more meaningful the location selection is, and the higher the score will be. The accident hot - spot and time situation can reflect the frequency and time of fire accidents within the jurisdiction of the fire station. This indicator is related to road factors and the impact on residents, but these two indicators do not overlap. The overall goal of constructing this index system is to determine the main factors affecting the fire station layout to ensure that the fire station layout can effectively respond to fires in the shortest possible time.

Table 9 Fire Station Location Selection Evaluation Index System

Secondary Indicators	Tertiary Indicators	Quaternary Indicators
A1 Qualitative Indicators	B1 Hot - Spot Factors	C1 Fire Risk Level within the Jurisdiction C2 Accident Hot - Spot Time Situation C3 Traffic Conditions
	B2 Social Factors	C4 Urban Population Density C5 Fire - Fighting Facility Construction Situation
	B3 Geographical Factors	C6 Terrain Conditions C7 Meteorological Conditions
A2 Quantitative Indicators	B4 Time Factors	C8 Maximum Time for Fire Trucks to Reach the Accident Site/min C9 Minimum Time for Fire Trucks to Reach the Accident Site/min
	B5 Economic Factors	C10 Fixed Cost of Fire - Fighting Facility Construction/10,000 yuan C11 Annual Total Operating Cost of the Fire Station/10,000 yuan

2.3 Construction of the Analytic Hierarchy Process (AHP) Model

2.3.1 Establishing the hierarchical structure model of the system

First, the decision - making problem is decomposed into different hierarchical structures, usually divided into the target layer, criterion layer, and alternative layer. The target layer is the ultimate goal of decision - making, the criterion layer contains the criteria affecting the decision - making, and the alternative layer is the specific decision - making options. There is a certain relationship between the elements of each layer and the elements of adjacent layers. Upper - layer elements are usually the synthesis or influencing factors of lower - layer elements, and lower - layer elements are the specific implementation or refinement of upper - layer elements. For example, in decision - making analysis, the upper - layer goal is ultimately decomposed into specific alternatives through the criteria and sub - criteria of the middle layer. Through the understanding and research of fire station construction location selection, the hierarchical structure established is shown in Figure 2.

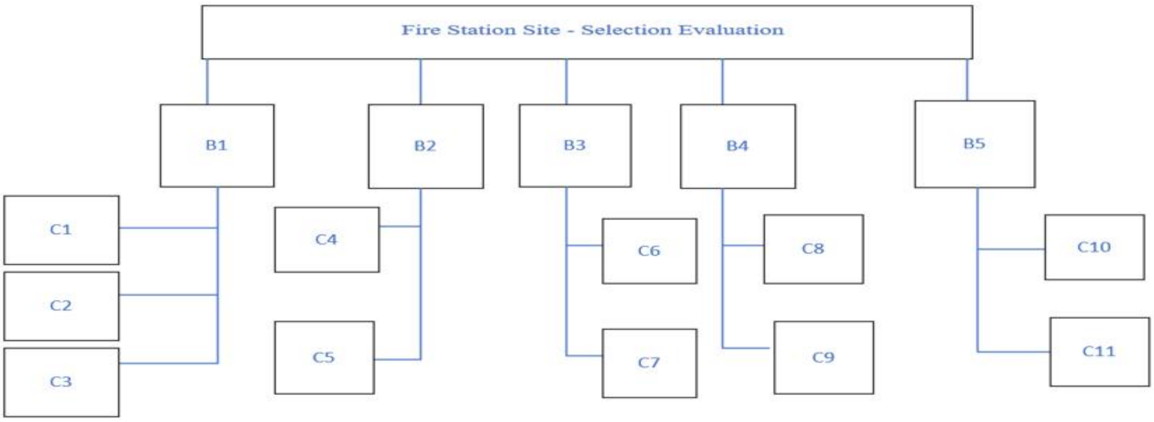


Figure 2 Hierarchical Structure

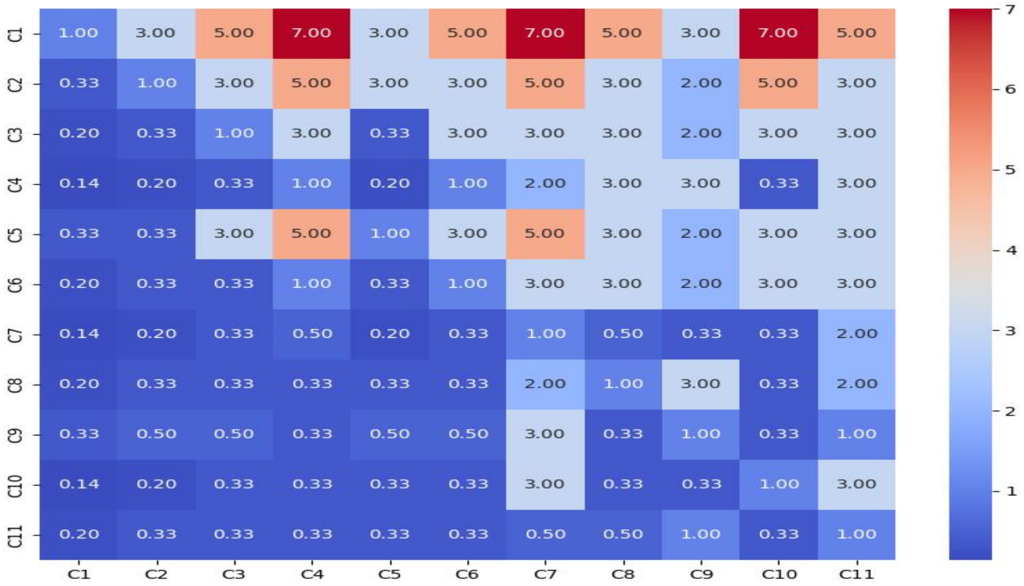
The scaling method is shown in Table 10:

Table 10 Scaling Method

Scaling Value	Importance Degree of One Element to Another
1	Equally important
3	Slightly important
5	Obviously important
7	Strongly important
9	Strongly important
2,4,6,8	Mid - values of the above - mentioned adjacent judgments

The elements in the judgment matrix represent the relative importance of factors with respect to the factors in the upper - level.

A judgment matrix system is established. According to the 11 evaluation indicators for fire station site - selection, the judgment matrix is constructed as shown in Figure 3 below:



	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
C1	1.00	3.00	5.00	7.00	3.00	5.00	7.00	5.00	3.00	7.00	5.00
C2	0.33	1.00	3.00	5.00	3.00	3.00	5.00	3.00	2.00	5.00	3.00
C3	0.20	0.33	1.00	3.00	0.33	3.00	3.00	3.00	2.00	3.00	3.00
C4	0.14	0.20	0.33	1.00	0.20	1.00	2.00	3.00	3.00	0.33	3.00
C5	0.33	0.33	3.00	5.00	1.00	3.00	5.00	3.00	2.00	3.00	3.00
C6	0.20	0.33	0.33	1.00	0.33	1.00	3.00	3.00	2.00	3.00	3.00
C7	0.14	0.20	0.33	0.50	0.20	0.33	1.00	0.50	0.33	0.33	2.00
C8	0.20	0.33	0.33	0.33	0.33	0.33	2.00	1.00	3.00	0.33	2.00
C9	0.33	0.50	0.50	0.33	0.50	0.50	3.00	0.33	1.00	0.33	1.00
C10	0.14	0.20	0.33	0.33	0.33	0.33	3.00	0.33	0.33	1.00	3.00
C11	0.20	0.33	0.33	0.33	0.33	0.33	0.50	0.50	1.00	0.33	1.00

Figure 3 Heat Map of the Judgment Matrix

1. Calculation of Index Weights: Calculate the maximum eigenvalue $\lambda_{\max}$ of each judgment matrix U and obtain the normalized eigenvector w. That is: $Uw = \lambda_{\max} w$

Here, the specific formula for calculating the eigenvector is not clearly given in the original text, so it is not translated in detail. The components of it are the weights we need $\left(w_1, w_2, \ldots, w_n\right)$. Under normal circumstances, it is rather troublesome to obtain the eigenvector, so the root - method is used for approximate calculation:

$$w_i = \left[\prod_{j=1}^{n} a_{ij} \right]^{\frac{1}{n}}, n = 1, 2, ..., n \tag{1}$$

The weight is the result of normalization processing, that is:

$$w_i^* = \frac{w_i}{\sum_{i=1}^{n} w_i}, s.t. \sum_{i=1}^{n} w_i^* = 1, w_i^* \geq 0, i = 1, 2, ..., n \tag{2}$$

2. Consistency Test of the Judgment Matrix:

Make a questionnaire and invite experts to score and judge. After calculating the weights according to the formula, it is also necessary to conduct a consistency test on the pairwise - comparison judgment matrix to reduce the inaccuracy and arbitrariness of subjective judgment. The consistency test adopts the following formula:

$$CR = \frac{CI}{RI} \tag{3}$$

In the formula:

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{4}$$

CI is the consistency index.

The values of the consistency index RI for different orders are shown in the following table 11:

Table 11 Values of the Consistency Index RI

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.52	0.90	1.12	1.24	1.32	1.41	1.45	1.49

If $CR < 0.1$, the consistency of the judgment matrix meets the requirements, that is, the judgment result is reliable; otherwise, the judgment matrix needs to be reconstructed.

The result of the model solution CR is 0.0626, which meets the consistency requirement, indicating that the judgment result in Table 11 is reliable. As can be seen from Table 12 below, according to the data, the fire risk level within the jurisdiction (C1) has the highest weight, indicating that it is a key factor in evaluating fire safety. The accident hot - spot situation (C2) and the fire - fighting facility situation (C5) are also very important, but to a lesser extent. The weights show that operational and response capabilities such as traffic conditions (C3), terrain (C6), and population density (C4) play an important role, but they are not as crucial as direct risk factors. The fastest and slowest arrival times of fire trucks at the accident site (C8 and C9), as well as the fire - fighting facilities and their operating costs (C10 and C11) are relatively less important but still relevant. The impact of weather conditions (C7) is the smallest. This may be because although weather affects fire risks and responses, compared with other factors in the model, it is uncontrollable and has a lower impact level. This fire station site - selection evaluation system emphasizes the multi - aspect nature of fire safety assessment, giving priority to direct risk factors while also considering operational conditions and costs. This comprehensive approach ensures that the developed fire safety strategies are both robust and take into account various key factors that may affect the effectiveness of fire prevention and emergency response measures.

Table 12 Corresponding Weights of Each Index

Index Meaning	Corresponding Weight	Weight Ranking
Fire Risk Level within the Jurisdiction (C1)	0.28	1
Accident Hot - Spot Time Situation (C2)	0.17	2
Fire - Fighting Facility Construction Situation (C5)	0.14	3
Traffic Conditions (C3)	0.10	4
Terrain Conditions (C6)	0.07	5
Urban Population Density (C4)	0.06	6
Maximum Time for Fire Trucks to Reach the Accident Site per Minute (C8)	0.05	7
Minimum Time for Fire Trucks to Reach the Accident Site per Minute (C9)	0.04	8
Fixed Cost of Fire - Fighting Facility Construction (in 10,000 yuan) (C10)	0.04	9
Annual Total Operating Cost of the Fire Station (in 10,000 yuan) (C11)	0.03	10
Meteorological Conditions (C7)	0.03	11

2.3 Sensitivity Analysis of the Model

In the problem of finding the optimal layout of fire stations, the initial average speed is set at 60 km/h and assumed to be constant. At the same time, it is set that all fire stations should reach the nodes within their jurisdiction as quickly as possible within 8 minutes. The goal of sensitivity analysis is to evaluate the impact of changes in different variables on the layout and jurisdiction scope of fire stations to ensure the robustness of the layout plan and its effectiveness in practical applications. This paper selects the average speed variable for sensitivity analysis. Keeping other relevant variables unchanged, the average speed is allowed to fluctuate by 5% up and down. The simulated annealing algorithm is used to calculate the optimal layout, and the Dijkstra algorithm is used to calculate the jurisdiction scope of each fire station. The Pycharm software is used to draw the graph of the number of nodes within the jurisdiction scope of each fire station changing with the speed under different speeds.

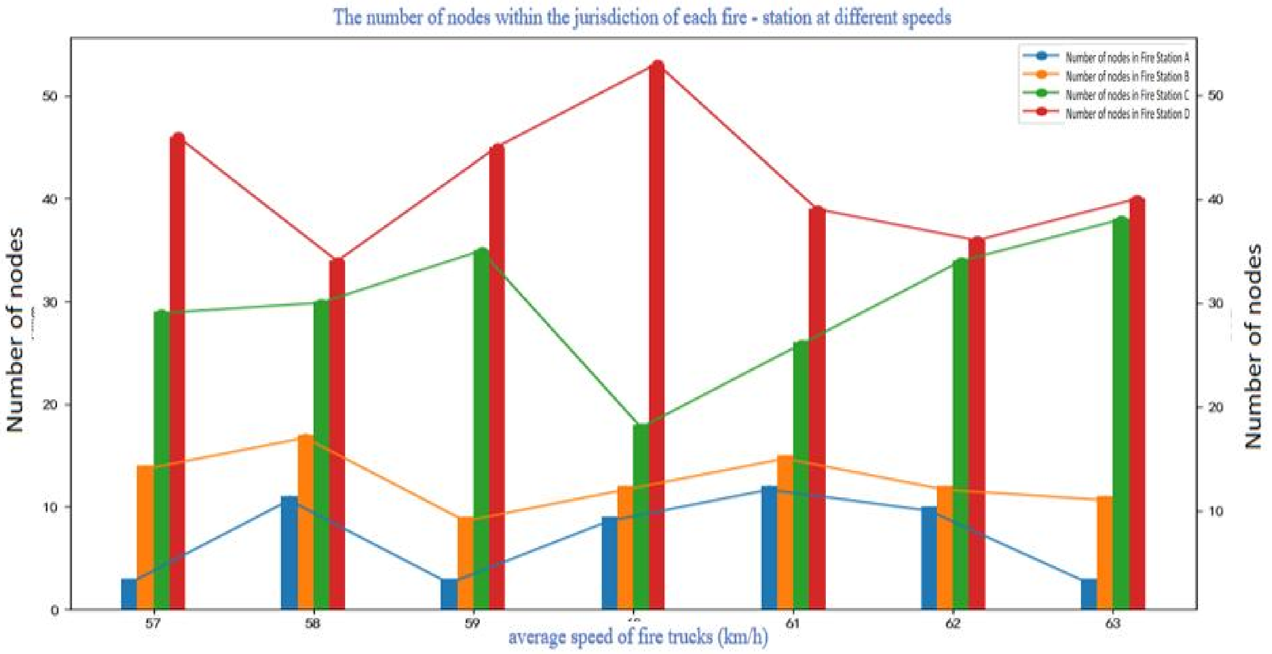


Figure 4 Number of Nodes within the Jurisdiction Scope of Each Fire Station under Different Speeds

As can be seen from Figure 4, assuming that the speed of fire trucks is constant at different speeds within the fluctuating range, the number of nodes within the jurisdiction scope of each fire station changes greatly, indicating that the layout and jurisdiction scope division of fire stations are relatively sensitive. This should be paid special attention to in the practical problem of fire station layout and jurisdiction scope division. Through sensitivity analysis, we can identify the key factors affecting the fire station layout and adjust the layout strategy according to the analysis results to ensure that fire stations can effectively respond to fire accidents in the shortest possible time under various possible circumstances.

3 CONCLUSION

This research focuses on the optimization of urban fire station layouts and conducts in - depth exploration by comprehensively applying multiple models and algorithms. Through the TOPSIS comprehensive evaluation model, the non - integer rank - sum ratio (RSR) model, and the entropy weight method, the defects of the existing fire station layout in City A in terms of node coverage and emergency response time are accurately identified. The layout is optimized by using the simulated annealing algorithm (SA), which significantly improves the emergency response speed and balances the jurisdiction scope of fire stations, verifying the effectiveness of this algorithm. The analytic hierarchy process (AHP) is used to clarify the key factors affecting the layout, providing theoretical support for optimization. The research results have high feasibility in practical applications. The optimized layout has obvious advantages in emergency response, node coverage, and resource allocation, effectively guaranteeing urban fire safety.

However, the simulated annealing algorithm has a slow convergence speed and high computational cost when dealing with large - scale problems. This problem will become more prominent with the growth of urban scale and data volume. Moreover, the model does not fully consider dynamic factors such as real - time traffic flow changes and dynamic adjustments of urban functional areas. Future research can introduce optimized algorithms such as the adaptive simulated annealing algorithm (ASA) to improve algorithm performance; strengthen the research on dynamic factors and integrate real - time data to construct dynamic models; expand the research scope to consider the coordinated operations of fire stations and the dynamic allocation of fire - fighting resources, and improve the urban fire safety guarantee system.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Wang Xianjie. Research on the Optimization of Urban Fire Station Location Selection Based on Hot - Spot Factor Analysis. South China University of Technology, 2014.

[2] Wang Yarong, Zhao Haipeng, Zhou Weitao, et al. Optimization Practice of Urban Fire Station Layout Integrating the Influence of Multiple Factors. Fire Science and Technology, 2024, 43(04): 551-556.

[3] Yao Yongfeng, Jia Hongchen, Zhang Yan, et al. Research on the Optimal Layout of Fire - Fighting Resources in Urban Areas. Fire Science and Technology, 2023, 41(02): 258-262.

[4] Jiang Yuncheng, Sun Lijian, Xiao Kun, et al. Research on the Location and Layout Optimization of Urban Fire Stations. Science of Surveying and Mapping, 2021, 46(09): 207-217.

[5] Guo Jingwen, Zhao Pengpeng, Ni Jiacheng. Fire Station Location Planning Model Based on Genetic Algorithm. Journal of Computer Applications, 2020, 40(S1): 41-44.

[6] Cao Yue, Li Jing, Gu Yu, et al. Optimization Method for Fire Station Location Based on the Lagrangian Relaxation Framework. Fire Science and Technology, 2023, 42(08): 1141-1146.

[7] Chaudhary P, Chhetri S K, Joshi K M, et al. Application of an analytic hierarchy process (AHP) in the GIS interface for suitable fire site selection: A case study from Kathmandu Metropolitan City, Nepal. Socio - Economic Planning Sciences,2016, 53, 60-71.

[8] Yu H, Feng X, Zhao, X. Application of the rank - sum ratio method in the comprehensive evaluation of the quality of life in different regions. Chinese Journal of Health Statistics, 2018, 35 (1): 10 - 13.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3048

PRODUCTION SCHEDULING OPTIMIZATION MODEL BASED ON DYNAMIC PROGRAMMING AND GENETIC ALGORITHM

XiangLong Huang*, ZhengTing Li, LiKe Zhong
School of Computer Science and Engineering, Guangdong Ocean University, Yangjiang 529500, Guangdong, China.
Corresponding Author: XiangLong Huang, Email: 13500020575@stu.gdou.edu.cn

Abstract: In view of the problem that enterprises can not improve their profits due to the difficulty in planning reasonable decision-making schemes in real life, this paper proposes a dynamic programming model of production decision-making based on genetic algorithm. Firstly, a joint target benefit model considering the detection cost of parts, the purchase price of parts, semi-finished products, the assembly cost of finished products, the loss of replacement, semi-finished products, the disassembly cost of finished products, the disassembly of semi-finished products into parts, the disassembly compensation of finished products into semi-finished products, and the market price is established. Then, the genetic algorithm is used to optimize the variables of whether to detect parts, whether to detect semi-finished products, and whether to disassemble finished products (0-1). Finally, taking an actual production plan as an example, the validity of the model is verified, and the decision-making plan when the profit is the largest is obtained. This paper proposes a genetic algorithm-based dynamic programming model for production decision-making, which comprehensively considers various cost and price factors to optimize decision variables, thereby improving production efficiency and profit margins, and is verified to be valid through an actual production plan.

Keywords: Production decision; Dynamic programming; Genetic algorithm

1 INTRODUCTION

The quality of production decision-making will directly affect the sales of products[1],and indirectly affect the economic benefits of enterprises. Only through scientific and reasonable decision-making, can enterprises maximize the efficiency of resources and improve their competitiveness and profitability. Therefore, enterprise decision makers need to have a huge information reserve, strong analysis and judgment ability, as well as keen market insight and extraordinary decision-making thinking, in order to make wise decisions and create more value for the enterprise. In order to better meet the needs of enterprises and achieve good economic and commercial benefits, it is necessary to solve the decision-making problems in the production process according to the actual situation[2].Through the establishment of some appropriate mathematical models, a more systematic and scientific method can be used to analyze and study the production products, formulate flexible and reasonable product processing strategies, and plan reasonable decision-making plans based on the analysis results, improve product sales, and bring higher profits to the enterprise[3]. In this paper, the dynamic programming model is optimized by genetic algorithm for the decision-making problem of enterprise production of electronic products. Then, the dynamic programming model is established and solved based on the lowest production cost and the highest economic benefit. Finally, the optimal enterprise production decision-making scheme is given[4]. In order to solve the problem that enterprises can not improve their income due to the difficulty of planning reasonable decision-making schemes in real life.

2 DYNAMIC PROGRAMMING MODEL OF PRODUCTION DECISION BASED ON MULTI-OBJECTIVE

2.1 Joint Benefit Objective Function

2.1.1 Definition of decision variables

Due to the increase in the number of spare parts, you can set 0-1 decision variables[5]x_i,i can be 1, 2,... In this paper, we consider the case of eight spare parts.$w_i$represents whether to detect semi-finished products,i= 1,2,3 represent three semi-finished products respectively,w_i = 0 means not to detect semi-finished products, and the detection of finished products still use y as the decision variable,z_i= 0 indicates no disassembly,i= 1,2,3 indicates whether the semi-finished products are disassembled, and i = 4 indicates whether the finished products are disassembled

$$x_i = \begin{cases} 0 & \text{Do not detect spare parts} \\ 1 & \text{detect spare parts} \end{cases} \quad (i = 1,2,3,\cdots,8)$$

$$w_i = \begin{cases} 0 & \text{Do not detect semi} - \text{finished products} \\ 1 & \text{detect semi} - \text{finished products} \end{cases} \quad (i = 1,2,3)$$

$$z_i = \begin{cases} 0 & \text{Not disassembled} \\ 1 & \text{disassembled} \end{cases} \quad (i = 1,2,3,4)$$

2.1.2 Establish the cost of testing

C_1 and C_2 are defined as the detection cost of spare parts 1 and 2, and C_i (i= 1,2,3,....,n) For the detection cost of the number of spare parts, the case of eight spare parts is also considered here.S_i (i = 1, 2, 3,...,n) is the detection cost of semi-finished products, C_f is the detection cost of finished products, and the detection cost is obtained as shown in Equation (1).

$$C = \sum_{i=1}^{8} x_i C_i + \sum_{j=1}^{3} w_j S_j + yC_f \tag{1}$$

Among them, C is the total inspection cost, which is composed of the inspection cost of spare parts, semi-finished products and finished products.

2.1.3 Determine the unit price cost

The purchase unit price is 2 yuan, 8 yuan, 12 yuan. Considering the unit price cost of the purchase, the purchase unit price of the 8 parts is summed to determine the unit price cost :

$$D = \sum_{i=1}^{8} m_i \tag{2}$$

Where m_i (i = 1, 2, 3....., 8) respectively represent the purchase unit price of parts 1-8, and D is the unit price cost.

2.1.4 Calculating assembly costs

The cost of assembly of semi-finished products and finished products can be expressed as Ca_i, Ca_1, Ca_2 and Ca_3 represent the assembly cost of semi-finished products, Ca_4 represents the assembly cost of finished products, and the assembly cost can be listed as Equation (3).

$$C_a = \sum_{i=1}^{4} C_{ai} \tag{3}$$

Where Ca is the total assembly cost, which is composed of the assembly cost of semi-finished products and the assembly cost of finished products.

2.1.5 Calculation of switching costs

The replacement loss needs to take into account the defective rate of the finished product, and the defective rate of the finished product is affected by the defective rate of the semi-finished product, and the defective rate of the semi-finished product is affected by the defective rate of the parts. The defective rate adjustment function is introduced to solve the new semi-finished product defective rate and finished product defective rate [6].

The adjustment function of the defective rate of spare parts is introduced :

$$Q_j = \sum_{i=1}^{8} (1 + P_i)^{(1-x_i)} \cdot Q_{sj} \qquad (j = 1,2,3) \tag{4}$$

where P_i (i = 1, 2,......,8)The defective rate of 8 spare parts is represented respectively. Q_{s1}, Q_{s2} and Q_{s3} represent the defect rate of the original semi-finished products respectively, and Q_j represents the defect rate of the new semi-finished products.

The adjustment function of the defective rate of spare parts is introduced :

$$P_f = \sum_{j=1}^{3} (1 + Q_j)^{(1-w_j)} \cdot P_s \tag{5}$$

P_s represents the defective rate of the original finished product, and P_f represents the defective rate of the new finished product.

$$L = (1-y) \cdot C_L \cdot P_f \tag{6}$$

Using the new finished product defect rate, combined with the exchange loss C_L, the total cost of the exchange can be obtained.

2.1.6 Calculation of disassembly cost and disassembly compensation

C_{di} is defined as the disassembly cost. C_{d1}, C_{d2} and C_{d3} represent the disassembly cost of semi-finished products. C_{d4} represents the disassembly cost of finished products. The disassembly cost can be expressed as the sum of the costs of semi-finished products or finished products to be disassembled, which can be expressed as Formula (7).

$$C_x = \sum_{i=1}^{4} z_i \cdot C_{di} \tag{7}$$

The dismantling of some unqualified products (finished or semi-finished products) may reduce the qualified probability of unqualified products. Here, the qualified rate is still defined as a decrease of 50 %. m_9 (22 yuan), m_{10} (22 yuan) and m_{11} (20 yuan) respectively represent the unit price of semi-finished products, which is obtained by summing the unit price of the assembled spare parts.

$$B_1 = \sum_{i=1}^{3}\sum_{j=1}^{3} z_j \cdot [(0.5 - P_{3j+i-3}) \cdot m_{3j+i-3}] \quad (3\mathrm{j} + \mathrm{i} - 3 < 9) \tag{8}$$

B_1 represents the disassembly compensation of semi-finished products into spare parts, and 3j + i-3 represents the number of spare parts respectively, which needs to be less than 9, because there are only 8 maximum spare parts.0.5-P_{3j+i-3} represents the qualified rate of spare parts, and the product with the unit price m_{3j+i-3} represents the return cost of qualified products , and the product with the decision variable z represents the disassembly compensation[7].

$$B_2 = \sum_{j=1}^{3} z_4 \cdot [(0.5 - Q_j) \cdot m_{j+8}] \tag{9}$$

B_2 represents the disassembly compensation of finished products into semi-finished products, and m_{j+8} represents the unit price of semi-finished products m_9, m_{10} and m_{11} respectively. The 0.5-Q_j represents the qualified rate of the semi-finished product, and the unit price m_{j+8} represents the return cost of the qualified product, and it is integrated with the decision variable z to represent the disassembly compensation.

2.2 Establishment of Objective Function

Considering the profit and cost, combined with the market price M = 200 yuan, the objective function for solving the optimal value can be listed[8], that is, $\mathrm{Max}\omega = \mathrm{M} - \mathrm{C} - \mathrm{D} - \mathrm{Cx} - \mathrm{Ca},$, $\mathrm{C} + \mathrm{D} + \mathrm{Cx} + \mathrm{Ca}$ represents the total cost, so that it is as small as possible. The objective function can also be converted into the maximum value of ω, and the constructed objective function is shown in Equation (10).

$$Max\ \omega = M - D - C - C_a - C_x + (B_1 + B_2) - L$$

$$st.\begin{cases} x_i, w_i, z_i, y = \begin{cases} 0 \\ 1 \end{cases} \quad (i = 1,2,3...) \\ Q_j = \sum_{i=1}^{8} (1+P_i)^{(1-x_i)} \cdot Q_{sj} \qquad (j = 1,2,3) \\ P_f = \sum_{j=1}^{3} (1+Q_j)^{(1-w_j)} \cdot P_s \\ D = \sum_{i=1}^{8} m_i \\ C = \sum_{i=1}^{8} x_i C_i + \sum_{j=1}^{3} w_j S_j + yC_f \\ C_a = \sum_{i=1}^{4} C_{ai} \\ C_x = \sum_{i=1}^{4} z_i \cdot C_{di} \\ B_1 = \sum_{i=1}^{3} \sum_{j=1}^{3} z_j \cdot [(0.5 - P_{3j+i-3}) \cdot m_{3j+i-3}] \quad (3\mathrm{j}+\mathrm{i}-3<9) \\ B_2 = \sum_{j=1}^{3} \mathrm{z}_4 \cdot [(0.5 - Q_j) \cdot m_{j+8}] \\ L = (1-y) \cdot C_L \cdot P_f \end{cases} \tag{10}$$

ω is the income of the enterprise, which is required to make the income as large as possible.

3 ANALYSIS PROCESS OF DYNAMIC PROGRAMMING MODEL BASED ON GENETIC ALGORITHM

3.1 The Working Principle of Genetic Algorithm

The basic principle of genetic algorithm is derived from the basic concepts of Darwin 's natural selection and genetics[9]. In the genetic algorithm, each instance of the solution is regarded as an ' individual ', and the entire solution space forms a ' population '. Each individual is represented by a string of ' genes ' that encode the specific parameters of the solution[10]. The genetic algorithm continuously improves the quality of the population and approaches the optimal solution through the iterative process. The core steps include selection, crossover and mutation.

Consider a simplified genetic algorithm model, whose fitness function $f(x)$ is used to evaluate the performance of each individual, where x is a vector that encodes individual features. The objective of the algorithm is to maximize the fitness function[11]. An iteration of genetic algorithm can be expressed as the following steps :

3.1.1 Selection

The probability that an individual is selected for reproduction is proportional to its fitness. If we set the probability that p_i the first i-individual is selected.

$$p_i = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_\mathrm{j})} \tag{11}$$

In formula (11), $f(x_i)$ is the fitness of the first i individual, and N is the total number of individuals in the population.

3.1.2 Crossing

The selected individuals generate new offspring through crossover operation. If the crossing point is k and two individuals are considered x_i and x_j, the descendant x_{new} can be expressed as :

$$x_{new} = \left(x_{i1}, x_{i2}, ..., x_{ik}, x_{j(k+1)}, ..., x_{jn}\right) \tag{12}$$

3.1.3 Mutation

Some genes of the newborn individuals are modified with a small probability μ to introduce variation and increase the diversity of the population. For gene x_{nk}, the mutation operation can be expressed as

$$x_{nk}^{'} = x_{nk} + \delta, \quad \text{with probability}\mu \tag{13}$$

In formula (13), δ is a small random variable and μ is the mutation rate.

By repeating these steps, the genetic algorithm can gradually improve the quality of the solution after multiple generations of iterations, close to the optimal solution[12]. The fitness function of each generation is usually improved, indicating the effectiveness of the algorithm in solving specific problems.

3.2 Dynamic Programming Model Solving Process

The solving process of the dynamic programming model is shown in Figure 1. The specific steps are as follows :

Step 1 : Construct a multi-objective normalization function. By setting the parameters of the objective function, the total profit and the defective rate penalty are combined into a multi-objective function. Normalized by the following formula : fitness value = total profit-defect rate penalty coefficient × defect rate

Step 2 : Define the optimization variables and algorithm parameters. A total of 16 binary decision variables including parts detection strategy (8 variables), semi-finished product detection strategy (3 variables), finished product detection strategy (1 variable), disassembly strategy (3 variables) and finished product disassembly strategy (1 variable) are taken as the optimization objectives. Set genetic algorithm parameters : population size, number of iterations, crossover probability, mutation probability. Variable upper and lower limits, etc.

Step 3 : Constraint condition verification and strategy generation. 1.Substandard product rate constraint : If the spare parts are not detected, the semi-finished product rate is cumulatively punished according to the formula : semi-finished product rate * = (1 + spare parts defective rate). If the semi-finished product is not detected, the finished product defective rate is cumulatively punished according to the formula : finished product defective rate * = (1 + semi-finished product defective rate). If the finished product is not detected, the total cost needs to be superimposed on the exchange loss × finished product defect rate. Disassembly compensation constraint : If you choose to disassemble semi-finished products or finished products, the total cost needs to be deducted from the compensation value. 3.Validation of policy effectiveness : Traverse all policy combinations and select the optimal policy that satisfies the constraints.

Step 4 : Joint optimization and result output : Combining genetic algorithm with dynamic programming to optimize the total profit maximization strategy. The maximum total profit strategy and the corresponding detection and dismantling scheme are output.

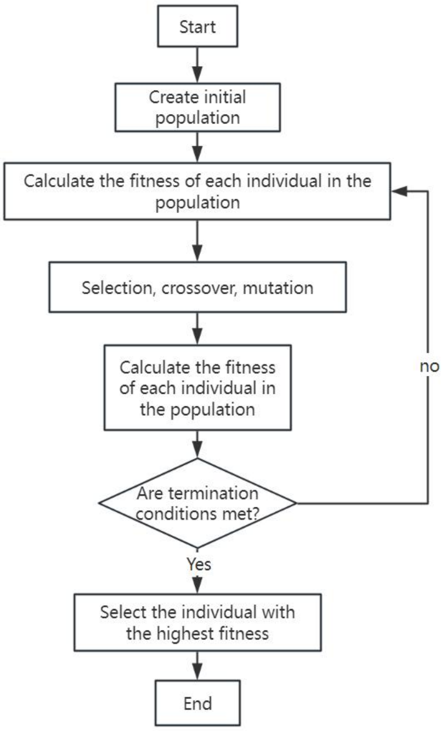

Start
Create initial population
Calculate the fitness of each individual in the population
Selection, crossover, mutation
Calculate the fitness of each individual in the population
no
Are termination conditions met?
Yes
Select the individual with the highest fitness
End

Figure 1 Dynamic Programming Model Solving process

4 AN INNOVATIVE SIMULATION - BASED SOLUTION APPROACH

The results of the improved fusion model : By initializing the population function, establishing the fitness function, evaluating the fitness of each individual, calculating the total profit and the defective rate, taking the total profit as the fitness value, and then performing a series of operations such as crossover and mutation of the genetic algorithm, 65536 different decision-making schemes are obtained after running the python software. Considering the large amount of sample data, there is no specific decision-making scheme here, only the total profit comparison under different strategies is shown, as shown in Figure 2.

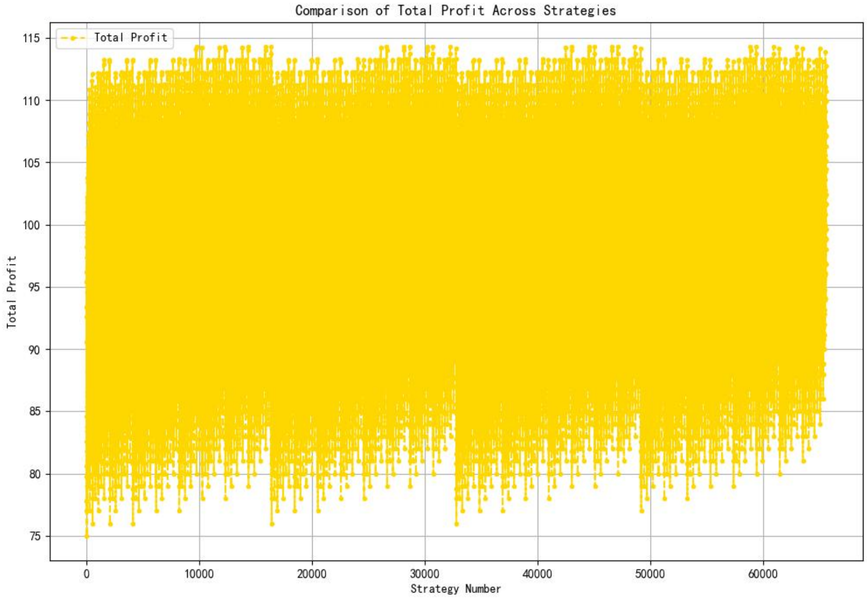


Figure 2 Total Profit Comparison under Different Strategies

Through the information in Figure 2, it can be seen that under more than 60,000 strategies, the total profit of the decision-making scheme shows a relatively regular change, and its total profit is roughly distributed between 83-112 yuan.

The highest total profit of 114.29 yuan is obtained by screening the output results in descending order. The decision-making scheme is to detect spare parts 1, 2, 4, not to detect spare parts 3, 5, 6, 7, 8, not to detect semi-finished products 1-3, detect finished products, disassemble unqualified semi-finished products 1-3, disassemble unqualified finished products. The minimum total profit is 75 yuan. The decision-making scheme is to detect parts 1-8, semi-finished products 1-3, finished products, unqualified semi-finished products 1-3, and unqualified finished products.

Table 1 Top 3 Decision Schemes of Total Profit

Policy number	decision plan	gross profit	Finished product rate
12257	testing spare parts 1,2,4, not testing spare parts 3,5,6,7,8, not testing semi-finished products 1-3, testing finished products, dismantling unqualified semi-finished products 1-3, dismantling unqualified finished products	114.2927	0.1565
14305	testing spare parts 1,2,5, not testing spare parts 3,4,6,7,8, not testing semi-finished products 1-3, testing finished products, dismantling unqualified semi-finished products 1-3, dismantling unqualified finished products	114.2927	0.1565
15841	testing spare parts 1,2,7, not testing spare parts 3,4,5,6,8, not testing semi-finished products 1-3, testing finished products, dismantling unqualified semi-finished products 1-3, dismantling unqualified finished products	114.2927	0.1565

Table 1 shows the decision-making schemes in the top 3 of the total profit. Through the data in Table 1, the decision-making scheme in the production process can be clearly provided for the enterprise. How to make decisions can maximize the benefits, and the finished product defect rate is taken as the corresponding index of the decision-making scheme.

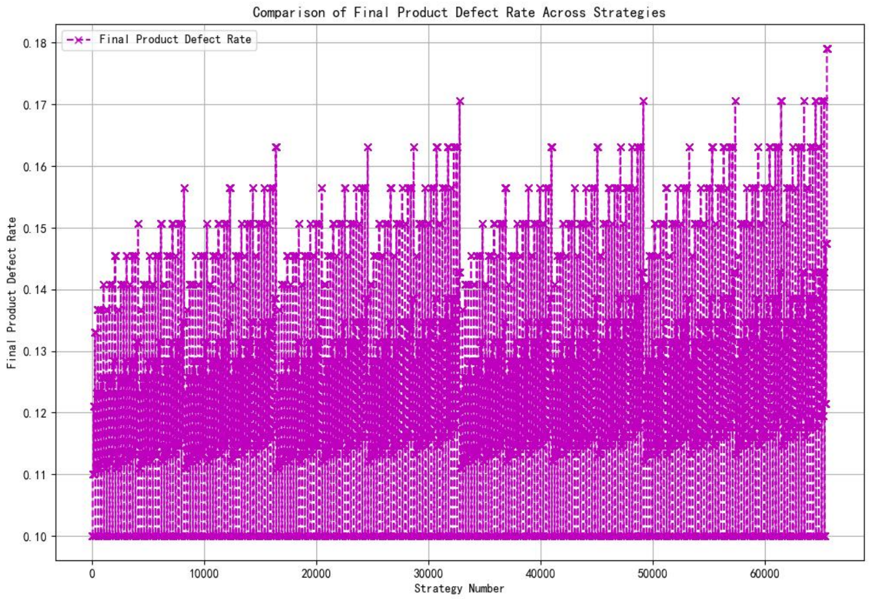


Figure 3 Comparison of Finished Product Defect Rate under Different Strategies

As depicted in Figure 3, the graph illustrates the comparison of the final product defect rate across various strategies. This visualization clearly demonstrates the fluctuation and distribution of defect rates under different strategies. It can be observed that the defect rate shows a certain degree of randomness and variability, without a clear - cut upward or downward trend. This further confirms that the defective rate of finished products under different strategies can indeed be used as the corresponding index result. After providing the specific decision - making scheme and its basis, analyzing this graph allows us to gain a more intuitive and comprehensive understanding of how different strategies impact the defect rate of finished products.

By establishing a production scheduling optimization model based on dynamic programming and genetic algorithm, remarkable results have been achieved : in 65536 decision-making schemes, the highest total profit reaches 114.29 yuan, and the corresponding decision-making scheme is to detect spare parts 1,2,4, do not detect spare parts 3,5,6,7,8, do not detect semi-finished products 1-3, detect finished products, disassemble unqualified semi-finished products 1-3, disassemble unqualified finished products ; the minimum total profit is 75 yuan, and the corresponding decision-making scheme is to detect all spare parts and semi-finished products, detect finished products, but not disassemble any non-conforming products.

5 CONCLUSION

This paper focuses on solving the problem that the profit cannot be improved due to the difficulty in formulating a reasonable decision-making plan in the actual operation of the enterprise. In view of the limitations of traditional dynamic programming methods in dealing with large-scale and complex problems, this paper introduces a dynamic programming model based on genetic algorithm. Genetic algorithm can effectively avoid local optimum and improve search efficiency. By combining genetic algorithm with dynamic programming, the advantages of both can be fully utilized to solve the problems that traditional dynamic programming is difficult to deal with. The actual production plan is taken as an example to verify the application. The results show that the model is effective and the corresponding decision-making scheme is successfully found when the profit reaches the maximum value.The total profit is roughly distributed between 83-112 yuan. Future work can consider using more intelligent optimization algorithms, such as particle swarm optimization algorithm, simulated annealing algorithm or ant colony algorithm, to further improve the efficiency of the model and the quality of the solution, and provide better production decision-making solutions for enterprises.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCE

[1] Chauhan M, Negi R, Ravi, et al. Empowering Small-Scale Vegetable Farmers with Drone-based Decision Support Systems for Sustainable Production. International Journal of Environment and Climate Change, 2025, 15(1): 203-216. DOI: 10.9734/ijecc/2025/v15i14685.

[2] Song H, Li X, Xin L, et al. Forging new pathways: How farmland abandonment affects decision-making of non-grain production - Insight from China's mountainous areas. Journal of environmental management, 2024, 373123753.

[3] Zhang L, Zhou J. Joint Pricing-Production Decisions for a Capital-Constrained Supplier in a Marketplace Platform. Journal of Theoretical and Applied Electronic Commerce Research, 2024, 19(4): 3547-3570.

[4] Liu T. The Research on Production Decision-Making Based on Cost-Benefit Analysis Models. Industrial Engineering and Innovation Management, 2024, 7(4): 28-36.

[5] Jin Z, Zhao Y, Sun Y, et al. Force feedback controller of a parallel haptic device via online adaptive dynamic programming. Mechatronics, 2025, 106103293-103293.

[6] Yang X, Wu C, Huang Z, et al. A two-stage dynamic planning for rural hybrid renewable energy systems under coupled carbon-green certificate trading. Energy, 2025, 316134336-134336.

[7] Wang K, Song H, Guo Z, et al. Automated multi-dimensional dynamic planning algorithm for solving energy management problems in fuel cell electric vehicles. Energy, 2025, 316134408-134408.

[8] Chen H, Lin Z, Chen Z, et al. Adaptive DWA algorithm with decision tree classifier for dynamic planning in USV navigation. Ocean Engineering, 2025, 321120328-120328.

[9] Qazi M I, Muneer A, Sonia R, et al. RDF Query Path Optimization Using Hybrid Genetic Algorithms: Semantic Web vs. Data-Intensive Cloud Computing. International Journal of Cloud Applications and Computing (IJCAC), 2022, 12(1): 1-16.

[10] Tian-Wei L, Jiang-Bo B, Nicholas F, et al. Multi-objective optimisation designs for thin-walled deployable composite hinges using surrogate models and Genetic Algorithms. Composite Structures, 2022, 280.

[11] Evandro M N, Bruno B G H, Sergio S G H, et al. Variable selection for estimating individual tree height using genetic algorithm and random forest. Forest Ecology and Management, 2022, 504.

[12] Genghan Y, Yang T, Peng W, et al. Optimizing 3D printing of chicken meat by response surface methodology and genetic algorithm: Feasibility study of 3D printed chicken product. LWT, 2022, 154.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3049

THE FUTURE DEVELOPMENT OF NEW ENERGY VEHICLES BASED ON ARIMA TIME SERIES PREDICTION MODEL

YiTong Liu[1*], YiLin Wang[2]
[1]*Department of business, Accounting, Xi'an International Studies University, Xi'an 710128, Shaanxi, China.*
[2]*Department of business, Business English, Xi'an International Studies University, Xi'an 710128, Shaanxi, China.*
Corresponding Author: YiTong Liu, Email: 18991002666@163.com

Abstract: This paper mainly adopts the evaluation model based on gray correlation method, multiple linear regression model, the secondary polynomial regression model, studied the influence of various factors of new energy vehicles in China and the influence of traditional fuel vehicles, and then choose the prediction of the next ten years, collect new energy vehicles in the past seven years, using the Pearson correlation coefficient test the development of new energy electric vehicles and predictor, found that the strong correlation, and derived the corresponding index of the correlation coefficient. And the ARIMA time series model is used to predict the trend of new energy electric vehicles in the next decade. Research shows that the research and development of new energy electric vehicles is very important for environmental protection. This paper calls on people to buy and ride in new energy electric vehicles to reduce greenhouse gas emissions and promote green development.

Keywords: New-energy electric vehicles; Multiple linear regression model; Pearson correlation analysis; ARIMA time series analysis

1 INTRODUCTION

These years has witnessed rapid economic growth and globalization process, coupled with environmental crises, posing a great threat to the world [1]. For China the world second large economy, it is essential to implement environmental protection measures, addressing the rapidly growing global energy demand [2]. Therefore, the New Energy Vehicles represent a choice dictated by history.

New energy vehicles (NEVs) are vehicles which feature advanced technical principles, novel technologies, and new structures, encompassing four main categories: hybrid electric vehicles (HEVs), pure electric vehicles (PEVs), fuel cell electric vehicles, and other types. NEVs utilize unconventional fuels (excluding gasoline and diesel) as their power source and incorporate advanced vehicle power control and drive technologies, noted for their low energy consumption and reduced pollution [3,4].

Previously, Huang, X., Zhang, D., and Zhang, X. proposed an intelligent building energy management strategy using deep reinforcement learning [5]. However, they overlooked external factors' impact on new energy vehicle development within this context. Zeng B ,Yin F ,Wang J , et al proposed an IBO and a three - parameter gray prediction model for Chinese NEV sales, tackling data and prediction trap issues.Yet, their research under-emphasizes product-related factors like performance and after-sales on sales [6]. Wang, Z. et al, based on deep neural networks, explored sustainable design factors in NEVs for promoting sustainable consumption [7-8], but didn't quantify economic, technological, and policy impacts or predict long - term market trends.

This study addresses these gaps through a combination of methods. By applying gray correlation analysis, the complex factors can be comprehensively analyzed, which influence the new energy vehicle industry, including economic, technological, and policy factors. This helps in understanding the relative importance of different factors and their impact on the industry's development. In addition, the use of ARIMA time-series prediction allows us to forecast new energy vehicle trends accurately. It helps us to not only predict sales but also project long-term market trends. Moreover, by integrating two methods to conduct a comprehensive analysis, the relationships between various factors and the development of new energy vehicles can be qualified, providing a more in-depth analysis of the quantitative impacts of economic, technological, and policy factors that the former studies failed to achieve.The results of this study not only provide a theoretical basis for the development of the new energy vehicle industry, but also provide an important reference for government decision-making and enterprise strategic planning.

2 FACTORS AFFECTING NEW ENERGY VEHICLE DEVELOPMENT

2.1 Variable Declaration

When analyzing the main factors affecting the development of new energy electric vehicles in China,consider the following factors: economy, science and technology, policy and infrastructure and many other aspects, which are interrelated and affect each other, so in order to clarify the impact of these factors on the development of China's new energy electric vehicles more clearly, establish a multivariate linear regression model.

Assuming that the development level of new energy electric vehicles is Y, and the factors of economy, science and technology, policy and infrastructure are X1, X2, X3 and X4 respectively, the multiple linear regression model can be expressed as follows:

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \beta 3X3 + \beta 4X4 + \varepsilon \tag{1}$$

Where $\beta 0$ is the constant term, β_1, β_2, β_3 and β_4 are the regression coefficients of each factor, and ε is the random error term.

Specifically, the impact of each factor on the development of new energy electric vehicles in China can be described as follows:

(1) The sales volume of new energy vehicles in China affects the market share occupied by new energy vehicle producers, which in turn affects consumer acceptance and producer production motivation. When the sales volume of new energy vehicles increases, the market share it accounts for will increase, which means that the influence and status of the enterprise in the industry will be enhanced.

(2) The energy density of Chinese new energy vehicles will greatly affect the range of new energy vehicles, which in turn affects the user satisfaction of new energy vehicles. The higher the energy density, the longer the range, the higher the consumer satisfaction and user experience, and the more inclined to buy new energy vehicles. Therefore, the regression coefficient β_2 of X2 should be positive.

(3) The cumulative amount of subsidies provided by the Chinese government for new energy vehicles will reduce the cost of purchasing new energy vehicles, promote consumers' motivation to buy new energy vehicles, stimulate market demand. The stronger the government subsidy, the faster the new energy industry will develop. Therefore, the regression coefficient β_3 of X3 should be positive.

(4) The degree of perfection of infrastructure related to new energy vehicles is inextricably linked to the user experience, which will greatly affect the market demand for new energy vehicles and other factors. For example, the more charging piles in the country, the more convenient it is to use new energy vehicles, and the greater the market demand, the regression coefficient β_4 of X4 should be positive.

2.2 Model Establishment

Determine the parent sequence and feature sequence, and prepare the data format. Discuss each major step in the comparative sequence analysis process as follows:

$$\left[X_1', X_2', \ldots, X_n'\right] = \begin{bmatrix} x_1'(1) & x_2^{‘}(1) & \cdots & x_n'(1) \\ x_1^{/'}(2) & x_2^{‘}(2) & \cdots & x_n'(2) \\ \vdots & \vdots & & \vdots \\ x_1^{/'}(m) & x_2^{‘}(m) & \cdots & x_n'(m) \end{bmatrix} \tag{2}$$

The parent sequence is as follows:

$$X_0' = \left(x_0'(1), x_0'(2), \ldots, x_0'(m)\right)^T \tag{3}$$

Carryout dimensional normalization of the index (usually required), in order to truly reflect the actual situation, eliminate the influence caused by the difference of each index unit and the disparity between the value orders, and avoid the occurrence of unreasonable phenomena, it is necessary to carry out dimensional normalization of the index.

Calculate the gray relational coefficient between the master sequence and the characteristic sequence. Calculate the correlation coefficient between each element in the comparison sequence and the corresponding element in the reference sequence according to the formula below:

$$\gamma\left(x_0(k), x_i(k)\right) = \frac{\Delta \min + \rho \Delta \max}{\Delta_{ik} + \rho \Delta \max} \tag{4}$$

$$\Delta \min = \min_i \min_k \mid x_0 (k) - x_i (k) \mid \tag{5}$$

$$\Delta \max = \max_i \max_k \mid x_0 (k) - x_i (k) \mid \tag{6}$$

$$\Delta_{ik} = \mid x_0 (k) - x_i (k) \mid \tag{7}$$

The resolution coefficient takes values within [0.5], where the smaller the resolution coefficient, the greater the difference between the correlation coefficients, indicating stronger discriminative ability. Usually, a resolution coefficient of 0.5 is chosen.

Solve the correlation degree value:

$$\gamma_{0i} = \frac{1}{m}\sum_{k=1}^{m} W_k \zeta_i(k) \tag{8}$$

Sort the correlation values to draw conclusions. Establish the correlation sequence of each evaluation object based on the size of the gray weighted correlation. The greater the correlation degree, the greater the importance of the evaluation object to the evaluation criteria.

2.3 Results

The related curves and bar charts derived from these data are as follows Figure 1 and Figure 2:

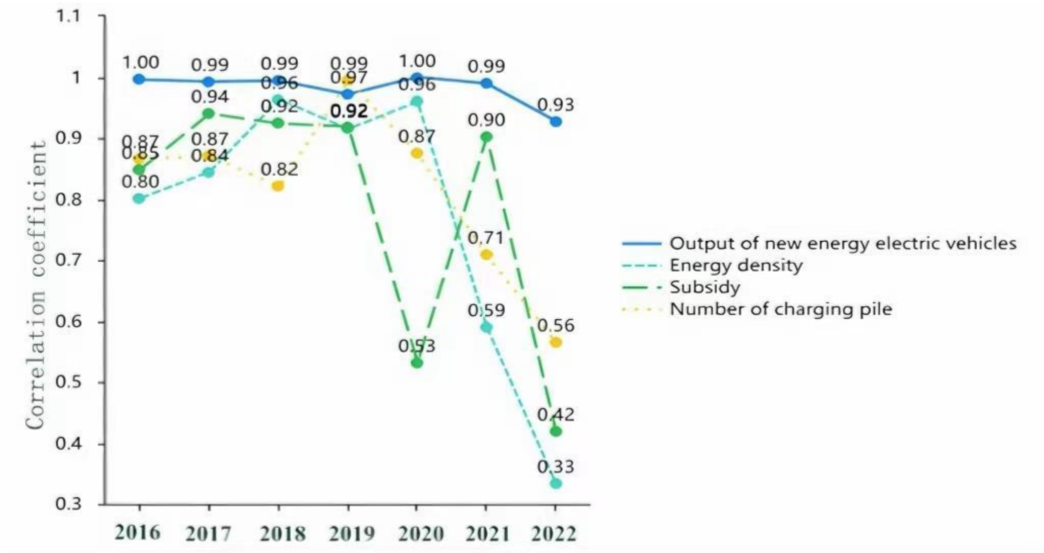


Figure 1 The Related Curves

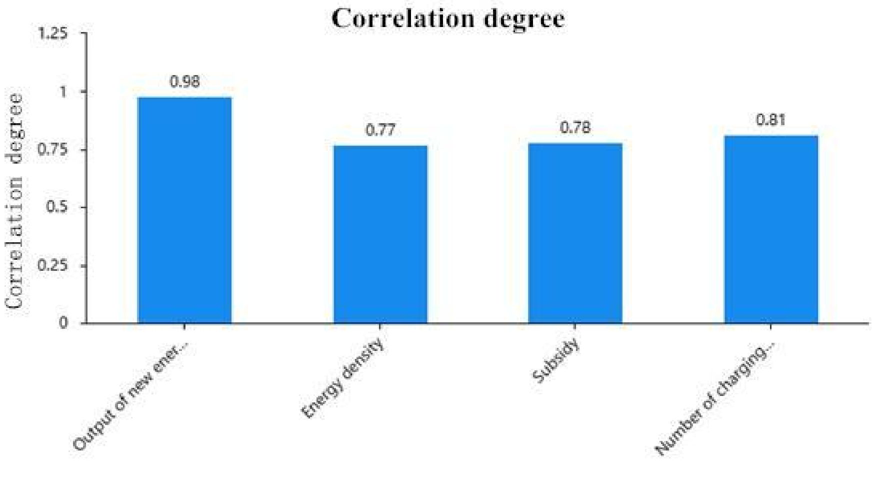


Figure 2 The Correlation Degree

According to the gray correlation algorithm,it is finally concluded that the production of new energy electric vehicles and the number of charging piles in China are the most important factors affecting the development of new energy electric vehicles in China, with the greatest degree of correlation. Therefore, according to the obtained data, in order to promote the development of the new energy electric vehicle industry, the enterprises should be incentivized to increase the cost of production investment and scientific and technological R&D costs of new energy vehicles, and continue to increase the new energy vehicle's market share, accelerate technology iteration, and develop new energy vehicles with longer range and higher performance as soon as possible. At the same time, more charging piles should be arranged to make the use of new energy vehicles more convenient.

3 PREDICTION ON THE FUTURE DEVELOPMENT OF NEW ENERGY VEHICLES

3.1 Basic Data Analysis

(1) x_1 is the utilization efficiency of each unit of charging fee for new energy electric vehicles.

(2) x_2 is the percentage of government subsidies for new energy EVs in the total subsidy amount, and a 1:100 ratio is adopted to standardize the order of magnitude;

(3) x_3 is the vehicle-to-pile ratio, representing the ratio of new energy vehicles per charging pile;

(4) x_4 is the ratio of new energy vehicle production to China's total automobile production.

3.2 ARIMA Forecasting in the Development Trajectory of Nevs

3.2.1 Pearson correlation analysis

Through Pearson correlation analysis, x_1 , x_2 , x_3 , x_4 have significant correlation on China's new energy vehicle market share.

(1) Market share(%) and Utilization efficiency of charging charges(%): correlation coefficient 0.981, significant at 0.01 level, significant positive relationship.

(2) Market share(%) and Subsidy ratio (1:100): correlation coefficient 0.778, significant at 0.05 level, significant positive correlation.

(3) Market share(%) and Vehicles per charging pile ratio (%): correlation coefficient -0.776, significant at 0.05 level, significant negative correlation.

(4) Market share(%) and proportion of new energy vehicle production: correlation coefficient 0.997, significant at 0.01 level, significant positive correlation.

3.2.2 Model establishment

Based on the foregoing, then formulate the subsequent equation by employing the correlation coefficient as the coefficient thereof, and subsequently derive the corresponding function F(x) for the period from 2016 to 2022.

$F_{(x)}$ is the comprehensive future development trend and make ARIMA forecast on it, which can lead to the development trend of new energy vehicle industry in the next 10 years.

$$\mathrm{F(x)} = \mathrm{ax}_1 + b\mathrm{x}_2 + c\mathrm{x}_3 + d\mathrm{x}_4 \tag{9}$$

The ARIMA model comprises three components: AR, I, and MA. AR stands for Auto Regression (autoregressive model), I for Integration (single integer). As a time series model, ARIMA requires the time series to be smooth for establishing an econometric model. Hence, a unit root test is initially performed on the time series. If non-smooth, differing is needed to convert it into a smooth series, and the number of times differing is applied is called the order of single-integer. MA represents the Moving Average (moving average model). Thus, the ARIMA model is essentially a combination of the AR and MA models.

AR model is used to describe the relationship between the current value and the historical value, and use the historical time data of the variable itself to forecast itself, and its formula is as follows:

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_t - i + \varepsilon_t \tag{10}$$

y_t denotes the current value, μ denotes the constant term, p denotes the order, γ_i denotes the autocorrelation coefficient, and ε_t denotes the error.

The MA model is concerned with the accumulation of error terms in the autoregressive model, which can effectively eliminate the random fluctuations in the prediction, and its formula is as follows.

$$y_t = \mu + \sum_{i=1}^{p} \theta_i \varepsilon_{t-i} + \varepsilon_t \tag{11}$$

Generally speaking, the MA(q) model must be smooth when q is constant, so for the ARIMA model, it is only necessary to verify the smoothness of the AR model.

The ARIMA autoregressive moving average model combines the advantages of both AR and MA models. In the ARMA model, the autoregressive process is responsible for quantifying the relationship between the current data and the previous data, and the moving average process is responsible for solving the problem of solving the term of the immediate change, and its formula is as follows.

$$y_t = \mu + \sum_{i=1}^{p} \gamma_i y_{t-i} + \varepsilon_t \sum_{i=1}^{p} \theta_i \varepsilon_{t-i} \tag{12}$$

3.2.3 Solutions

Since the smoothness of the original data is poor, the original data are processed by first-order differing: the purpose of differing is to eliminate the instability of the series and make its fluctuation curve smoother. Through the time series of first-order difference and second-order difference, to carry out the ADF unit root test for the ARIMA model, this paper utilizes SPSS PRO software to carry out first-order and second-order difference processing on the data. The results are shown in the Table 1: The result of ADF test below:

Table 1 The Result of ADF Test

f(x)-ADF test					
Difference in order	t	p	Threshold value		
			1%	5%	10%
0	0.079	0.965	-5.354	-3.646	-2.901
1	-2.964	0.038	-6.045	-3.929	-2.987
2	-3.235	0.018	-7.355	-4.474	-3.127

As it can be seen from the above table, the p=0.965>0.1, the original hypothesis cannot be rejected and the series is not smooth. The first-order difference is performed on the series and then the ADF testis performed.
The ADF test result of the data after the first-order difference shows p=0.038<0.05, there is a higher than 95% certainty that the original hypothesis is rejected, and the series is smooth at this time.
Therefore, the 1st difference order in ARIMA model is chosen.
Then, perform the white noise residual test, as the Table 2: Model Q statistic result shown:

Table 2 Model Q Statistic Result

Model Q statistic table		
Items	Statistic	*p*
Q1	2.314	0.128
Q2	2.336	0.311
Q3	2.377	0.498
Q4	2.377	0.667
Q5	2.377	0.795

From the results of Q statistics, the p-value of Q5 is greater than 0.1, then the original hypothesis cannot be rejected at the significance level of 0.1, the residuals of the model are white noise, and the model basically meets the requirements. According to the ARIMA prediction model, this paper predicts the development trend of new energy electric vehicles in the global decade as shown in Figure 3:

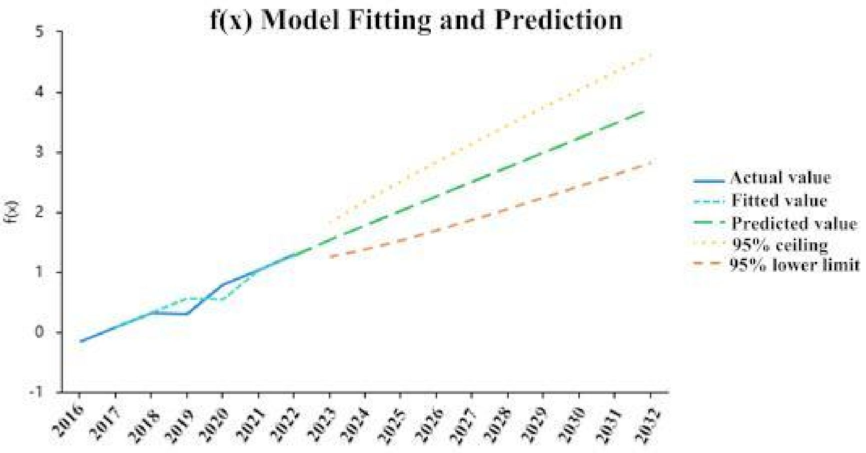


Figure 3 f(x) Fitting and Prediction

The LM test fails in the ARIMA model testing, and the test model residual series do not have serial correlation, which may be caused by the data volume is too small. However, in the partial autocorrelation plot, the p-value greater than 0.05 does not directly determine whether ARIMA forecasting is possible.The selection of ARIMA model needs to consider the characteristics of autocorrelation plot (ACF) and partial autocorrelation plot (PACF). As the result above, combined the ADF test and partial autocorrelation analysis to finally decide to use the ARIMA (0, 1, 0) model to predict f(x) and arrive at the above results.

4 CONCLUSION

The results show that among the factors affecting the development of new energy vehicles, policy subsidies, production of new energy vehicles play a key role in the number of new energy vehicles. The output of new energy vehicles and the number of charging piles are the most closely related to the development of new energy vehicles in China, indicating that these factors are the main driving force for the development of the new energy vehicle industry. Secondly, new energy vehicles will show a growth trend in the next decade. With the enhancement of global awareness of environmental protection and the transformation of energy structure, new energy vehicles will gradually become the mainstream of the automobile market with their characteristics of low carbon, environmental protection, energy saving and high efficiency. Governments of various countries have introduced relevant policies to increase the support for new energy vehicles and promote their industrialization process. Technological innovation will continue to promote the development of new energy vehicles, and the popularization of new energy vehicles will promote the optimization of the energy structure. To sum up, China's new energy vehicle industry has broad development prospects and huge market potential. The results of this study not only provide a theoretical basis for the development of the new energy vehicle industry, but also provide an important reference for government decision-making and enterprise strategic planning. Research shows that new energy vehicles are of great significance to environmental protection, and China's new energy vehicle industry has broad development prospects and huge market potential in the future.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Zhang Q, Du M A, Lin B. Driving total factor productivity: The spillover effect of digitalization in the new energy supply chain. Research in International Business and Finance, 2025, 75, 102764-102764.

[2] Fan B, Wen Z, Qin Q. Competition and cooperation mechanism of new energy vehicle policies in China’s key regions. Humanities and Social Sciences Communications, 2024, 11(1): 1640-1640.

[3] Liu B, Song C, Wang Q, et al. Research on regional differences of China’s new energy vehicles promotion policies: A perspective of sales volume forecasting. Energy, 2022, 248: 123541.

[4] Liu Q, Wen X, Cao Q. Multi-objective development path evolution of new energy vehicle policy driven by big data: From the perspective of economic-ecological-social. Appl Energy, 2023, 341: 121065.

[5] Huang Xiaoqing, Zhang Dongliang, Zhang XiaoSong. Energy management of intelligent building based on deep reinforced learning. Alexandria Engineering Journal, 2021, 60(1): 1509-1517.

[6] Zeng B, Yin F, Wang J, et al. Prediction and rationality analysis of new energy vehicle sales in China with a novel intelligent buffer operator. Engineering Applications of Artificial Intelligence, 2025, 143, 110030-110030.

[7] Wang Z, Niu S, Hu S, et al. How to promote sustainable consumption and development of NEV? Decoding complex interrelationships in consumer requirements and design practice. Journal of Cleaner Production, 2025, 486, 144524-144524.

[8] Jiang Jiachen, Mei Yuxin, Pan Yaoyao, et al. Development evaluation and sales forecast of new energy vehicles. Journal of Taizhou University, 2024, 46(03): 9-15.

Journal of Computer Science and Electrical Engineering
ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3050

NETWORK INTRUSION DETECTION METHODS BASED ON MACHINE LEARNING

JingYa Sun[1*], ZiJie Cao[2]
[1]Mathematics and Physics Teaching Department, Hebei GEO University, Shijiazhuang 050030, Hebei, China.
[2]School of Information Engineering, College of Science & Technology Ningbo University, Ningbo 315300, Zhejiang, China.
Corresponding Author: JingYa Sun, Email: sunjingya2021@163.com

Abstract: With the rapid development of Internet technology, traditional intrusion detection methods have limitations. Although machine learning technology provides new ideas for intrusion detection, its efficiency and accuracy in large-scale data environment still need to be optimized. This study aims to propose an intrusion detection method that combines feature selection with optimized machine learning algorithms to solve the problems of data redundancy and category imbalance, and to reduce the false alarm rate. Based on the UNSW-NB15 dataset, ANOVA, chi-square test and Gini coefficient are used for feature selection, combined with principal component analysis (PCA) dimensionality reduction technique. The model is constructed by algorithms such as logistic regression and random forest, and hyperparameter optimization is carried out using GridSearchCV, and data imbalance and outliers are handled by stratified sampling and RobustScaler. The experiments show that the balanced accuracy of the logistic regression model is 70%, and the accuracy of the random forest model is 67.33%. Feature selection significantly improves the model performance. The method proposed in this study demonstrates high efficiency and reliability in large-scale network data and provides a technical basis for the design of real-time intrusion detection systems.

Keywords: Network intrusion detection; Feature selection; ANOVA; Logistic regression

1 INTRODUCTIONS

The rapid evolution of network technology demands advanced intrusion detection systems (IDS) addressing high-dimensional data, redundancy, and class imbalance. Machine learning (ML) offers innovative solutions through feature engineering, model optimization, and data preprocessing.

Wei Jintai and Gao Qiong [1] integrated information gain with a random forest classifier using Quantum Particle Swarm Optimization (QPSO) and ReliefF for dimensionality reduction, while Zhu Linjie et al. [2] applied mutual information (MI) to filter low-redundancy features, achieving 99.8%/99.6% accuracy on NSL-KDD. He Hongyan et al. [3] improved small-sample attack recall via ET-RFE feature selection and optimized LightGBM. Samantaray et al. [4] advocated MaxAbsScaler for IoT feature scaling, enhancing generalization. Ali et al. [5] developed a genetically optimized ensemble model achieving 0.00145 MAE on Kaggle. Nabi and Zhou [6] reduced CICIDS2017 features to 10 dimensions using autoencoders and PCA, retaining 99.6% accuracy with random forests. Arco et al. [7] proposed a two-step training framework to mitigate dataset contamination. Sarhan et al. [8] emphasized standardized feature sets for IoT detection. Ren Jiadong et al. [9] validated Pearson correlation-based feature selection across classification tasks.

Despite advancements, lightweight models, dynamic adaptability, and cross-domain transferability remain challenges. This study integrates ANOVA, chi-square, and Gini coefficient-based feature selection with PCA, optimizes logistic regression and random forest via GridSearchCV, and applies stratified sampling and RobustScaler to enhance detection efficiency and reduce false alarms in large-scale networks.

2 ALGORITHMIC PRINCIPLES FOR NETWORK INTRUSION DETECTION BASED ON MACHINE LEARNING

With the rapid development of network technology, network security is facing increasingly severe challenges. Network intrusion detection methods based on machine learning have become a research hotspot, but still face the problems of high dimensionality, redundancy and imbalance of data features. In order to overcome these problems, feature selection, dimensionality reduction techniques, and hyperparameter optimization have become important means to improve the model performance.

2.1 Feature Selection and Dimension Reduction Techniques

The purpose of feature selection is to select the most relevant features for the target variable from the original feature set. In machine learning-based network intrusion detection, the original feature set often contains a large amount of redundant and irrelevant information, and feature selection aims to select the most relevant features for the target variable (intrusion judgement) from the original feature set to reduce the spatial dimensionality of the features while retaining the classifier's ability to effectively interpret the data patterns. The most relevant features for the target variable (intrusion judgement) are selected from these original features to reduce the spatial dimension of the features, while retaining the classifier's ability to effectively interpret the data patterns. Common feature selection methods can be divided into the following categories:

(1) Filtering method: By calculating the correlation between features and target variables, features are filtered based on the statistical relationship between features and target variables. This method does not rely on specific learning algorithms and therefore has a high computational efficiency.

The chi-square test measures the degree of influence by calculating the chi-square value between the characteristics and the target variable, and the central idea is to compare the difference between the observed and expected values, which is given by the formula:

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \# \tag{1}$$

Where O_i is the observed value and E_i is the expected value. When the chi-square statistic is larger, it indicates a stronger correlation between the feature and the target variable.

(2) Packing method: the best performing feature subset is selected by constructing several different feature subsets and evaluating them directly using a predictive model. Recursive Feature Elimination (RFE) is a typical representative, which is based on the initial full set of features and evaluates the performance of feature subsets by constructing a prediction model, the core steps of which are to train the model, compute the feature importance scores, and gradually eliminate the weakest features. Each iteration removes the least contributing features and re-evaluates them until the preset conditions are satisfied. In addition, forward stepwise selection and backward stepwise elimination are also common packaging methods, with the former gradually adding features starting from the empty set and the latter gradually removing features starting from the full set. This method is more computationally intensive, but it can obtain the feature combination with optimal performance.

(3) Embedding method: feature selection is performed during model training, and feature importance assessment and model construction are completed at the same time, which has higher efficiency. Commonly used embedding methods include:

Decision tree: the decision tree selects the optimal split features based on indicators such as information gain, information gain rate or Gini index when constructing the tree structure, so as to automatically screen out the features that play an important role in classification; the information gain is calculated as:

$$IG(T,X) = H(T) - \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i) \# \tag{2}$$

where $H(T)$ is the entropy before feature splitting, $H(T_i)$ is the conditional entropy after splitting, and $|T_i|$ and $|T|$ are the sizes of the subset and the full set, respectively.

Lasso regression achieves feature selection by adding an L1 regularization term to the loss function, which makes the coefficients of some of the features zero. Its loss function is:

$$L = \frac{1}{2n} \sum (y_i - \hat{y_i}) + \lambda \sum_{j=1}^{p} |\beta_j| \# \tag{3}$$

where λ is the regularisation factor that controls the strength of the contraction on the coefficients.

Dimensionality reduction techniques aim to reduce the dimensionality of the data and reduce the computational complexity while retaining as much of the main information of the data as possible. Common methods of dimensionality reduction are Principal Component Analysis (PCA).

(1) Principal Component Analysis (PCA), which projects the data into a low-dimensional space consisting of principal components by constructing the covariance matrix and extracting its eigenvalues and eigenvectors. The covariance matrix of the original data is first calculated, which reflects the correlation between the features. The covariance matrix is then eigenvalue decomposed to obtain the eigenvalues and eigenvectors. The eigenvalues represent the variance of the data in each direction, the larger the variance is, the more information is available. PCA sorts the eigenvectors according to the eigenvalues from the largest to the smallest, and selects the first k eigenvectors (k is the dimensionality after dimensionality reduction), and then projects the original data to the low-dimensional space composed of these k eigenvectors to achieve dimensionality reduction. The principal component corresponds to the eigenvector with the largest eigenvalue.

Feature selection and dimensionality reduction are closely related techniques that can often be used in combination to improve the efficiency and accuracy of intrusion detection systems. In addition, these techniques can reduce noise interference and improve the generalization ability and reliability of the model.

2.2 Model Training and Optimization

In the development of machine learning frameworks, the pivotal stages of training and optimization play a critical role, as they are intrinsically linked to the model's predictive accuracy and its ability to generalize to unseen data. The selection of suitable algorithms, coupled with effective optimization strategies, is essential for ensuring that the intrusion detection system operates with both efficiency and precision. Machine learning methodologies can be broadly categorized into linear models, nonlinear models, and ensemble learning techniques. Among the prevalent machine learning algorithms are:

Logistic Regression: This methodology is primarily employed for binary classification tasks and is particularly effective for linear separable problems. The fundamental principle involves utilizing a logistic function (Sigmoid function) to constrain the output of linear regression within the range of 0 to 1, thereby facilitating the classification of samples into positive (anomalous) or negative (normal) categories. Logistic regression is characterized by its low computational overhead, straightforward training process, and the ability to determine weight parameters through optimization techniques such as gradient descent, which can achieve rapid convergence even with extensive datasets. Moreover, the interpretability of logistic regression is notable, as the weight parameters provide clear insights into the impact of individual features on the predictive outcomes.

Decision Tree: This algorithm constructs a hierarchical tree structure by iteratively partitioning the dataset to perform classification or regression tasks. The criteria for partitioning include metrics such as information gain, gain ratio, and Gini impurity. The tree is built upon the distinct values of the input features, where information gain quantifies the contribution of these features to the classification task by assessing the variability in information entropy before and after the split; a higher information gain indicates a more significant capability of the feature to segregate the data effectively.

K-Nearest Neighbor Algorithm (K-NN): This classification technique operates by assessing the proximity between test samples and training samples, subsequently identifying the K nearest neighbors for classification. It is particularly advantageous in scenarios involving small datasets or when there is a strong correlation among features. The K-NN algorithm is noted for its simplicity and intuitive nature, requiring no complex model training, and exhibits commendable performance with limited sample sizes and intricate data distributions. However, it is computationally intensive, necessitating the retention of all training samples in memory, and it can be sensitive to the selection of the K parameter, impacting classification outcomes significantly.

3 EXAMPLE ANALYSES

3.1 Data Sources

In terms of data scale (Data sources: https://research.unsw.edu.au/projects/unsw-nb15-dataset), the dataset has more than 80,000 records to provide rich support for model training and testing, which is conducive to the model learning normal and abnormal network behaviour patterns, and improving generalization and detection accuracy. In terms of feature completeness, 45 features cover multiple dimensions of network connections, which can comprehensively describe network traffic characteristics, enable the model to identify intrusion behaviours in multiple dimensions, and enhance the detection effect. On the target variable setting, the fields attack_cat (attack category) and label (whether it is an attack) indicate that it is designed for intrusion detection, and is suitable for multi-classification and bi-classification tasks respectively, which is convenient for researchers to model and evaluate. In summary, although this dataset has the potential problem of category imbalance, it is suitable for the research needs of intrusion detection in terms of data size, feature completeness and target variable setting, and has research value.

3.2 Data Pre-Processing

Firstly, data import and merge, read the training set and test set data from the specified path and merge them for subsequent unified processing. Then variable categorization is performed to clearly classify the category variables, numeric variables and target variables. After that, the category variables are encoded using LabelEncoder to convert the category data in string form into numeric values. Then data partitioning was performed to divide the combined data into training and test sets through stratified sampling. Finally, data exploration was performed to analyze the dependent variables as shown in Figure 1:

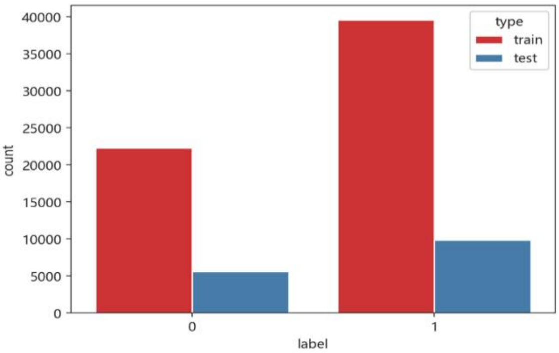


Figure 1 Analysis of Dependent Variables

The data was found to have a significant category imbalance problem, which may affect the training and generalization ability of the model; also, outliers were observed in the numerical variables in the independent variables, and a large number of outliers were found, therefore, data processing with RobustScaler was chosen to reduce the impact of outliers.

3.3 Feature Engineering

*3.3.*1 *Correlation analysis*

Analysis of variance (ANOVA): used to analyze the relationship between continuous features and discrete target variables, calculating the F-value and P-value for each numerical feature and ranking the features according to the F-value, as shown in Figure 2

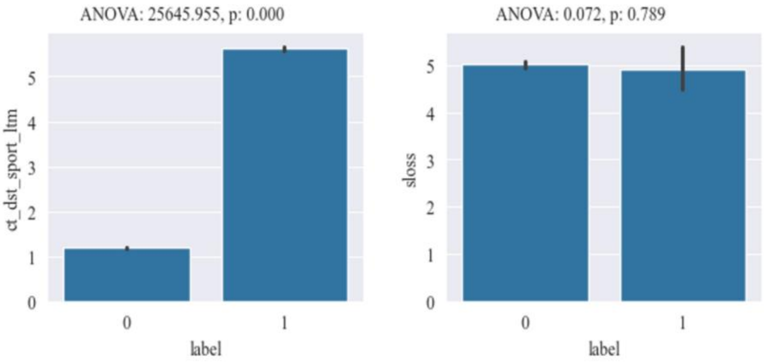


Figure 2 Analysis of Variance

The resultant visualization shows strong correlations between multiple numerical features and the target variable.
Chi-square test: used to assess the correlation between discrete features and discrete target variables, the chi-square and p-value of each categorical feature is calculated and ranked as shown in Figure 3:

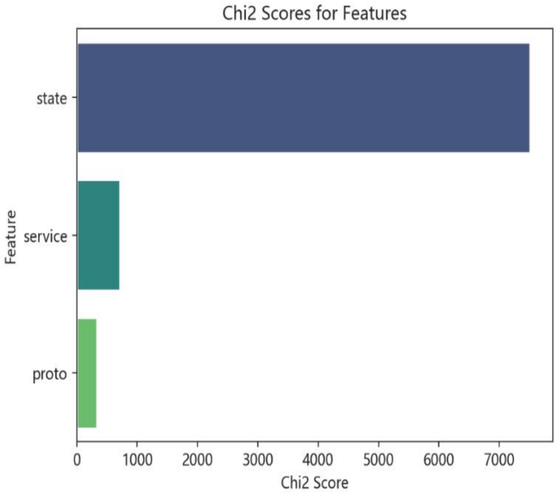


Figure 3 Chi-Square Test

Visualization through bar charts revealed a high correlation between the characteristics of STATE, SERVICE and PROTO and the target variables.

*3.3.*2 *Feature Selection and Dimension Reduction*

Gini coefficient feature selection, The Gini coefficient of importance of features is calculated with the help of Random Forest model to identify features of lower importance as shown in Figure 4.

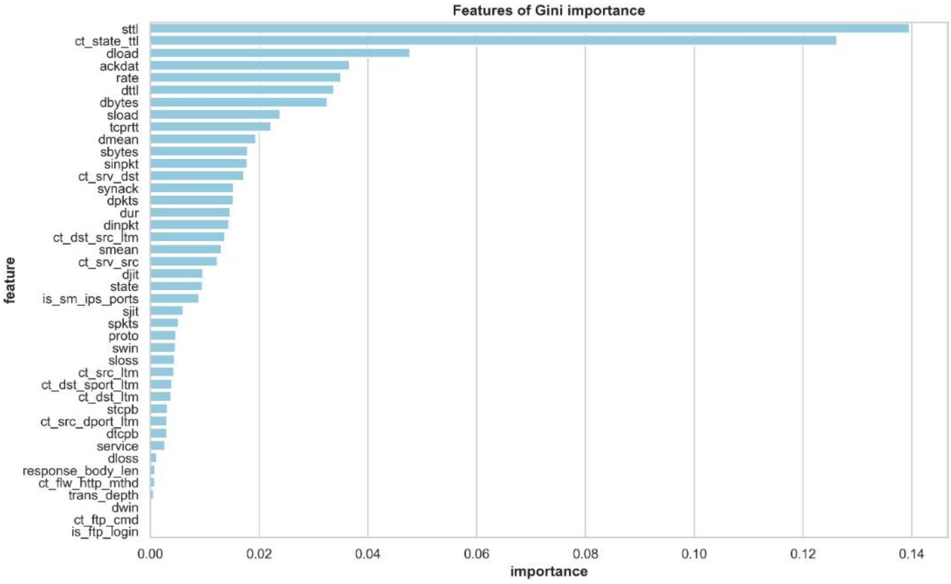


Figure 4 Gini Coefficient Feature Selection

The “sttl” corresponds to the highest importance value, close to 0.12, with the longest bar. The other features have decreasing importance values in order and the bars are correspondingly shorter in length.

3.4 Model Construction and Evaluation

Firstly, model construction and optimization are carried out, adopting logistic regression model and systematically tuning the hyperparameters of the model with the help of GridSearchCV and cross-validation techniques to seek the best combination of hyperparameters, so as to improve the overall performance of the model. Then the model evaluation is carried out to evaluate the performance of the model on the test set using the balanced accuracy as the evaluation criterion, and the confusion matrix and classification report are calculated at the same time in order to comprehensively analyze the classification effect of the model as shown in Figure 5:

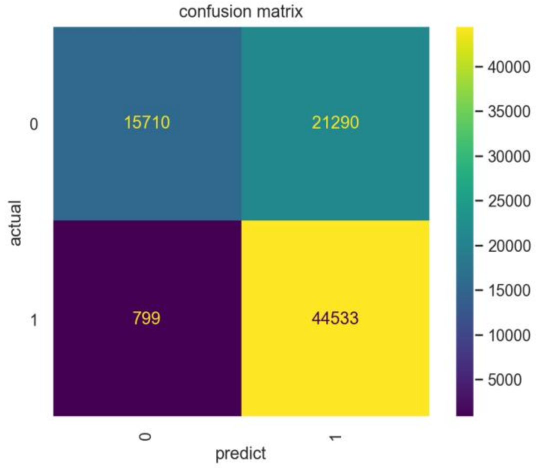


Figure 5 Logistic Regression Model

The results show that the logistic regression model has an equilibrium accuracy of 0.70 on the test set and that there are significant differences in precision, recall and F1 values across categories.

Random Forest Model, the accuracy of the model on the test set was calculated by “accuracy-score” as 0.67, which indicates that the model is correct in about 67.33% of its predictions on the test set, but this accuracy is not considered high and there may be room for optimization. Feature Importance, the importance of each feature is calculated and shown in Figure 6.

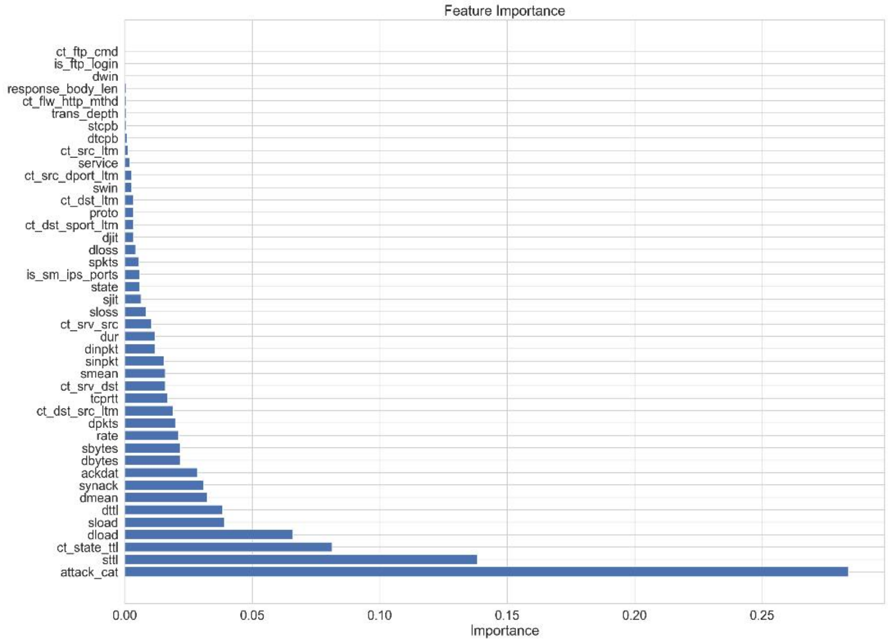


Figure 6 Random Forest Model

The importance of the “attack-cat” feature is 0.218024, which is relatively high among all the features, indicating that it plays a large role in the model decision-making process; while the importance of features like “ct-ftp-cmd” and “is-ftp-login” is only 0.000007, which has less impact on the model decision.

This paper focuses on the task of classifying network traffic data, and integrates a variety of technical tools such as data processing, feature engineering, and model training and evaluation, with the goal of constructing effective classification models to identify cyber-attacks.

Through multiple analysis methods in feature engineering, the importance and relevance of different features to the target variables are clarified, providing a clear direction for subsequent model optimisation, e.g., feature screening can be performed based on feature importance, or other more complex models can be tried to improve classification performance.

4 CONCLUSIONS

This study proposes efficient machine learning-based intrusion detection method, an intrusion detection method that combines feature selection and optimized machine learning algorithms, which can operate efficiently in large-scale data environments with high accuracy and low false alarm rate. A real-time intrusion detection framework is created to design and implement a real-time intrusion detection system that can efficiently respond to changes in network traffic, ensuring a balance between detection accuracy and real-time performance.

From the evaluation results of the classification model, the logistic regression model achieves a certain degree of accuracy in dealing with this network traffic classification task, but there is still room for further improvement in the model performance due to the category imbalance of the data and the complexity of the network traffic data.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Wei Jintai, Gao Qiong. Research on intrusion detection system based on information gain and random forest classifier. Journal of North Central University (Natural Science Edition), 2018, 39(01): 74-79+88.

[2] LJ Zhu, G P Zhao, L H Kang. Intrusion detection method based on the combination of MI feature selection and KNN classifier. Gansu Science and Technology, 2022, 38(15): 33-36.

[3] Hongyan He, Guoyan Huang, Bing Zhang, et al. Anomaly detection model based on recursive elimination of limit tree features and LightGBM. Information Network Security, 2022(1): 64-71.

[4] Samantaray M, Barik C R, Biswal K A. A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems. Decision Analytics Journal, 2024: 11100478.

[5] Ali M A, Owais M S Q, Andleeb M S, et al. Robust genetic machine learning ensemble model for intrusion detection in network traffic. Scientific Reports, 2023, 13(1): 17227-17227.

[6] Nabi F, Zhou X. Enhancing intrusion detection systems through dimensionality reduction: a comparative study of machine learning techniques for cyber security. Cyber Security and Applications, 2024: 2100033.

[7] Arco M G J, Carrión M R, Gómez R A R, et al. Methodology for the Detection of Contaminated Training Datasets for Machine Learning-Based Network Intrusion-Detection Systems. Sensors, 2024, 24(2).

[8] Sarhan M, Layeghy S, Moustafa N, et al. Feature extraction for machine learning-based intrusion detection in IoT networks. Digital Communications and Networks, 2024, 10(01): 205-216.

[9] Ren Jiadong, Zhang Yafei, Zhang Bing, et al. A feature selection-based classification method for industrial internet intrusion detection. Computer Research and Development, 2022, 59(05): 1148-1159.