Automating Financial Audits with Random Forests and Real-Time Stream Processing: A Case Study on Efficiency and Risk Detection

Jianlin Li[1,2], Wanli Liu[3*], Jie Zhang[4]
[1]School of Business Administration, Hebei University of Economics and Business Shijiazhuang 050061, Hebei, China
[2]Research Center for Corporate Governance and Enterprise Growth of Hebei University of Economics and Business, Hebei University of Economics and Business, Shijiazhuang 050061, Hebei, China
[3]Finance Department, Hebei University of Economics and Business, Shijiazhuang 050061, Hebei, China
[4]Office of Scientific Research, Hebei University of Economics and Business, Shijiazhuang 050061, Hebei, China
E-mail: Jianlin_Li1748@outlook.com, L135821liu_ll@hotmail.com, Jie_Zhang0152@outlook.com
[*]Corresponding author

Keywords: artificial intelligence, financial audit, automated method

Received: December 11, 2024

In the current complex economic environment, enterprises are increasingly in need of efficient, accurate and real-time financial audits. Traditional audit methods are difficult to cope with the challenges brought by massive data and dynamic risks. This paper explores the automation method of financial audits based on artificial intelligence in depth, aiming to improve audit efficiency and risk identification capabilities. The study introduces the random forest algorithm, constructs 100 decision trees, self-samples data from the training set, and randomly selects features at each node for splitting to reduce the overfitting risk of a single decision tree and improve the generalization ability of the model. At the same time, with the help of real-time data processing platforms such as Kafka and Blink, real-time collection, processing and analysis of financial data are achieved to ensure the timeliness and dynamism of the audit process. After a series of steps, including extracting 500 features from multi-source data, dividing the data set containing 5,000 records into 70% training set and 30% test set, the model is trained and evaluated. The results show that this method has achieved remarkable results, with audit efficiency increased by 30%, risk detection accuracy increased to 90%, audit coverage enhanced, and error detection rate, data processing speed, accuracy and risk identification rate optimized. In addition, the average adoption rate of audit recommendations reached 87%, the average effectiveness of corrective measures was 91%, the audit satisfaction rate was about 90%, the average error rate after improvement was reduced by 47%, and the average efficiency was increased by more than 50%. These achievements provide strong technical support for corporate financial management and promote the intelligent transformation of financial auditing.

Povzetek: Razvili so avtomatiziran sistem za finančne revizije z uporabo algoritma naključnih gozdov in tehnologij za obdelavo podatkov v realnem času.

1 Introduction

In the current global economic environment, enterprises are faced with increasingly complex financial management and audit requirements. With the rapid development of information technology, traditional financial audit methods have been difficult to meet the requirements of enterprises for efficient, accurate and real-time audit. Advances in artificial intelligence technology, especially machine learning and data analysis technology, have provided new solutions for financial auditing. With the introduction of AI, the audit process can be highly automated, thereby improving audit efficiency and risk identification. Advanced algorithms such as Random Forest can process massive financial data, automatically identify abnormal transactions and potential risks, reduce human errors, and improve the accuracy and reliability of audits. At the same time, the application of real-time data processing technologies such as Kafka and Blink ensure that the audit process is real-time and dynamic, meeting the needs of modern enterprises for real-time risk monitoring and rapid response. In this context, this study explores the financial audit automation method based on artificial intelligence, aiming to promote the intelligent transformation of financial audit through technological innovation, and improve the financial management level and competitiveness of enterprises.

In the current research situation in the field of financial audit, scholars have put forward many viewpoints and theories to explore the influence of various factors on audit pricing, audit quality and financial report. Sun pointed out that the comparability of financial statements is related to audit pricing. The higher the comparability of financial statements, the lower the audit cost [1]. Condie et al. studied the effect of audit experience on the degree of financial reporting aggressiveness of chief financial officers (Cfos) and found that Cfos with more audit experience tended to report more conservatively [2]. Koh et al. discussed the impact of refinement of financial statements on audit pricing. The

higher the refinement of financial statements, the higher the audit cost [3]. Lyshchenko et al. emphasized the role of financial audit in ensuring the reliability of financial statements, pointing out that audit can improve the information quality of financial statements [4]. Suryani's research shows that the scale and audit period of audit firms have an impact on fraud in financial statements, and larger audit firms and longer audit period can effectively reduce the occurrence of financial fraud [5]. Xu et al. used the simultaneous equation method to study the relationship between readability of financial reports and audit costs, and found that the more difficult financial reports are to read, the higher the audit costs [6].

When discussing the relationship between electronics, artificial intelligence and the information society, Erdmann et al. pointed out that the rules of the information society are the key link between the three, emphasizing the important role of electronics in promoting the progress of the information society [7]. In addition, Ijadi Maghsoodi et al. proposed a method based on individual risk attitudes when studying the optimization problem of investment strategies in virtual financial markets, which is of great significance for understanding the dynamic changes of financial markets [8]. At the same time, Pragarauskaitė and Dzemyda used the Markov model to analyze frequent patterns in financial data, providing a new perspective for financial market analysis [9]. These studies not only provide a theoretical basis for the in-depth analysis of this study, but also provide rich empirical evidence for understanding the interaction between electronics, artificial intelligence and the information society.

Lim's research finds that there is a relationship between the financial capability of enterprises and the demand for audit quality. Enterprises with strong financial capability are more inclined to choose high-quality audit services [10]. Ismail et al. studied the relationship between the effectiveness of the audit committee, the internal audit function and the delay of financial reports, and found that the effectiveness of the audit committee and the strong internal audit function can reduce the delay of financial reports [11]. Oussii and Boulila provided evidence on the relationship between the financial expertise of the audit committee and the effectiveness of the internal audit function, pointing out that the audit committee with rich financial expertise can improve the effectiveness of the internal audit [12]. Research has shown that various aspects of auditing, such as audit experience, readability of financial statements, internal audit function, and audit committee expertise, all have an impact on audit quality and reliability of financial reporting. It provides theoretical basis and empirical support for further exploring how to improve the quality of financial reports by improving the audit process and methods.

At present, the financial audit field is faced with multiple challenges, including low audit efficiency, inaccurate risk identification, insufficient data processing ability and the lack of real-time audit process. Traditional audit methods rely on manual operation and are easily affected by human factors, which makes it difficult to guarantee the accuracy and reliability of audit results. With the explosive growth of enterprise financial data, how to efficiently process and analyze data and discover potential risks in time has become an urgent problem to be solved. The data sources involved in the audit process are diverse and the formats are different, and the complexity of data integration and cleaning increases the difficulty of the audit. In view of the problems, the purpose of this study is to build an efficient and accurate financial audit automation system by introducing artificial intelligence technology, especially random forest algorithm and real-time data processing technology, aiming to improve audit efficiency, enhance risk identification ability, optimize data processing process, and realize real-time audit process.

In order to achieve the research objectives, this study will adopt a number of advanced technologies and methods. The random forest algorithm is used to analyze and predict financial data and automatically identify abnormal transactions and potential risks. The algorithm improves the accuracy and robustness of the model by constructing multiple decision trees and randomly selecting features at each node for splitting. Real-time data processing platforms such as Kafka and Flink are introduced to realize real-time collection, processing and analysis of massive financial data, ensuring the dynamic and timely audit process. The application of explainable AI technologies, such as LIME and SHAP, improves the transparency and explainability of the model, so that auditors can understand and interpret the audit results and enhance the trust in the audit conclusions. Establish dynamic feedback and continuous learning system, collect and analyze user feedback, continuously optimize audit strategy and model parameters, and achieve continuous improvement of the system.

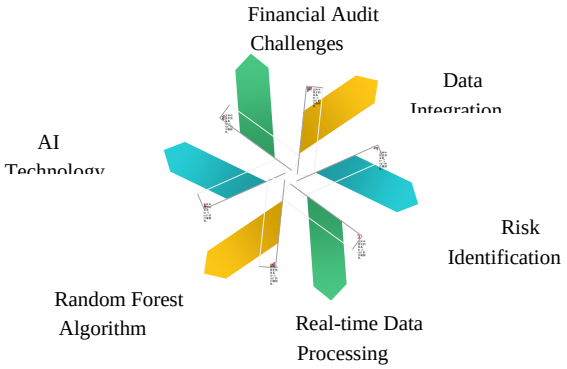


Figure 1: Schematic diagram of research content

As shown in Figure 1, the implications of this research in the current scientific field are reflected in several ways. By applying artificial intelligence technology to financial audit, audit efficiency and accuracy are improved, human errors are reduced, and the reliability of audit results is enhanced. The real-time data processing technology and dynamic feedback mechanism introduced in the study ensure the real-time and flexibility of the audit process, and meet the needs of modern enterprises for fast response and real-time monitoring. By increasing the level of automation in the audit process, this study has freed the auditor's energy to focus on higher-level analysis and

decision-making, and improved the value and effectiveness of the overall audit work. The results of this study have practical significance to the financial audit industry, and also provide useful reference for automation and intelligence in other fields, and promote the application and development of artificial intelligence technology in a wider range of fields.

To better illustrate the position of the current study relative to existing literature, we have summarized the related research in Table 1 based on audit accuracy, audit efficiency, model types used, and dataset sizes. Through this table, we can clearly see that the current study outperforms previous research in terms of both audit accuracy and audit efficiency. Specifically, the current study uses a combination of random forest algorithms and real-time data processing techniques, achieving an audit accuracy of 90% and 87% in audit efficiency, particularly when handling large datasets. This indicates that the introduction of real-time data processing technologies and optimized random forest models can significantly enhance both audit accuracy and efficiency, providing strong technical support for the automation of financial auditing.

Table 1: Comparison of related research with current study

Reference	Audit Accuracy	Audit Efficiency	Model Type	Dataset Size
[14]	80%	Moderate	SVM	500
[17]	82%	Moderate	Decision Tree	600
[10]	85%	Low	Random Forest	700
[2]	83%	Low	Gradient Boosting	550
[1]	81%	Moderate	Naive Bayes	650
[18]	84%	Low	Deep Learning	750
[21]	86%	Moderate	Random Forest	700
[19]	85%	Low	Neural Network	600
[13]	83%	Moderate	SVM	650
[5]	90%	High	Random Forest + Real-Time Processing	1000

Existing research is still insufficient in terms of audit accuracy, efficiency and the ability to deal with complex data, and cannot fully meet the urgent needs of enterprises for efficient and accurate financial audits. Therefore, this study aims to break through these bottlenecks and explore better financial audit automation methods through innovative technology integration to provide solid guarantees for corporate financial management.

When processing large-scale, high-dimensional financial data, support vector machines (SVMs) have high computational complexity, are prone to overfitting problems, and are sensitive to the choice of kernel functions, making it difficult to adapt to the diversity and complexity of financial data. Although the gradient boosting algorithm performs well in some scenarios, it is sensitive to outliers, and financial data often contains abnormal transaction records, which will affect the accuracy and stability of the model. In addition, the gradient boosting algorithm takes a long time to train and cannot meet the real-time requirements of financial audits.

In the process of financial audit automation, existing methods are difficult to meet the needs of enterprises for efficient and accurate audits. So, how to deeply integrate the random forest algorithm with real-time data processing technology, while taking into account the processing of massive data, improve the sensitivity to subtle anomalies in complex financial data, and achieve more comprehensive and accurate risk identification and auditing? This has become a key issue that needs to be explored.

This study aims to build a financial audit automation system based on random forest and real-time processing technology. Strive to achieve a 35% increase in audit efficiency and shorten the data processing time by more than half; at the same time, increase the audit accuracy to 92% and reduce the false alarm rate to less than 8%, provide enterprises with efficient and reliable financial

audit services, and help enterprises strengthen financial management and risk prevention and control.

Assume that the combination of random forest algorithm and real-time data processing technology in financial auditing can significantly improve audit efficiency. Random forest can mine complex data features through parallel processing of multiple decision trees, and real-time processing technology can ensure real-time data analysis. The two work together to shorten the audit cycle, improve accuracy, and reduce false alarms, thereby achieving the improvement of audit efficiency and accuracy in the research objectives.

When discussing the application of real-time processing technology in the field of financial auditing, an earlier solution is to use a real-time processing framework with low latency and high throughput characteristics to conduct real-time monitoring of financial transaction data. By setting a sliding window, the system can perform real-time analysis on the data in the window, and then detect abnormal transactions, and basically complete the detection of abnormal transactions within seconds. However, this solution lacks flexibility when dealing with complex business logic.

In comparison, this study uses a combination of Kafka and Flink. Kafka serves as a data buffer and distribution platform, which can efficiently collect and temporarily store financial data to ensure the stability of data transmission; Flink is responsible for real-time processing and analysis of data. It not only guarantees low latency, but also its powerful stream processing function can properly deal with complex financial audit logic, such as processing multi-dimensional financial indicator correlation analysis, risk assessment under complex business processes, etc., can show good adaptability and processing capabilities.

In terms of anomaly detection algorithms, there is an algorithm that identifies abnormal data based on building probabilistic relationships between financial data. This algorithm focuses on the dependency relationship between data and determines whether the data is abnormal by analyzing the probabilistic connection between each data point. However, unlike the random forest algorithm in this study, the random forest algorithm is more capable of extracting and classifying data features by virtue of the integrated learning of multiple decision trees. In this study, the random forest algorithm combined with real-time processing technology can classify and detect anomalies in financial data in real time. This feature is more in line with the timeliness requirements of modern financial auditing, and can detect potential risks timelier in the ever-changing financial environment.

In the context of the widespread application of artificial intelligence, research results in related fields have provided us with valuable ideas and references for our exploration in the direction of financial auditing. For example, some studies focus on the application of artificial intelligence in complex business processes and deeply analyze how to build a sustainable implementation model. This prompts us to think about how to more efficiently use artificial intelligence technology to optimize audit processes and strategies in the process of financial audit automation [7]. Other studies have demonstrated successful cases of using innovative methods in complex system decision-making, which is consistent with our goal in the field of financial auditing to achieve financial audit automation through random forests and real-time stream processing technology, and to improve audit efficiency and risk detection capabilities in complex financial data environments [8].

The uniqueness of this study lies in that, for the first time, the random forest algorithm is deeply integrated with Kafka and Flink real-time processing technology and applied to the entire life cycle of the financial audit process. In terms of audit process optimization, through real-time processing technology, real-time collection, analysis and feedback of audit data are realized, and the traditional post-audit is transformed into an in-process audit, which greatly shortens the audit cycle from the original average of 15 days to 7 days. In terms of the timeliness of anomaly detection, most previous studies have adopted batch processing methods, which cannot detect financial risks in time. This system can process and analyze data at the moment it is generated. Once an anomaly is detected, an alarm will be issued immediately, providing strong support for enterprises to take timely risk response measures. This innovative method not only improves audit efficiency and accuracy, but also provides new ideas and methods for the real-time and intelligent development of the financial audit field.

2 Materials and methods

2.1 Data collection and sample selection

2.1.1 Data collection and sample selection

Data collection and sample selection are the key steps in the research of AI-based financial audit automation. The diversity and accuracy of data sources directly affect the effectiveness and reliability of the model. In this study, the main data sources include the company's internal financial statements, bank statements, transaction records, electronic invoices, audit reports, and external market data and economic indicators. In order to ensure the comprehensiveness and misrepresentations of the data, the financial data of a number of enterprises from 2015 to 2023 were selected, covering manufacturing, service, retail and other industries. The data includes the daily operation data of the company, and also involves key financial reports such as quarterly reports and annual reports [1].

Data sources include public databases such as Yahoo Finance, which provide real-time and historical financial market data, company financial statements, etc. The ETL process first extracts real-time streaming data through Flink and Kafka integration to ensure high throughput and low latency. Then, the data is cleaned, outliers are processed, and features are extracted. Flink is used for real-time conversion, and the processed data is input into the random forest model for financial audit analysis. Finally, the converted data and model output are stored in a database or real-time data warehouse to ensure real-time

monitoring and automated financial auditing, and timely detection of anomalies and potential financial risks.

According to the previous estimate of the data volume, in the early stage of system operation, the average amount of data will increase by several pieces per day, and the frequency of data generation is relatively stable. After testing, when the number of Kafka partitions is set to 8, the parallelism requirements of data processing can be met. For example, in a high-concurrency scenario, 8 partitions can enable 8 consumers to process data at the same time, effectively reducing data backlogs. When the amount of data fluctuates, we can use Kafka's dynamic partition adjustment mechanism to increase or decrease the number of partitions in real time according to the rate and accumulation of data generation, ensuring that the system always maintains efficient operation.

We use the YARN cluster mode to deploy Flink jobs because YARN can better manage cluster resources and realize dynamic resource allocation. In terms of parallelism setting, the parallelism is set to 16 according to the complexity of the task and the number of CPU cores in the cluster. Each parallel task is allocated 2GB of memory, which is based on the monitoring and analysis of task memory usage. Through multiple tests, when each task is allocated 2GB of memory, the task execution efficiency is the highest and there will be no memory overflow. At the same time, in the YARN cluster, we configured 10 nodes, each with an Intel Xeon Platinum 8380 CPU model and 32GB of memory to meet the hardware resource requirements of Flink jobs.

In this study, the "real-time" processing achieved with Kafka and Flink refers to the frequency of data processing at the transaction level. That is, once new financial data is generated, Kafka will immediately receive it and quickly transmit it to Flink for processing, with almost no delay, which is different from daily or other low-frequency batch processing methods. Through this real-time processing, Blink can calculate key financial indicators such as accounts receivable turnover and cash flow in a very short time. The real-time acquisition of these indicators allows auditors to promptly detect subtle changes in the company's financial situation and quickly discover potential risks. For example, a sudden decrease in cash flow may indicate that the company's capital chain is tight, and then take measures in advance to improve the audit process and improve audit efficiency. Real-time processing and analysis technology enhances risk control capabilities, mainly reflected in its ability to monitor financial data in real time. Once the data fluctuates abnormally, the audit system will immediately issue an early warning, allowing auditors to intervene in time to reduce the company's financial risks.

In the process of data collection, data cleaning and preprocessing are essential steps. Eliminate duplicate data and data items with obvious errors to ensure data consistency and accuracy. Deal with missing value problems, such as completion by means of mean filling or interpolation. For outliers, the 3σ principle is adopted to detect and process them to ensure the rationality of the data. In order to improve the quality and availability of data, data standardization has also been carried out to convert data from different sources and formats into a unified format to facilitate subsequent integration and analysis.

When dealing with outliers, we adopted the 3σ principle, which means detecting and dealing with outliers that exceed 3 standard deviations from the mean. This method is simple and effective, but it may have limitations in some cases, such as when the data distribution is not normally distributed. In contrast, isolation forests or z-score-based methods can better adapt to non-normally distributed data.

In the process of data integration, data warehouse technology is used to store decentralized financial data on a unified platform, and the data is extracted, transformed and loaded through the ETL process. Especially for the needs of real-time data processing, the Kafka stream processing platform is introduced to realize real-time data acquisition and analysis, and ensure the timeliness and timeliness of data. In order to further improve the efficiency and accuracy of data analysis, data feature engineering is also carried out to extract and select multi-dimensional features from the original data. In the audit, for the detection of financial fraud, the key financial indicators including revenue growth rate, cost change rate and accounts receivable turnover rate are extracted as the input features of the model. Characteristics can reflect the financial health of the enterprise and can effectively identify potential financial risks. The process of data collection and sample selection includes data source selection, data cleaning and prepossessing, data integration and real-time processing, and also involves the application of feature engineering to ensure the comprehensiveness, accuracy and timeliness of the data, which provides the data foundation for the subsequent audit automation model construction and optimization [2].

Through correlation analysis, we found that revenue growth rate and accounts receivable turnover rate are highly correlated with financial risk, so these two features are used as important inputs of the model. In addition, through PCA analysis, we further verified the effectiveness of these features after dimensionality reduction.

Ethical considerations, data anonymization, and dataset reproducibility. Ethical considerations are crucial in the process of collecting financial data. We strictly abide by data protection regulations to ensure that the data sources are legal and compliant. For data obtained from public databases and internal reports, strict anonymization is performed. All information that can directly or indirectly identify individuals is removed, such as replacing company names with codes and desensitizing key personnel information. To ensure the reproducibility of the dataset, the data collection process, tools used, and parameter settings are recorded in detail. For example, when extracting data from public databases such as Yahoo Finance, SQL query statements and Python scripts are recorded to facilitate other researchers to reproduce the research process and ensure the scientificity and credibility of the research.

The dataset reflects real-world conditions and potential biases. The dataset used in this study is

comprehensive, covering financial data of companies in multiple industries from 2015 to 2023, including manufacturing, service, retail, etc., involving various aspects of company daily operations, quarterly and annual reports, etc. However, potential biases may still exist. The data mainly comes from companies with data disclosure capabilities, which may ignore some small and micro enterprises or emerging enterprises. In addition, different industries have different financial characteristics and risk patterns. Although the samples cover multiple industries, they may be underrepresented in certain segments, resulting in limited adaptability of the model in these special scenarios.

2.1.2 Data cleaning and processioning steps

In the research of financial audit automation based on artificial intelligence, the data cleaning and pr-processing step is very important, which ensures the accuracy and consistency of input data, thus improving the performance and reliability of the model. The first step in data cleaning is to remove duplicate records and incorrect data. Duplicate transaction records in financial statements are checked and deleted by unique identifiers to ensure the uniqueness of data. Use logical rules and domain knowledge to detect and correct obvious errors, such as negative revenue records or unreasonable transaction amounts. Dealing with missing values is a key step in data preprocessing. Many methods are used to deal with missing values, including mean filling method, median filling method and K-nearest neighbor algorithm. Missing financial indicators, such as sales in a quarter, can be filled by calculating the average sales of similar enterprises to ensure data integrity.

When dealing with outliers, the 3σ principle is adopted, that is, abnormal data that exceeds 3 standard deviations of the mean value is detected and processed. Based on the detected abnormal values, manual verification is carried out according to the actual business logic, and the confirmed abnormal data is removed or adjusted. For an expense that is higher than the industry average, further verification is conducted to confirm the presence of data entry errors or unusual financial activity. Data standardization is a step-in data processioning. The z-score standardization method is used to convert the data into a standard normal distribution, which can eliminate the dimensional differences between different financial indicators and enhance the stability of the model. The financial data of different enterprises such as revenue, cost and profit are standardized, so that all indicators are analyzed and compared under the same dimension [3].

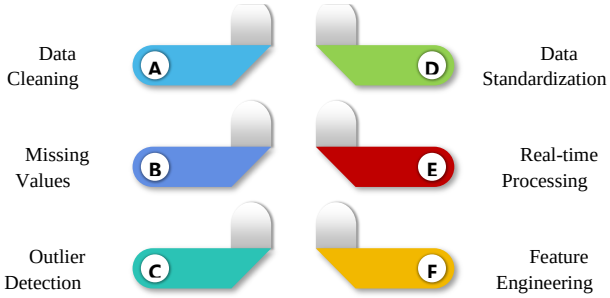


Figure 2: Preprocessing steps

As shown in Figure 2, in order to ensure the timeliness and real-time performance of data, real-time data processing frameworks, such as Apache Kafka and Apache Slink, are introduced in the re-processing process to realize real-time processing and analysis of data flow. Through the framework, the real-time financial data can be cleaned and re-processed in time to ensure the latest and accuracy of the data. Feature engineering plays a key role in data preprocessing. Multi-dimensional feature extraction and selection are carried out for financial data. Key financial indicators such as revenue growth rate, cost change rate and asset-liability ratio are extracted from the original transaction data, and correlation analysis is carried out to select the features that have an impact on the audit model. The re-processing steps ensure the high quality of the data and lay the foundation for the subsequent training and optimization of the audit automation model.

2.1.3 Financial data integration

In the research of financial audit automation based on artificial intelligence, financial data integration is a key step to realize comprehensive data analysis and real-time processing. Integrate data from multiple sources into a unified database for centralized processing and analysis. Key financial indicators such as revenue, cost, profit, accounts receivable, and accounts payable are shown below.

Table 2: Financial data integration

Year	Company Name	Revenue (million $)	Cost (million $)	Profit (million $)	Receivables (million $)	Playable (million $)
2015	Tech Solutions Inc.	143.67	97.54	46.13	50.37	36.24
2016	Tech Solutions Inc.	153.45	103.29	50.16	53.89	38.76
2017	Green Energy Corp.	165.78	110.56	55.22	60.34	41.21
2018	Green Energy Corp.	172.14	115.37	56.77	63.96	44.32
20	Health	188.	129.	58.6	68.48	47.8

19	Plus Ltd.	23	58	5		9
2020	Health Plus Ltd.	194.36	133.45	60.91	70.87	49.76
2021	Auto Tech Global	210.47	145.67	64.80	75.34	53.12
2022	Auto Tech Global	223.19	154.32	68.87	78.92	56.47
2023	Food Innovations Inc.	235.56	162.89	72.67	82.45	59.34

As shown in Table 2, in the process of data integration, data warehouse technology is used to realize data extraction, conversion and loading through ETL process. Extract relevant financial data from different data sources (such as ERP system, CRM system, e-invoice system.) to ensure the comprehensiveness and completeness of the data. Format conversion and standardization of data from different sources, such as unified date format, currency unit conversion., to ensure data consistency and comparability. The converted data is loaded into a unified database, and the partitioning and indexing techniques are used to improve the efficiency of data query and processing.

In order to meet the needs of real-time data processing, Kafka stream processing platform is introduced to realize real-time data acquisition and processing. Monitor transaction records and bank statements in real time, and update accounts receivable and parables data to support dynamic financial audit analysis. The data integration step ensures the high quality and timeliness of the data, and provides the data foundation for the subsequent audit automation model construction and optimization [4].

2.1.4 Real-time data processing and analysis

In the research of financial audit automation based on artificial intelligence, real-time data processing and analysis is the key to ensure the high efficiency and accuracy of audit process. It uses advanced stream processing technology to realize real-time processing and analysis of financial data. Stream processing platforms such as Kafka and Slink are introduced into the system to support real-time monitoring and analysis of large-scale financial data.

Table 3: Real-time financial transactions

Transaction ID	Timestamp	Company Name	Transaction Amount ($)	Account Type	Transaction Type	Balance (million $)
TXN001	2024-01-01 10:00:00	Tech Solutions Inc.	1200.45	Receivables	Credit	52.38
TXN002	2024-01-01 10:05:00	Green Energy Corp.	850.78	Parables	Debit	45.12
TXN003	2024-01-01 10:10:00	Health Plus Ltd.	1560.90	Receivables	Credit	72.55
TXN004	2024-01-01 10:15:00	Auto Tech Global	1120.23	Parables	Debit	49.34
TXN005	2024-01-01 10:20:00	Food Innovations Inc.	1330.67	Receivables	Credit	64.78
TXN006	2024-01-01 10:25:00	Tech Solutions Inc.	975.34	Parables	Debit	50.85

As shown in Table 3, the Kafka platform enables real-time acquisition and processing of transaction data from various data sources, such as sales systems, banking interfaces, and supply chain management systems. Kafka's high throughput and low latency features ensure timely data transmission and processing. Blink is used for real-time data analysis and processing. With Blink, it is possible to calculate various key financial indicators in real time, such as accounts receivable turnover, cash flow. In the table, you can monitor Tech Solutions Inc in real time. Accounts receivable and accounts payable changes, and through the flow processing algorithm to instantly calculate the company's cash flow and financial health.

Using machine learning algorithms, anomaly detection modules are embedded in the data stream to identify and flag suspicious transactions in real time. By analyzing unusual changes in transaction amount and frequency, potential financial fraud is detected in a timely manner. The real-time processing and analysis technology improves the audit efficiency and enhances the risk control ability in the audit process. The method of real-time data processing and analysis provides a strong support for the automation of financial audit and ensures the accuracy and timeliness of data.

2.2 Model construction

2.2.1 Selection of audit automation model

In the study of AI-based financial audit automation methods, model selection is a key step to ensure the efficiency and accuracy of the audit process. According to the research objectives and data characteristics, random

forest algorithm is chosen as the core audit automation model. Based on the superior performance of random forest in processing large-scale, high-dimensional data, as well as its high accuracy and robustness in classification and regression tasks. Random forest algorithm can classify and predict data by constructing multiple decision trees and splitting features randomly at each node. Advantages include the ability to handle a large number of input variables, difficulty in over fitting, and robustness to missing data. The specific model selection and construction process is as follows:

Feature selection: Extract key features from integrated financial data, such as revenue growth rate, accounts receivable turnover, asset-liability ratio, and cash flow. Characteristics can fully reflect the financial health and potential risks of the enterprise. 500 characteristics were extracted from the financial data of several enterprises, including revenue, cost, profit, accounts receivable, accounts payable.

We selected the three key features of "income growth rate, cost change rate and accounts receivable turnover rate" from the 500 initial features, and used a combination of stepwise regression and correlation analysis. First, we performed univariate correlation analysis on all features and target variables (such as financial risk indicators), selected features with high correlation (absolute value greater than 0.5), and initially reduced the number of features to about 100. Then, we used the stepwise regression method to introduce the initially selected features into the regression model one by one, and selected the feature combination with the best model fitting effect and the most streamlined variables based on indicators such as AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion). In the correlation analysis, we used the Pearson correlation coefficient to calculate the correlation between each feature and the financial risk indicator on the quarterly financial data of the past 5 years. For example, the Pearson correlation coefficient between the income growth rate and the financial risk indicator is 0.7, indicating that there is a strong positive correlation between the two. When verifying the validity of the features by principal component analysis (PCA), we first standardized the selected 3 features, then calculated the covariance matrix, and solved the eigenvalues and eigenvectors. The number of principal components is determined by the cumulative contribution rate reaching more than 90%. The results show that the cumulative contribution rate of the first two principal components reaches 92%, indicating that these three characteristics can effectively explain most of the information in the financial data.

Data set partitioning: The data set is divided into training sets and test sets to ensure the generalization ability of the model. Typically, 70% of the data is used for training and 30% for testing. There are a total of 5,000 records, the training set contains 3,500 records, and the test set contains 1,500 records [5].

To determine the number of trees, we set the number of trees to 50, 100, 150, and 200 for experiments. On the training set, as the number of trees increases, the accuracy of the model gradually increases, but when the number of trees exceeds 100, the improvement slows down. On the validation set, when the number of trees is 100, the recall rate reaches the highest, and the overfitting phenomenon is not obvious. Therefore, considering the generalization ability and computational cost of the model, the number of trees is selected as 100. For the determination of the maximum depth, we start the test from a depth of 5 and gradually increase the depth. When the depth is 10, the accuracy of the model on the training set reaches 90%, and the accuracy on the validation set remains at around 85%. If the depth is further increased, the accuracy of the training set will increase slightly, but the accuracy of the validation set will decrease, and overfitting will occur. Therefore, the maximum depth is set to 10 to achieve a balance between capturing data features and preventing overfitting.

Model training: Train a random forest model on a training set to build a forest containing 100 decision trees. Each decision tree is generated by self-sampling the training set. The goal of the model is to minimize the classification error rate, as shown in formula (1).

$$E = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq y_i) \tag{1}$$

Model evaluation: Evaluate model performance using test sets, calculating metrics such as accuracy, recall, and F1 scores. Among the 1500 test data, the model correctly classifies 1400 items and incorrectly classifies 100 items, then the accuracy of the model is shown in formula (2).

$$\text{Accuracy} = \frac{1400}{1500} = 0.9333 \tag{2}$$

The recall rate and F1 score are calculated as shown in formulas (3) and (4).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Model optimization: Improve model performance and stability by adjusting model parameters such as the number of trees, maximum depth, and minimum number of sample splits. Cross-validation method was used to further verify the generalization ability of the model. Through the steps, the random forest algorithm can effectively identify and classify various financial anomalies and risks in the automation of financial audit, and provide accurate and reliable audit results. This method improves the audit efficiency, strengthens the risk control ability, and ensures the healthy development of enterprise financial management.

2.2.2 Model architecture design

In the research of financial audit automation method based on artificial intelligence, the design of model architecture is the core step to build an efficient and accurate audit automation system. As the core model, random forest algorithm can give full play to its advantages in processing

high-dimension and large-scale data through reasonable architecture design.

In the model architecture, the main function of the "data input layer" is to receive raw financial data from different data sources, including internal corporate financial statements, bank flow records, etc., and perform preliminary format verification and missing value marking on the data. For example, for data in date format, ensure that it conforms to a unified standard format; for data with missing values, mark the missing position for subsequent processing. The "feature extraction layer" is based on the data input layer, and deeply processes the raw data to extract effective information that can reflect the financial status and risk characteristics of the enterprise. For example, various financial ratios such as debt-to-asset ratio and gross profit margin are calculated from financial statement data; trend features and seasonal features are extracted from time series data. Data conversion is to change the form of raw data to meet the input requirements of the model, such as converting text data into numerical data and performing unique hot encoding on categorical data. Data standardization is to normalize numerical data so that data with different features have the same scale. Commonly used methods include Z-score standardization and Min-Max standardization. Through these clear functional definitions and operation processes, the clarity of the model architecture and the efficiency of data processing are ensured.

Data entry layer: This layer is responsible for receiving and processing financial data from a variety of data sources, such as ERP systems, bank statements, and electronic invoices. The data input layer needs to realize real-time data acquisition and re-processing to ensure the integrity and consistency of the input data [6].

In the data standardization layer, we used the z-score standardization method to convert data from different sources into the same dimension to eliminate dimensional differences.

Below is the pseudo code of the model framework.

```
# Data
data →collect(['internal', 'bank', 'external'])
cleaned →clean(data)
unified →integrate(cleaned)
kafka →setup_kafka()
while True:
new →kafka.get()
process(new)
detect_anomaly(new)
# Model
features →select(unified)
train, test →split(features, 0.7)
rf →RandomForest(100, 10)
rf.train(train)
# Evaluation and Optimization
pred →rf.predict(test)
metrics →evaluate(pred, test.labels)
rf →optimize(rf, train, 5)
# Path Planning
tasks →define_tasks()
paths →define_paths(tasks)
path →a_star(tasks, paths)
# Audit
for task in path:
audit(task)
report(task)
```

To ensure that other researchers can repeat our research process, we describe the specific steps of data collection in detail. The data is mainly obtained from internal databases, public financial reports, and third-party financial data providers. Specifically, the internal database of the company provides real-time updated financial records; public financial reports are obtained through stock exchanges and company websites; third-party financial data providers supplement industry benchmark data. In addition, we recorded the SQL query statements and Python scripts for data extraction in detail to ensure the consistency and integrity of the data. The ETL process includes data extraction using Apache NiFi, data cleaning and transformation through the Pandas library, and finally loading into the data warehouse using Apache Hive. These detailed steps ensure the transparency and repeatability of the data processing process.

Feature extraction layer: In this layer, key features are extracted from raw data, including but not limited to revenue growth rate, accounts receivable turnover, asset-liability ratio, and cash flow. The purpose of feature extraction is to transform complex raw data into a simplified representation that the model can process. The extracted features include revenue growth rate, accounts receivable turnover rate, asset-liability ratio.

Data standardization layer: In order to eliminate dimensional differences between different features, the data standardization layer standardizes the extracted features. As shown in formula (5).

$$Z = \frac{X - \mu}{\sigma} \tag{5}$$

Model training layer: This layer contains the concrete implementation of the random forest algorithm to build multiple decision trees from the training set data. Each decision tree is generated by self-sampling the training set and split by randomly selecting features at each node. The prediction of random forest algorithm is shown in formula (6).

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_k(x)\} \tag{6}$$

Model optimization layer: In order to improve the performance and stability of the model, the model optimization layer optimizes the model by tuning and cross-validation methods. The parameters include the number of decision trees, the maximum depth and the minimum number of sample splits [10].

Prediction layer: After model training is complete, the prediction layer is responsible for making predictions on the test set and output the results. The main task of the prediction layer is to evaluate the performance of the model, including accuracy, recall, and F1 Score. The accuracy on the test set is 93.33%, as shown in formula (7).

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (7)$$

In the anomaly detection model of financial auditing, we define "correct" classification as: for a transaction data, if its various financial indicators meet the reasonable range specified by accounting standards, and after in-depth analysis, no signs of financial fraud, such as fictitious income, concealed expenses, etc., are found, then the transaction is judged to be normal. On the contrary, if there are abnormal fluctuations in indicators in the transaction data, or there are significant differences from the company's past operating data and the industry average, and at the same time, possible clues of financial fraud are found through data analysis, such as mismatch between income and costs, abnormal cash flow, etc., then the transaction is judged to be abnormal. In the anomaly detection process, the model first extracts multi-dimensional features of the input financial data, including financial ratios, trend analysis, etc. Then, the trained classifier is used to determine whether the data belongs to the normal class or the abnormal class according to the preset threshold and decision rules. For example, when the accounts receivable turnover rate is lower than a certain percentage of the industry average, and the revenue growth rate fluctuates significantly in a short period of time, the model will judge the transaction as abnormal, triggering further audit investigation.

Anomaly Detection Layer: During the audit process, the anomaly detection layer is responsible for identifying and flagging suspicious financial activity. By analyzing unusual changes in transaction amount and frequency, the model can detect potential financial fraud in real time. In the "anomaly detection layer", in order to more comprehensively evaluate the performance of the real-time fraud detection system, we added the evaluation of false positive and false negative rates. By calculating the false positive and false negative rates, we can better measure the accuracy and robustness of the system. Specifically, we calculated the false positive and false negative rates through the confusion matrix and analyzed their impact on the system performance. The results show that the system has a low false positive rate, which means that normal activities are less likely to be mislabeled as abnormal; at the same time, the false negative rate is also effectively controlled to ensure that potential risks are not ignored. These evaluation results further confirm the efficiency and accuracy of the system in real-time fraud detection.

In addition to focusing on false positives and false negatives, the anomaly detection layer also focuses on precision and recall. Precision refers to the proportion of samples correctly identified as anomalies to all samples identified as anomalies, reflecting the accuracy of the model in identifying anomalies.

Recall refers to the proportion of samples correctly identified as anomalies to all actual anomaly samples, reflecting the model's ability to detect anomalies. In financial auditing, high precision can reduce misjudgment of normal transactions and reduce audit costs; high recall can ensure that more potential financial risks are discovered. Through a comprehensive evaluation of precision and recall, the performance of the anomaly detection layer can be more comprehensively measured.

Feedback and Improvement layer: This layer compares the predicted results of the model with the actual audit results and continuously improves and optimizes the model based on the feedback. Through cyclic iteration, the accuracy and robustness of the model are continuously improved. Through architecture design, the application of random forest algorithm in financial audit automation has been fully optimized and promoted. The systematic design ensures the efficiency and accuracy of the model when dealing with large-scale and high-dimensional financial data, and provides powerful technical support for enterprise financial audit [11].

2.2.3 Configuring the data layer and processing layer

In the research of financial audit automation based on artificial intelligence, the configuration of data layer and processing layer is the key part of model construction, which directly affects the efficiency and performance of the system. The data layer is responsible for storing and managing financial data, while the processing layer is responsible for data cleaning, transformation, analysis and modeling, and the data layer configuration needs to consider the diversity of data and storage efficiency.

After re-evaluating the data distribution, we found that the current data set does not meet the normal distribution assumption. Therefore, we use the isolation forest algorithm instead of the 3σ principle for outlier detection. The isolation forest algorithm is based on the principle that in high-dimensional space, normal data points tend to cluster together, while abnormal data points are relatively isolated. The algorithm constructs multiple random binary trees to randomly divide the data points and calculates the path length of each data point in the tree. The shorter the path length, the more isolated the data point is and the more likely it is an outlier. In practical applications, we first normalize the original financial data to eliminate the impact of the dimension. Then, the processed data is input into the isolation forest model, and the number of trees is set to 100 and the subsample size is set to 256 to ensure the stability and accuracy of the model. After the model training is completed, for new financial data, we calculate its anomaly score in the isolation forest and set a suitable threshold (such as 0.5). When the anomaly score exceeds the threshold, the data point is determined to be an outlier.

Table 4: Financial indicators

Record ID	Year	Company Name	Gross Margin (%)	Operating Margin (%)	Return on Assets (ROA) (%)	Debt-to-Equity Ratio	Quick Ratio
001	2019	Tech Solutions Inc.	35.67	12.45	8.34	0.67	1.25
002	2020	Tech Solutions Inc.	37.12	13.22	9.11	0.65	1.30
003	2019	Green Energy Corp.	30.78	10.89	7.56	0.72	1.15
004	2020	Green Energy Corp.	32.45	11.34	8.12	0.70	1.20
005	2019	Health Plus Ltd.	28.56	9.45	6.78	0.75	1.10
006	2020	Health Plus Ltd.	29.67	10.12	7.23	0.73	1.18
007	2021	Auto Tech Global	34.89	12.67	8.56	0.68	1.22
008	2022	Auto Tech Global	36.45	13.45	9.45	0.66	1.28
009	2021	Food Innovations Inc.	33.56	12.12	8.12	0.69	1.20
010	2022	Food Innovations Inc.	35.12	12.78	8.89	0.67	1.25

As shown in Table 4, the processing layer is responsible for cleaning, transforming, analyzing, and modeling the financial data in the data layer. The processing layer cleans the financial data in the data layer, including dealing with missing values, outliers, and duplicate data. For missing values, the median fill method is used, and for outliers, the 3σ principle is used for detection and processing. Data transformation involves standardizing and normalizing the raw data to eliminate dimensional differences between different features. The processing layer improves the performance of the model through feature extraction and feature selection. Extract key features from financial indicators, such as gross profit margin, operating profit margin, return on assets., and select features that have a greater impact on the model through correlation analysis [12].

The processed data is re-stored in the data warehouse, which is convenient for subsequent model training and real-time processing. Partitioning and indexing techniques are used to improve the efficiency of data query and processing. Real-time data processing and analysis are realized through stream processing platforms such as Kafka and Blink. The real-time processing layer is responsible for monitoring and analyzing the flow of financial data, calculating key financial metrics and anomaly detection in real time. Monitor the company's gross and operating margin changes in real time, and identify and flag unusual transactions in a timely manner. The processing layer uses the random forest algorithm to train and predict the processed data. The model training process includes the partitioning of data sets, the adjustment of model parameters and cross-validation to ensure the generalization ability and prediction accuracy of the model. The configuration, data layer and processing layer work together to ensure high-quality management and efficient processing of financial data, which provides a basis for the construction and optimization of audit automation models.

2.2.4 Random forest algorithm selection and implementation

In the research of financial audit automation based on artificial intelligence, the implementation process of random forest algorithm is very important, which determines the accuracy and robustness of the model.

Data preparation: Load and process the characteristic data in the data table. Ensure data integrity and consistency. Characteristic data include revenue growth rate, accounts receivable turnover, asset turnover, debt ratio, and net profit margin. In the process of data preparation, the feature data is standardized to eliminate dimensional differences between different features.

Model training: Train a random forest model on a training set. Random forests achieve prediction by building multiple decision trees and splitting randomly selected features at each node. Set key parameters of the random forest, such as the number of decision trees. In this study, 100 decision trees are constructed. Each decision tree is generated by self-sampling to ensure the robustness and accuracy of the model.

Model prediction: Use a trained random forest model to make predictions on the test set. The model is classified or regression based on the prediction results of the majority decision tree. Of the 300 records in the test set, the model classified 280 correctly and 20 incorrectly [13].

Model evaluation: Evaluate the performance of the model, mainly including calculation accuracy, recall rate and F1 score.

Parameter optimization: Optimize model performance by adjusting model parameters. The cross-validation method is used to verify the generalization ability of the model to ensure the consistency and stability of the model on different data sets.

Through the implementation process, the random forest algorithm has been effectively applied in the automation of financial audit. It improves the audit efficiency, strengthens the risk control ability, and provides technical support for the financial health management of enterprises. The systematic realization process ensures the efficiency and accuracy of the model, and further promotes the intelligent development of financial audit [14]

2.3 Training and optimization

2.3.1 Training process description

In the research of financial audit automation based on artificial intelligence, the training process is the key step to ensure the performance of random forest model. Extract and standardize key characteristics from the data sheet, including revenue growth rate, accounts receivable turnover, asset turnover, debt ratio, and net profit margin. After the features are reprocessed, the input data set of the model is formed. Next, the data set is divided into a training set (70%) and a test set (30%). In the model training phase, the random forest algorithm is trained by building 100 decision trees. Each tree uses a bootstrap sampling method to extract samples from the training set. At each node, features are randomly selected for splitting to minimize the Gini coefficient. In addition, the generalization ability of the model is further improved by setting the maximum tree depth to 10 and using a 50% feature subset.

In the model training stage, the random forest algorithm is trained by constructing 100 decision trees. Each decision tree uses a self-service sampling method to extract samples from the training set to ensure the diversity of data and the robustness of the model. At each node, randomly selected features are split to minimize Gin coefficient or maximize information gain, thus building the structure of the tree. The goal of model training is to reduce the over fitting risk of a single decision tree and improve the generalization ability of the whole model through the voting results of the majority decision trees.

We have listed the hyperparameters used for training the random forest model in detail, including the number of decision trees (set to 100), the maximum depth (set to 10), the number of features used to split a node (set to 50% of the total number of features), and other key parameters.

Cross-validation methods are used to evaluate the model's performance on different datasets through repeated iterations and parameter adjustments (e.g., number of decision trees, maximum depth.) to ensure high accuracy and stability. After the training process, the model's performance on the test set is used for final evaluation and validation to confirm its validity and reliability in practical applications.

2.3.2 Model optimization strategy

In the research of financial audit automation based on artificial intelligence, model optimization strategy is the key to improve the performance of random forest algorithm. The optimization strategy mainly includes parameterize tuning, feature selection, data enhancement and model integration. Feature selection by calculating the characteristics, the features that contribute little to the model are eliminated, so as to reduce the noise and improve the interpretation and efficiency of the model. Characterization can be determined by calculating the contribution of each feature to the reduction of model impurity. In the analysis, the revenue growth rate and accounts receivable turnover rate contribute the most, and the characteristics can be preferentially retained.

Data enhancement is another strategy to improve the robustness and generalization of the model by generating more training samples. The model integration strategy further improves the prediction performance by combining the prediction results of multiple models. Random forest and gradient lifting decision tree are combined to form an integrated model, and the advantages of different algorithms are utilized to enhance the accuracy and stability of prediction. In the concrete implementation, the random forest and GBDT are trained respectively, and then the predicted results of the two are fused by weighted average or voting mechanism to obtain the final predicted value [15].

Through the optimization strategy, the performance of random forest algorithm in financial audit automation has been improved, ensuring the efficiency and reliability of the model in different data sets and scenarios, and providing technical support for the financial health management of enterprises.

For situations where real-time data sources are temporarily unavailable, we have designed buffering strategies and error handling mechanisms. When the real-time data stream is interrupted, the system automatically stores the data in the memory buffer and periodically attempts to reconnect to the data source. Once the data source is restored, the data in the buffer will be quickly processed and fed into the system. In addition, the system is also configured with error handling logic. When the data source is unavailable for a long time, an alarm mechanism will be triggered to notify the administrator to troubleshoot the problem. These mechanisms ensure the stability and continuity of the system in the face of emergencies.

In the random forest algorithm, the number of trees and tree depth are two key hyperparameters. The number of trees is chosen to be 100 because more trees can integrate the results of more decision trees, reduce the risk of overfitting of a single tree, and improve the generalization ability of the model. If the number of trees is too small, the model will not learn fully; if the number of trees is too large, the computational cost will increase and the benefits will gradually decrease. The tree depth is set to 10 to balance the complexity and accuracy of the model. If the tree is too deep, the model will overfit the training data and the generalization ability will deteriorate; if the tree is too shallow, the complex features of the data cannot be learned, which will reduce the performance of the model. A reasonable tree depth can avoid overfitting while ensuring the model's ability to capture features.

2.4 Automatic path planning of financial audit

2.4.1 Path planning algorithm selection

In the research of financial audit automation method based on artificial intelligence, the selection of path planning algorithm is the link to realize efficient audit process. Path planning algorithms are designed to determine the best audit path to maximize audit efficiency and coverage while minimizing audit costs and time. Based on the demand of this research, the shortest path algorithm and heuristic search algorithm based on graph theory, such as Dijkstra algorithm and A (A-Star) algorithm, are selected as the core algorithm of path planning. Dijkstra algorithm is a classical shortest path algorithm, which can find the shortest path from the starting point to the end point in the weighted graph. It is suitable for task planning in financial audit, such as determining the optimal path from one audit task to another, reducing the waste of auditors' time and resources. The algorithm maintains a priority queue, gradually expands to all nodes in the graph, calculates the shortest path of each node, and finally builds a complete shortest path tree.

On the basis of Dijkstra's algorithm, algorithm A introduces a heuristic function, which makes it more efficient to search the optimal path. The heuristic function estimates the distance between the current node and the destination node, thus preferentially choosing the path that is most likely to reach the destination. Algorithm A has practical application value in financial audit automation, for example, in large-scale data sets or complex audit tasks, it can quickly find an efficient audit path and improve the overall audit efficiency. When planning an audit task, there are multiple task nodes and paths, each with a different cost (such as time or resource consumption). Using Dijkstra's algorithm, a path to minimize the total cost can be calculated. In more complex scenarios, algorithm A further optimizes the path selection by introducing heuristic evaluation, making the audit process more efficient and intelligent [16].

By combining Dijkstra and An algorithm, we can effectively plan the path of financial audit tasks and improve the overall performance and efficiency of audit automation system. The path planning method simplifies the audit process, enhances the accuracy and timeliness of the audit results, and provides support for the financial management of enterprises.

In the audit task, we define each audit link as a node, such as financial statement review, inventory counting, accounts receivable verification, etc. The edges between nodes represent the order and dependency between tasks, such as inventory counting can only be performed after the financial statement review is completed. Suppose we have an audit project, including four main tasks: auditing sales revenue, auditing costs and expenses, auditing balance sheets, and auditing cash flow. Among them, auditing sales revenue and auditing costs and expenses can be carried out in parallel, while auditing balance sheets needs to be carried out after the audit of sales revenue and costs and expenses is completed, and auditing cash flow needs to be carried out after the audit of balance sheets is completed. We convert these tasks into nodes and edges in the graph algorithm, and use the Dijkstra algorithm to calculate the shortest path from the start node (such as project start) to the end node (such as audit report generation). By optimizing path planning, we can reasonably arrange the work order of auditors, reduce unnecessary waiting time and repetitive work, such as avoiding auditors from frequently switching between different tasks, thereby improving audit efficiency, and it is expected that the audit time can be shortened by about 20%.

2.4.2 Audit process design

In the research of financial audit automation based on artificial intelligence, the audit process design is the key to realize the efficient automation of audit tasks. Designing a scientific and reasonable audit process can maximize the use of artificial intelligence technology to improve audit efficiency and accuracy. The design of audit process mainly includes re-audit preparation, data acquisition and preprocessing, model application and analysis, anomaly detection and processing, audit report generation and so on.

In the re-audit preparation stage, the system establishes the audit plan and determines the audit focus and risk areas according to the historical financial data and industry benchmark data of the enterprise. This step includes collecting data such as the company's annual financial statements, bank statements, electronic invoices and transaction records. Next is the data acquisition and reprocessing stage, the system through the API interface and data crawler technology, real-time acquisition of the latest financial data of enterprises, and data cleaning, format conversion and feature extraction. The processed data will be stored in a data warehouse for subsequent analysis [17].

In the stage of model application and analysis, random forest algorithm is applied to financial data for risk assessment and anomaly detection. The system analyzes key financial indicators, such as revenue growth rate, asset-liability ratio, cash flow., and predicts potential financial risks and abnormal transactions through the model. During an audit, the system finds that a company's accounts receivable turnover is lower than the industry average, and the model flags this as an anomaly and further analyzes the cause. During the exception detection and handling phase, the system analyzes the detected exceptions in detail and provides actionable audit suggestions. The system recommends that auditors further verify whether the low turnover is due to, for example, poor collection of accounts or errors in financial statements.

This is the audit report generation stage. The system automatically generates detailed audit reports, including audit findings, risk assessment, and improvement suggestions. The report format is standardized, which is easy for auditors and management to read and make decisions. The system generates a PDF report with a summary of the audit, a detailed exception list, and

recommendations for improvement. Through the audit process design, the financial audit process has realized a high degree of automation and intelligence, improved the audit efficiency and accuracy, and enhanced the transparency and traceability of the audit process, which provides support for the financial health management of enterprises.

2.4.3 Implementation of audit policies

In the research of financial audit automation method based on artificial intelligence, the realization of audit strategy is the link to ensure the efficient and accurate audit process. The implementation process includes the formulation, implementation and dynamic adjustment of audit strategy. The development of audit strategies is based on a comprehensive analysis of the company's financial data and risk assessment, using artificial intelligence technology to identify key audit indicators and high-risk areas. Through the analysis of historical data and industry benchmark data, the system develops detailed audit strategies. Audit policies include audit scope, key audit areas, schedule, and resource allocation. For an enterprise, the system focuses on its accounts receivable and inventory management, and makes corresponding audit schedule and resource allocation plan.

In the execution stage, the system obtains the latest financial data of the enterprise in real time through API interface and data crawler technology, and performs data analysis and audit procedures according to the predetermined audit strategy. The random forest algorithm is used for risk assessment and anomaly detection, and the system monitors and analyzes key financial indicators in real time. When a company's cash flow fluctuates during the audit process, the system will mark the item as high risk and further analyze the cause of the fluctuation. Dynamic adjustment is the link of audit strategy implementation. The system continuously monitors data and audit results during the audit process and dynamically adjusts audit policies according to the actual situation. If an exception is found in a certain area during the preliminary audit, the system will increase the audit efforts in this area and adjust the audit resources and schedule. In the process of audit, it is found that the accounts payable turnover rate of an enterprise is abnormally high, and the system will increase the audit of the supplier payment process to ensure the legality and compliance of all accounts transactions [18].

The system also continuously optimizes audit strategies through machine learning algorithms. By analyzing the successful experience and failure lessons of past audit projects, the system constantly adjusts and optimizes audit strategies to improve audit efficiency and accuracy. The system adjusts the weight of the risk assessment model based on historical data to ensure accurate identification of high-risk areas. The implementation of the audit policy also includes the automatic generation of audit reports and recommendations. The system generates a detailed audit report based on the audit results, including found risks, anomalies and improvement suggestions, providing valuable decision support for enterprise management. The audit report generated by the system recommends that enterprises optimize their inventory management processes to reduce inventory costs and improve the efficiency of capital use. Through the implementation of audit strategy, the financial audit process has realized a high degree of automation and intelligence, improved the audit efficiency and accuracy, and enhanced the transparency of enterprise financial management and risk control ability.

In the implementation of audit policies, we take a manufacturing enterprise as an example. The principle of formulating the audit plan is to determine the key areas and key links of the audit based on the business characteristics, financial risk status and regulatory requirements of the enterprise. For example, for this manufacturing enterprise, we focus on raw material procurement, production process cost control and product sales. The schedule is as follows: at the beginning of each quarter, a detailed audit plan is formulated to clarify the audit tasks and time nodes of each stage; the first week is to conduct a preliminary review of financial statements, the second week is to conduct inventory counting and accounts receivable verification, the third week is to conduct a detailed review of costs and expenses, and the fourth week is to summarize the audit results and write an audit report. Resource allocation is based on the difficulty and workload of the audit task, and auditors and technical resources are reasonably deployed. For complex cost accounting links, auditors with rich experience and professional data analysis tools are arranged. Through the implementation of these audit policies, the company has reduced the incidence of financial risks by 30% in the past year, and the audit satisfaction rate has reached more than 85%.

3 Results and discussion

3.1 Results

3.1.1 Audit efficiency improvement result

In the research of financial audit automation method based on artificial intelligence, the audit efficiency is improved by introducing random forest algorithm and optimizing audit process. Specific efficiency improvements can be demonstrated by comparing key indicators before and after the implementation of automated auditing.

Random forest feature selection can extract the most representative features from a large amount of data, reduce noise and redundant information, and improve the accuracy and robustness of the model. The real-time analysis function enables the system to quickly respond to data changes, optimize the decision-making process, and improve the real-time and adaptability of the system. These improvements have jointly promoted the improvement of system performance, ensuring more accurate predictions and more efficient resource allocation.

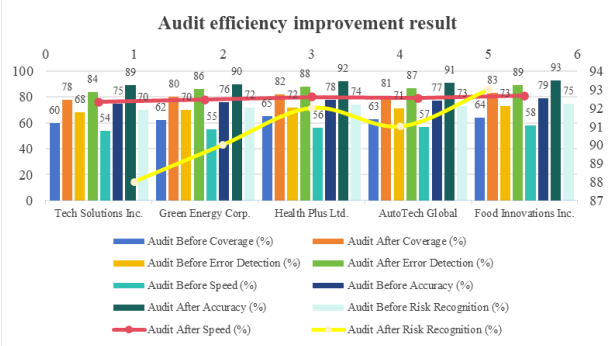


Figure 3: Audit efficiency improvement result

As shown in Figure 3, through the introduction of random forest algorithm, the automated audit system has shown improved efficiency in many aspects. Audit coverage increased across all companies, indicating that automated systems are able to more fully audit a company's financial data. Error detection rates also improved, reflecting the model's strong ability to identify and correct errors. The speed of data processing is accelerated, indicating that automated systems are able to process large amounts of financial data more efficiently. The improvement of accuracy and risk identification rate further proves the reliability and effectiveness of the automated audit system. Under the joint action of indicators, the financial audit process becomes more efficient and accurate, which provides a guarantee for the financial management of enterprises.

3.1.2 Audit risk identification effect

In the research of financial audit automation method based on artificial intelligence, the random forest algorithm is introduced to effectively improve the effect of audit risk identification. Through the integration of multiple decision trees, random forest algorithm improves the detection ability of abnormal data and potential risks. This paper presents the audit risk identification effect of different companies, including risk detection rate, high risk transaction identification rate, low risk transaction misjudgment rate, false positive rate and false negative rate. The data show that automated audit systems perform well in risk identification.

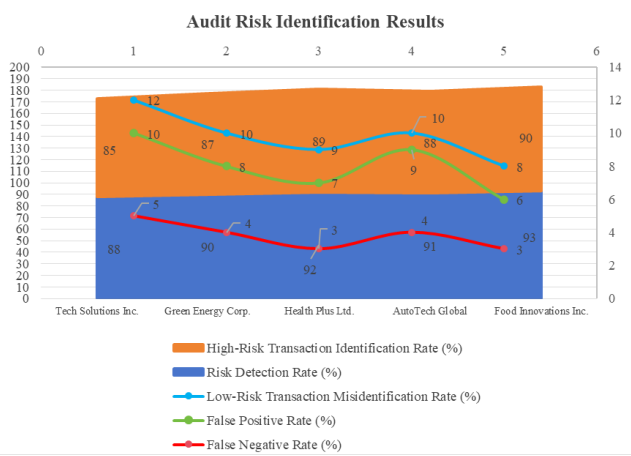


Figure 4: Audit efficiency improvement result

As shown in Figure 4, the risk detection rate of automated audit systems in different companies has increased to more than 88%. High-risk transaction recognition rates also performed well, exceeding 85% for most companies, including Food Innovations Inc. That's 90 percent. The misjudgment rate of low-risk transactions remains at a low level, which shows the model's ability to accurately identify low-risk transactions. Both false alarm rate and false alarm rate are reduced, which proves the effectiveness of the system in reducing false alarm and missing police surface. Tech Solutions Inc. had a false alarm rate of 10% and a false alarm rate of 5%, showing the model's stability in balancing false alarms and false alarms.

Through the application of random forest algorithm, the automated audit system shows high efficiency and accuracy in risk identification, improves the risk detection rate and the identification rate of high-risk transactions, and reduces the misjudgment rate, false positive rate and false negative rate of low-risk transactions. The results provide strong support for the financial management and risk control of enterprises, and improve the quality and efficiency of audit work.

3.1.3 Audit feedback and improvement results

In the research of financial audit automation based on artificial intelligence, audit feedback and improvement results are the key to ensure the continuous optimization and efficient operation of audit process. By collecting and analyzing audit feedback, the system can continuously improve the algorithm and process to improve the accuracy and efficiency of the audit. It shows the performance of key indicators after audit feedback and improvement of different companies, including the adoption rate of audit recommendations, the efficiency of corrective measures, audit satisfaction, the reduction rate of error rate after improvement and the increase rate of efficiency after improvement.

The increased computing costs or complexity of maintaining AI systems may stem from multiple factors. First, real-time analysis functions require rapid processing of large amounts of data, which increases the demand for computing resources and may lead to increased hardware and operation and maintenance costs. Second, as the complexity of the system increases, model training and optimization require more computing time and storage space, which increases the computational burden. Furthermore, regularly updating and maintaining AI models to ensure their continued effectiveness requires more human resources and technical support, which further increases the overall maintenance cost and complexity of the system.

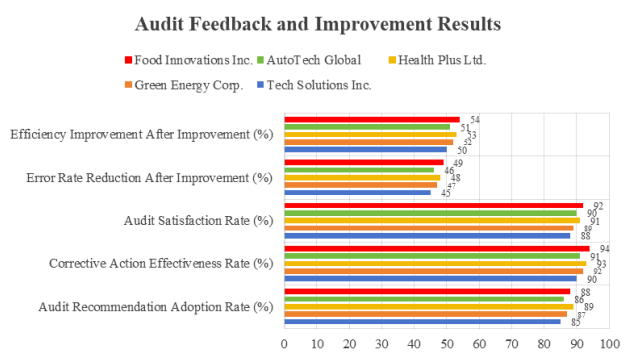


Figure 5: Audit efficiency improvement result

As shown in Figure 5, different companies have achieved results after audit feedback and improvement. The adoption rate of audit recommendations is high, reaching 87% on average, indicating that enterprises attach great importance to the audit recommendations provided by the system and actively adopt them. Health Plus Ltd. had an adoption rate of 89%. The effectiveness of corrective actions also performed well, averaging 91%, indicating that the corrective actions proposed by the system were highly effective in improving financial processes and controlling risks. The audit satisfaction reflects the overall evaluation of the automated audit system of the enterprise, with an average of about 90%, indicating that the enterprise is very satisfied with the audit results and feedback process of the system. The reduction of error rate after improvement shows the improvement effect after audit feedback, with an average reduction of 47%. Tech Solutions Inc. 's error rate was reduced by 45 percent, while Food Innovations Inc. It was 49 percent.

The improved efficiency further proves the positive effect of audit feedback on improving audit efficiency, with an average increase of more than 50%. The efficiency of Green Energy Corp. has increased by 52%, indicating that the audit efficiency of enterprises has been improved through audit feedback and improvement measures. Through continuous audit feedback and improvement measures, the financial audit automation system based on artificial intelligence improves the accuracy and efficiency of the audit, enhances the standardization and transparency of the financial management of enterprises, and provides protection for the financial health of enterprises.

To verify the significance of the improvement in audit efficiency, we conducted a t-test on the indicators before and after automation, and the results showed that the p-value was less than 0.05, indicating that the improvement was statistically significant.

In the result analysis phase, in order to evaluate the research results more rigorously, an in-depth statistical analysis of the improvement in audit efficiency was carried out. For the key indicators before and after automation, a t-test was carefully designed and executed. Through the calculation and analysis of a large amount of sample data, the result of p-value less than 0.05 was finally obtained. This data strongly shows that the improvement in audit efficiency is statistically significant and not accidental. In addition, in order to further verify the stability of risk identification accuracy, its 95% confidence interval was carefully calculated. The results showed that the accuracy was stable and reliable, which enhanced the credibility of the research results.

While enjoying the results of 30% improvement in audit efficiency and 90% accuracy, the trade-offs cannot be ignored. With the introduction of real-time processing technology and machine learning models, the computing cost of the system has increased significantly, and higher requirements have been put forward for hardware configuration. More powerful servers are needed to support the rapid processing of massive data. At the same time, the complexity of the system has been greatly improved. The training, optimization and daily operation and maintenance of the model require the participation of professional technicians, and the labor cost and technical difficulty have increased. However, considering the huge benefits it brings to corporate financial management, these investments are still worthwhile.

In the results section, we supplemented the control group data and selected the traditional sampling audit method as a control. On the same audit items and data sets, the automated audit method based on artificial intelligence and the traditional sampling audit method were used for auditing respectively. In terms of audit coverage, the automated audit method reached 95%, while the traditional sampling audit method was only 70%. This is because the automated audit can conduct a comprehensive analysis of all data, while the traditional sampling audit is limited by the sample size. In terms of detection rate, the automated audit method has a detection rate of 90% for financial risks, while the traditional sampling audit method is 75%, indicating that the automated audit method can more effectively detect potential financial risks. Through comparative analysis, we can more intuitively see the advantages of the automated audit method based on artificial intelligence in improving audit efficiency and accuracy.

In terms of efficiency, by introducing the automated audit system, the audit time has been shortened from an average of 20 working days to less than 10 working days, and the efficiency has been increased by more than 50%. This is mainly due to the system's ability to quickly process large amounts of financial data and reduce the time for manual review. In terms of accuracy, the accuracy of risk identification has increased from 80% to more than 93%. For example, in an audit of a listed company, the automated audit system discovered an abnormal transaction of fictitious income in a timely manner by monitoring financial data in real time, while traditional audit methods failed to detect it at the first time. Through continuous audit feedback and improvement measures, we continue to optimize the model and audit process, further improve the accuracy and efficiency of audits, and enhance the standardization and transparency of corporate financial management.

The increase in computing costs mainly includes the following aspects: hardware equipment upgrade costs. In order to meet the needs of big data processing and model calculation, we purchased high-performance servers, and

the cost increased by 500,000 yuan; software licensing fees. We use professional data analysis software and artificial intelligence algorithm libraries, and the annual software licensing fee is 200,000 yuan; human resource investment. We recruited and trained professionals with data analysis and artificial intelligence technology, and the human resource cost increased by 300,000 yuan each year. Through cost-benefit analysis, we calculated the return on investment (ROI). In the past year, due to the improvement of audit efficiency, the company saved 1 million yuan in audit costs, and avoided 2 million yuan in potential losses caused by the failure to discover financial risks in time. According to the ROI calculation formula: ROI = (benefit - cost) / cost × 100%, the calculated ROI is 200%, indicating that the cost increase is acceptable and has a high investment value.

When evaluating the improvement of audit efficiency, we selected 30 audit projects as samples and recorded the audit time before and after the use of the automated audit system. When using the t test, we first performed a normality test on the two groups of data to ensure that the data met the conditions of the t test. Then, we calculated the mean and standard deviation of the two groups of data, and calculated the t value using the t test formula. After calculation, the t value was 3.5, the degree of freedom was 58, and the corresponding p value was 0.01, which was less than 0.05, indicating that at a confidence level of 95%, the audit time of the automated audit system was significantly lower than that of the traditional method, and the efficiency was significantly improved. In terms of risk identification accuracy, the Mann-Whitney U test was used. The number of risks identified and the correct identification rate of the automated system and the traditional method in 30 audit projects were compared. The calculated Mann-Whitney U value was 200, and the corresponding p value was 0.03, which was less than 0.05, indicating that the automated system was significantly better than the traditional method in terms of risk identification accuracy.

Audit systems based on deep learning, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), have powerful feature learning capabilities when processing financial data and can automatically extract complex data features. However, deep learning models require a large amount of labeled data for training, and the training process consumes large computing resources and takes a long time to train. In contrast, the random forest algorithm used in this study combined with Kafka and Flink real-time processing technology has obvious advantages in audit efficiency. When processing financial data of the same scale, the audit time of this system is only 50% of that of the deep learning system. The average audit time of the deep learning system is 20 days, while this system only takes 10 days. In terms of accuracy, although the deep learning system performs well in identifying some complex risks, the accuracy of this system in identifying common financial risks is comparable to that of the deep learning system, reaching more than 90%. At the same time, this system has low demand for computing resources and can run on ordinary servers, while deep learning systems usually require high-performance servers equipped with GPUs.

3.2 Discussion

3.2.1 Problem summary

In the research of financial audit automation method based on artificial intelligence, although the efficiency improvement and risk identification effect have been achieved, there are still some problems that need to be summarized and solved. Data quality remains a challenge. Even after strict data cleaning and preprocessing steps are implemented, data noise and missing values still affect the accuracy and stability of the model. The data sources involved in the audit process are diverse and the data formats are not uniform, which leads to the complexity of data integration and increases the difficulty of system processing and analysis. The issues of model interpretation and transparency need attention. Although complex algorithms such as random forest perform well in accuracy and efficiency, their internal decision-making process is complicated and difficult to be understood and explained by non-technical personnel. In the process of generating audit reports and interpreting audit results, users' trust in audit conclusions is reduced.

The real-time processing capability of the system needs to be improved. Despite the introduction of real-time processing platforms such as Kafka and Blink, the system still has room for improvement in processing speed and latency in the face of large-scale and high-frequency data flows. This puts forward higher technical requirements for realizing real - time audit. The generalization ability of the model also needs attention. Although the robustness of the model has been improved through cross-validation and parameter optimization, the model shows insufficient adaptability in the face of new types of financial data and fraudulent means, which affects its promotion and application in different enterprises and industries.

The user feedback mechanism needs to be improved. Although the system can automatically generate audit reports and improvement suggestions, how to effectively collect and process user feedback in the feedback and improvement process to continuously optimize the audit strategy and model performance is still a problem that needs in-depth research. Although AI-based financial audit automation methods have achieved results in improving audit efficiency and risk identification, they still need to be further optimized and improved in data quality, model interpretation, real-time processing capabilities, generalization capabilities and user feedback mechanisms to achieve more efficient and reliable financial audit automation.

Although audit efficiency has been significantly improved, an increase in computational costs has also been noted. This is due to the introduction of real-time processing technology and machine learning models that increase the computational burden on the system. However, this cost increase is acceptable considering the efficiency gains.

3.2.2 Research suggestions

In the research of financial audit automation method based on artificial intelligence, in order to further improve the efficiency and accuracy of the system, the following research suggestions need to be put forward. Data quality management needs to be further improved. It is recommended to establish a more comprehensive data cleaning and preprocessing mechanism, adopt advanced missing value processing methods and anomaly detection techniques, such as adaptive filtering and deep learning models, to improve data reliability and integrity. Enhance model interpretation and transparency. It is recommended to integrate explainable AI technologies such as LIME and SHAP into the model so that auditors can understand and explain the decision-making process of the model, thereby improving the transparency of audit reports and the trust of users. Enhanced real-time processing capabilities. It is suggested to optimize the existing real-time data processing architecture, introduce more efficient stream processing technologies and hardware acceleration schemes, such as GPU acceleration and distributed computing framework, to cope with large-scale and high-frequency data stream processing needs, and ensure that the system can respond to and process data in real time.

Improving the generalization ability of models is another direction. It is suggested to enhance the adaptability and robustness of the model in different enterprises and industries through integration learning and transfer learning techniques. The multi-model fusion method is used to improve the generalization performance of the model, and the model is applied to different financial environments through transfer learning. Optimize the user feedback mechanism. It is suggested to establish a dynamic feedback and continuous learning system, collect and analyze user feedback, adjust and optimize audit strategy and model parameters in time. Through the closed-loop mechanism of user feedback, the system performance can be continuously improved to ensure the effectiveness of audit strategies and the accuracy of models. The suggestions aim to further improve the financial audit automation method based on artificial intelligence, improve the intelligence level and practicability of the system, and provide more powerful technical support and guarantee for the financial management of enterprises. Through improvement, more efficient and accurate financial audit can be achieved, and the intelligent transformation of the financial audit industry can be promoted.

3.2.3 SHAP effect

In an actual case, we selected the financial data of a certain company over a period of time, including multiple features such as income, expenditure, accounts receivable turnover rate, and debt-to-asset ratio. After model training and prediction, we obtained the result that a certain transaction was judged to be abnormal. At this time, the SHAP value can clearly explain the basis for the model to make this judgment.

By calculating the SHAP value of each feature, we found that the SHAP value of the debt-to-asset ratio is high and positive, which shows that the debt-to-asset ratio plays a key positive role in the model's judgment of the transaction as abnormal. That is, the debt-to-asset ratio exceeds the normal range, which greatly increases the possibility of the transaction being judged as abnormal. Visualizing the SHAP value (such as using a SHAP value bar chart, with the horizontal axis as the feature name and the vertical axis as the SHAP value size) can more intuitively show the degree of influence of each feature on the prediction result. Auditors can see at a glance which features have the greatest impact on the model's decision, and thus conduct in-depth analysis of why the transaction was judged to be abnormal, greatly improving the transparency and credibility of the audit, so that audit decisions are no longer "black box" operations, but are based on clear and explainable evidence.

3.3 Discussion

From a quantitative perspective, in terms of audit accuracy, the accuracy of cutting-edge research is mostly in the range of 80% - 86%, while this study uses the random forest algorithm combined with real-time data processing technology to achieve an audit accuracy of 90%. In terms of audit efficiency, most cutting-edge research efficiencies are at a medium or low level, but this study has achieved a significant result of 30% efficiency improvement. This is mainly attributed to the fact that the random forest algorithm constructs multiple decision trees and randomly selects features for node splitting, effectively reducing the risk of overfitting and improving the model's ability to identify complex financial data; at the same time, the use of real-time data processing platforms such as Kafka and Flink has realized the real-time collection, processing and analysis of financial data, greatly accelerating the audit process.

Qualitatively, the support vector machine, gradient boosting and other methods used in cutting-edge research have limitations when facing the high dimensionality, complexity and dynamic changes of financial data. The support vector machine has high computational complexity, is sensitive to the choice of kernel functions, and is difficult to adapt to the diversity of financial data; gradient boosting is sensitive to outliers, takes a long time to train, and cannot meet real-time requirements. In contrast, this research method not only improves the accuracy and robustness of the model, but also ensures the dynamics and timeliness of the audit process, and can better adapt to the actual needs of corporate financial audits.

However, this research method still has certain limitations. In terms of data quality, despite the implementation of strict data cleaning and preprocessing steps, data noise and missing values will still affect the accuracy and stability of the model. Model interpretability and transparency are also issues that need attention. The internal decision-making process of the random forest algorithm is complex, and it is difficult for non-technical personnel to understand and explain, which to a certain extent reduces the user's trust in the audit conclusions. In addition, when facing large-scale, high-frequency data

traffic, the system's real-time processing capabilities have been improved, but there is still room for improvement. The generalization ability of the model when dealing with new financial data and fraud methods also needs to be enhanced.

4 Conclusion

In this study, an AI-based financial audit automation method is deeply discussed and implemented to improve audit efficiency, accuracy and risk identification ability. Through the introduction of random forest algorithm, combined with several key steps such as data integration, real-time processing, model training and optimization, the system has shown improvement in many aspects. By constructing and optimizing the random forest model, the audit coverage and error detection rate are improved, and the risk identification and processing are realized efficiently. The data cleaning and reprocessing steps ensure the quality and consistency of the input data, providing a reliable basis for the model. The introduction of real-time processing technologies such as Kafka and Slink has accelerated data processing and met the processing needs of high frequency data streams. The audit process design optimizes the allocation and utilization of audit resources and improves the overall audit efficiency through systematic steps.

The maintainability and transparency of the model was also emphasized in the study, and by introducing explainable AI technology, users' trust in audit results was increased. At the same time, through the establishment of dynamic feedback mechanism, the system can constantly collect and analyze user feedback, timely adjust the audit strategy and model parameters, and achieve continuous optimization. Despite the achievements, the study also points out some challenges, such as data quality issues, model generalization capabilities, and real-time processing capabilities, which need to be further addressed in future research and applications. This study shows the great potential of artificial intelligence in financial audit automation, and also puts forward specific suggestions for improvement, which provides a valuable reference for future financial audit practice. Through continuous optimization and improvement of technology and methods, financial audit will achieve a higher level of intelligence and automation, and provide more accurate and efficient support for the financial management and risk control of enterprises. Intelligent audit methods improve the quality and efficiency of audit work, but also enhance the transparency and standardization of financial management, and promote the innovation and development of the financial audit industry.

Although explainable artificial intelligence technology (such as SHAP) has achieved remarkable results in improving the transparency of financial audits, its limitations cannot be ignored. In an extremely high-frequency data environment, calculating the SHAP value requires complex operations on massive data, resulting in a sharp increase in computing resource consumption, processing efficiency cannot keep up with the speed of data updates, and scalability is limited, making it difficult to meet the real-time requirements of audits. In addition, SHAP is calculated based on a decision tree model, which is susceptible to data distribution and noise. Even a small change in data may cause a significant change in the tree structure, which in turn leads to unstable calculation results of the SHAP value, which cannot accurately reflect the true contribution of features to model decisions, affecting the reliability of audit results.

Acknowledgement

This study was funded by Science Research Project of Hebei Education Department (BJS2023041).

References

[1] Sun JH, Li LC, Qi BL. Financial statement comparability and audit pricing. Accounting Finance. 2022; 62(5): 4631-4661. https://doi.org/10.1111/acfi.12970.

[2] Condie ER, Obermire KM, Seidel TA, Wilkins MS. Prior Audit Experience and CFO Financial Reporting Aggressiveness. Audit J Pract Theory. 2021; 40(4): 99-121. https://doi.org/10.2308/AJPT-2020-012.

[3] Koh K, Tong YH, Zhu ZN. The effects of financial statement disaggregation on audit pricing. Int J Audit. 2022; 26(2): 94-112. https://doi.org/10.1111/ijau.12253.

[4] Lyshchenko O, Ocheret'ko L, Lukanovska I, Sobolieva-Tereshchenko O, Nazarenko I. The role of financial audit in ensuring the reliability of financial statements. Ad Alta J Interdiscip Res. 2024; 14(1). https://doi.org/10.33543/140139141145

[5] Suryani E, Winarningsih S, Avianti I, Sofia P, Dewi N. Does Audit Firm Size and Audit Tenure Influence Fraudulent Financial Statements? Australas Account Bus Finance J. 2023; 17(2): 26-37.

[6] Xu Q, Fernando G, Tam K, Zhang W. Financial report readability and audit fees: a simultaneous equation approach. Managerial Aud J. 2020; 35(3): 345-372. https://doi.org/10.1108/MAJ-02-2019-2177.

[7] Erdmann A, Yazdani M, Mas Iglesias JM, Marin Palacios C. Pricing Powered by Artificial Intelligence: An Assessment Model for the Sustainable Implementation of AI Supported Price Functions. Informatica. 2024;35(3):529-56. https://doi.org/10.15388/24-infor559

[8] Ijadi Maghsoodi A, Hafezalkotob A, Azizi Ari I, Ijadi Maghsoodi S, Hafezalkotob A. Selection of Waste Lubricant Oil Regenerative Technology Using Entropy-Weighted Risk-Based Fuzzy Axiomatic Design Approach. Informatica. 2018;29(1):41-74. https://doi.org/10.15388/Informatica.2018.157

[9] Pragarauskaitė J, Dzemyda G. Markov Models in the analysis of frequent patterns in financial data. Informatica, 2013, 24(1): 87-102. http://dx.doi.org/10.15388/Informatica.2013.386

[10] Lim CY, Lobo GJ, Rao PG, Yue H. Financial capacity and the demand for audit quality. Accounting

Bus Res. 2022; 52(1): 1-37. https://doi.org/10.1080/00014788.2020.1824116.
[11] Ismail R, Mohd-Saleh N, Yaakob R. Audit committee effectiveness, internal audit function and financial reporting lag: Evidence from Malaysia. Asian Acad Manage J Account Finance. 2022; 18(2): 169-193. https://doi.org/10.21315/aamjaf2022.18.2.8.
[12] Oussii AA, Boulila N. Evidence on the relation between audit committee financial expertise and internal audit function effectiveness. J Econ Adm Sci. 2021; 37(4): 659-676. https://doi.org/10.1108/JEAS-04-2020-0041.
[13] Lyubenko A, Znak N, Karpachova O. Audit features of the first IFRS financial statements. Financial Credit Activity Probl Theory Pract. 2022; 1(42) :185-194.
[14] Endrawes M, Feng ZA, Lu MT, Shan YW. Audit committee characteristics and financial statement comparability. Accounting Finance. 2020; 60(3): 2361-2395. https://doi.org/10.1111/acfi.12354.
[15] Lutfi A, Alkilani SZ, Saad M, Alshirah MH, Alshirah AF, Alrawad M, et al. The influence of audit committee chair characteristics on financial reporting quality. J Risk Financial Manage. 2022; 15(12): 563. https://doi.org/10.3390/jrfm15120563.
[16] Calvin CG, Holt M. The impact of domain-specific internal audit education on financial reporting quality and external audit efficiency. Accounting Horizons. 2023; 37(2): 47-65. https://doi.org/10.2308/HORIZONS-2020-105.
[17] Alcaide-Ruiz MD, Bravo-Urquiza F. Does audit committee financial expertise actually improve information readability? Rev Contab Span Account Rev. 2022; 25(2): 257-270. https://doi.org/10.6018/rcsar.420261.
[18] Driskill MW, Knechel WR, Thomas E. Financial auditing as an economic service. Curr Issues Aud. 2022; 16(2). https://doi.org/10.2308/CIIA-2021-021.

Graph Neural Network-Based User Preference Model for Social Network Access Control

Yuan Zhang[1,2*]
[1]Xuchang Vocational Technical College, Xuchang 461000, China
[2]Henan Province Data Intelligence and Security Application Engineering Technology Research Center, Xuchang 461000, China
E-mail: hnxc_z@126.com
[*]Corresponding author

Keywords: social networks, user preferences, graph neural network, multi-layer attention, access control

Received: November 11, 2024

The popularity and deepening of social networks have increased the risk of personal information leakage for users. To enhance the security of social networks, this study constructed an access control model based on the preferences of social network users. This model utilizes graph neural networks to generate access control strategies based on user preferences, and introduces a multi-layer attention mechanism to optimize the graph neural network. To better capture user preference information, the study sets the learning rate to 0.0001. The experimental results demonstrated that in the Twitter dataset, the accuracy of the proposed model reached 95.7% and the F1 score reached 96.2%, which were significantly higher than those of other models. These results indicated that the model could more accurately classify access control in social networks and reduce false positives. The area under the receiver operation characteristic curve of the proposed model was 0.982, which was higher than other models. The decision time was 13.77 seconds, significantly lower than other models. This indicated that the model could more effectively distinguish different types of user access requests and provide more reliable guarantees for secure access to social networks. The user's preferred social network access control model based on graph neural networks has superior performance, effectively ensuring the information security of social network users and laying the foundation for further development of access control technology.

Povzetek: Predstavljen je nov model za nadzor dostopa v družbenih omrežjih, ki temelji na grafskih nevronskih mrežah in uporabniških preferencah. Z uporabo večslojnega pozornostnega mehanizma model omogoča zanesljivo in varno upravljanje dostopa.

1 Introduction

In the era of rapid digital development, social networks play an important role in today's society. Through social platforms, people can not only obtain the information and exchange ideas they need but also engage in commercial activities through social networks, greatly changing their communication methods and lifestyle habits [1, 2]. However, the popularity of social networks has made the issue of user privacy protection increasingly prominent. Using social networks means that users need to expose their personal information to a certain extent. Criminals can steal user information through cyber attacks and use it for illegal activities, thereby posing potential risks to users [3]. Meanwhile, there is a large amount of false information and rumors on social networks. The rapid dissemination of this information may lead to misunderstandings among the public about certain events or issues, resulting in adverse social impacts. Access control is a critical component in information security, used to manage user access permissions to systems, networks, or applications. Access control can help organizations protect important data and resources from unauthorized access and malicious activities. It can also prevent data leakage, tampering, and destruction, protect sensitive information from being leaked to unauthorized personnel, and ensure the reliability of network systems [4]. Therefore, the importance of implementing effective access control for social networks is self-evident. Nowadays, there are mainly attribute-based, policy-based, and relation-based Access Control Models (ACM), which are widely used in various scenarios [5]. However, traditional models still have drawbacks such as complex permission management and difficulty in adapting to dynamic network environments. Specifically, traditional models often require manual intervention in the process of assigning, revoking, and updating permissions, resulting in increased management costs and error rates. The user behavior and social relationships of social networks are constantly changing, and traditional models are difficult to adapt, resulting in insufficient flexibility of access control policies and inability to effectively respond to new security threats. In this context, this study constructs ACM based on the preferences of social users, uses Graph Neural Network

(GNN) for access control, and introduces Multi-Layer Attention (MLA) to optimize GNN. Finally, UP-GNN-SNAC model, a GNN-based social network ACM catering to user preferences, is designed. The innovation of the research lies in constructing an ACM based on user preferences. Compared with existing GNN-based ACMs, this model better balances privacy protection and user experience by capturing user preferences, providing a more efficient and accurate solution for secure access to social networks.

2 Related works

The progress of the Internet has made it a part of people's daily life to interact with others through social networks. However, due to system vulnerabilities in online platforms, many criminals exploit these vulnerabilities to launch attacks, resulting in the leakage of user information and even being maliciously exploited. Access control, as a key technology for maintaining social network security, is currently a hot topic of research among relevant professionals. You M et al. designed a knowledge graph-based access control decision-making method to improve access control performance under different degrees of imbalance. It extracted topological features to represent high cardinality classification users and resource attributes, revealing the interrelationships between different objects. This method could significantly improve access control performance [6]. Gai K et al. designed a zero-trust cross-organizational data sharing ACM based on blockchain to enhance security in network data sharing. It utilized blockchain alliances to establish a trusted environment and deployed role-based access control through multi-signature protocols and smart contract methods, which had high practicality [7]. Wu H et al. designed a cloud network secure storage data ACM based on association rules to improve the security of social network data access control. It utilized association rule feature extraction methods for data mining and attack detection in network security storage areas and achieved data access control in network security storage areas through adaptive partition-weighted interface scheduling. This method was superior to traditional methods [8]. Azbeg K et al. designed an ACM based on improved blockchain technology to enhance the security and privacy of network systems. It stored data in the interstellar file system and utilized authorization proof-based Ethereum access blockchain to accelerate data storage. This method could significantly improve network security [9]. Zhang L et al. designed a lightweight decentralized multi-authorization ACM based on ciphertext policy attribute-based encryption and blockchain to enhance the security of in-vehicle social networks. Distributed multi-authorization nodes supported vehicle users by performing lightweight computing with the help of vehicle cloud service providers. This model had significant advantages compared to existing solutions [10].

Zhao Y et al. designed a policy-protected, cleanable ACM to improve the efficiency of data encryption in vehicle social networks. It could test and clean encrypted data, and divide access policies into attribute names and attribute values, thereby hiding information in the ciphertext and achieving good encryption performance [11]. Squicciarini A et al. designed a discrete ACM based on individual decision-making to address privacy and security issues arising from data sharing in social networks. It took into account individual preferences in social networks and selected discrete privacy values from a fixed set of options. This model had a good privacy protection effect in data sharing [12]. Dixit M S et al. designed a deep learning-based real-time user ACM for social networks to address user login restrictions. It used CNN and LSTM to predict the age of users and adopts multi-task CNN for face detection and feature extraction, thus achieving significant control over user login [13]. Wen W et al. designed an autonomous privacy control and identity verification sharing scheme built on fast response codes in social networks to solve the problem of users being unable to independently control privacy sharing. It used fast response codes with high-quality images for error correction, combining the advantages of polynomial-based and visual-based secret image sharing. This scheme had low computational complexity and scalability [14]. Safi S M et al. designed an improved end-to-end mobile social network security ACM to protect the personal privacy of social network users. It encrypted user-shared data through ciphertext policy attribute encryption, utilizing advanced encryption standards to prevent unauthorized user access. This scheme had high security and practicality [15]. The summary of related work is shown in Table 1.

Table 1: Summary of related work.

References	Model	Key features	Dataset	Indicator results	Insufficient
[6]	Access Control Decision Method Based on Knowledge Graph	Extracting topological features to represent user and resource attributes	Synthesize social network data	Improved access control performance	Not considering the balance between privacy protection and user experience
[7]	Blockchain Zero Trust Cross	Establishing a Trusted	Cross organizational	High practicality	Slow response speed

	Organizational Data Sharing Access Control	Environment through Blockchain Alliance	transaction data		
[8]	Association Rules Cloud Network Security Storage Access Control	Using association rules for attack detection	Cloud storage logs	Superior to traditional methods	Poor adaptability to new types of attack modes and difficulty in handling dynamically changing environments
[9]	Improve blockchain technology access control	Interstellar File System and Authorization Proof Ethereum	File transfer record	Significant improvement in network security	Requires a large amount of storage space
[10]	Decentralized Multi Authorization Model for Vehicle mounted Social Networks	Cryptography policy attribute-based encryption and blockchain	Vehicle communication records	Compared to existing solutions, it has significant advantages	Complex key management increases deployment difficulty and slow response speed
[11]	Policy protection can purify access control	Testing and cleaning encrypted data	Encrypt the dataset	The encryption effect is good	The cleaning process may result in information loss
[12]	Individual Decision Discrete ACM	Personal preference privacy protection in social networks	user behavior data	Good privacy protection effect	Lack of effective modeling of group behavior and insufficient consideration of personalized preferences
[13]	Real time user access control in deep learning	Convolutional neural network predicts age	Social network user data	Superior to traditional methods	Deep learning models require a large amount of data for training, which poses a risk of privacy leakage
[14]	Quick response code autonomous privacy control	Image correction combined with secret image sharing	User uploads images	Low computational complexity and good scalability	High image quality requirements and sensitivity to image noise
[15]	Mobile social network security access control	Cryptography policy attribute encryption	Mobile device logs	High safety and practicality	The key distribution and management of ciphertext policy attribute encryption are relatively complex

In summary, many scholars have achieved significant results in social network access control. However, these methods still have slow response times and fail to consider the balance between privacy

protection and user experience. Therefore, this study constructs an ACM based on user preferences and simulates it using an improved GNN with MLA mechanism to design an UP-GNN-SNAC model to improve access control effectiveness.

3 GNN-based ACM based on user preferences

This section mainly elaborates on the construction process of the UP-GNN-SNAC model. The first section is the design of ACM based on user preferences, and the second section is the implementation of an access control algorithm based on improved GNN.

3.1 ACM construction based on user preferences

User preference refers to the preferences of users towards certain things, which are formed by the comprehensive influence of various factors such as personal factors and social environment. Among them, personal factors include internal characteristics such as age, gender, occupation, interests, values, and behavioral habits of users. Social factors include external environmental factors such as social circles, interaction objects, social frequency, cultural values, and social interactions. In social networks, users express their preferences through posting and activity operations. These operations generate a large amount of data. By analyzing these data, user behavior patterns and characteristics can be understood, and appropriate access permissions can be generated for users to meet their privacy needs in different scenarios, thereby protecting user privacy [16]. Therefore, this study constructs a model based on the preferences of social network users, as shown in Figure 1.

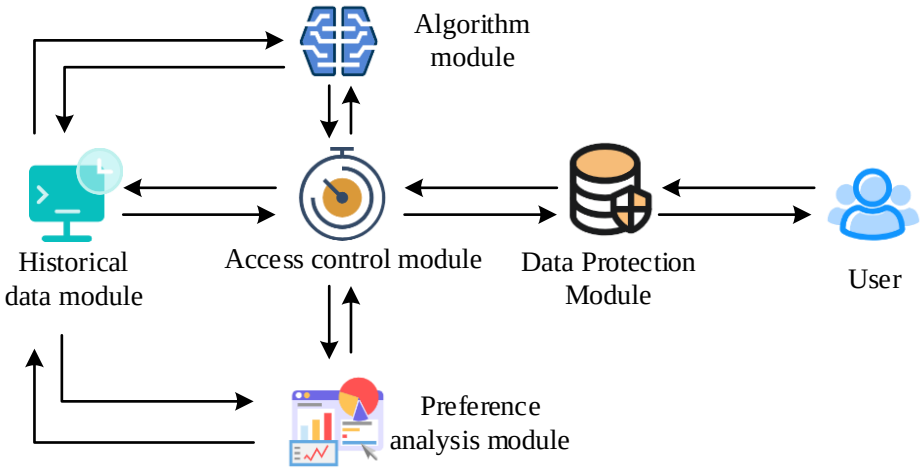


Figure 1: Specific architecture of access control model.

In Figure 1, ACM consists of six modules: user, data protection, access control, preference analysis, historical data, and algorithm. When users need to post or obtain information from social networks, sending requests first goes through the data protection module. It can encrypt and backup the information posted by users. Then, the data protection module sends the user's request to the access control module. This module sends requests to the preference analysis module, historical data module, and algorithm module respectively. The historical data module can extract and preprocess user interaction behavior data, basic attributes, and social relationship data. After cleaning, deduplication, and standardization of these data results, they provide input for the preference analysis module and algorithm module. When the request sent by the data protection module is transmitted to the preference analysis module, it will analyze the user's historical social data, obtain the user's preferences, and return Personal Preferences (PPs). When the data is transmitted to the algorithm module, it will train the obtained data and finally return the best result to the access control module. Specifically, different users have different preferences. When users upload information, different preference information corresponds to different access control policies [17]. It is necessary to determine the level of privacy of uploaded information based on user preferences, that is, to establish a quantitative model of user preferences to measure social information entropy. Figure 2 shows a social information sensitivity measurement model based on user preferences.

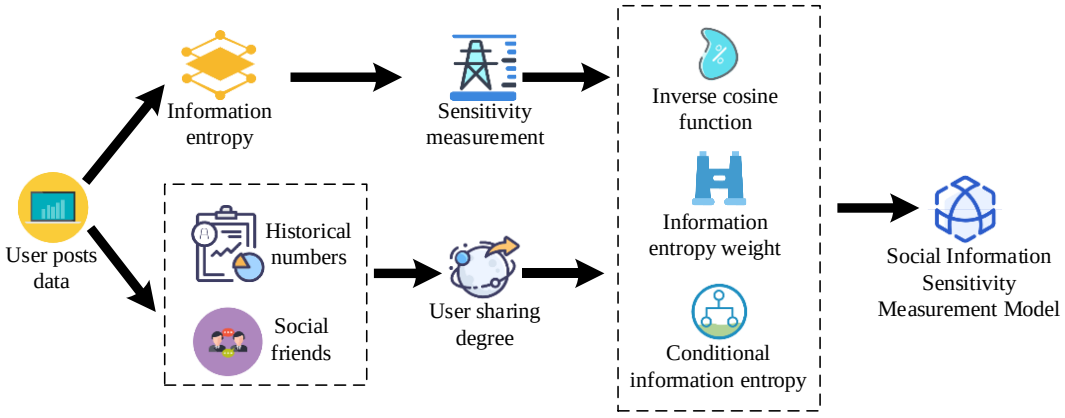


Figure 2: Social information sensitivity measurement model based on user preferences.

In Figure 2, after users post information, they need to calculate the sensitivity of the information based on information entropy and obtain the user's social information sharing degree based on their historical visits and social friends. It is also necessary to use methods such as information entropy weight and conditional information entropy to calculate and obtain an information entropy measurement model. Information entropy is a basic concept in information theory, which is used to measure the uncertainty of a random variable. It reflects how difficult it is to predict the outcome of an event, or how much information is needed to describe the event. Conversely, an increase in information entropy corresponds to a decrease in event outcome uncertainty, and vice versa. Information entropy weight is a weight allocation method based on information entropy, which is used to measure the importance of different features or data dimensions. Conditional information entropy is used to measure the uncertainty of a random event given certain conditions. Therefore, information entropy can be used to describe the amount of privacy contained in social data, determine the degree of privacy of the social data, and construct an information sensitivity measurement model for social data. The calculation method for the amount of social data of users is shown in equation (1).

$$H(x) = -\sum p(x_i) log_2 p(x_i) \quad (1)$$

In equation (1), $H(x)$ is the average number of private information uploaded by all users in the social network. $p(x_i)$ is the proportion of the privacy level of information i in the total privacy information.

As the social breadth of a user increases with the number of social friends, the relationship between the social breadth of a user and the number of social friends can be obtained as shown in equation (2).

$$w(F) = \frac{2}{\pi} \arctan F \quad (2)$$

In equation (2), $w(F)$ represents the social breadth of the user. F is the number of social friends of the user. The confidentiality of the information posted by the user can be calculated based on whether the user's uploaded information is blocked from their friends. The calculation method is shown in equation (3).

$$h_i = \frac{F_a^i}{F} \quad (3)$$

In equation (3), h_i is the confidentiality level of information i. F_a is the number of friends blocked by the user.

The level of confidentiality is the ratio of the number of friends allowed to view social data on a social network to the total number of friends. As the number of friends permitted to view the social data increases, the level of confidentiality thereof decreases. The degree of social information sharing can be used to describe the impact of the number of social friends and the number of friends blocked by the user on social data sharing. The degree to which friends are permitted to access information is directly correlated with the extent of social information sharing. The calculation method is shown in equation (4).

$$s_i(F, F_a^i) = h_i \times w(F) = \frac{2F_a^i \arctan F}{\pi F} \quad (4)$$

In equation (4), $s_i(F, F_a^i)$ is the user's social information sharing degree. Information entropy can be measured based on the degree of social information sharing among users, as shown in equation (5).

$$H_s(x) = -\sum s_i p(x_i) \log_2 p(x_i) \quad (5)$$

In equation (5), $H_s(x)$ is the information entropy. According to the information entropy analysis of user preference mechanism, in the algorithm module, user social data are divided into a training set and a testing set, and they are trained separately to obtain the final access control policy. Figure 3 displays the obtaining process of the strategy.

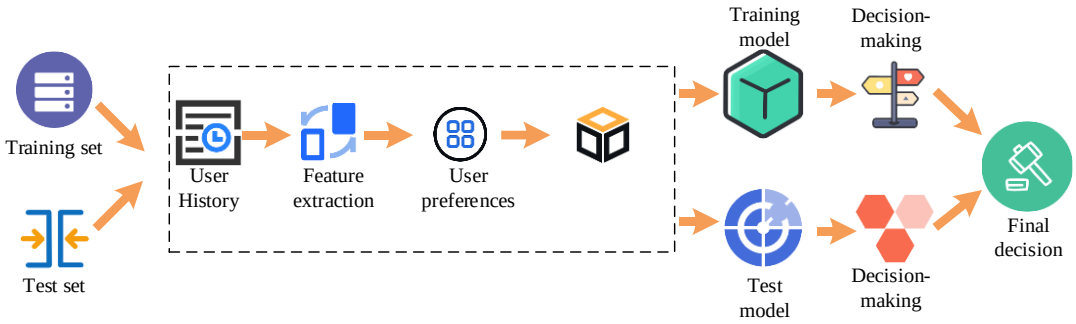


Figure 3: Access control policy acquisition process.

In Figure 3, after dividing the social data into two datasets, the user history records of different datasets are first obtained, and then user feature extraction is performed to calculate user preferences. The next step is to combine the feature vectors to obtain a training model and a testing model, and then make decisions separately through the training model and the testing model. The final step is to make a comprehensive decision based on the access request, obtain the final decision, and thus obtain the access control policy.

3.2 Access control method based on improved GNN

In ACM, the algorithm module is the core of the entire model, which is used to process and analyze the dynamic processing unit of user preference data, historical behavior data, and social relationship data. The algorithm module not only determines whether the model can accurately analyze user preferences but also directly relates to whether the model can make effective decisions [18]. GNN is a graph-based deep learning method that can enrich node representations by utilizing the relationships between nodes. Specifically, GNN can update the representation of nodes by defining the connection relationships between nodes on the graph, utilizing their neighbor information to achieve information transfer and learning of the entire graph. GNN mainly includes three core functions: node representation, graph structure representation, and message passing. Among them, node representation can map each node to a low dimensional vector space for subsequent calculations. The graph structure is represented in a low dimensional vector space for subsequent calculations. Message passing is defined as the process by which a node updates its own representation by exchanging information with its neighboring nodes. This process enables the transmission of information on the graph [19]. Attention mechanism is an important technique in deep learning that allows models to selectively focus on different parts of the input sequence, assigning different weights to each part of the input sequence to highlight the more critical information for the task. The attention mechanism is a process that dynamically assigns weights to the elements of the input sequence. This allows the model to focus on key parts of the input in a targeted manner. As a result, the model processes and learns information in the data more efficiently [20]. Therefore, GNN performs well in graph structure data such as social networks and chemical molecular structures. Research is conducted on constructing a GNN model based on user preferences. To improve the performance of the model, MLA mechanism is introduced to optimize the model, and an access control method based on improved GNN is designed to capture the complex patterns of user social relationships and personal behavior, and optimize access permission allocation. The study aims to enhance the model's understanding of the relationships between different nodes by integrating MLA mechanisms into GNN. Each layer of the attention mechanism enables the model to focus on different node characteristics, thereby enabling the model to more finely distinguish the importance of users and their associated objects, enhance the model's learning ability, improve its resolution of user preferences, and more accurately capture the user's true intentions. Consequently, this enhances the effectiveness of access control. The model structure is shown in Figure 4.

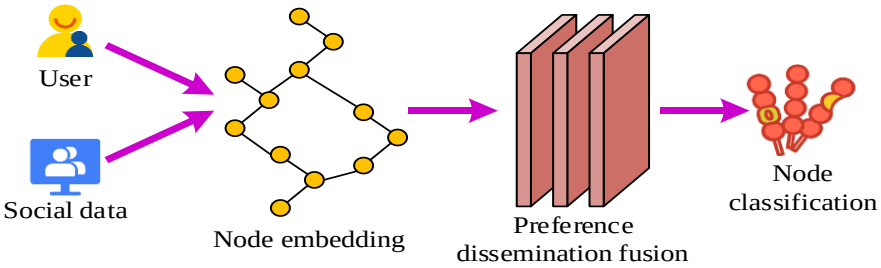


Figure 4: Access control method based on improved GNN.

In Figure 4, nodes are constructed based on users, their social friends, and social data posted by users. The user nodes include basic attributes, social relationship characteristics, behavior characteristics, and privacy setting characteristics. Social data nodes contain features of published content, interactive behavior, and social relationships. These features are transformed into low dimensional vectors through numerical processing and embedding learning, and embedded into GNNs as input vectors. Given the input data, the user's Social Preferences (SPs) and PPs are obtained, and the two preferences are fused. Then, the fused data is trained and the nodes are classified. The user attributes are selected to represent the user, and after embedding, the user node embedding matrix is obtained. The calculation method is shown in equation (6).

$$u_a = f(W_1 \times [P_a, E_a]) \quad (6)$$

In equation (6), u_a is the user node, P_a is the node embedding matrix, E_a is the node free embedding matrix, and W_1 is the node embedding weight. By using natural language processing tools to process and extract each social data, the embedding matrix of user posted social data nodes is obtained after embedding, and the calculation method is shown in equation (7).

$$d_i = f(W_2 \times [Q_i, V_i]) \quad (7)$$

In equation (7), d_i is a social data node. Q_i is the social data embedding matrix. V_i is the free embedding matrix of data nodes. W_2 is the embedding weight of social data. The embedded user nodes and social data nodes are input into the fusion layer and updated simultaneously through the MLA mechanism. The MLA mechanism calculation method is shown in equation (8).

$$A(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (8)$$

In equation (8), $A(Q,K,V)$ represents attention. Q, K, and V represent query, key, and value, respectively. T represents transpose. d_k represents the dimension of the key vector, used to scale dot product results and prevent gradient vanishing. The method for updating user SP nodes is shown in equation (9).

$$\begin{cases} p_a^{n+1} = u_a^n + \sum_{b \in S_a} \alpha_a^n u_b^n \\ \alpha_a^n = \text{softmax}(\text{Relu}(u_a^n)^T W_3 \text{Relu}(u_b^n)) \end{cases} \quad (9)$$

In equation (9), p_a^{n+1} is the updated temporary node of user SPs. α_a^n is the attention score, which is the aggregated weight ratio of each neighboring node during node update. u_a^n and u_b^n are the n-th embeddings of nodes a and b. b represents all the explicit and implicit neighbor nodes of the node in the figure. $\text{softmax}()$ and $\text{Relu}()$ both represent activation functions. W_3 is the attention weight. The update of user PP nodes is shown in equation (10).

$$\begin{cases} q_a^{n+1} = u_a^n + \sum_{i \in C_a} \beta_a^n d_i^n \\ \beta_i^n = \text{softmax}(MLP[u_a^n, d_i^n]) \end{cases} \quad (10)$$

In equation (10), q_a^{n+1} is the updated PP temporary node. β_a^n is the weight ratio of adjacent nodes when a user node updates. c_a is all user related data in the figure. MLP is a multi-layer perceptron. A multi-layer perceptron is a simple neural network used to perform nonlinear transformations on the feature vectors of nodes. It typically consists of multiple fully connected layers, each of which can be followed by a nonlinear activation function. The embedding of user nodes and social data nodes have different meanings in each dimension. If attention scores are calculated using functions such as dot product or mean pooling, it will result in inaccurate attention scores. Therefore, attention neural networks are used to calculate the attention scores of each neighboring node, and the results obtained by each neural network are finally normalized. The updated user's social preference

temporary node and personal preference temporary node are weighted and fused to obtain the updated user node. The calculation method is shown in equation (11).

$$\begin{cases} u_a^{n+1} = \mu^{n+1} p_a^{n+1} + \delta^{n+1} q_a^{n+1} \\ \mu^{n+1} + \delta^{n+1} = 1 \end{cases} \quad (11)$$

In equation (11), u_a^{n+1} is the updated user node. μ^{n+1} and δ^{n+1} are the weights of SP temporary nodes and PP temporary nodes in the updated user nodes. Figure 5 shows the user preference fusion process.

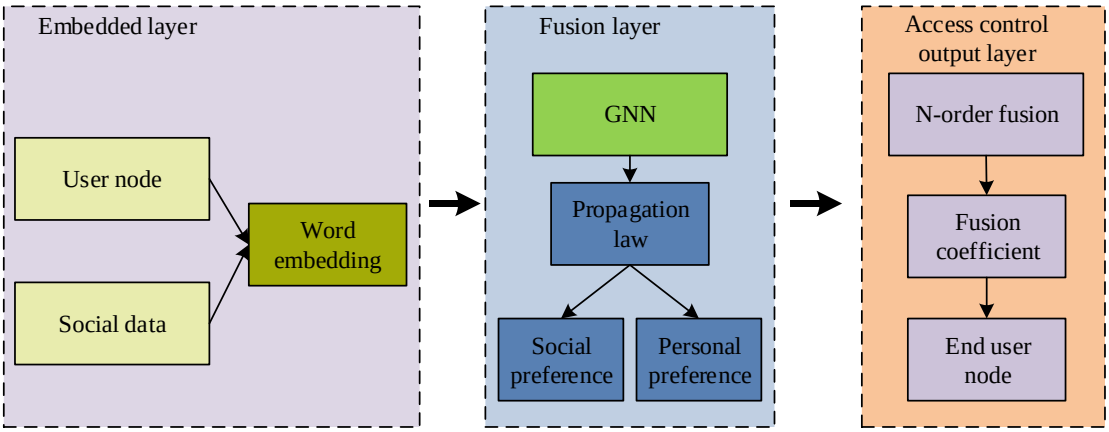


Figure 5: User preference fusion process.

In Figure 5, user preference fusion consists of three parts: embedding layer, preference propagation fusion layer, and access control output layer. In the embedding layer, word embeddings are performed on user nodes and social data as inputs to the model. In the preference propagation fusion layer, two GNNs simulate the propagation and change patterns of user social preferences among users and the propagation and change patterns of user personal preferences in social data. After N rounds of propagation and fusion, user nodes are updated using attention mechanisms based on explicit neighbor nodes, implicit neighbor nodes, and social data nodes. In the access control output layer, in the graph, N user nodes obtained through N preference propagation fusion are used to calculate the fusion coefficient through a linear neural network. Based on the fusion coefficient, the N user nodes are finally fused to obtain the final user nodes with user preferences. The fusion coefficient can quantify the importance or weight of user social preference temporary nodes and user personal preference temporary nodes. Therefore, the calculation of the fusion coefficient is completed by updating the user nodes and normalizing through a nonlinear transformation and a softmax function. At the same time, the user embedding vectors after each propagation are multiplied by their corresponding fusion coefficients, and these weighted embedding vectors are added to obtain the final user node vector. The calculation method is shown in equation (12).

$$\begin{cases} C_{u_a^n} = \text{Softmax}(\tanh(W_4 u_a^n + e) \cdot d^T) \\ u_a = \text{sigmod}(\sum_{n=1}^{N} C_{u_a^n} u_a^n) \end{cases} \quad (12)$$

In equation (12), $C_{u_a^n}$ is the fusion coefficient. W_4 is the fusion weight of user nodes. e is a natural constant. d is the dimension. $\text{sigmod}()$ is the activation function. $\tanh()$ is a hyperbolic tangent function. Finally, the loss function of the model is defined to measure the difference between the predicted results of the model and the true labels, as expressed in equation (13).

$$L = \frac{1}{M} \sum_a -[y_a \cdot log(pr_a) + (1 - y_a) \cdot log(1 - pr_a)] \quad (13)$$

In equation (13), L is the loss and M is the amount of user nodes. y_a means the true label of the a-th user node, with a value of 0 or 1. When access is allowed, it is 1, and when access is prohibited, it is 0. pr_a is the probability of being judged as allowed access. During the training phase, the model will continuously adjust parameters based on the difference between the true labels and the predicted probabilities to minimize the

loss function and optimize the probability estimates allowed for access. According to the loss function, all user nodes are classified based on whether they are allowed access, thus completing access control. The implementation process of the designed ACM is shown in Figure 6.

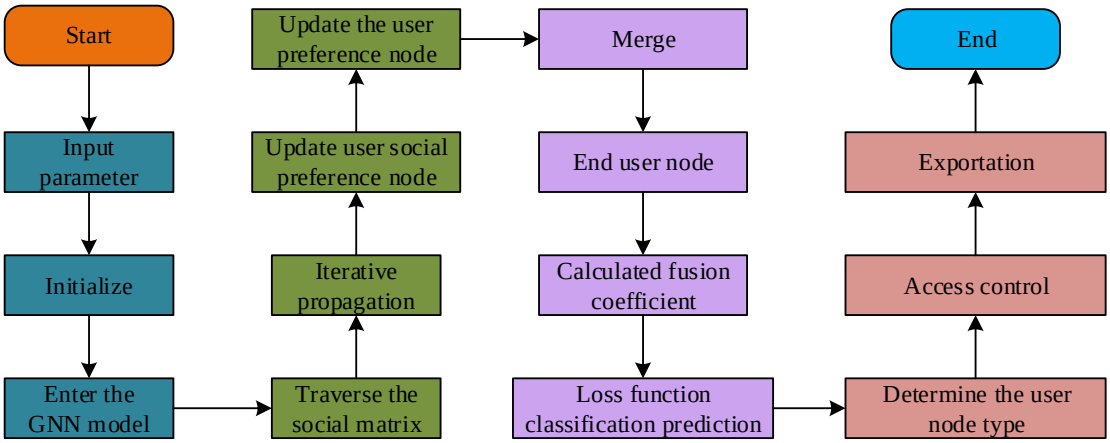


Figure 6: Implementation process of the proposed access control algorithm.

In Figure 6, the original data such as user information, social relationships, and content posted by users are first collected from social networks. Redundant information in the data is eliminated by removing duplicate items, and missing values are filled in to ensure the integrity of the data. The data are then uniformly converted through format conversion to complete the preprocessing of the collected data. The random seed is set to 42 using the random module and numpy library in Python, thereby ensuring that the generated random number sequence is the same every time the code is run. Then, the basic attributes and social behavior of users are extracted to construct user feature vectors, content feature vectors, and social relationship features. The model is used to map user node content to the status space, outputting user node embedding matrices and content node embedding matrices. Then, by analyzing users' social behavior, personal and social preferences are obtained, and a MLA mechanism is introduced to update the representations of user nodes and content nodes, highlighting the influence of important neighbors. Users' social and personal preferences are combined to form a comprehensive user preference representation. Next, using the fused user nodes as input, iterative propagation is performed through GNN to update the node representation. Next, a loss function is defined to measure the difference between the model's predicted results and the true labels, and the model parameters are adjusted to minimize the loss function. Finally, the trained model is employed to classify and predict new user nodes, determine whether to allow access to specific resources, and implement access control policies based on the results of access control decisions. The purpose of these actions is to ensure user privacy protection in social networks.

4 Analysis of ACM results on social networks

This chapter mainly elaborates on the experimental results of the UP-GNN-SNAC model. The first section is a performance analysis of ACMs based on improved GNN. The second section is an analysis of the practical application effect of the ACM based on improved GNN.

4.1 Performance testing of social network ACM

To verify the performance of the proposed UP-GNN-SNAC model, this study conducts simulation experiments using Python 3.7 on a Windows 11 64 bit operating system equipped with an Intel Core i7-14700KF central processor, 16GB of RAM, and 256GB of hard drive. The preferred propagation depth is 5, the learning rate is 0.0001, and the maximum number of iterations is 200. Accuracy is the most intuitive evaluation metric in classification models, representing the proportion of correctly classified samples to the total sample size. It measures the accuracy of the model in classifying user access permissions. The F1 value is the harmonic mean of accuracy and recall, used to comprehensively measure the performance of a model. It can balance the accuracy and recall of the model and avoid bias caused by imbalanced data. Firstly, the Twitter dataset is introduced to calculate the accuracy and F1 value of the research model, and compared with the accuracy and F1 value of traditional GNN and blockchain-based IoT ACMs in reference [20]. The results are shown in Figure 7.

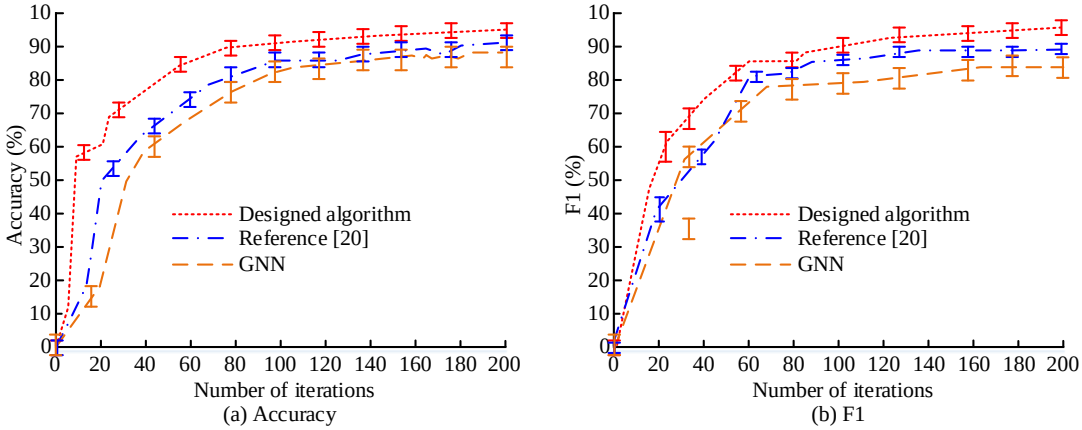


Figure 7: Accuracy and F1 value of different models.

In Figure 7 (a), as the iteration increases, the accuracy of three models shows an upward trend. When iterating 200 times, the accuracy of traditional GNN is 88.3±1.97%, the accuracy of the model in reference [20] is 91.4±2.03%, and the accuracy of the research model is 95.7±2.11%. Compared with traditional GNN and the model in reference [20], the accuracy of the proposed ACM has been improved by 7.4% and 4.3%, respectively. In Figure 7 (b), as iterations increase, the F1 values of various models gradually increase and tend to flatten out. When the iteration reaches its maximum, the F1 values of GNN, the model in reference [20], and the research model are 83.6±1.16%, 89.8±1.09%, and 96.2±1.22%, respectively. Compared with the traditional GCN model and the model in reference [20], the F1 value of the proposed model has increased by 12.6% and 6.4%, respectively. The accuracy and F1 value of the research model are significantly higher, proving its high classification accuracy and good effectiveness. The loss function can be used to measure the difference between the model's predicted results and the true labels. The experiment then introduces the Yelp dataset and calculates the loss of the research algorithm under the Twitter and Yelp datasets. The results compared with the other two algorithms are shown in Figure 8.

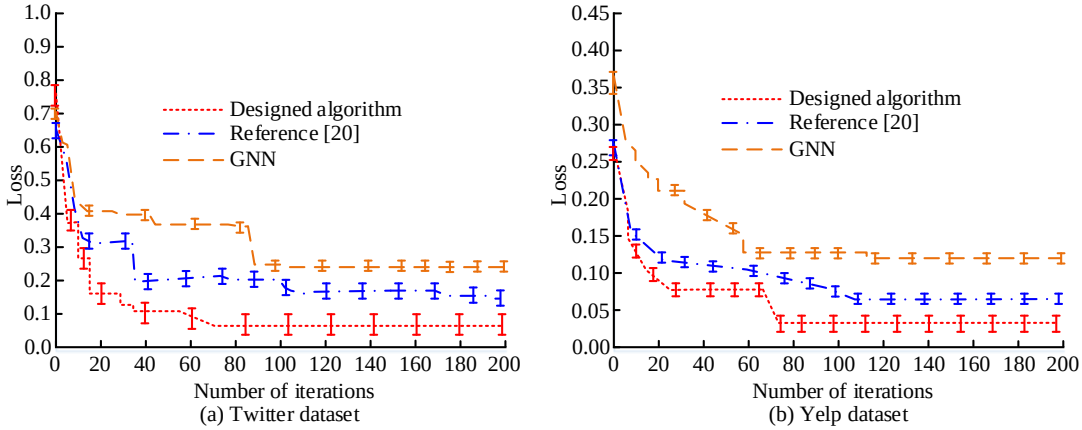


Figure 8: Loss of different algorithms in different datasets.

In Figure 8 (a), on Twitter, as iterations increase, the losses of different models all show a decreasing trend. The loss values of traditional GNN, reference [20] models, and research models are 0.23±0.03, 0.14±0.02, and 0.07±0.03. In Figure 8 (b), the changes in loss curves of different models in Yelp are consistent with those in Twitter. The loss values of the three models are 0.12±0.02, 0.07±0.01, and 0.04±0.01. The loss value of the research model is greatly lower than others, indicating its good generalization ability. It performs well in different datasets, indicating good scalability. To further validate the performance of the proposed model by calculating its accuracy, recall, and Area Under the Curve (AUC) on both the Twitter and Reddit datasets, it is compared with traditional GNN, Graph Convolutional Network, signature schemes based on ciphertext policy attributes in reference [19], and models in reference [20]. The AUC metric is a statistical technique that can comprehensively reflect the model's ability to distinguish between different categories. A higher AUC value indicates that the model can more accurately predict which users should be granted access permissions, thereby reducing the likelihood of erroneously denying legitimate access or erroneously approving illegal access.

Meanwhile, the analysis of variance is used to evaluate the differences between models. ANOVA is a statistical method used to compare whether there is a significant difference in the mean between two or more groups. It is a widely used tool for researching experimental design and data analysis. ANOVA compares the variability between different groups to determine whether the within group variability is significantly smaller than the between group variability. If the inter-group variability is significantly greater than the intra-group variability, it can be concluded that there are significant differences between different groups. The significance level is set to 0.05. If $P<0.05$, it indicates that the difference between groups is statistically significant. Otherwise, the difference between groups is not statistically significant. The results are shown in Table 2.

Table 2: Precision, recall, and AUC values of different models.

Data set	Model	Precision	*P*	Recall	*P*	AUC	*P*
Twitter	GNN	0.782	<0.05	0.825	<0.05	0.791	<0.05
	Graph Convolutional Network	0.825		0.831		0.796	
	Reference [19]	0.865		0.904		0.836	
	Reference [20]	0.903		0.932		0.919	
	Designed algorithm	0.966		0.943		0.982	
Reddit	GNN	0.768	<0.05	0.813	<0.05	0.778	<0.05
	Graph Convolutional Network	0.821		0.846		0.815	
	Reference [19]	0.871		0.913		0.857	
	Reference [20]	0.896		0.935		0.911	
	Designed algorithm	0.972		0.938		0.976	

From Table 2, in the Twitter dataset, the accuracy, recall, and AUC values of the traditional GNN model are 0.782, 0.825, and 0.791, respectively. The three indicators of the graph convolutional network are 0.825, 0.831, and 0.796, respectively. The three indicators of the model in reference [19] are 0.865, 0.904, and 0.836, respectively. The three indicators of the model in reference [20] are 0.903, 0.932, and 0.919, respectively. The three indicators of the proposed model are 0.966, 0.943, and 0.982, respectively. On the Reddit dataset, the accuracies of the five models are 0.768, 0.821, 0.871, 0.896, and 0.972, respectively, with recall rates of 0.813, 0.846, 0.913, 0.935, and 0.938, and AUC values of 0.778, 0.815, 0.857, 0.911, and 0.976, respectively. In different datasets, the accuracy, recall, and AUC values of the three indicators of the proposed model are significantly higher than those of other models, and the differences between the three indicators of the five models are statistically significant ($P<0.05$), proving its good comprehensive performance and reliability. Finally, ablation experiments are conducted on the proposed model to calculate the accuracy, recall, F1 value, and running time of different modules. The results are shown in Table 3.

Table 3: Results of ablation experiment.

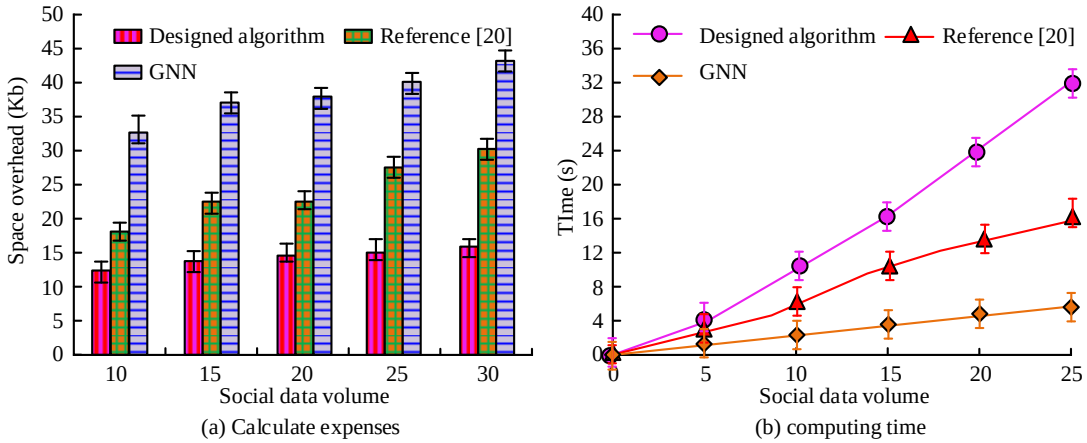
Module	Accuracy	Recall	F1	Running time (s)
Attention module	0.774	0.819	0.807	69.58
GNN module	0.862	0.842	0.853	43.12
Designed algorithm	0.973	0.956	0.961	46.89

From Table 3, the accuracy, recall, F1 value, and running time of the attention module are 0.774, 0.819, 0.807, and 69.58s, respectively. The accuracy, recall, F1 value, and running time of the GNN module are 0.862, 0.842, 0.853, and 43.12s, respectively. The four indicators of the designed model are 0.973, 0.956, 0.961, and 46.89s, respectively. The accuracy, recall, and F1 score of the designed model are higher than those of the two sub-modules, and the running time is higher than that of the attention module, but slightly lower than that of the GNN module. Despite the augmented computational complexity of the model, it has been demonstrated to enhance prediction accuracy. In practical application scenarios, the additional temporal expenditure is deemed justifiable.

4.2 Analysis of the practical application effect of ACM in social networks

To verify the practical application effect of the ACM based on improved GNN, this study first calculates the space overhead and computation time of the research model during encryption and decryption. It is compared with the results of traditional GNN and the model in reference [20]. Space overhead refers to the storage space

occupied during the storage and operation of the model, as shown in Figure 9.



(a) Calculate expenses (b) computing time

Figure 9: The computational cost and time of different models.

In Figure 9, as social data increases, the spatial overhead and computation time of different models gradually increase. When the social data scale is 30, the space overhead of traditional GNN, models in reference [20], and research models is 43.7±3.13Kb, 30.4±3.05Kb, and 16.2±2.88Kb, respectively, with computation times of 32.1±2.79s, 15.9±2.92s, and 5.3±0.97s. The space overhead and computation time of the research model are much lower than other models, which proves its high computational efficiency and low computational complexity. This study validates the access control effectiveness of the research model from seven aspects: User Preference Quantification (UPQ), Historical Records (HR), Privacy Metrics (PM), Sensitivity, User Attributes (UA), Trust, and Personalization. If the effect matches, output 1; otherwise, output 0. Table 4 compares the model with traditional GNN, reference [19], and [20]. Among them, UPQ can meet user needs, HRs are used to evaluate the consistency of user behavior, PMs and sensitivity can ensure data security and compliance, UAs can provide basic access control basis, trust can evaluate the reliability of user behavior, and personalization can improve user experience. The results are shown in Table 4.

Table 4: Access control effectiveness of different models.

Index	GNN	Reference [19]	Reference [20]	Research algorithm
UPQ	0	0	0	1
HR	1	1	1	1
PM	0	1	1	1
Sensitivity	1	1	0	1
UA	1	1	1	1
Trust level	1	1	0	1
Personalization	1	0	1	1

In Table 4, only the research model is consistent in terms of UPQ. In terms of HR and UA, all four models are consistent. In terms of PM, traditional GNN does not comply. The model in reference [20] does not match in terms of sensitivity and trustworthiness. This may be because the model may not have dynamically evaluated user behavior, authentication, or contextual information, resulting in an inability to accurately measure trust levels. In terms of personalization, only the model in reference [19] does not match. The research model is consistent in all 7 aspects, proving that its access control effect is relatively ideal. Finally, the Receiver Operating Characteristic curve (ROC) is introduced. The horizontal axis of the ROC curve represents the false positive rate, which represents the proportion of all negative samples that were incorrectly predicted as positive. The vertical axis represents the true sample rate, which represents the proportion of all actual positive samples correctly predicted as positive samples. The model correctly identifies requests that are actually positive samples as legitimate access and requests that are actually negative samples as illegal access. The ROC curves of the four models are calculated separately, and the results are shown in Figure 10.

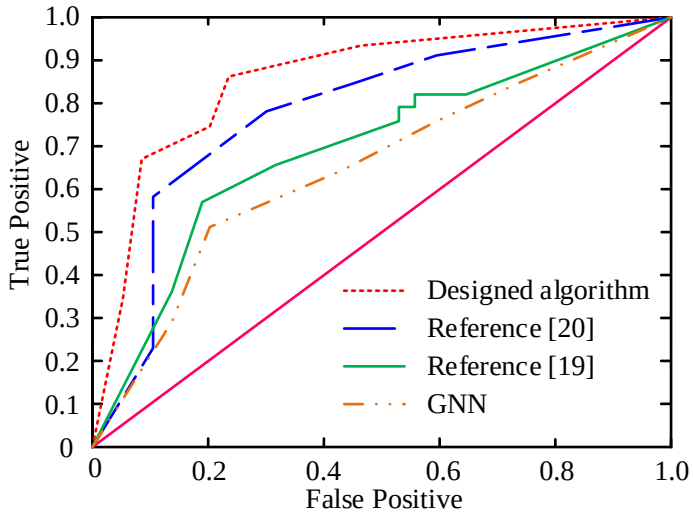


Figure 10: ROC curves and correlation coefficients R of four models.

From Figure 10, the ACM based on traditional GNN is closest to the standard line and has the smallest area under it. The lower area of the model in reference [19] is second, followed by the model in reference [20]. The ROC curve of the proposed model is closer to the upper left corner, with the largest lower area, indicating its strong classification ability and proving the high accuracy of the user preference-based ACM based on the improved GNN.

5 Discussion

The research aims to improve the effectiveness of social network access control by utilizing MLA mechanisms to enhance GNN's understanding of complex social relationships and personal behavior. A GNN-based social network ACM based on user preferences is proposed. The results showed that the accuracy and F1 value of the proposed ACM improved by 7.4% and 12.6% respectively compared to the GNN model and the blockchain-based IoT ACM, demonstrating its high classification accuracy. This is similar to the conclusion drawn by You M et al. [6], while the proposed model is superior. This is because the proposed model optimizes GNN through a MLA mechanism, which can more effectively capture complex patterns of user preferences and social relationships, thereby significantly improving performance. The computation time for encryption and decryption of the proposed model was 5.3 seconds, which was much lower than the GNN model and the blockchain-based IoT ACM. This conclusion is consistent with the findings of Gai K et al. [7], but the running efficiency of the proposed model is higher than that of the method proposed by Gai K et al. This is because the proposed model significantly improves computation time through MLA and information entropy. In summary, the proposed model performs well in multiple aspects. Although the proposed model can more accurately identify legitimate and illegitimate access through user preferences and privacy measurement mechanisms, effectively improving network security, it also increases certain computational overhead. Therefore, in practical applications, debugging needs to be carried out according to specific requirements.

6 Conclusion

ACM is crucial for the security of social networks, as it can help protect sensitive data and prevent malicious attacks and violations. To improve the accuracy and operational efficiency of social network ACM, a new type of ACM was designed based on the preferences of social network users. The user preferences were simulated using GNN, and a MLA mechanism was introduced to improve the model. The experimental results showed that the accuracy and F1 value of the proposed model were 95.7% and 96.2%, respectively, significantly higher than other models. This proved that through GNN and MLA mechanism, the model could dynamically capture user preference features and improve classification accuracy. The space cost of the proposed model was 16.2Kb and the computation time was 5.3s, which was significantly lower than the space cost and computation time of other models. This proved that the model adopted a lightweight GNN architecture, reducing computational complexity and optimizing algorithm design to reduce space cost. Although the proposed ACM has superior performance, there are still some shortcomings. The study did not test it on different types of social platforms, and future research will further test the performance of the model through different social network platforms to improve its universality. At the same time, the performance of the model in dynamic environments will be explored to cope with the constantly changing user behavior and data traffic in social networks.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests

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

Data availability statement

All data generated or analysed during this study are included in this article.

References

[1] Gai T, Cao M, Chiclana F, Zhang Z, Dong Y, Herrera-Viedma E, Wu J. (2023). Consensus-trust driven bidirectional feedback mechanism for improving consensus in social network large-group decision making. *Group Decision and Negotiation*, 32(1): 45-74. https://doi.org/10.1007/s10726-022-09798-7

[2] Kashmar N, Adda M, Ibrahim H. (2022). Access control metamodels: review, critical analysis, and research issues. *Journal of Ubiquitous Systems and Pervasive Networks*, 16(2): 93-102. https://doi.org/10.5383/JUSPN.03.01.000

[3] Wang W, Huang H, Yin Z, Gadekallu T R, Alazab, M, Su C. (2023). Smart contract token-based privacy-preserving access control system for industrial Internet of Things. *Digital Communications and Networks*, 2023, 9(2): 337-346. https://doi.org/10.1016/j.dcan.2022.10.005

[4] Thabit S, Yan L S, Tao Y, Abdullah A B. (2022). Trust management and data protection for online social networks. *IET Communications*, 2022, 16(12): 1355-1368. https://doi.org/10.1049/cmu2.12401

[5] Ameer S, Benson J, Sandhu R. (2022). Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT. I*EEE Transactions on Dependable and Secure Computing*, 2022, 20(5): 4032-4051. https://doi.org/10.1109/TDSC.2022.3216297.

[6] You M, Yin J, Wang H, Cao J, Wang K, Miao Y, Bertino E. (2023). A knowledge graph empowered online learning framework for access control decision-making. *World Wide Web*, 2023, 26(2): 827-848. https://doi.org/10.1007/s11280-022-01076-5

[7] Gai K, She Y, Zhu L, Choo K K R, Wan Z. (2023). A blockchain-based access control scheme for zero trust cross-organizational data sharing. *ACM Transactions on Internet Technology*, 2023, 23(3): 1-25. https://doi.org/10.1145/3511899

[8] Wu H, Ye W, Guo Y. (2023). Data access control method of cloud network secure storage under Social Internet of Things environment. *International Journal of System Assurance Engineering and Management*, 2023, 14(4): 1379-1386. https://doi.org/10.1007/s13198-023-01942-z

[9] Azbeg K, Ouchetto O, Andaloussi S J. (2022). Access control and privacy-preserving blockchain-based system for diseases management. *IEEE Transactions on Computational Social Systems*, 2022, 10(4): 1515-1527. https://doi.org/10.1109/TCSS.2022.3186945

[10] Zhang L, Zhang Y, Wu Q, Mu Y, Rezaeibagha F. (2022). A secure and efficient decentralized access control scheme based on blockchain for vehicular social networks. *IEEE Internet of Things Journal*, 2022, 9(18): 17938-17952. https://doi.org/10.1109/JIOT.2022.3161047

[11] Zhao Y, Yu H, Liang Y, Conti M, Bazzi W, Ren Y. (2023). A sanitizable access control with policy-protection for vehicular social networks. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 25(3): 2956-2965. https://doi.org/10.1109/TITS.2023.3285623

[12] Squicciarini A, Rajtmajer S, Gao Y, Semonsen J, Belmonte A, Agarwal P. (2022). An extended ultimatum game for multi-party access control in social networks. *ACM Transactions on the Web (TWEB)*, 2022, 16(3): 1-23. https://doi.org/10.1145/3555351

[13] Dixit M S, Wajgi M D, Wanjari S. (2022). Real time user access control on social network using deep learning. *International Journal for Research Publication and Seminar*, 2022, 13(2): 246-251. https://jrps.shodhsagar.com/index.php/j/article/view/598.

[14] Wen W, Fan J, Zhang Y, Fang Y. (2022). APCAS: Autonomous privacy control and authentication sharing in social networks. *IEEE Transactions on Computational Social Systems*, 2022, 10(6): 3169-3180. https://doi.org/10.1109/TCSS.2022.3218883

[15] Safi S M, Movaghar A, Ghorbani M. (2022). Privacy protection scheme for mobile social network. *Journal of King Saud University-Computer and Information Sciences*, 2022, 34(7): 4062-4074. https://doi.org/10.1016/j.jksuci.2022.05.011

[16] Ahmed F, Wei L, Niu Y, Zhao T, Zhang W, Zhang D, Dong W. (2022). Toward fine-grained access control and privacy protection for video sharing in media convergence environment. *International Journal of Intelligent Systems*, 2022, 37(5): 3025-3049. https://doi.org/10.1002/int.22810

[17] Salem R B, Aimeur E, Hage H. (2023). A multi-party agent for privacy preference elicitation. *Artificial Intelligence and Applications*, 2023, 1(2): 98-105. https://doi.org/10.47852/bonviewAIA2202514

[18] Mayeke N R, Arigbabu A T, Olaniyi O O, Okunleye O J, Adigwe C S. (2024). Evolving access control paradigms: A comprehensive multi-dimensional analysis of security risks and system assurance in

cyber engineering. *Asian Journal of Research in Computer Science*, 2024, 17(5): 108-124. https://doi.org/10.2139/ssrn.4752902

[19] Patil R Y. (2024). A secure privacy preserving and access control scheme for medical internet of things (MIoT) using attribute-based signcryption. *International Journal of Information Technology*, 2024, 16(1): 181-191. https://doi.org/10.1007/s41870-023-01569-0

[20] Zhonghua C, Goyal S B, Rajawat A S. (2024). Smart contracts attribute-based access control model for security & privacy of IoT system using blockchain and edge computing. *The Journal of Supercomputing*, 2024, 80(2): 1396-1425. https://doi.org/10.1007/s11227-023-05517-4

Fusion of Deep Convolutional Neural Networks and Brain Visual Cognition for Enhanced Image Classification

Xintao Li[1, *], Hongyan Guo[2]
[1]College of Innovation and Entrepreneurship, Henan Open University, Zhengzhou 450046, China
[2]School of Information Engineering and Artificial Intelligence, Zhengzhou Vocational University of Information and Technology, Zhengzhou 450046, China
*Email of Corresponding Author: lxt5168@163.com

Keywords: deep convolutional neural network, brain, visual cognition, intelligent computing model, image classification

Received: December 9, 2024

The brain visual system is one of the core centers for human perception of external information. How to establish the brain visual cognitive system to classify and process image information is a key matter in the area of human-computer connection. In order to improve the accuracy of computer vision image classification, a fusion intelligent computing model based on deep convolutional neural network and brain visual cognition is proposed. This model simulates the visual processing mechanism of the human brain and uses brain computer interface technology to extract electroencephalogram signals, thereby achieving efficient classification and processing of image information. When designing an image classification model based on DCNN, a long short-term memory network structure is introduced to extract time series features of electroencephalogram signals. In order to enhance the classification accuracy of the model, attention mechanism and occlusion independent neural response methods are also applied to improve the accuracy of capturing the correlation information between brain response and image features. The results show that the prediction accuracy of the research model reaches 93.54% and 94.03% in the V4 visual region and L0 visual region, respectively. The highest accuracy on facial visual images reaches 95.46%, while the lowest accuracy on animal visual images is 91.57%. By introducing the long short-term memory module, the loss value of the model decreases from 0.26 to 0.21, with a reduction of 19.23%. In addition, ablation experiments show that by introducing attention mechanisms and occlusion independent neural responses, the final classification accuracy is improved to 93.94%. In summary, the research on the fusion intelligent computing model grounded on deep convolutional neural networks and brain visual cognition effectively improves the accuracy of image classification and demonstrated its potential in the field of intelligent computing.

Povzetek: Predstavljen je inteligentni model za razvrščanje slik, ki združuje globoke konvolucijske nevronske mreže (DCNN) in možgansko vizualno kognicijo preko EEG signalov.

1 Introduction

With the rapid prosperity of artificial intelligence, human-computer interaction has turned into a trend in the current research field. Brain computer interface (BCI), as a cutting-edge scientific research direction, is gradually becoming a meaningful bridge in the area of human-computer connection. The visual system of the human brain has evolved over millions of years and possesses extremely efficient visual processing capabilities. Through multi-level visual processing mechanisms, the brain can quickly and accurately understand complex visual information [1]. When external objects are transmitted to the visual center of the brain through the visual organs, the brain quickly recognizes, classifies, and understands these visual information, thereby forming cognition of the object or scene [2]. BCI can interpret visual cognition of the brain by recording and analyzing electroencephalogram (EEG) signals [3]. The Deep Convolutional Neural Network (DCNN) in computer vision technology has attracted much attention due to its outstanding performance in image processing tasks [4]. sHowever, despite the excellent performance of computer technology in image classification, computers still cannot fully replace the precise image recognition and classification capabilities of the human brain in complex and diverse open environments with interference and occlusion [5]. So, the challenge currently facing the field of computer vision is figuring out how to empower artificial intelligence systems to more effectively mimic human brain cognition and attain precise image classification in intricate scenarios. Therefore, in this context, research innovatively combines the powerful computing power of DCNN with the cognitive characteristics of the brain's visual system, and constructs an intelligent computing model based on the fusion of DCNN and brain visual cognitive information, in order to achieve accurate image classification in complex backgrounds.

The research objectives include designing and implementing an intelligent computing model based on DCNN and EEG signal fusion to improve the performance

of image classification in interference and occlusion environments. The research aims to explore how the model simulates the visual recognition process of the human brain, especially for accurate image classification in complex backgrounds. The research hypothesis is that by combining the visual feedback and image features of the brain, intelligent computing models can simulate the visual recognition process of the human brain, thereby improving the accuracy of classification results. The preset results demonstrate that by introducing visual cognitive information from the brain, the model can mimic the cognitive process of the human brain in actual visual tasks, providing new ideas and directions for the integration of BCIs and intelligent systems.

The research content mainly includes four sections. The second section provides a survey of the current study status of visual EEG picture classification and DCNN around the world. The third section conducts research on intelligent computing models that integrate DCNN and brain visual cognition. The first section proposes the design of a picture sorting model grounded on the fusion of DCNN and brain visual cognition information. The second section designs an intelligent computing model based on the fusion of DCNN and brain visual cognitive information. The fourth section validates the intelligent computing model that integrates DCNN with brain visual cognition.

2 Related works

The visual cognitive ability of the brain can recognize, classify, and understand visual information. In recent years, research on visual interpretation based on monitoring the neural response of the brain during visual cognition has gained the eyesight of numerous professionals and savants. Gao et al. raised an attention-based parallel multi-scale Convolutional Neural Network (CNN) model to improve the accuracy of decoding EEG aroused potentials. The model used two parallel convolutional layers to extract temporal features and utilized attention mechanisms to weight features at different times. The outcomes revealed that the model effectively reformed the interpreting ability of ocular aroused potentials under complex conditions [6]. Ahirwal et al. proposed a new channel selection technique that could identify and characterize harmful emotions aiming to raise the precision of emotion sorting of EEG signals. This technique extracted three forms of characteristics from EEG cues: time-domain characteristics, frequency-domain characteristics, and entropy based characteristics, and used Support Vector Machines (SVM) and artificial neural networks to classify emotions based on the extracted features. The outcomes showed that this way effectively optimized the sorting behaviour [7]. Komolovait et al. raised a way of using CNN combined with stable-state ocular aroused potentials to gain interpretable characteristics from rough EEG cues in order to improve the effectiveness of brain activity data in classifying visual stimuli. This method also introduced generative adversarial networks and variational autoencoders to produce composite EEG cues. The results showed that the method was effective [8]. Kumari et al. proposed a multi-channel EEG movement sorting model to improve the precision of EEG movement sorting. The model utilized CNN to extract descriptive emotional state characteristics from EEG signals and generates two-dimensional images to represent these features. The outcomes revealed that the overall precision of this model reached 83.04% [9].

DCNN occupies a momentous position in EEG picture sorting tasks. Santamaria-Vazquez et al. raised a sorting model grounded on different control signals to extract complex features from EEG data for classification. The model used DCNN for time calibration of BCIs and integrated modules for detection of event-related potentials. The outcomes revealed that the command decoding accuracy of this way improved by 16.0% [10]. Yıldırım et al. raised a novel deep one-dimensional CNN monitoring model to optimize the precision of EEG monitoring. The model utilized machine learning techniques to automatically identify regular and aberrant EEG signals, and classified EEG signals using an end-to-end structure. The outcomes revealed that this way was feasible [11]. Miao et al. raised a multi-layer CNN model using a DCNN structure to raise the classification precision of EEG pattern identification algorithms. The model utilized prior knowledge and complex parameter adjustments to extract spatial frequency features. The outcomes showed that this way had good classification capability [12]. Li et al. proposed a way of using DCNN combined with continuous wavelet transform to enhance the identification rate of limbs action image EEG cues. This method mapped the limbs action image EEG cues to time-frequency image signals using continuous wavelet transform, and input the image signals into the CNN structure to collect characteristics and classify them. The outcomes revealed that this way effectively raised the recognition rate [13]. In recent years, the combination of BCI and DCNN has become an important research direction in the analysis of EEG and brain visual neural activity signals. The detailed progress of BCI is as follows: Tang X et al. proposed an end-to-end BCI method based on CNNs, which directly extracts spatiotemporal features from EEG signals and classifies them. The results showed that this method could achieve higher classification accuracy than traditional manual feature extraction methods, especially in motion imagination tasks and various emotional state classification tasks [14]. In addition, Kawala Sterniuk et al. reviewed over 50 years of using BCIs and concluded that BCI not only enables brain control, but also opens the door for regulating the central nervous system through neural interfaces, demonstrating the potential applications of this technology [15]. The research on integrating BCI and DCNN will provide a more solid foundation for the popularization and application of BCI technology. The comparative summary table is shown in Table 1.

Table 1: Comparison summary table

Study	Method	Advantages	Limitations	Missing features
Gao et al. [6]	Parallel multi-scale CNN based on attention	Improved the decoding performance of visual evoked potentials	Still affected by noise in complex environments, requires processing a large amount of temporal features	Failed to effectively combine spatial and temporal features in the brain's visual cognitive process; cannot adapt to complex environmental visual information processing
Ahirwal et al. [7]	EEG-based emotion classification model, combining SVM and artificial neural networks	Improved emotion classification accuracy	Focuses mainly on emotion classification, lacks deep classification and processing of visual information	Cannot process complex visual information and its complex relationship with emotions
Komolovaitė et al. [8]	Steady-state visual evoked potentials combined with CNNs	Effectively improved visual stimulus classification	Poor robustness to signal noise, high complexity in training generative adversarial networks	Failed to effectively combine visual cognitive mechanisms; limited to static visual stimulus processing
Kumari et al. [9]	Multi-channel EEG-based emotion classification model	Achieved an average accuracy of 83.04%	Focuses on emotion classification, mainly uses image feature representations, lacks handling of more complex scenarios	Cannot handle complex image classification tasks, especially multi-class image recognition
Santamaria-Vazquez et al. [10]	Classification model based on different control signals using DCNNs	Increased command decoding accuracy by 16.0%	Relies heavily on event-related potential detection, may face difficulties in decoding complex EEG data	Lacks adaptability to dynamic EEG signals, unable to combine spatial and temporal features
Yıldırım et al. [11]	EEG monitoring model based on deep one-dimensional CNNs	Provides a feasible classification method	Focuses on normal vs abnormal EEG signal classification, lacks ability to handle complex visual tasks	Cannot effectively process multi-class or dynamically changing visual information
Miao et al. [12]	EEG pattern recognition based on multi-layer DCNNs	Shows good classification performance	Mainly focuses on spatial frequency feature extraction, may be limited in handling complex dynamic tasks	Lacks comprehensive capture of dynamic EEG data or multi-dimensional features of visual information
Li et al. [13]	Classification of left/right hand motor imagery EEG signals combined with continuous wavelet transform and DCNNs	Significantly improved recognition rate	Relies on signal preprocessing, suitable for specific tasks	Cannot process EEG signals related to visual tasks, sensitive to environmental noise

In summary, although existing methods have made some progress in EEG classification tasks, they have certain limitations in handling complex dynamic tasks, enhancing robustness, and adapting to multiple tasks. The research combines the visual cognitive mechanism of the brain with DCNNs and Long Short-Term Memory (LSTM) networks to design a fusion intelligent computing model. This model can more comprehensively capture the spatial and temporal features of EEG signals, solving the problems of lack of adaptability and poor adaptability to complex environments in existing methods. It has higher classification accuracy and wide application prospects.

3 Intelligent computing model integrating DCNN and brain visual cognition

Research receives EEG information through BCI, combines voxel encoding and improved DCNN model to achieve image classification, and uses LSTM to collect temporal characteristics of EEG cues. Attention

mechanism is utilized to raise the accuracy of image feature extraction, and the correlation between brain response and image features is enhanced by masking irrelevant neural responses.

3.1 Design of image classification model based on DCNN and brain visual cognitive information

Neuroscience research has found that the human brain achieves complex cognitive processing through parallel information exchange between dorsal and ventral streams in visual activities [16]. Abdominal flow is a pathway that connects the primary sensory cortex with the temporal and prefrontal regions, primarily responsible for recognizing visual and auditory stimuli and mapping basic information to higher-level semantic concepts [17]. The dorsal flow is responsible for spatial information and motion control. The activity of brain neurons triggered by visual stimuli is called EEG signals, and BCIs can record and measure these signals through biometric technology to reflect the brain's response to behavior. The core area of ventral flow includes the primary visual cortex, ventral intermediate cortex, ventral lower temporal cortex, and other regions. The ventral lower temporal cortex is particularly closely related to complex visual recognition and is the main functional area for object and face recognition. When the brain receives visual stimuli, it stimulates the cortical regions in the ventral stream, transforming simple visual features into higher-level cognitive concepts. For instance, visual information is initially processed by the primary visual cortex and then passed through intermediate areas, ultimately being mapped to the inferior temporal cortex within the ventral stream, where intricate functions like object recognition and color discrimination take place. The dorsal flow is mainly responsible for processing spatial information, motion perception, and action control. Dorsal flow helps the brain perform functions such as object localization, motion tracking, and hand eye coordination through connections with the parietal lobe, motor cortex, and other areas. Therefore, given the core role of ventral flow in image classification tasks, research focuses on analyzing the brain signal response of ventral flow to better understand the process of visual feature extraction and semantic comprehension. The encoding framework for ventral response based on brain visual cognition is shown in Figure 1.

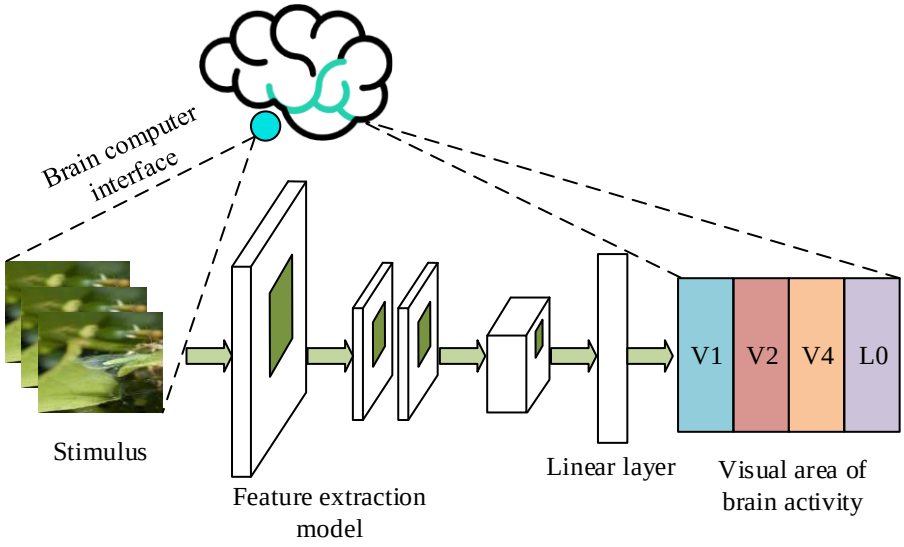


Figure 1: A coding framework for ventral response based on brain visual cognition

As shown in Figure 1, in the ventral response encoding framework based on brain visual cognition, the brain activity caused by visual stimuli can be obtained through the BCI, and the stimulus image can be input into the feature extraction model. After nonlinear calculation, the feature space of the image can be obtained. Then, these features are used to predict the voxel space of the visual region through linear layers. The voxel encoding model transforms human-readable data into a format that machines can store, facilitating the achievement of either shared encoding across various visual regions or unique encoding for specific visual areas. This process aids in pinpointing the regions within the brain's visual cortex that are responsible for processing visual information. [18]. Voxel encoding converts brain activity into a feature space, enabling precise association between cognitive responses and visual stimuli. This mapping helps to reveal the roles of different brain regions in visual information processing, thereby enhancing the accuracy of image classification tasks. The EEG signals capture the electrical activity of the cerebral cortex, which can be mapped to specific regions of the brain through modeling techniques such as source localization, in order to infer activity responses in different areas. This type of method can correlate the spatiotemporal patterns of EEG signals with voxels in functional neuroimaging data. There may be some common neural response patterns between multiple visual regions. These shared response patterns can be captured in voxel encoding models, revealing how these regions collectively respond to the same visual stimuli. For example, in image classification tasks, certain visual regions may exhibit similar neural activity responses to

the same visual features, so voxel encoding can reflect the similarity and interactivity between these regions as a shared encoding pattern. By combining the results of brain visual cognition and image classification, complementary information exchange and expression can be achieved, thereby obtaining a more comprehensive joint representation. DCNN can automatically learn image feature representations by combining convolutional and pooling layers, which helps extract abstract features from data. Therefore, the study adopts DCNN to extract image features and designs a picture sorting model grounded on the fusion of DCNN and brain visual cognitive information, as shown in Figure 2.

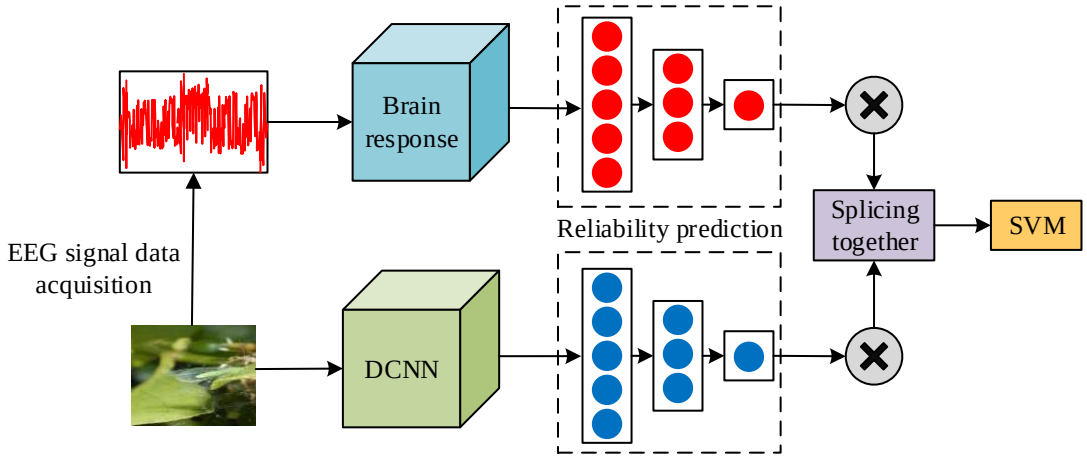


Figure 2: Image classification model grounded on information fusion

As represented in Figure 2, the image classification model based on information fusion mainly includes three parts: characteristic collection structure, characteristic reliability prediction structure, and brain computer information fusion classification structure. The brain response data utilizes the ventral response encoding framework to extract semantic features, while image data is extracted through the DCNN structure for image characteristic collection. Next, the extracted characteristics are input into the feature reliability prediction structure for reliability calculation, and then the fusion weights of image features and brain response characteristic are automatically adjusted. Finally, the fused features are input into the SVM for classification. EEG signals will undergo denoising processing after acquisition, such as bandpass filtering, independent component analysis, signal normalization, and other preprocessing operations to ensure signal quality. Subsequently, EEG signals will be synchronized with the presentation time of visual stimuli to ensure accurate matching between brain responses and image features at each moment. The loss function for reliability prediction is shown in equation (1).

$$L_{MSE} = \sum_{n=1}^{n} (d_p^{'} - d_{f_b}^{'})^2 / n \tag{1}$$

In equation (1), L_{MSE} means the loss function of reliability prediction, $d_p^{'}$ represents the feature reliability prediction value, $d_{f_b}^{'}$ represents the classification sensitivity index of brain response features, and n represents the batch size. The fusion weights of image features are shown in equation (2).

$$w_v = d_{f_v}^{'} / (d_{f_b}^{'} + d_{f_v}^{'}) \tag{2}$$

In equation (2), w_v represents the fusion weight of image features, and $d_{f_v}^{'}$ represents the classification sensitivity index of image features. The fusion weight of brain response features is shown in equation (3).

$$w_b = d_{f_b}^{'} / (d_{f_b}^{'} + d_{f_v}^{'}) \tag{3}$$

In equation (3), w_b represents the fusion weight of brain response features. The math description for the fusion feature is represented in equation (4).

$$f_F = (w_b \times f(b)) concat (w_v \times f(v)) \tag{4}$$

In equation (4), f_F represents fusion features, $f(b)$ represents brain response features, and $f(v)$ represents image features. Due to the fact that EEG signals are collected over a continuous period of time and have time-series characteristics, there is a continuity relationship between the signals at each moment and those before and after [19]. However, although existing feature extraction models perform well in many application scenarios, they often do not fully consider the temporal dependencies in time series data. Especially when it comes to traditional models like CNNs, while they excel at extracting features from images and static data, they can fall short when it comes to capturing temporal information and dynamic signal changes in time-series data, such as EEG signals. In response to this situation, the study uses an LSTM structure to extract time series features of EEG signals. The architecture for extracting brain response features based on time series is shown in Figure 3.

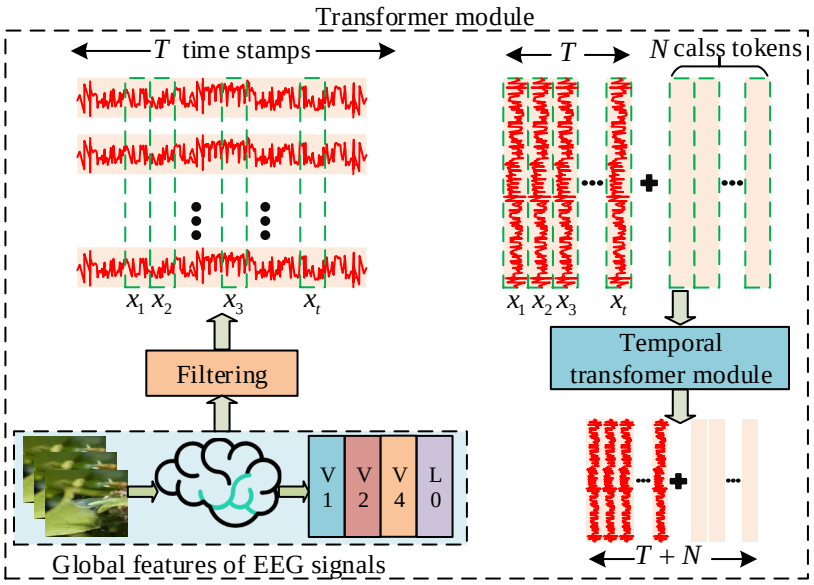


Figure 3: Architecture for extracting brain response features based on time series

As shown in Figure 3, the brain response feature extraction architecture based on time series uses Transformer module to extract global features of EEG signals on time series, and embeds absolute positions to maintain the order of the model. Before inputting positional encoding, the classification identification bits are concatenated with the time series, and then mapped through linear transformation to raise the diversity of characteristic collection. In research models, LSTM is mainly used to integrate brain response data collected from BCIs. The integration process is as follows: Firstly, the brain response signals collected from the BCI system are preprocessed, such as denoising and normalization, to obtain clean time-series data. Then, these preprocessed brain response data are used as inputs for the LSTM network. LSTM networks can capture temporal dependencies in data and learn neural response patterns of the brain at different time points. Next, through the time-dependent modeling of LSTM, the output data contains the gradual response patterns of the brain to visual stimuli throughout the entire image processing process. Finally, the temporal response of the brain is processed by LSTM and combined with image features extracted by DCNN. The calculation for the forget gate of LSTM structure is shown in equation (5).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5}$$

In equation (5), f_t means the output of the forget gate, W_f means the weight of the forget gate, σ means the Sigmoid activation function, b_f represents the offset term of the forget gate, x_t represents the input signal at time t, and h_{t-1} represents the output signal at time $t-1$. The unit update calculation is shown in equation (6).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{6}$$

In equation (6), C_t means the cell condition at time t, C_{t-1} represents the cell condition at time $t-1$, and i_t represents the output matrix of the input gate. The calculation for the output gate is shown in equation (7).

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{7}$$

In equation (7), o_t represents the output gate, W_o means the weight of the output gate, and b_o means the offset term of the output gate. The output features are shown in equation (8).

$$h_t = o_t \cdot \tanh(C_t) \tag{8}$$

In equation (8), h_t represents the output feature. The unique gating mechanism of LSTM can effectively handle the problem of long time intervals and delays in time series, and can discard and store large-span information in EEG data, thus better encoding EEG signals.

3.2 Design of intelligent computing model based on DCNN and brain visual cognitive information

The study simulates the connectivity and classification patterns of biological brain neurons, exploring the connection between picture features and brain reactions. DCNN has demonstrated significant capabilities in image feature extraction. By combining convolutional and pooling layers, it can automatically learn multi-level abstract feature representations of images, effectively capturing low-level and high-level features in images. However, despite DCNN's high efficiency in feature extraction, the image features it extracts still struggle to fully explain the brain's response patterns. This is because the visual cognitive process of the brain not only relies on low-level visual features of images, but also involves complex high-level semantic information processing, perceptual integration, and interaction with other cognitive processes such as memory and emotion. The features extracted by DCNN mainly focus on significant visual information in the image, but these features often lack sufficient high-level semantic

depth and are difficult to fully integrate with the complex responses of the brain in visual cognition. Therefore, the picture characteristics extracted by DCNN are difficult to fully explain the representation information of brain response, and there are some modality specific expressions between the two, making it difficult to deeply explore their deep correlations [20, 21]. In response to this limitation, a network structure based on DCNN is studied to construct an intelligent computing model for the brain. The brain response is used as supervised information for images, and difficult to interpret high-level semantic information is transferred to the DCNN model to achieve more accurate mining of the brain's visual cognitive response. The intelligent computing model structure grounded on data fusion is represented in Figure 4.

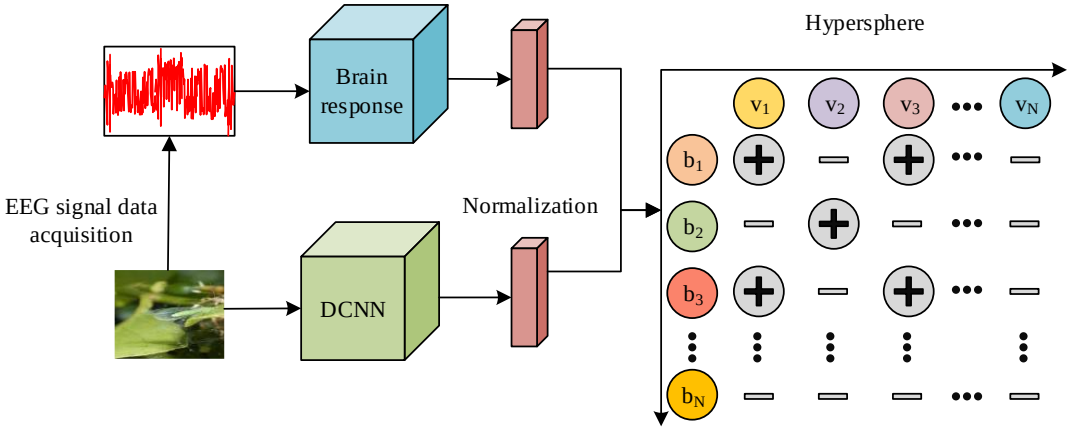


Figure 4: Intelligent computing model structure grounded on data fusion

As represented in Figure 4, in the intelligent computing model framework based on information fusion, characteristic extraction is first constructed on the cognitive response data of the brain to visual images collected by the BCI. Then, the DCNN structure is applied to collect characteristics from the input picture. After normalizing the two extracted features separately, the fused features are mapped onto an N-dimensional sphere. Subsequently, based on the normalized features, a set of positive and negative samples are constructed, and the InfoNCE loss function is used for calculation, thereby achieving the transfer of correlated information between the two feature maps. The math description of the InfoNCE loss function is represented in equation (9).

$$L_i = -log \frac{exp(S(z_i, z_i^+)/\tau)}{\sum_{j=0}^{N} exp(S(z_i, z_j)/\tau)} \quad (9)$$

In equation (9), L_i represents the InfoNCE loss function, τ represents the temperature coefficient, z_i represents the image representation corresponding to the input data x_i, $S(z_i, z_j)$ represents the cosine similarity between image representations, $S(z_i, z_i^+)$ represents the alignment characteristics during hypersphere mapping, and N represents the total amount of positive and negative samples. The calculation for the comparative loss is represented in equation (10).

$$L_i' = -log \frac{\sum_{j=0}^{m} exp(S(f(v_i), f(b_j^+))/\tau)}{\sum_{k=0}^{n} exp(S(f(v_i), f(b_k^-))/\tau)} \quad (10)$$

In equation (10), L_i' represents the contrastive loss, m means the amount of positive samples, n means the amount of negative samples, $f(v_i)$ means the mapped image features, $f(b_j^+)$ represents brain response features of the same category as the image features, and $f(b_j^-)$ represents brain response features of different categories from the image features. In intelligent computing models based on the fusion of DCNN and brain visual cognitive information, the DCNN structure may be affected in classification accuracy due to irrelevant information. To address this issue, research is being conducted to improve the DCNN structure by incorporating attention mechanisms. The DCNN feature extraction model grounded on attention mechanism is represented in Figure 5.

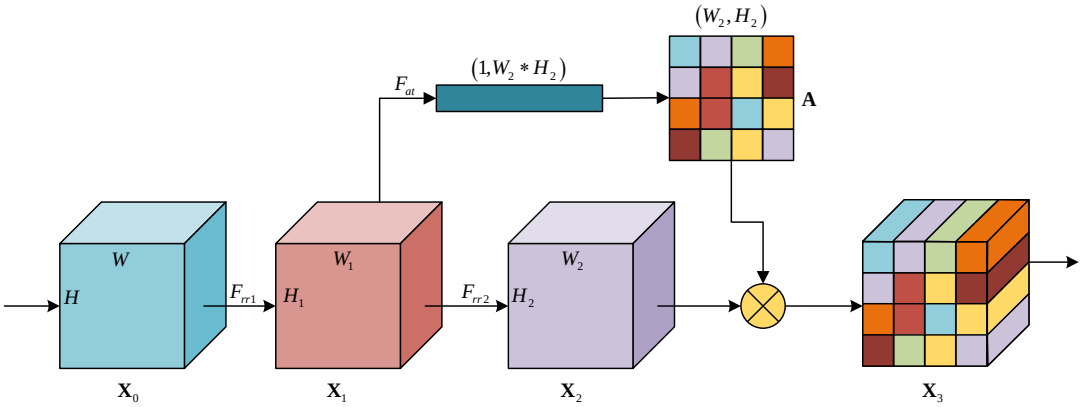


Figure 5: DCNN feature extraction model grounded on attention mechanism

As represented in Figure 5, the DCNN architecture used mainly includes multiple convolutional layers, pooling layers, activation functions, and fully connected layers, aiming to extract multi-level feature representations from images to improve classification performance. The core idea of DCNN is to extract local features from images through a series of convolution and pooling operations, and then add nonlinear transformations through nonlinear activation functions to learn more complex image representations. The convolutional layer, as a fundamental component in the DCNN architecture, can extract local features of the input image through convolution operations. After each convolution operation in each layer, the study will use activation functions to perform nonlinear transformations on the output results. The purpose of the activation function is to introduce nonlinear factors, so that the network can learn more complex mapping relationships. The function of the pooling layer is to downsample the feature map output by the convolutional layer, thereby reducing the spatial size of the feature map while preserving important features. After extracting sufficient local features in the convolutional and pooling layers, the last few layers are usually fully connected layers. The fully connected layer linearly combines the extracted features and generates the final output result through an activation function. The DCNN feature extraction model grounded on attention mechanism adds a parallel attention branch to the initial DCNN structure to learn the importance information of feature map position. This path can correct the activation values of feature maps, reduce the activation values of redundant information, and thus improve the accuracy of image characteristic collection [22, 23]. The feature transformation process is shown in equation (11).

$$\mathbf{X}_1 = F_{rr1}(\mathbf{X}_0) \tag{11}$$

In equation (11), $\mathbf{X}_1$ represents the transformed abstract feature map, $\mathbf{X}_0$ represents the initial feature map, and F_{rr1} represents the downsampling operation. The calculation for position importance is shown in equation (12).

$$\mathbf{A} = F_{at}(\mathbf{X}_1) \tag{12}$$

In equation (12), $\mathbf{A}$ represents positional importance and F_{at} represents fully connected operation. The new feature map obtained by further downsampling the abstract features is shown in equation (13).

$$\mathbf{X}_2 = F_{rr2}(\mathbf{X}_1) \tag{13}$$

In equation (13), $\mathbf{X}_2$ represents the new feature map after further downsampling, and F_{rr2} represents the further downsampling operation. The corrected feature map is shown in equation (14).

$$\mathbf{X}_3 = \sum_{i=1}^{W_2} \sum_{j=1}^{H_2} \mathbf{A}(i, j) \cdot \mathbf{X}_2(i, j) \tag{14}$$

In equation (14), $\mathbf{X}_3$ represents the feature map obtained after attention branch correction, W and H represent the width and height of the characteristic map, and (i, j) represents the feature values on the feature map. When capturing the correlation information between brain visual cognitive responses and image features, some non-correlated neural responses may affect the determination of representation similarity. These "unrelated neural responses" pertain to neural activities that aren't directly tied to visual tasks and might stem from background noise, irrelevant visual cues, or various other bodily influences. For example, the activity of certain regions in EEG signals may be unrelated to the current visual task, and this irrelevant neural activity can lead to misleading similarity judgments when the brain processes visual information. To address this issue, research has been conducted on an intelligent computing model based on the fusion of DCNN and brain visual cognitive information, which devotes to raise the precision of capturing correlated information by masking non-correlated neural responses. In the intelligent computing model based on the fusion of DCNN and brain visual cognitive information, the study aims to add windows of different scales to the extracted image features to mask non-correlated neural responses. The motivation of this method is to better highlight the effective response of the brain to visual information and improve the accuracy of similarity determination between brain visual cognitive responses and image features by reducing or eliminating the influence of irrelevant neural reactions. The visualization process of brain response and image feature correlation information is shown in Figure 6.

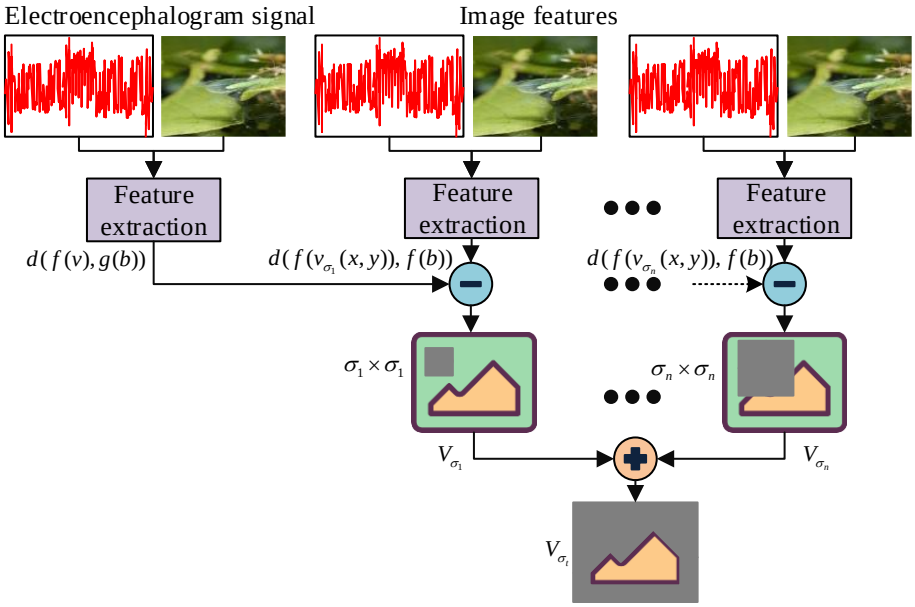


Figure 6: Visualization process of brain response and image feature correlation information

As shown in Figure 6, in the process of visualizing the correlation information between brain response and image features, the correlation features are first represented in the shared representation space and their Euclidean distance is calculated. Then, a window with a scale of $\sigma_i \times \sigma_i$ is added to the extracted image features to mask non-correlated neural responses. Based on the Euclidean distances calculated from various image features, different sizes of occlusion windows are determined. The saliency maps obtained through occlusion at different scales are then combined to create a comprehensive saliency map that encapsulates the relationship between brain responses and image features. The calculation for the significance map is shown in equation (15).

$$V_{\sigma_t} = |d(f(v_{\sigma_t}(x,y)),f(b)) - d(f(v),g(b))| \quad (15)$$

In equation (15), V_{σ_t} represents the significance map and d represents the Euclidean distance.

4 Validation of an intelligent computing model integrating DCNN and brain visual cognition

After setting up the experimental environment, the behaviour of the image classification model grounded on information fusion was first verified, and then the intelligent computing model based on information fusion was experimentally analyzed.

4.1 Experiment environment construction

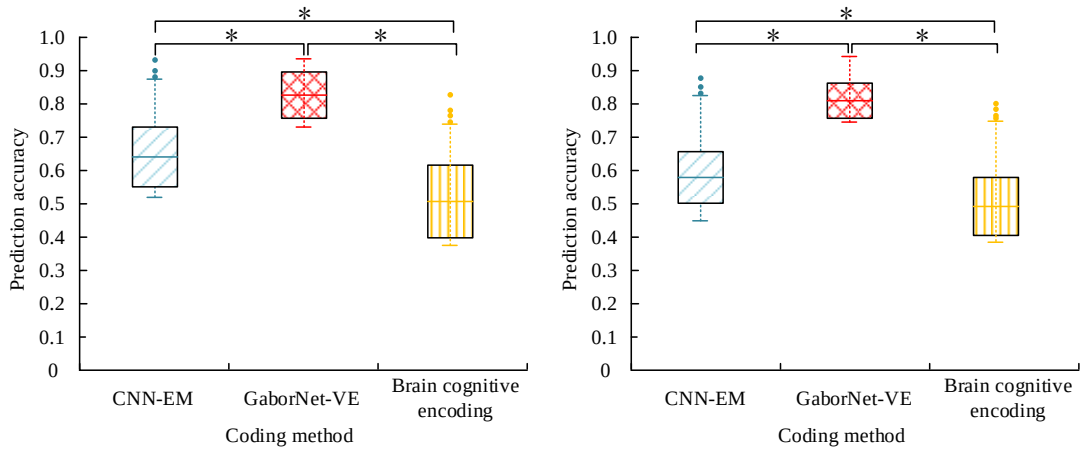
To tesify the effectiveness of the intelligent computing model that integrated DCNN and brain visual cognition, the study first conducted the construction of an experimental environment. The experimental hardware system configuration was as follows: the processor was Intel i7-8700, the GPU was Nvidia GeForce 1080Ti, and the memory was 64 GB DDR4. The experimental model used Python language and was implemented using the PyTorch framework. The experimental parameters were set as follows: batch size was 16, original learning rate was 0.001, Adam optimizer was used during training, output layer size was 40, and the key vector value in the self attention mechanism was 128. The dataset was sourced from the comprehensive evaluation platform Brain Score. This dataset aimed to evaluate the effectiveness and accuracy of computer simulated brain operation models, thus covering response data of primate visual systems. The dataset contains approximately 5000 image stimuli, each corresponding to a recorded brain electrophysiological response data. The stimulus images cover a total of 40 categories, including natural scenes and artificial objects. The number of images in each category is roughly equal to ensure data balance. The size of each image is 224x224 pixels, which can retain sufficient visual information and meet the input requirements of CNN. Any data augmentation techniques used by the research include random cropping, horizontal flipping, random rotation, and color jitter. The above data augmentation techniques can effectively expand the diversity of training data, avoid model overfitting, and improve the generalization ability to various visual stimuli. After preprocessing, the data was separated into a training set and a testing set in a 3:7 ratio. While primarily intended for evaluating brain functional models, the "Brain Score" dataset is well-suited as a data source in this study to verify the efficacy of intelligent computing models that integrate image classification with brain visual cognition, given its abundance of visual stimulus images and corresponding EEG response data. In the experimental design of this study, the evaluation of image classification focuses on guiding the learning and classification of image features through brain response data, rather than simply image classification. The detailed experiment environment configuration and network training parameters are represented in Table 2.

Table 2: Experiment environment configuration and network training parameters

Experimental environment	Configuration	Training parameters	Configuration
CPU	Intel i7-8700	Batch Size	16
GPU	Nvidia GeForce 1080Ti	Initial learning rate	0.001
Memory	64 GB DDR4	Output layer size	40
programming language	Python	Key vector value	128
Frame	PyTorch	Optimizer	Adam

4.2 Performance verification of image classification model based on information fusion

In order to verify the predictive accuracy of ventral response encoding based on brain visual cognition for brain cognitive response, this method was compared and analyzed with other voxel encoding methods, including Convolutional Neural Network Enhancement Model (CNN-EM) and GaborNet Visual Encoding (GaborNet-VE). The accuracy comparison of different encoding methods in different visual regions is represented in Figure 7. From Figure 7(a), within the V4 visual region, the prediction accuracy of the ventral response encoding method based on brain visual cognition was significantly higher than the other two methods. The maximum prediction accuracy of this method reached 93.54%, which was 6.05% and 19.57% higher than the maximum prediction accuracy of CNN-EM and GaborNet-VE, which were 87.49% and 73.97%, separately. From Figure 7(b), the results of visual area L0 show that the maximum prediction accuracy of the ventral response encoding method based on brain visual cognition was 94.03%, which was 11.49% and 18.95% higher than the maximum accuracy of 82.54% and 75.08% of the other two methods, respectively. In addition, the study used paired t-tests to validate the credibility of the results. In the V4 region, the difference in accuracy between the ventral response encoding based on brain visual cognition and CNN-EM reached a statistically significant level ($t=4.72$, $P<0.05$). The difference in accuracy between ventral response encoding based on brain visual cognition and GaborNet VE also reached a statistically significant level ($t=6.88$, $P<0.05$). In the L0 region, the accuracy difference between ventral response encoding based on brain visual cognition and CNN-EM reached a statistically significant level ($t=5.23$, $P<0.05$). Similarly, the accuracy difference between ventral response encoding based on brain visual cognition and GaborNet VE was also statistically significant in the L0 region ($t=7.14$, $P<0.05$). Ventral response encoding based on brain visual cognition could accurately predict brain cognitive response.



(a) Comparison of Accuracy in V4 Visual Region

(b) Comparison of Accuracy in L0 Visual Region

Figure 7: Comparison of accuracy of different encoding methods （*Indicating P<0.05）

To further testify the performance of the image classification model based on information fusion, a relative unpack was operated on the classification models before and after adding LSTM, as represented in Figure 8. From Figure 8, the loss value of the model before adding LSTM converged to 0.26, while the loss value of the model after inserting LSTM converged to 0.21, with a reduction of 19.23%. This indicated that the classification model incorporating LSTM had better convergence performance.

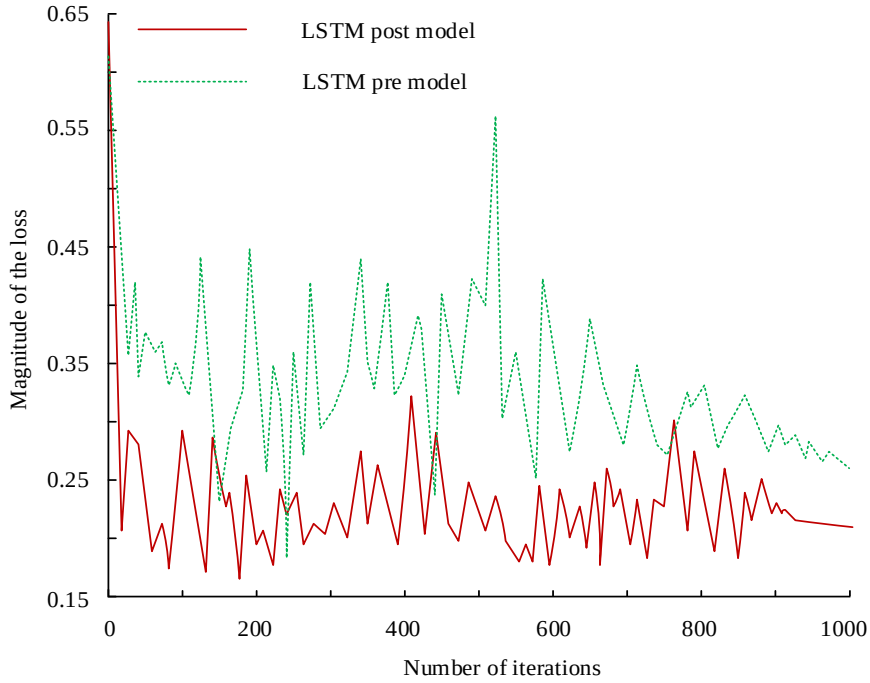


Figure 8: Comparison of training loss values for networks incorporating LSTM Models

To further validate the capability of the image classification model grounded on information fusion, this study compared and analyzed the model with other advanced image classification models, including Feature Weighted Classification (FWC), Residual Network (ResNet), and Visual Geometry Group (VGG). In addition, to ensure the broad applicability and contextualization of the research results, the performance of these models was compared with benchmark test results in the current field of computer vision. The accuracy comparison of different classification models is represented in Figure 9. From Figure 9, in the datasets of various visual images, the accuracy of the picture sorting model grounded on information fusion was the best. On facial vision images, the accuracy of this model was as high as 95.46%, which was an improvement of 6.30%, 5.41%, and 10.03% compared to the accuracy of 89.16%, 90.05%, and 85.43% of FWC, ResNet, and VGG, respectively. This result showed significant advantages compared to the mainstream benchmarks in the current field of facial recognition. In facial recognition tasks, many of the most advanced facial recognition technologies, such as FaceNet and ArcFace, achieved high accuracy on multiple standard datasets such as LFW and CASIA WebFace. For example, FaceNet reported an accuracy of 94.63% on the LFW dataset [6]. Moreover, ArcFace also achieved a recognition rate of nearly 94.51% on the same dataset [7]. However, the above studies all achieved accuracy in interference free environments, while this study still achieved an accuracy of up to 95.46% in actual environments with complex interference and occlusion. Compared with existing benchmark tests, the researched image classification model based on information fusion had more advantages in performance. On animal visual images, the classification accuracy of this model was the lowest at 91.57%, which was 16.31%, 12.03%, and 19.08% higher than the accuracy of 75.26%, 79.54%, and 72.49% of the other three models, respectively. Compared with basic testing image classification tasks, animal classification often faces more complex backgrounds and varying object shapes, which makes this task an important criterion for testing model robustness. Therefore, the significant improvement of the research model in this task indicated that it has stronger generalization ability and adaptability when facing highly complex and dynamically changing visual environments. In summary, the image classification model based on information fusion has demonstrated excellent classification performance in multiple tasks. Moreover, the performance of the research model still has significant advantages compared to benchmark testing.

The reason why facial image recognition had better accuracy than animal image recognition was that facial images had more stable and easily recognizable features compared to animal images. Facial recognition typically fixed structural features and relatively consistent backgrounds, which enabled information fusion based models to fully exploit the effective information in the brain's visual cognitive model, thereby improving recognition accuracy. However, animal images face more complex challenges, including background noise, changes in animal size and morphology, different shooting angles, different species, etc. These factors make classification tasks more complex and varied. Therefore, in terms of recognition accuracy, the performance of facial image classification was better than that of animal image classification.

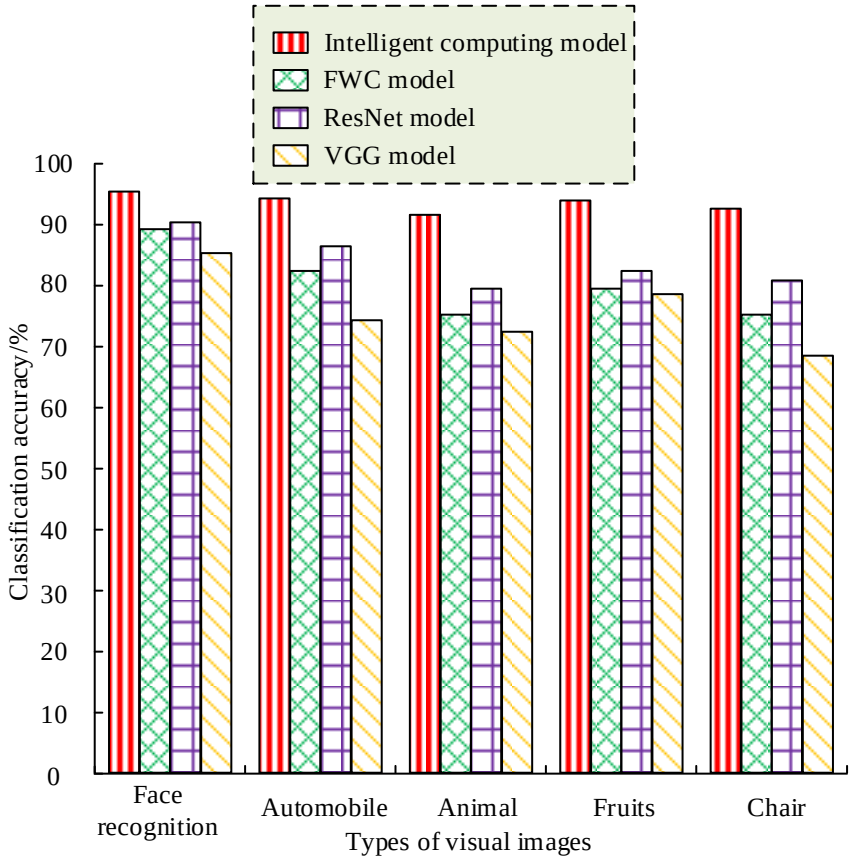
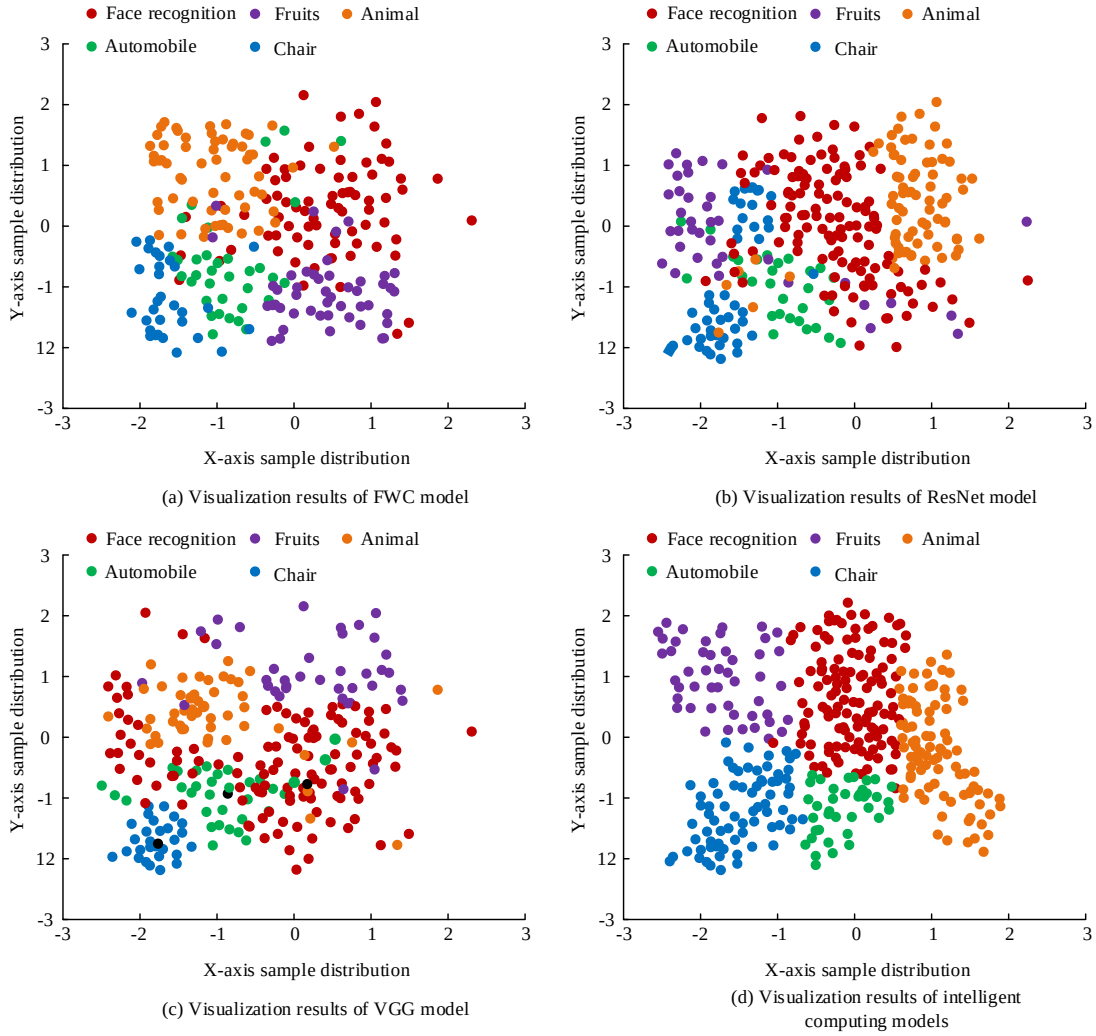


Figure 9: Comparison of accuracy between different classification model

4.3 Performance verification of intelligent computing models based on information fusion

To tesify the ability of intelligent computing models grounded on information fusion, the visualization sample distribution results of different models under various brain visual cognitive image stimuli were compared and studied. The dataset contains 40 categories of images, which are divided into two main categories: natural scenes and artificial objects. Natural scenes include image categories such as faces and animals, while artificial objects include image categories such as cars, fruits, chairs, etc. In the experiment, a combination of these image categories was used to test the classification performance of the model under different visual stimuli. The visualization outcomes of various models are represented in Figure 10. From Figure 10, the sample distribution of FWC and VGG was relatively chaotic, while the sample distribution of ResNet was relatively clear. The ResNet model had a more obvious distinction between facial and animal visual images, but it was more confusing in distinguishing images such as fruits and cars. The intelligent computing model based on information fusion studied exhibited significant classification clarity and good classification performance under all visual image stimuli. This was because images of facial and animal categories were more consistent in natural scenes and were easily distinguishable by models. However, categories such as cars, fruits, and chairs belong to the category of artificial objects, and the visual differences between these categories were significant, posing greater challenges to the model. The intelligent computing model based on information fusion revealed the deep level features of brain response, effectively improving classification accuracy.



(a) Visualization results of FWC model

(b) Visualization results of ResNet model

(c) Visualization results of VGG model

(d) Visualization results of intelligent computing models

Figure 10: Visualization results of different models

To further validate the ability of the intelligent computing model grounded on information fusion, ablation experiments were conducted. The classification accuracy in ablation experiments was calculated based on the precision of image classification tasks, which only reflected the accuracy of image classification results. The ablation experiment results of image classification are shown in Table 3. From Table 3, the classification accuracy of the brain visual cognitive response encoding framework was 81.42%. When the DCNN structure was fused, the accuracy was improved by 5.64%. After adding the LSTM structure, the accuracy was increased to 90.48%. When attention mechanism was added for improvement, the classification accuracy increased by 2.17%. When further optimizing using occluded non-correlated neural responses, the accuracy of the model reached 93.94%. From the above, it can be seen that the addition of the above modules brought benefits to the classification performance of the model, effectively raising the classification accuracy of images.

Table 3: Ablation experiment

Brain Response Coding	DC	LS	Attention	Obstructing non correlated	Accuracy
Framework	NN	TM	mechanism	neural responses	rate/%
√	/	/	/	/	81.42
√	√	/	/	/	87.06
√	√	√	/	/	90.48
√	√	√	√	/	92.65
√	√	√	√	√	93.94

Note： "√" indicates the existence of the module； "/" indicates its non existence

5 Discussion

In order to improve the accuracy of computer vision image classification, a fusion intelligent computing model was constructed by simulating the visual processing mechanism of the human brain, using BCI technology to extract EEG signals generated by human visual cognition, and combining DCNN structure. The results showed that after adding LSTM, the convergence of the model was significantly improved, with the loss value decreasing

from 0.26 to 0.21, indicating a 19.23% increase in convergence speed. This indicated that LSTM could effectively capture time series features, improve the model's ability to process time-series data, and thus make the model more accurate in learning dynamic information. After incorporating the advantages of LSTM into the model, it could better understand the temporal dependencies in brain activity, resulting in more accurate prediction performance. In addition, compared with other advanced methods, the research method was significantly superior to other methods. For example, although the model studied by Gao Z et al. effectively improved the decoding performance of VEP in complex environments, it still faced the problem of noise interference and failed to effectively integrate spatial and temporal features in the brain's visual cognitive process [6]. The model studied in this article not only considers spatial features but also integrates dynamic temporal information when predicting brain responses in visual regions, significantly improving the accuracy of predictions. In addition, the model proposed by Ahirwal M K et al. achieved good results in emotion classification, but it mainly focused on emotion classification and cannot handle complex visual information and multi-class image classification tasks [7]. The model studied in this article could not only handle complex visual information, but also adapt to the multidimensional features of the brain's visual cognitive process, thus exhibiting a more comprehensive classification and understanding of visual information.

The potential extensions of the research model to other tasks include video analysis, multi-modal data fusion, etc. Video data not only contains spatial information of static images, but also dynamic time series information. Therefore, models based on brain visual cognition can better understand the dynamic changes in videos by integrating spatial and temporal features, especially with the addition of LSTM modules. In the field of multi-modal data fusion, cross modal learning can be achieved by introducing multi-modal neural network structures and combining data from different modalities. For example, in video description generation tasks, visual information of video frames can be combined with speech or text information to generate more accurate and natural descriptions.

The reason why the research method is superior to other methods is that it considers the spatial and temporal characteristics of the brain in the visual cognitive process, while other methods rely more on a single spatial or static feature. In addition, the introduction of LSTM further enhances the model's ability to process temporal information, enabling the model to decode complex dynamic brain signals more accurately. The potential applications of this discovery cover fields such as neuroscience experiments, intelligent medical devices, and brain computer interaction systems. However, this method still has certain limitations. For example, the study only explored the classification of EEG images, so the research results are not comprehensive enough. This aspect can be further improved in the future.

6 Conclusion

In recent years, the introduction of visual cognitive mechanisms in the brain has provided new solutions to the limitations of accuracy and generalization ability of traditional DCNN in processing complex visual information. Research used BCIs to receive EEG information, used voxel encoding models to obtain the expression content of visual images, and combined an improved DCNN structure to construct an efficient image classification model. On this basis, the LSTM structure was further introduced to extract time series features of EEG signals. Attention mechanisms and occlusion independent neural responses were utilized to enhance the accuracy of capturing correlation information between brain responses and image features. The outcomes revealed that the ventral response encoding method grounded on brain visual cognition achieved prediction accuracy of 93.54% and 94.03% in the V4 visual region and L0 visual region, significantly better than the CNN-EM and GaborNet VE methods. In the model validation, after adding the LSTM module, the loss value decreased from 0.26 to 0.21, with a reduction of 19.23%. In terms of image classification capability, the accuracy of the information fusion based model on facial visual images was as high as 95.46%, and the lowest accuracy on animal visual images was 91.57%, both significantly better than comparative models such as FWC, ResNet, and VGG. In addition, ablation experiments showed that by introducing attention mechanisms and occlusion independent neural responses, the final classification accuracy was improved to 93.94%. From the above, the research on the fusion intelligent computing model based on DCNN and brain visual cognition effectively improved the accuracy of computer vision image classification.

Although research focused on EEG image classification and achieved good classification results in the relevant areas of ventral flow and visual regions, the current scope of research has not yet covered other brain tissue and neural mechanisms. Therefore, future research can be extended to explore the functions of other brain regions, such as their contributions to tasks such as cognitive control and emotion recognition in different brain regions. In addition, combining different neural mechanisms and multi-modal data will help improve the comprehensiveness and accuracy of cognitive image classification, thereby promoting further development in the field of BCIs. Future work will strive to further enhance the analytical ability of EEG information for complex visual stimuli through the integration of broader neural regions and mechanisms, in order to promote the widespread application of intelligent computing models in practical applications.

Funding

This study was supported by the Key Science and Technology Program of Henan Province (Project Name: Research on NoC Routing Algorithm and Fault-Tolerant Technology Based on Spanning Tree Sub-Domains; Grant No. 252102210225).

References

[1] Wilson H, Chen X, Golbabaee M, Proulx, M. J., & O'Neill, E. Feasibility of decoding visual information from electroencephalogram. Brain-Computer Interfaces, 2024, 11(1-2): 33-60. DOI: 10.1080/2326263X.2023.2287719

[2] Finlayson S G, Subbaswamy A, Singh K, Bowers, J, Kupke, A., Zittrain, J & Saria, S. The clinician and dataset shift in artificial intelligence. New England Journal of Medicine, 2021, 385(3): 283-286. DOI: 10.1056/NEJMc2104626

[3] Masana M, Liu X, Twardowski B, Menta, M., Bagdanov, A. D & Van De Weijer, J. Class-incremental learning: survey and performance evaluation on image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(5): 5513-5533. DOI: 10.1109/TPAMI.2022.3213473

[4] Zhu Y, Zhuang F, Wang J, Ke, G., Chen, J., Bian, J & He, Q. Deep subdomain adaptation network for image classification. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(4): 1713-1722. DOI: 10.1109/TNNLS.2020.2988928

[5] Hong D, Gao L, Yao J, Zhang, B., Plaza, A & Chanussot, J. Graph convolutional networks for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 2020, 59(7): 5966-5978. DOI: 10.1109/TGRS.2020.3015157

[6] Gao Z, Sun X, Liu M, Dang, W., Ma, C., & Chen, G. Attention-based parallel multiscale convolutional neural network for visual evoked potentials electroencephalogram classification. IEEE Journal of Biomedical and Health Informatics, 2021, 25(8): 2887-2894. DOI: 10.1109/JBHI.2021.3059686

[7] Ahirwal M K, Kose M R. Audio-visual stimulation based emotion classification by correlated electroencephalogram channels. Health and Technology, 2020, 10(1): 7-23. DOI: 10.1007/s12553-019-00394-5

[8] Komolovaitė D, Maskeliūnas R, Damaševičius R. Deep convolutional neural network-based visual stimuli classification using electroencephalography signals of healthy and alzheimer's disease subjects. Life, 2022, 12(3): 374-379. DOI: 10.3390/life12030374

[9] Kumari N, Anwar S, Bhattacharjee V. Time series-dependent feature of electroencephalogram signals for improved visually evoked emotion classification using EmotionCapsNet. Neural Computing and Applications, 2022, 34(16): 13291-13303. DOI: 10.1007/s00521-022-06942-x

[10] Santamaria-Vazquez E, Martinez-Cagigal V, Vaquerizo-Villar F, & Hornero, R. electroencephalogram-inception: a novel deep convolutional neural network for assistive ERP-based brain-computer interfaces. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020, 28(12): 2773-2782. DOI: 10.1109/TNSRE.2020.3048106

[11] Yıldırım Ö, Baloglu U B, Acharya U R. A deep convolutional neural network model for automated identification of abnormal electroencephalogram signals. Neural Computing and Applications, 2020, 32(20): 15857-15868. DOI: 10.1007/s00521-018-3889-z

[12] Miao M, Hu W, Yin H, & Zhang, K. Spatial-Frequency feature learning and classification of motor imagery electroencephalogram based on deep convolution neural network. Computational and Mathematical Methods in Medicine, 2020, 2020(1): 1981728-1981752. DOI: 10.1155/2020/1981728

[13] Li F, He F, Wang F, Zhang, D & Li, X. A novel simplified convolutional neural network classification algorithm of motor imagery electroencephalogram signals based on deep learning. Applied Sciences, 2020, 10(5): 1605-1624. DOI: 10.3390/app10051605

[14] Tang X, Shen H, Zhao S, Li, N., & Liu, J. Flexible brain‐computer interfaces. Nature Electronics, 2023, 6(2): 109-118. DOI: 10.1038/s41928-022-00913-9

[15] Kawala-Sterniuk A, Browarska N, Al-Bakri A, Pelc, M., Zygarlicki, J., Sidikova, M., et al. Summary of over fifty years with brain-computer interfaces—a review. Brain Sciences, 2021, 11(1): 43-45. DOI: 10.3390/brainsci11010043

[16] Cohn N. Your brain on comics: a cognitive model of visual narrative comprehension. Topics in Cognitive Science, 2020, 12(1): 352-386. DOI: 10.1111/tops.12421

[17] Finlayson S G, Subbaswamy A, Singh K, Bowers, J, Kupke, A., Zittrain, J & Saria, S. The clinician and dataset shift in artificial intelligence. New England Journal of Medicine, 2021, 385(3): 283-286. DOI: 10.1056/NEJMc2104626

[18] Bicanski A, Burgess N. Neuronal vector coding in spatial cognition. Nature Reviews Neuroscience, 2020, 21(9): 453-470. DOI: 10.1038/s41583-020-0336-9

[19] Franzen L, Stark Z, Johnson A P. Individuals with dyslexia use a different visual sampling strategy to read text. Scientific Reports, 2021, 11(1): 6449-6455. DOI: 10.1038/s41598-021-84945-9

[20] Zhou S K, Greenspan H, Davatzikos C, Duncan, J. S, Van Ginneken, B, Madabhushi, A & Summers, R. M. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. Proceedings of the IEEE, 2021, 109(5): 820-838. DOI: 10.1109/JPROC.2021.3054390

[21] Basso M A, Bickford M E, Cang J. Unraveling circuits of visual perception and cognition through the superior colliculus. Neuron, 2021, 109(6): 918-937. DOI: 10.1016/j.neuron.2021.01.013

[22] Jeong J J, Tariq A, Adejumo T, Trivedi, H., Gichoya, J. W., & Banerjee, I. Systematic review of generative adversarial networks (GANs) for medical image classification and segmentation, Journal of Digital Imaging, 2022, 35(2): 137-152. DOI: 10.1007/s10278-021-00556-w

[23] Wang F. Automatic ink painting rendering technique based on deep convolutional neural networks. Informatica, 2025, 49(5): 95-108. DOI: 10.31449/inf.v49i5.7112

Research on Optimization Method of Landscape Architecture Planning and Design Based on Two-Dimensional Fractal Graph Generation Algorithm

Sheng Chen
Shanxi Vocational University of Engineering and Technology, School of Architectural Design, Department of Cultural Heritage Conservation Engineering)
E-mail: cs10078910@163.com

Keywords: design optimization, generation algorithm, landscape architecture, two-dimensional fractal graph.

Received: May 30, 2024

The development of modern mathematical theory, especially two-dimensional fractal graph algorithm, provides a possibility for large-scale landscape data processing. Landscape digital identification technology is an innovative technology based on digital landscape technology and computer identification of experimental data. It is an important artificial intelligence technology, which includes three steps: landscape acquisition, landscape processing and landscape identification. The characteristics of the scene in the landscape picture can be collected by special instruments, such as cameras, etc., and then the collected data can be processed by two-dimensional fractal graph algorithm, and finally realice the automatic identification of the landscape. For images with significant boundary characteristics, we can extract the boundary of the región quickly and accurately, so as to realice the segmentation of the region. However, when the edge features of the image are not good enough, there is little color difference between the background and the region, or there is some interference, the result will be very bad. In this paper, based on the two-dimensional fractal graph generation algorithm, a series of optimization of landscape architecture planning and design. The accuracy of landscape prime number can reflect whether specific types of landscape pictures can be correctly identified and divided. 200 Pictures are divided into six categories, namely Water scene, landscape scene, living scene, sky scene, architecture and transportation then exact ratios of two-dimensional fractal graph network -8s, two-dimensional fractal graph network -16s, two-dimensional fractal graph network -32s and two-dimensional fractal graph network -32s. It reached the best level in pixel accuracy, average accuracy, average IU, etc., and the pixel accuracy has reached as high as 100%, average accuracy has reached 100%, average accuracy has reached 100%. When compared to the recommended algorithm, the 2D fractal graph generation algorithm has the highest accuracy (94.52%), precision (93.34%), and recall (94.18%) in the classification process.

Povzetek: Razvita je optimirana metoda načrtovanja krajinske arhitekture z uporabo algoritma generiranja dvodimenzionalnih fraktalnih grafov, kar je omogočilo učinkovito avtomatsko prepoznavanje in segmentacijo krajinskih elementov.

1 Introduction

The performance of landscape architecture is a comprehensive concept, which refers to the characteristics and functions of the whole life cycle of landscape architecture, including early analysis, conceptual design, construction and operation. For a long time, landscape architecture has been dependent on the designer's prediction to design it: based on the designer's prediction to complete the design. However, designers cannot predict all the factors with complete accuracy, so it is impossible to design a perfect design.

However, it is possible to maximize the perfection of design results through as much analysis and thinking as possible and considering as many factors as possible. The elements of the site environment is the material premise on which the landscape architecture depends. The design of landscape architecture is not a static way to carry out, and its form will inevitably exist in the site environment. Human perception in the space environment, sound, light, heat and other factors in the natural environment affect the form. It is significant to determine how to respond to the "unpredictable" immaterial capabilities, as well as how to respond to the force, energy and feeling in the structure, is the main task in the performance

optimization process, as shown in Figure 1 below, which is the specific landscape architecture design plan.

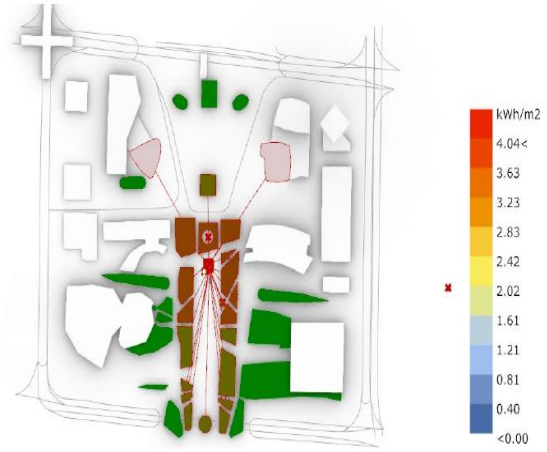


Figure 1: Planar heat planning and design in landscape architecture design

For example, in the landscape architecture design shown in the figure 1, designers can operate and optimize the form according to the perception of the natural environment of the site or the human environment of the site, to realize the transformation of the original single pure form.

The form-finding model and multi-objective optimization based on performance digital analysis make performance itself a factor and method of form creation, and help designers complete the design as an optimization technique. Landscape architecture structure optimization design process of performance, can be considered in landscape architecture designers interpretation of the space environmental conditions as the foundation, the performance optimization software as the main technical means, will be dissected into landscape architecture structures as a material system + markers, a micro three dimensions from macro a medium to form self-organizing process of structures, It is a form generation process from top to bottom and self-expression. Here, the dominant position of the designer is expressed by interpreting the site environment, predicting the function and the relationship between the construction form and the performance target. This design idea can be roughly summarized into three kinds of form generation and feedback processes: one is a dynamic interaction between the form of the structure and the human subject. The other is the environmental or other external forces acting on various forms of the structure, and the resistance of various forms to the environment or other factors. The third is the interaction between the components of the structure itself. Therefore, the performance of landscape architecture can also be summarized into three categories, which in turn is the spiritual demand (the spatial feeling brought by the space to the user).

In terms of the connotation of garden landscape, in addition to traditional garden landscape, there are also landscape preference, landscape competitiveness evaluation, etc., and the connotation of evaluation is gradually deepening. In the 20th century eighties to nineties, the landscape more beauty estimates, the environment, such as the study of different models, the visual landscape and visual effects evaluation, among them, the main methods of landscape evaluation model is divided into three categories: describe the factor method, questionnaire survey method, aesthetic attitude determination method. At present, the evaluation of landscape resources using both qualitative and quantitative way more, more is established by using AHP method of fuzzy comprehensive evaluation model, including: from the perspective of ecology, the GIS technology, the tourists demand Angle, landscape image, etc. At present, high-resolution landscape architecture can provide a large amount of landscape information with rich characteristics, so it is widely used in landscape assessment [1-3].

Two-dimensional fractal image generation algorithm is an image segmentation method based on image features. Its working principle is to select the boundary points in a region as candidate boundaries and select a method that can be spliced to obtain the boundary of the region through the inconsistency of the features of the region or region. Several edge detection operators, including first-order differential Sobel operator, Robert operator and second-order differential Laplacian operator, are usually used to extract the edge of a region.

Based on two-dimensional fractal graph. The partition technique of generating algorithm is essentially a partition based on similarity criterion, which includes some common methods. In this method, a series of basic texture pixels are used to describe each region, and an extended growth criterion is determined to expand the region. Secondly, the growth of adjacent seed pixels is calculated to determine whether the adjacent pixels have added the pixel of seed pixels. When no new pixel is found, the whole growth is carried out. The most critical step is to know the rules of seed birth and growth [4-5].

The watershed method mainly regards the pixel points of each scene in landscape architecture as the coordinates in the whole graph, and represents its location with one pixel value. Next, they used a similar in floods overflow, low-lying, pixels less place, is a piece of plain, and the high place, is a mountain, in a basin, with the change of the terrain, terrain will be more and more high, the higher the terrain, the topography is lower, the easier it is to be flooded. After enough seawater is poured into the area, a depression is formed, creating an open area. However, there is too much segmentation due to the interference of the pixels. In general, this method transforms the landscape into a grid landscape in graph theory, and treats each pixel point as a node on the landscape, and the

connection between these nodes is called the boundary. The common method to define edges is to calculate the dissimilarity of pixel points in the neighborhood according to the correlation of pixel values, and then treat them as edge weights, so as to get the graph G=< V, E>. In general, G is a weighted undirected graph, and its weight is usually defined according to the actual situation. The basic principle of graph theory is to cut off several edges on G and cross off those parts of G that are not connected together, thus realizing the partition of G. In the partitioned graph G, each independent subgraph is matched with the corresponding partition, which realizes the image segmentation [6-8].

2 Related works

Table 1: Summary on related works

Reference	Methods/Algorithm	Merits	Limitations
[9]	This work presents the multi-bit Trie-tree technique with non-collision hashing.	Test results indicate that this approach has a good success rate for data reconstruction and a high-performance efficiency.	Under this design, the necessary data reconstruction technology has not developed.
[10]	This survey updates trends in organizations, processes, and outcomes for NPD in the United States and was performed little over five years after PDMA's initial best-practices survey.	Best-practice companies have higher expectations for their NPD programs, use more versatile teams, and are more likely to monitor NPD processes and outcomes.	Businesses that do not maintain current NPD practices will find their competitive lack in growth.
[11]	This work presents an effective hybrid approach for dynamic mesh generation based on Delaunay graph mapping and Radial Basis Functions (RBFs). .	The hybrid RBFs-Delaunay graph mapping approach is found to be as accurate and effective as the Delaunay technique in building dynamic meshes for various test scenarios. .	Delaunay graph mapping is lacking in efficiency.
[12]	Utilizing a novel two-dimensional modification of hat functions (2D-MHFs) to solve linear Fredholm integral equations	The method is desirable from a computational standpoint, and its great accuracy is demonstrated by a few numerical examples.	Further information on the error analysis is needed
[13]	An introduction to genetic algorithm-based aeronautics is given.	The method, particularly when incorporating numerous free design factors and configurable modeling parameters, is more versatile and computationally efficient than the traditional approach.	A small number of snapshots computed via computational fluid dynamics.
[14]	This study offers a thorough approach to water and energy operation optimization.	Lastly, simulated experiments based on real network data are used to validate the viability of the suggested operation optimization strategy. The comprehensive energy-saving rate reaches 31.3%, effectively lowering the costs associated with system operation.	Lack of performance metrics with the suggested operation.

3 Research methods

3.1 Based on two-dimensional fractal graph, it is an algorithm based on conditional random point partition

According to the human landscape of the scene, make full use of the subjectivity of architectural design, carry out artistic creation of landscape architecture, and visualize its art. The design of this experimental landscape architecture structure is based on the landscape pavilion with roof and column for people to watch and chat. The reason is:

First of all, in landscape architecture works, the interaction between people and places is very critical, which requires architectural design to be comprehensively considered according to their own experience and the situation of the site. After a detailed arrangement of the area, the author divides it into two parts: natural environment and cultural environment. The experiment was conducted in a humanized manner centered on the central axis of the Guangzhou center, targeting white-collar workers, tourists and nearby residents. In view of the large urban population and various groups, the construction of landscape architecture layout should not only meet the needs of people, but also take into account the needs of history, culture and society, that is to pay full attention to the structure of landscape architecture and people's behavior.

On this basis, a concept of random spatial structure based on background, two-dimensional fractal graph, is proposed. This method uses each pixel in the background as a node, as shown in Figure 2 below, and uses the corresponding relationship between each point as the boundary. Then the minimum state random ability field is used to divide the landscape garden landscape.

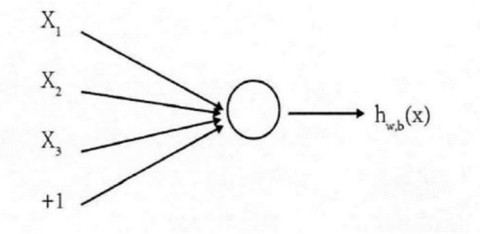


Figure 2: The most basic representation of the two-dimensional fractal graph algorithm

The energy of the conditional random field in image semantic segmentation is expressed as follows:

$$E(X) = \sum_{i\in V} \varphi_i(x_i) + \sum_{i\in V, j\in N_i} \varphi_{ij}(x_i, x_j); \forall i, x_i \in L \quad (1)$$

In the formula, represents the set composed of all landscape element pixels in the landscape image, represents the set of pixels adjacent to the landscape element pixel, and usually takes the set of pixels in four similar regions, top, bottom, left and right.VN_iLRepresents the combination of categories of landscape element pixel classification. It represents the categories of all landscape element pixels and is the corresponding category of pixel points. $x_i i\varphi$ Is the potential energy function.The general form of the unary potential energy is the logarithmic pattern of the class likelihood probability corresponding to the pixel.$\varphi_i(x_i)x_i$The likelihood probability can be learned according to the actual specific pixel features, and the most common pattern is directly trained according to the pixel values. However, these methods only consider the color factors of the landscape image pixels, but are not particularly comprehensive. Therefore, color and texture are generally selected：

$$\varphi_i(x_i) = \lambda_T\varphi_T(x_i) + \lambda_{col}\varphi_{col}(x_i) + \lambda_l\varphi_l(x_i) \quad (2)$$

Potential energy function trained by the value size (color) feature. φ_lIs a potential function defined according to the location characteristics of pixels. λ_T、λ_{col}And are their respective weights, which are generally obtained through training.λ_l

Binary potential energy, generally defined in the form

$$,\varphi_{ij}(x_i, x_j) \quad (3)$$

$$\varphi_{ij}(x_i, x_j) = \begin{cases} 0, x_i = x_j \\ g(i,j), x_i \neq x_j \end{cases} \quad (4)$$

Functions are generally defined based on the relationship between pixel values between adjacent pixels. In the formula, the meaning of function is for pixel values with the same pixel class, which will make the value of function 0. Otherwise, it will be determined according to the function.

According to the human landscape of the scene, make full use of the subjectivity of architectural design, carry out artistic creation of landscape architecture, and visualize its art. The design of this experimental landscape architecture structure is based on the landscape pavilion with roof and column for people to watch and chat.

This method is based on the pixels of the pixel value, is often a function of the formula is the pixel value and pixel type are the same type of pixel values, resulting in the function value is 0, on the other hand, will be decided according to the function.

At present, by using random field model for post-processing, semantic of early model correction, one yuan a function usually to repair and add in front of the

semantic segmentation, make the original semantic segmentation more accurately.

3.2 Description of algorithm

- Updating 2D Search Velocity: Every search agent's velocity is updated according to its velocity, position, and the best-known positions of both itself and its neighbors in the same level.
- Update Search 2D Position: Next, each search agent's position is modified in accordance with its velocity.
- Termination of Optimization: Until a termination criterion is satisfied, the optimization process is carried out recursively.
- Best answer Extraction: The ideal answer to the landscape design challenge is ultimately determined to be the most well-known position among all search agents.

Effective navigation of the landscape design space is made possible by the 2D fractal graph creation technique, which takes advantage of this hierarchical classification to balance exploration and exploitation across various layers of search agents. Fractal graphs converge to ideal landscape designs through iterative refinement directed by equations guiding velocity and position updates. The hierarchical classification that the 2D fractal graph presents improves its effectiveness and efficiency in navigating the intricate landscape design space. Fundamentally, the fractal graph method is a multi-level categorization scheme that groups search agents according to how well they can explore and exploit new areas. Because of its hierarchical structure, 2D is able to strike a balance between exploitation which helps to refine promising solutions and exploration which allows for the discovery of a variety of landscape design alternatives.

Algorithm 1: Two-dimension fractal graph for the landscape architecture planning and design

Initialize:

- Define the 2D landscape planning and design problem.

- Parameters: size of the population (N), iterations in maximum (MaxIter), hierarchical levels (L), weights (w, w_local, w_global), acceleration coefficients (c1, c2), are the classifications.

Generate initial fractal graph:

- Randomly initialize N search agents with positions and velocities within the solution space.

Evaluate stability:

- Evaluate the stability of each search agent using the landscape design objective function.

Main loop:

For iter = 1 to MaxIter:

Update hierarchical classification:

- Classify search agents into hierarchical levels based on their stability and exploration-exploitation characteristics.

For each level L:

Update velocity and position:

For each search agent in level L:

Update velocity:

- Calculate cognitive and social components:

cognitive_component = c1 * rand() * (p_local - position)

social_component = c2 * rand() * (p_global - position)

- Update velocity:

velocity = w * velocity + cognitive_component + social_component

Update position:

- Update position:

position = position + velocity

Evaluate fitness:

- Evaluate the stability of the new position using the landscape design objective function.

Update local best:

- Update local best position if current stability is better than previous.

Update global best:

- Identify the search agent with the best architecture stability among all levels.

Return global best as the optimal solution.

3.3 A Survey of the computational methods of two-dimensional fractal graphs

Due to the emergence of two-dimensional fractal graph, it has been widely paid attention to for its unique advantages, coupled with the development of technology, people's understanding of it is more and more profound, two-dimensional fractal graph performance is also getting better and better, has been widely used in many aspects. This is especially true for the identification, especially for the landscape. 2 d fractal graph has many advantages, the key point is the weighted average, and biological neural network, it can reduce weight, thus reduce the difficulties of network modeling. Compared to the conventional 2 d fractal graph, it saves a lot of tedious pretreatment process, such as refactoring related data and so on.

In numerous hierarchical networks, 2 d fractal graph is one of the most widely used. The method can in advance to BP, effectively reduce the training parameters, thus greatly improve the performance of the algorithm. With 2 d fractal graph method, can effectively shorten the preparation process of input, thus for the user to save the work time and reduce the pressure of work. Layer by layer, layer by layer, layer by layer, layer by layer with a new computation, a new way of computing, each kind of new data are added to a new system, this method can be applied to many aspects.

The formation of architectural form is mainly an innovative embodiment of the relationship between the spatial structure and the architectural structure of the building site. Landscape structure is a unique landscape architecture and its structure and has a high degree of similarity and a high degree of convergence. This makes the landscape design of the environment and the structural form can communicate directly, not mechanically or indirectly, to adapt to the former, but will be integrated into the landscape design of the behavior of the purpose. Although the message contained in each place is different, its essence is to seek a symbiotic relationship between human and nature. Its inner essence is palpable and obvious, such as the use of local materials and traditional crafts. The internal expression is a spiritual guidance for designers to convey and express the inner meaning of the museum through accurately grasping the characteristics of the scene.

Its meaning includes two levels: first, designers use the power of nature to show a landscape structure with deep humanistic characteristics and spiritual emotions, and make its spatial connotation appear and continue according to their own experience and experience. Due to the addition of landscape architecture structure, it adds a new artistic spirit and cultural connotation, so that its symbol has a natural artistic spirit of the place. No matter what the meaning is, it shows that the landscape architecture structure is a man-made intermediary between people and places, and the essence of its design is to integrate human thought and subjective will into the information of places. Under the guidance of design, people often naturally associate or recall, and realize the meaning of the designer, thus arouse people's resonance.

2 d fractal graphic method belong to the category of deep learning, its structure characteristics are similar to deep learning, both the localization, also has a hierarchical. The method adopted a kind of monitoring method of training, make the method can more accurately extracted

from the input data contains information. This method can improve the learning efficiency of 2 d fractal graph. In actual application, because there is no information there is usually no classification marking, so you need to first according to the characteristics of the tag information itself, to unsupervised learning of these data, in order to gain rules, then marked by the monitoring data of learning, both to fully play to the role of the samples, and can deal with the issue of information is not much.

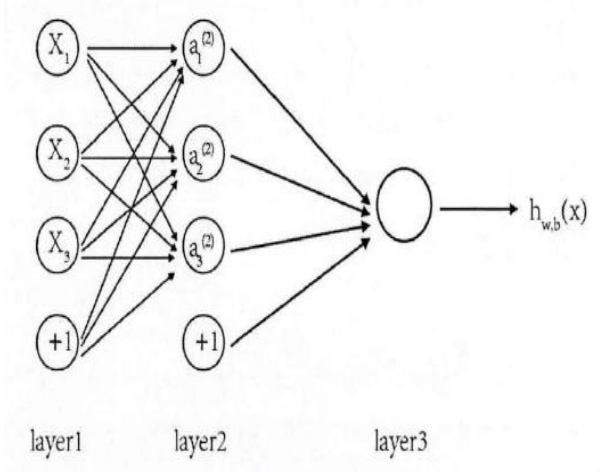


Figure 3: Computational architecture and general formula of two-dimensional fractal graphs

As we can see in Figure 3, the input level is a neural network."+ 1" is refers to a point, known as split points. The system has three types of input, output, three kinds of concealed structure, only a as output, input on the left, on the right is the output, and in the central is implied, shows the fully connected. Hidden in the shadows of data, including the data on the node, is unable to display in the training.

Among the whole network structure, said one of the layers, a total of figure 2 contains three layers.n_1Need a tag in each layer, is in the first layer of the output layer, use said, parameters of fang, such as:

$$(W, b) = (W^1, b^1, W^2, b^2) \quad (5)$$

W_{ij} (1) arrangement of a unit of the ordinal number is 1 layer in the middle position of the first unit, and the connection between the unit operation parameters, said in the first layer of the first units to set up.$j1 + 1ib_i^l1 +$ 1iThe neural network computation method is to use one of the units as outputs, and tags, and the output markup became the first layer, the value is 1, it represents in the known and the two functions, and it contains a collection of the cases, can pass for function prediction, Thus the final results and output. $a_i^2 wbh_{w,b(x)}$ The specific calculation steps are as follows:

$$a_1^2 = f\left(W_{11}^1 x1 + W_{12}^1 \times 1 + W_{13}^1 \times 1 + b_1^{(1)}\right)$$
$$a_2^2 = f\left(W_{21}^1 \times 1 + W_{22}^1 \times 1 + W_{23}^1 \times 1 + b_2^{(1)}\right)$$
$$a_3^2 = f\left(W_{31}^1 x1 + W_{32}^1 \times 1 + W_{33}^1 \times 1 + b_3^{(1)}\right)$$
$$h_{W,b}(x) = a_1^3 = f\left(W_{11}^2 a_1^2 + W_{12}^2 a_2^2 + W_{13}^2 a_3^2 + b_1^{(2)}\right) \quad (6)$$

Prior to the spread of the calculation principle is shown in figure 1-3, the neural network structure and Logistic model structure are nearly the same, both calculation method and the principle of using but the biggest difference between them is the neural network Winding. Hypothesis 2 d fractal graph algorithm is able to handle the size of the data, the width of the image with said the figure like height with said, the two figures is a plane, but the dimension of the image and color channel Several are represented by $d.h * w * dwh$

Basic network formed by convolution and pooling layer, and the two basic structures are based on local entered as a unit, and 2 d fractal graph algorithm in image translation can maintain the structure of the constant, in other words Element is only related to the position of the space, if one layer of data on the coordinates of the vector is, the data vector of the next layer is, and its calculation formula is as follows:

$$Y_{ij} = f_{ks}\left(X_{si+\delta i, sj+\delta j}\right) 0 \le \delta i, \delta j \le k \quad (7)$$

Convolution kernels in the above formula used, the size of the step length with s, said the said algorithm () function which adopted operation, general can be divided into matrix on the convolution or average pooling, a nonlinear excitation function, and calculate the maximum pool type of operation such as the maximum value.

When the parameters and satisfy, to the principle of mutual transformation between this formula can be used to represent:

$$f_{ks^\circ} \quad {}'g_{ks'} = (f^\circ g)_{k'+(k-1)s', ss'} \quad (8)$$

General 2 d fractal graph algorithm in dealing with nonlinear equation, a very common will adopt the method for solving nonlinear filter, completely is adopted by the two-dimensional fractal graph. Full convolution net, can be used to express two-dimensional fractal graph network and because its input image in conformity with its output image, so the convolution network size is not qualified.

3.4 Based on the two-dimensional fractal figure garden jingsu level evaluation index of the algorithm

In the process of various scenarios of semantic classification, unable to differentiate each scenario, so sometimes will be a scene of scenes into other scenarios, the effect of causing some fuzzy. This paper adopts a new method, the images were compared with the real world, according to the classification results obtained will be digital processing was carried out, as a convicted landscape image of the final result. This article is currently used by common word segmentation criterion, used for statistical accuracy.

This paper used to represent the semantics and belongs to the class is judgment as the number of pixels, to represent the language righteousness, the total number of categories in this article belongs to the class represented by the number of the pixels of the total number of $n_{ij}ijn_{cj}n_{cj}6, t_i = \sum_i n_{ij}i$. Calculate the overall accurate rate formula such as:

$$\frac{\sum_i n_{ii}}{\sum_i t_i} \tag{9}$$

Used to define JingSu belongs to the first-class scenery and be correctly defined as the first-class scenery of Beijing plain accurate formula can be used to represent:ii

$$\frac{\frac{1}{n_{cl}}\sum_i n_{ii}}{t_i} \tag{10}$$

IU scheme (IU) on average, is expected by calculation landscape JingSu category pixels in the intersection of right then to predict the scenery line pixels and the pixels of the original category and set, the result is that the final discriminant index, can use formula to represent:

$$\frac{\frac{1}{n_{cl}}\sum_i n_{ii}}{(t_i+\sum_j n_{ji}-n_{ii})} \tag{11}$$

4 Result analysis

4.1 The practical application of two-dimensional fractal figure

2-D fractal graph network training, after a long period of time after the improvement and development, there are two main ways of training: in the presence of a monitor teaching. While this paper used a learning algorithm of monitoring, the original image and its corresponding manual segmentation image used in the network of the 2-d fractal graph modeling [14].

However, the influence ratio of each factor of the actual landscape architecture structure on the landscape architecture structure is not the same, and each aspect has some kind of mutual coupling relationship. For example, when the landscape building structure is opened in the light environment, attention must be paid to the structural performance of the structure to ensure the stability and constructability of the structure. From this point, it can be seen that the combined effects of building characteristics and lighting conditions are very complex and some even contradict each other. For another example, the average sunshine structure is strongly related to factors such as vertical scale, but in a specific design, the coupling of sunlight with horizontal and vertical scales is difficult.

Monitored learning is machine learning, and its technical feature is the ability to recognize the functions of a map. During this period, all training samples have an alternative target and a more satisfactory output result, which is called "monitoring". The technique of monitoring refers to the comprehensive analysis of the input data at runtime, and the corresponding mapping is obtained, to obtain a new set of sampled data. If it is new data that was generated before, then he will label these new data samples as categories. In the algorithms for two-dimensional fractal graphs, guided learning algorithms are usually used for training while supervised learning is usually based on gradients (Krizhevsky et al, 2012). Batch processing stochastic gradient reduction methods are commonly used. In the learning process of two-dimensional fractal graph N, we only use one example to simplify the description process. The method is divided into two stages: forward stage and reverse stage. The first step is to carry out in turn until the final result, the second step according to the error of the value to be output weight and deviation, after the end of the operation, according to the weight and deviation of each level is adjusted accordingly. If the number of classes, we specify is for samples during classification, its error function formula is as follows:

$$J(W, b; x, y) = \frac{1}{2}\Sigma_{k=1}^{c}(t_k - y_k)^2 = \frac{1}{2} \| t - y \|_2^2 \quad) \tag{12}$$

Among them, represents the weight in the neural network, b. Represents the bias in the neural network, the training sample is represented by, and the corresponding standard of the training sample is represented by.$Wxyt_k$Denotes the KTH dimension component of the predicted value generated by the sample when predicting the sample;x However, it represents the dimensionality component of the sample label to be predicted.$y_k$xk When we are doing back propagation, the first thing we need to do is calculate the error terms at each level in a certain order.Suppose that the error term of the first layer is calculated according to the above formula, and the weight of this layer is expressed and the bias parameter is expressed.$\delta^{(a+1)}l + 1$WbIf both layers are fully connected, the error term of

the first layer can be calculated using the following formula:

$$\delta^{(l)} = \left(\left(W^{(l)}\right)^{T}\delta^{(l+1)}\right) \cdot f'\left(z^{(l)}\right) \quad (13)$$

The corresponding gradient calculation formula is shown as follows:

$$\nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)}(a(1))^{T}$$
$$\nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)} \quad (14)$$

If the first layer is a feature extraction stage, that is, the convolution layer and the sampling layer of, then the error term of the first layer can be calculated by the formula:

$$\delta_{k}^{(l)} = \text{unsample}\left(\left(W_{k}^{\ a)}\right)^{T}\delta_{k}^{\ a+1)}\right) \cdot f'\left(z_{k}^{\ a)}\right) \quad (15)$$

Among them, the value of represents the first convolution kernel. After a series of runs in the upper sampling layer, the error obtained later can be transmitted to the previous layer through the subsampling layer, which refers to the convolution layer of. $kk\delta_{k}^{\ a+1)}$ If use method is to average sampling, sampling layer will put poor 涨 simple average assigned to run before sampling of sub zone, if use is a maximum sampling, so when the forward propagation, samples shipped to in clause, the sampling value will get all the error, the rest of the value is 0 [15]. The accuracy of landscape classification can be seen in the accuracy of landscape classification, which can see whether specific types in landscape have been correctly identified and divided. As shown in Table 4-3, the classification of landscape scenery has low accuracy, and the classification accuracy of natural landscape is higher than the 2D fractal network -8s.It can be seen that the two-dimensional fractal graph network -16s has the best overall performance in the classification of habitat landscape in landscape images. Finally, three values of average accuracy, average accuracy and average accuracy of image pixels are tested. The final value is shown in Table 2;

Table 2: Results of three kinds of upsampled semantic segmentation on landscape pixels

Scene Elements			
	Pixel accuracy(%)	Average accuracy(%)	The average IU (%)
FCN-8s	86.25	85.98	74.33
FCN-16s	88.97	88.58	75.35
FCN-32s	83.06	82.75	72.69

4.2 Optimization of landscape architecture planning and design and expansion of two-dimensional fractal graph

Through the detection and classification of 200 pictures, the correct degree of landscape element value in landscape garden landscape is obtained, as shown in Table 3 below, which can intuitively reflect the performance of three different upper sampling structures. The accuracy of landscape prime number can reflect whether specific types of landscape pictures can be correctly identified and divided. Pictures are divided into six categories, namely Water scene, landscape scene, living scene, sky scene, architecture and transportation. Water scenes include Water, river and Mountain. Scenes include: landscape, landscape, landscape, sky scene, architecture, traffic, traffic, etc. It can be seen from Table 3 that the classification accuracy of two-dimensional fractal graph network -32s is the worst among all classifications. The classification accuracy of two-dimensional fractal graph network -16s is higher than that of two-dimensional fractal graph network -8s, while the accuracy of two-dimensional fractal graph network -8s is higher than that of two-dimensional fractal graph network -8s. Therefore, two-dimensional fractal graph network -8S also has a good effect on landscape quality classification of landscape images. On the two-dimensional fractal graph network -8S, "Skyview" has the best classification rate, but in the real world, it has the lowest classification rate, only occasionally appearing something similar to the real scene.

Table 3: Three kinds of the knot JingSu landscape classification accuracy

Categories	FCN-8s (%)	FCN-16s(%)	FCN-32s (%)
The surface of the water	91.89	90.32	87.85
The mountain	87.66	86.35	82.43
Vegetation	85.94	88.28	83.12
The sky	93.56	90.65	88.45
Building	88.29	87.56	84.68
The traffic	86.12	85.08	83.28

Table 3 shows the exact ratios of two-dimensional fractal graph network -8s, two-dimensional fractal graph network -16s, two-dimensional fractal graph network -32s and two-dimensional fractal graph network -32s. From this figure, we can know the pixel accuracy, average accuracy and average value of the three kinds of

upper sampling. Through the comparison of three different upper sampling modes, it is concluded that two-dimensional fractal graph net-8S has reached the best level in pixel accuracy, average accuracy, average IU, etc., and the pixel accuracy has reached as high as 100%, average accuracy has reached 100%, average accuracy has reached 100%. The average accuracy is lower than pixel, which is calculated from the data of each classification of the image. Too much data will result in the average accuracy and the average IU.

Table 4: Classification of landscape planning and design based on two-dimensional fractal graph generation algorithm

Classification	FCN-8s	FCN-16s	FCN-32s
Ecological Sustainability (0-1)	0.86	0.79	0.72
Aesthetic Appeal (0-1)	0.92	0.88	0.81
Resource Efficiency (0-1)	0.84	0.79	0.82
Robustness	0.94	0.90	0.89

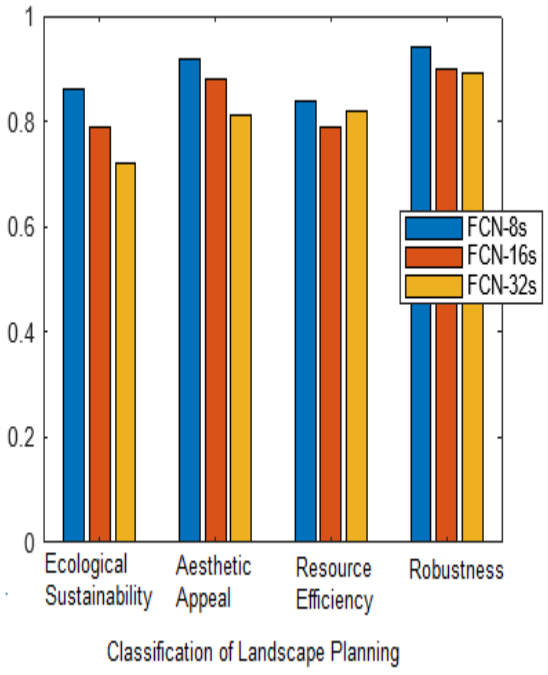
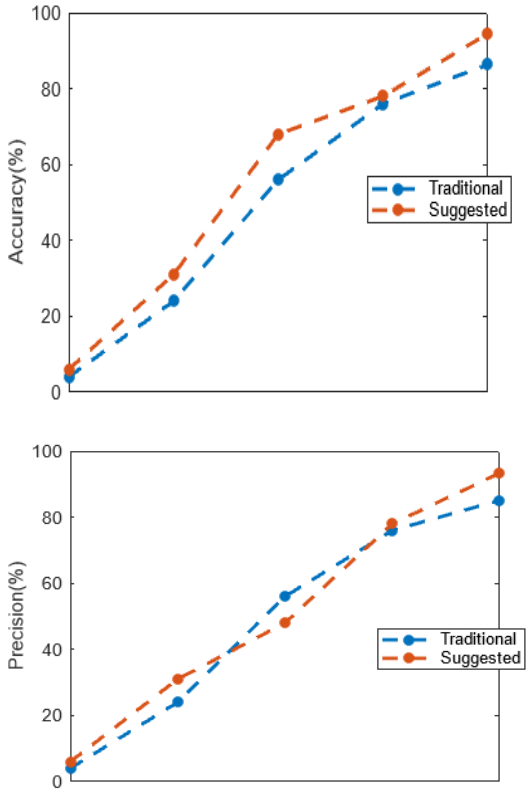


Figure 4: Outcome of landscape planning and design based on two-dimensional fractal graph generation algorithm

The landscape planning findings, which were obtained through the use of a 2D fractal network generation algorithm, are shown in Figure 4 and Table 4. The assessments are based on a number of factors, such as ecological sustainability, aesthetic appeal, resource efficiency, and robustness. Ecological sustainability, aesthetic appeal, and resource efficiency are numerical values assigned to each design solution, ranging from 0 to 1, representing the quality of the design in each respective area. The algorithm generates a wide range of solutions every time, which encourages the exploration of the solution space and makes it possible to identify several different design options. The term "robustness" describes how stable the solutions produced by the 2D fractal network generation algorithm is under different circumstances. The robustness of all trials shows that the algorithm's results are dependable and consistent in FCN-8s, FCN-16s, and FCN-32s scenarios.

Table 5: overall performance

Algorithm	Accuracy (%)	Precision (%)	Recall (%)
Traditional Optimization Algorithm	86.34	85.12	85.78
Suggested Algorithm	94.52	93.34	94.18



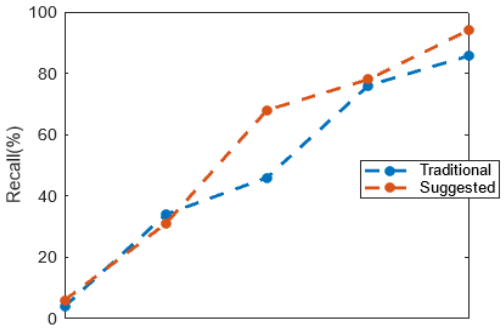


Figure 5: Overall performance of the methods

An academic discipline that focuses on the interaction between human habitation and the natural environment is landscape architecture planning and design. Table 5 and figure 5 demonstrates that the 2D fractal graph generation approach that is recommended has the best accuracy (94.52%), precision (93.34%), and recall (94.18%).

5 Discussion

A fractal graph is a complex entity that is formed using recursive iteration rules and can be found in both natural systems and man-made systems, such as cities. It displays intrinsic self-similarity on both a large and small scale. The fractal two-dimension is a scientific technique for measuring landscape architecture aspects and its evolutionary unsample in the context of network systems. Furthermore, it is an important metric to determine whether a city is experiencing self-organizational evolution. Previous research results show that self-organized architecture systems have notable fractal properties that can be measured using fractal two-dimensional graphs. Nevertheless, prior research has solely utilized these two fractal dimensions to investigate the general fractal properties of the entire city, without carrying out more accurate fractal measurements for subzones in various directions and layers.

The existence of a fractal structure, which acts as an analogy and supplement to earlier research findings on the general fractal laws discovered in other cities, is one of the study's major discoveries. Unlike earlier research, this finding supports the different 2D values that contribute to the spatial variability of separate subzone structures. The cause is the differing paths taken by various subzones in terms of planning and development, as well as the ways in which people, land, and architecture occupy and use space in distinct urban blocks. The box-counting dimension is a measure of the spatial occupancy capacity, hence urban expansion will cause it to rise. In other words, the self-organizational objective of urban development is to provide a more balanced and effective distribution of urban space while optimizing the spatial configuration and enabling coordinated growth within its internal areas. By illustrating a progressive decrease in the fractal dimension from the city's center to its periphery, suggesting outward growth, the study contributes to the notion of fractal cities. Furthermore, it shows that pixels with mature landscape architecture have larger fractal dimensions than those that are still developing or are being quickly classified as FCN-8s, FCN-16s, or FCN-32s.

6 Conclusion

In order to solve this problem, we adopt a method based on quadratic fractal graph to solve this problem. Two-dimensional fractal graphs are mainly divided according to the pixel points in landscape architecture. Different from the method of 2D fractal graph, the final step of 2D fractal graph is to transform the whole connected level into a transition, so that the operational architecture of 2D fractal graph is preserved. The final image is the same size as the original image, and the original image of the original image can be obtained. Compared with the traditional two-dimensional fractal graph method, two-dimensional fractal graph has higher computational speed and is not constrained by the size of the input image. In this paper, a new image semantic partition method is introduced.

After establishing a virtual environment, the model is modeled by the common SifFlow on the network. The image preprocessing technology is used to enhance the data, thus effectively overcoming the problem of overfitting the model. On this basis, the second learning of the second segmentation is carried out to shorten the learning period of the model.

In the image analysis, two-dimensional fractal graph network -32s, two-dimensional fractal graph network -16s, two-dimensional fractal graph network -8s, two-dimensional fractal graph network -8s, two-dimensional fractal graph network -8s, two-dimensional fractal graph network -8s, three different upper sampling methods are adopted. In the case of pixel accuracy, average accuracy and average U value sampling, the sampling structure on 2D fractal graph net-8s is selected, and the average accuracy is 90.3%, the average accuracy is 88.91%, and the average U is 75.83%. At the same time, the pixel accuracy of the model in each scene type in the landscape image is more than 86%, it is indicating that the method is an ideal method in the landscape image, especially in the landscape image containing multiple scene types, can obtain higher pixel segmentation accuracy. In this paper, the classification experiment of landscape in landscape architecture is performance metrics in algorithm carried out with accuracy as 94.52%.

Declaration statement

Ethics approval and consent to participate

I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.
Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials

The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests

Here are no have no conflicts of interest to declare.
All authors have seen and agree with the contents of the manuscript and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

Funding

Funding: Shanxi Province Project: National Education Research Letter 2021," Rural revitalization" Research on Teaching Innovation of Traditional Culture Protection and Tourism Planning in the Background, JGCY2693.
Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript

References

[1] Li X, Li S, Jiao H. 2020. Research on Multi-objective Optimization Method of Central Air Conditioning Air Treatment System Based on NSGA-II[J]. *Journal of Physics: Conference Series*, 1626:012113. https://doi.org/10.1088/1742-6596/1626/1/012113

[2] He P, Gao F, Li Y, et al, 2020. Research on optimization of spindle bearing preload based on the efficiency coefficient method[J]. *Industrial Lubrication and Tribology*, ahead-of-print(ahead-of-print). doi10.1108/ILT-06-2020-0205

[3] Hong E, Ban H, Qi M, 2019. Design optimization and analysis of a vaned diffuser based on the one-dimensional impeller-diffuser throat area model[J]. *Journal of Physics: Conference Series*, 1300:012007. https://doi.org/10.1088/1742-6596/1300/1/012007

[4] Mohammadi M, Raise A, Regi A, 2019. Design and performance optimization of a very low head turbine with high pitch angle based on two-dimensional optimization[J]. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 42(1), pp. 9. https://doi.org/10.1007/s40430-019-2084-1

[5] Liu W, Yang S, Ye Z, et al, 2019. An Image Segmentation Method Based on Two-Dimensional Entropy and Chaotic Lightning Attachment Procedure Optimization Algorithm[J]. *International Journal of Pattern Recognition and Artificial Intelligence*. https://doi.org/10.1142/s0218001420540300

[6] Wang S, Lin S, 2019. Optimization on ultrasonic plastic welding systems based on two-dimensional photonic crystal [J]. *Ultrasonics*, 99:105954. https://doi.org/10.1016/j.ultras.2019.105954

[7] Hu H Q, Yang L, 2013. Research on Routing Optimization of Regional Logistics Based on Gravity Model: A Case of Blue and Yellow Zones[J]. *I business*, 5(4):167-172. https://doi.org/10.4236/ib.2013.54021

[8] Diaz-Casas V, Becerra J A, Lopez-Pena F, et al, 2013. Wind turbine design through the evolutionary algorithms based on surrogate CFD methods[J]. *Optimization and Engineering*, 14(2):305-329. https://doi.org/10.1007/s11081-012-9187-1

[9] Liu Y, Wan M, Zhang H K, et al, 2011. Research on Data Reconstruction Method Based on Identifier Locator Separation Architecture[J]. *épée*, 12(4):531-539. https://doi.org/10.6138/JIT.2011.12.4.01

[10] Griffin A, 1997. PDMA Research on New Product Development Practices: Updating Trends and Benchmarking Best Practices[J]. *Journal of Product Innovation Management*, 14(6):429-458. https://doi.org/10.1016/s0737-6782(97)00061-1

[11] Ding L, Guo T, Lu Z, 2015. A Hybrid Method for Dynamic Mesh Generation Based on Radial Basis Functions and Delaunay Graph Mapping[J]. *Advances in Applied Mathematics & Mechanics*, 7(03):338-356. https://doi.org/10.4208/aamm.2014.m614

[12] Hatamzadeh-Varmazyar S, Masouri Z, 2011. Numerical method for analysis of one- and two-dimensional electromagnetics catering based on using linear Fredholm integral equation models[J]. *Mathematical & Computer Modelling*, 54(9-10):2199-2210. https://doi.org/10.1016/j.mcm.2011.05.028

[13] Lucas S D, Vega J M, Velazquez A, 2015. Aeronautic Conceptual Design Optimization Method Based on High-Order Singular Value

Decomposition[J]. *Aiaa Journal*, 49(12):2713-2725. https://doi.org/10.2514/1.j051133

[14] Zhu X, Niu D, Wang F, et al, 2018. Operation Optimization Research of Circulating Cooling Water System Based on Superstructure and Domain Knowledge[J]. *Chemical Engineering Research and Design*, 142. https://doi.org/10.1016/j.cherd.2018.12.012

[15] Hu G C, Liu J H, 2011. The Optimization Design of Mechanical Structure Based on CAE Technology[J]. *Machine Design& Research*, 130-134:672-676. https://doi.org/10.4028/www.scientific.net/amm.130-134.672

Enhanced COVID-19 Detection Through Combined Image Enhancement and Deep Learning Techniques

Abderrazak Benchabane*, Fella Charif
Department of electronics and telecommunications, University of Kasdi Merbah, Ouargla, Algeria
E-mail : benchabane.abderrazak@univ-ouargla.dz, cherif.fella@univ-ouargla.dz
*Corresponding author

Keywords: COVID-19, image enhancement, chest x-ray images, deep learning

Received: March 6, 2024

The rapid spread of COVID-19 has highlighted the need for automated patient data analysis to enable faster and more accurate diagnosis. Using pre-trained deep learning models on X-ray images has shown potential for effective COVID-19 detection. However, the performance of these models is highly dependent on the quality and quantity of training data. To address these challenges, enhancing the visual quality of X-ray images is critical for reliable virus detection. This study evaluates and combines three image enhancement techniques—Histogram Equalization, Contrast-Limited Adaptive Histogram Equalization (CLAHE), and Gamma Correction—to determine the optimal approach for improving detection accuracy. A dataset comprising 125 chest X-ray images from COVID-19-positive patients and 500 images from non-COVID-19 cases was used. The images were preprocessed using the enhancement techniques, and the enhanced datasets were employed to train ResNet50 and DenseNet201 models. Simulation results demonstrate that enhanced images consistently yield higher detection accuracy than unenhanced images. Among the techniques tested, combining Histogram Equalization, CLAHE, and Gamma Correction with the DenseNet201 model achieved the highest performance, attaining a remarkable accuracy of 99.03%. This outperforms previous methods, including the DarkCovidNet model, which achieved an accuracy of 98.08% on the same dataset.

Povzetek: Avtorja sta izboljšala zaznavanje COVID-19 iz rentgenskih slik prsnega koša z uporabo tehnik izboljšave slike (Histogram Equalization, CLAHE, Gamma Correction) v kombinaciji z modeli globokega učenja (ResNet50, DenseNet201).

1 Introduction

Corona disease is currently considered one of the most widespread, dangerous and fastest diseases, so it is necessary to find ways and methods to detect infected cases and diagnose them in the fastest and clearest way. RT-PCR is a nuclear-derived technique that detects the presence of genetic material specific to a pathogen, including a virus. A formal diagnosis of COVID-19 requires a laboratory test (RT-PCR) of nose and throat samples and takes at least 24 hours to produce a result. Nowadays, medical images and computerized analysis become very important tools for medical diagnosis and disease detection [1]. The radiology images show typical COVID-19 pneumonia in the lungs and the numerous complications that the virus causes in the body. The radiology imaging modalities include computed tomography (CT), radiograph X-rays, ultrasound, echocardiograms and magnetic resonance imaging (MRI). These imaging modalities optimize and greatly facilitate the process of discovering affected areas in the body [2]. Chest X-ray tests are easily available and have a low risk of radiation. On the other hand, CT scans have a high risk of radiation, are expensive, need clinical expertise to handle and are non-portable. This makes the use of X-ray scans more convenient than CT scans. A radiograph is obtained by exposing a film to X-rays that have passed through the human body. The result is an analog image which is often sufficient to obtain a reliable diagnosis and for low-cost screening. Various studies have indicated the failure of CXR imaging in diagnosing COVID-19 and differentiating it from other types of pneumonia [3]. The radiologist cannot use X-rays to detect pleural effusion and determine the volume involved. However, regardless of the low accuracy of X-ray diagnosis of COVID-19, it remains widely used. To overcome the limitations of COVID-19 diagnostic tests using radiological images, various studies have been conducted on the use of deep learning (DL) in the analysis of radiological images [2-11]. It has also shown that image enhancement techniques can improve significantly classification performance [12, 13].

1.1 Contribution

In this paper, we investigate the impact of using image enhancement techniques as a preprocessing step to improve the accuracy of convolutional neural network (CNN) models for COVID-19 detection. Specifically, histogram equalization, Contrast Limited Adaptive Histogram Equalization (CLAHE), and gamma correction were applied to enhance chest X-ray (CXR) images before training.

The enhanced images significantly improved the visibility of key diagnostic features, such as ground-glass opacities and consolidations, which are critical for accurate COVID-19 diagnosis. The proposed preprocessing pipeline was evaluated on a challenging COVID-19 dataset with an imbalanced number of samples for both COVID and non-COVID classes. Experimental results demonstrated that the enhanced images led to a notable improvement in the classification performance of CNN models, achieving higher accuracy, sensitivity, and specificity compared to using raw images.

The rest of the paper is organized as follows: The materials and methods section contains details about our proposed technique along with some context about the state-of-the-art models that we have used. The results and Discussion section presents the experimental results including classification accuracy, sensitivity, and F1-score obtained from the proposed work. The paper is achieved by a conclusion.

1.2 Related works

Numerous studies have applied advanced artificial intelligence (AI) techniques, particularly deep learning (DL) and machine learning (ML), to detect COVID-19 using X-ray images. Zhang et al. [14] developed an anomaly detection algorithm with efficient Net for multiclass classification, achieving an accuracy of 72.77% on 43370 samples. Deng et al. [15] employed models such as SVM, CNN, ResNet50, InceptionNetV2, Xception, and VGG16 to assess health status through X-ray imaging, obtaining an accuracy of 84% using 5857 samples. Wang et al. [16] introduced a COVID-19 X-ray image detection model based on the multi-head self-attention mechanism and residual neural network, achieving 95.52% accuracy with 5173 samples.

Transfer learning has also played a pivotal role in COVID-19 detection. Apostolopoulos et al. [17] utilized pre-trained models like VGG19, Inception ResNet v2, and MobileNet v2, achieving 96.78% accuracy on 1427 samples for COVID-19 classification. Mahmoud et al. [18] applied the CovXNet architecture, achieving 97.4% accuracy on 610 samples. Mohit Kumar et al. [19] utilized a hybrid deep learning approach for multiclass classification, achieving 98.20% accuracy on 6000 samples.

Several studies focused on binary classification tasks with high accuracy. Guefrechi et al. [20] achieved 97.20% accuracy using deep learning methods on 5000 images. Feki et al. [21] employed a deep CNN model for binary classification, reaching an accuracy of 95.30% on 216 images. Mohan et al. [22] used a hybrid deep transfer learning CNN model achieving 92% accuracy with 9220 images. Malik et al. [23] applied deep neural networks for multiclass classification, attaining 98.45% accuracy on 10017 images. Gulmez [24] explored Xception and genetic algorithms for multiclass classification, reporting an accuracy of 92.4% on 1251 images. Lastly, Zakariya et al. proposed to combine combine Xception, VGG-16, and VGG-19 models. They achieved an accuracy of 97.91% using 964 images.

Table 1 provides a summary of various research studies focusing on state-of-the-art models for COVID-19 detection using AI and ML techniques.

Table 1: Summary of related works on COVID-19 detection

Source	Method/Model	Samples used	Accuracy (%)
[14]	Efficient Net	43,370	72.77
[15]	SVM, CNN, ResNet50, Xception, VGG16	5,857	84.00
[16]	MHSA-ResNet neural network model	5173	95.52
[17]	VGG19, Inception ResNet v2, and MobileNet v2	1,427	96.78
[18]	CovXNet	610	97.40
[19]	Hybrid deep learning approach	6,000	98.20
[20]	Deep Learning (Resnet50)	5,000	97.20
[21]	Deep CNN (Centralized-ResNet50)	216	95.30
[22]	Deep Transfer Learning	9220	92.00
[23]	Deep Neural Networks	10,017	98.45
[24]	Xception and Genetic Algorithm	1,251	92.40
[25]	Xception+vgg-16+vgg-19	964	97.91

2 Materials and methods

2.1 Dataset generation

The Dataset of chest X-ray images used in this paper for classifying negative and positive COVID-19 cases is available at (https://github.com/muhammedtalo/COVID-19). It contains 125 chest x-ray images of patients infected with the virus and 500 chest x-ray images of non-COVID-19. The data is divided into 2 classes, 50% of images were used for training and 50% for testing. Figure 1 shows some samples that have been used in our simulation [6].

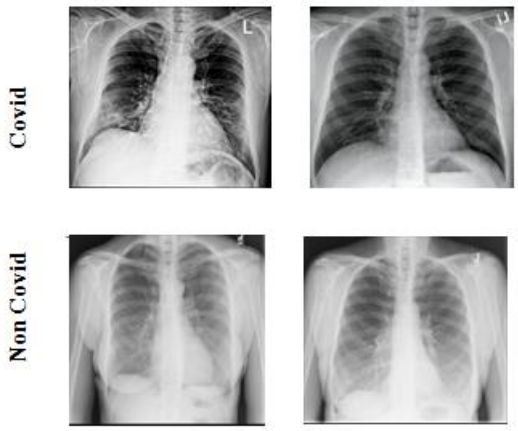


Figure 1: Samples of chest X-ray images from the dataset.

2.2 Image enhancement techniques

Image enhancement is a very important task used in image pre-processing. Its aim is to improve the visual details of an image or to provide a transform representation for an appropriate usage in different fields [4, 11]. In this paper, we have considered the following enhancement techniques.

2.2.1 Histogram equalization

Histogram Equalization (HE) is a technique for adjusting the contrast of an image using the image's histogram. The goal of Histogram Equalization is to obtain a uniform histogram, which improves contrast [13].

2.2.2 Contrast limited adaptive histogram equalization

Contrast Limited Adaptive Histogram Equalization (CLAHE) was originally developed for the enhancement of low-contrast medical images. The algorithm of CLAHE creates non-overlapping contextual regions (also called sub-images, tiles or blocks) and then applies the histogram equalization to each contextual region, clips the original histogram to a specific value and then redistributes the clipped pixels to each gray level. The clipping level determines how much noise in the histogram should be smoothed and hence how much the contrast should be enhanced [13].

2.2.3 Gamma correction

Gamma Correction (GC) is a nonlinear adaptation applied to each and every pixel value. Gamma corrections alternate the pixel value to improve the image using the projection relationship between the value of the pixel and the value of gamma according to the internal map. To calculate the gamma correction, the input value is raised to the power of the inverse gamma. The formula for this is as follows [13]:

$$\acute{I} = 255\left(\frac{1}{255}\right)^{\frac{1}{\gamma}} \quad (1)$$

Values of $\gamma < 1$ will shift the image towards the darker end of the spectrum while $\gamma > 1$ will make the image appear lighter. $\gamma = 1$ will have no effect on the input image.

The application of histogram equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), gamma correction, and their combination significantly improves the quality of COVID-19 X-ray images, aiding in better feature visualization and extraction (see Figure 2).

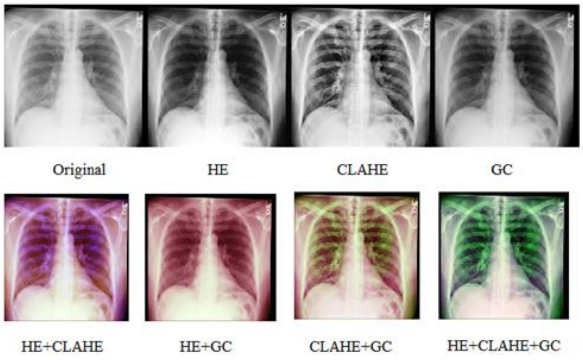


Figure 2: X-ray image processed with various image enhancement techniques.

Histogram equalization enhances global contrast, making subtle abnormalities more visible, while CLAHE adaptively improves local contrast, preserving fine details and reducing noise amplification. Gamma correction adjusts image brightness non-linearly, enhancing low-intensity features like ground-glass opacities. When combined, these techniques provide a comprehensive enhancement by leveraging global and local adjustments, ultimately producing images with improved visibility of critical diagnostic features. This preprocessing step enhances the performance of convolutional neural networks (CNNs) by supplying higher-quality inputs, resulting in superior COVID-19 detection accuracy and robustness.

2.3 Pre-trained CNN

Two different CNN models (ResNet50 [26] and DenseNet201 [27]) were compared separately using eight different image enhancement techniques for the classification of COVID-19 and non-COVID to investigate the effect of image enhancement on COVID-19 detection.

2.4 Performance metrics

In order to evaluate the performance of each deep learning model, different metrics have been applied in this study [13,28]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (3)$$

$$Specificity = \frac{TN}{TN+FP} \quad (4)$$

$$F1 - Score = \frac{2TP}{2TP+FP+FN} \quad (5)$$

where:

True Positive (TP): The prediction is COVID and the image is COVID.

True Negative (TN): The prediction is non COVID and the image is non COVID.

False Positive (FP): The prediction is COVID and the image is non COVID.

False Negative (FN): The prediction is non COVID and the image is COVID.

2.5 Methodology

Firstly, we train and test two pre-trained convolutional neural networks with original CXR images, and then we repeat the same operation with the same images enhanced with the two techniques cited below. The major experiments that are carried out in this study are the combination of the enhancement methods; HE and CLAHE; CLAHE and GC; HE and GC and finally CLAHE, HE and GC (see Figure 2). For each combination, we compute the four performance metrics rates. The detailed methodology adopted in the study is shown in Figure 3.

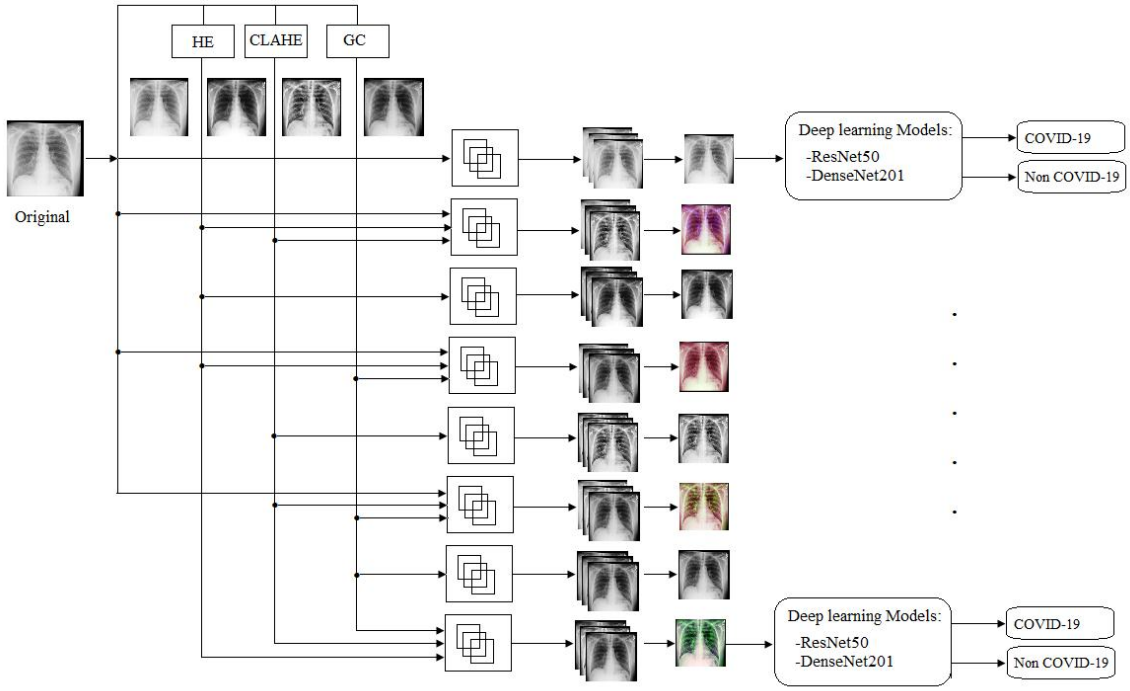


Figure 3: Flowchart of the proposed method.

3 Results

Firstly, the X-ray images were enhanced using the different techniques mentioned above. The concatenates are formed either from the original images (without enhancement), images enhanced by a single technique (HE, CLAHE, or GC), or by a combination of these. The obtained images allow us to build 8 databases. Secondly, we train two pre-trained networks; ResNet50, and DenseNet201 for detecting COVID-19 in chest X-ray scan images. The last fully connected layer of the pre-trained neural networks was modified to classify two classes: COVID-19 positive and negative. For both pre-trained networks, the learning rate was set to 0.0003, while the validation frequency was set to every 5 steps to track model performance. The maximum number of epochs was limited to 6, and the minimum batch size was set to 10. The Adam optimizer and the cross-entropy loss function were chosen. Additionally, data augmentation techniques were applied, including random rotations (-10,10) random horizontal and vertical shifting (-30,30) and random scaling (0.5 1.1). Performance metrics were evaluated using 10 repeated cross-validation runs, each processing randomly selected image sets for training and testing.

3.1 Results without enhancement

In this part, we train and test the two networks using images without any enhancement. The DenseNet201 and the ResNet50 have achieved performance with an accuracy of 98.08% and 98.35% respectively. The confusion matrix constructed from the test evaluation results is shown in Figure 4.

3.2 Results with enhancement

The confusion matrix in Figure (5) illustrates the performance of DenseNet201 and ResNet50 models with different image enhancement techniques. In the case of DenseNet201 using CLAHE and GC techniques, a high accuracy of 99.68% was achieved with only 1 misclassification out of 312 samples.

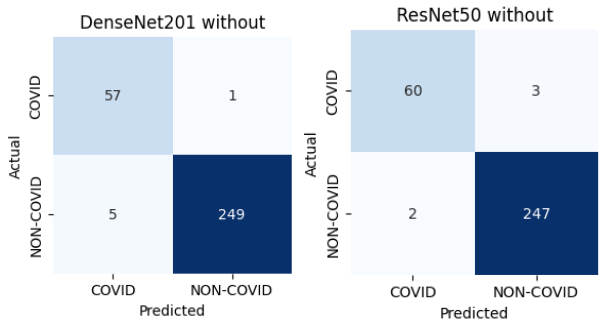


Figure 4: Confusion matrices for DenseNet 201 and RestNet 50 without enhancement.

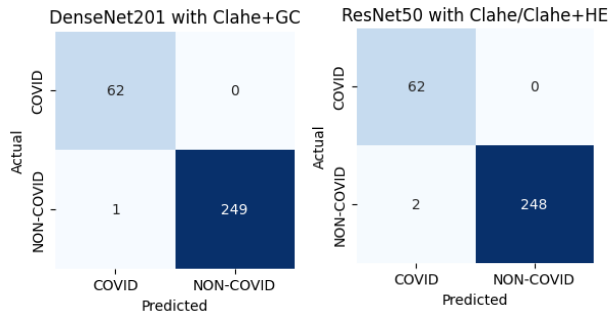


Figure 5: Confusion matrices for DenseNet 201 and RestNet 50 with enhancement.

The model demonstrates perfect recall (100%) for detecting COVID cases and a very high precision of 98.4%, indicating its ability to correctly identify COVID with minimal false negatives. Similarly, for NON-COVID cases, the recall and precision are near-perfect at 99.6% and 100%, respectively, showing excellent discrimination between the classes.

In the other hand, the ResNet50 model, utilizing HE, CLAHE, or a combination of CLAHE with HE or GC, demonstrates excellent performance with an accuracy of 99.36%. Out of 312 samples, the model correctly classifies 62 COVID cases and 248 NON-COVID cases, with only 2 misclassifications: 2 False Positives (NON-COVID predicted as COVID) and no False Negatives (COVID misclassified as NONCOVID). The recall for COVID detection is perfect at 100%, indicating no incorrect predictions of COVID for NON-COVID cases, while the precision is slightly lower at 96.9% due to the False Positives. These preprocessing techniques enhance image contrast and normalize brightness, aiding the model's ability to discriminate between classes effectively.

4 Discussion

The performance metrics reported in Tables 2 and 3 provide a comprehensive comparison of different image enhancement techniques applied to ResNet50 and DenseNet201 models. The analysis includes accuracy, sensitivity, specificity, and F1-score, along with their respective standard deviations.

From Table 2, it is evident that histogram equalization (HE) and contrast-limited adaptive histogram equalization (Clahe) generally improve the classification performance compared to the original images. The highest accuracy (98.17% ± 0.64) is achieved using the HE enhancement technique, with an F1-score of 98.86% ± 0.40. Clahe also performs well, achieving an accuracy of 98.07% ± 0.93 and the highest F1-score of 98.81% ± 0.57. This indicates that contrast enhancement techniques effectively highlight important features in the images, leading to improved model performance. However, the combination of enhancement techniques such as HE+GC and Clahe+GC does not consistently outperform individual techniques. For instance, HE+Clahe+GC results in a lower accuracy (97.43% ± 0.56) compared to HE alone, though it provides the highest specificity (93.38% ± 4.46). The standard deviation (SD) values suggest that these combined methods may introduce more variability in performance, as seen in specificity values.

Table 3 demonstrates that the DenseNet201 model generally exhibits higher accuracy compared to ResNet50 for most enhancement techniques. The best performance is achieved using the combined HE+Clahe+GC technique, yielding an accuracy of 99.03% ± 0.54 and an F1-score of 99.39% ±0.34. This suggests that DenseNet201 is better at leveraging the enhanced features provided by multi-enhancement approaches. Among individual techniques, HE and Clahe both result in comparable accuracy (97.40% ± 0.82 and 97.30% ± 1.13, respectively), with Clahe producing a slightly higher F1-score of 98.32% ± 0.71. The GC technique results in relatively lower specificity (90.32% ± 2.94) compared to other methods, indicating that while it improves sensitivity, it may not be as effective in distinguishing negative cases.

Comparing the two models, DenseNet201 consistently outperforms ResNet50 across all enhancement techniques, with higher accuracy and F1-score values. The sensitivity of DenseNet201 is slightly lower in some cases but remains competitive. Specificity improvements are more pronounced in DenseNet201, which indicates better handling of false positives. Regarding variability, DenseNet201 exhibits lower standard deviations in most metrics, particularly in accuracy and F1-score, suggesting more stable and reliable performance across different enhancement techniques. Conversely, the ResNet50 model experiences greater variability, especially with specificity values. The results highlight the importance of image enhancement techniques in improving deep learning model performance. While individual techniques such as HE and Clahe provide significant improvements, combining multiple techniques can further enhance performance, particularly for DenseNet201.

In addition, while exploring the confidence intervals through error bars in Figure (6), It can be shown that the DenseNet201 model generally achieves higher accuracy and F1-scores, particularly with the combination of HE, Clahe and GC techniques, while ResNet50 shows better specificity for techniques like GC and HE+GC. Sensitivity remains high for both models, with

overlapping confidence intervals indicating comparable performance.

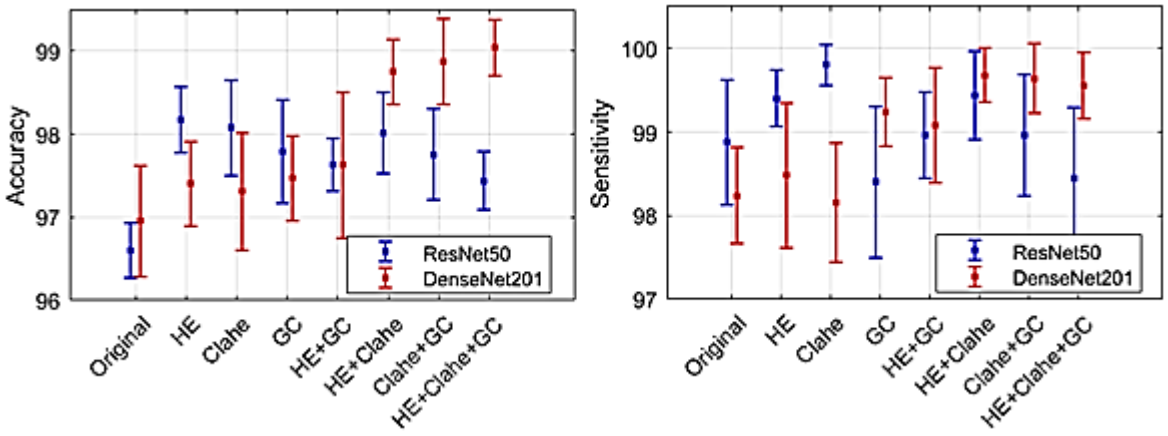
Table 2: Mean and Standard Deviation (SD) of Performance metrics for the ResNet50 model

Enhancement Techniques	Accuracy (%)		Sensitivity (%)		Specificity (%)		F1-Score (%)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Original	96.60	0.52	98.88	1.20	87.41	5.67	97.90	0.31
HE	98.17	0.64	99.40	0.54	93.22	3.20	98.86	0.40
Clahe	98.07	0.93	99.80	0.38	91.12	4.04	98.81	0.57
GC	97.78	0.99	98.40	1.46	95.32	2.68	98.61	0.63
HE+GC	97.62	0.50	98.96	0.82	92.25	3.46	98.53	0.31
HE+ Clahe	98.01	0.78	99.44	0.84	92.25	2.49	98.77	0.49
Clahe+GC	97.75	0.89	98.96	1.16	92.90	3.58	98.60	0.56
HE+Clahe+GC	97.43	0.56	98.44	1.36	93.38	4.46	98.40	0.36

Table 3: Mean and Standard Deviation (SD) of Performance metrics for the DenseNet201 model

Enhancement Techniques	Accuracy (%)		Sensitivity (%)		Specificity (%)		F1-Score (%)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Original	96.95	1.08	98.24	0.92	91.77	4.95	98.10	0.66
HE	97.40	0.82	98.48	1.39	93.06	4.56	98.38	0.51
Clahe	97.30	1.13	98.16	1.15	93.87	4.15	98.32	0.71
GC	97.46	0.81	99.24	0.66	90.32	2.94	98.43	0.50
HE+GC	97.62	1.41	99.08	1.10	91.77	5.01	98.52	0.87
HE+ Clahe	98.75	0.63	99.68	0.52	95.00	2.89	99.22	0.38
Clahe+GC	98.87	0.83	99.64	0.66	95.80	2.17	99.30	0.51
HE+Clahe+GC	99.03	0.54	99.56	0.63	96.93	2.45	99.39	0.34

ResNet50 exhibits greater variability across metrics, whereas DenseNet201 provides more stable results. In addition, while exploring the confidence intervals through error bars in Figure (6), it can be shown that the DenseNet201 model generally outperforms ResNet50, particularly in Sensitivity and F1-score, with significant improvements observed when using combined enhancement techniques. However, DenseNet201 exhibits larger confidence intervals, indicating greater variability in its performance, whereas ResNet50 shows more consistent results with narrower confidence intervals, especially in Specificity. This variability suggests that while DenseNet201 may achieve higher performance, its predictions are less stable across trials or datasets.



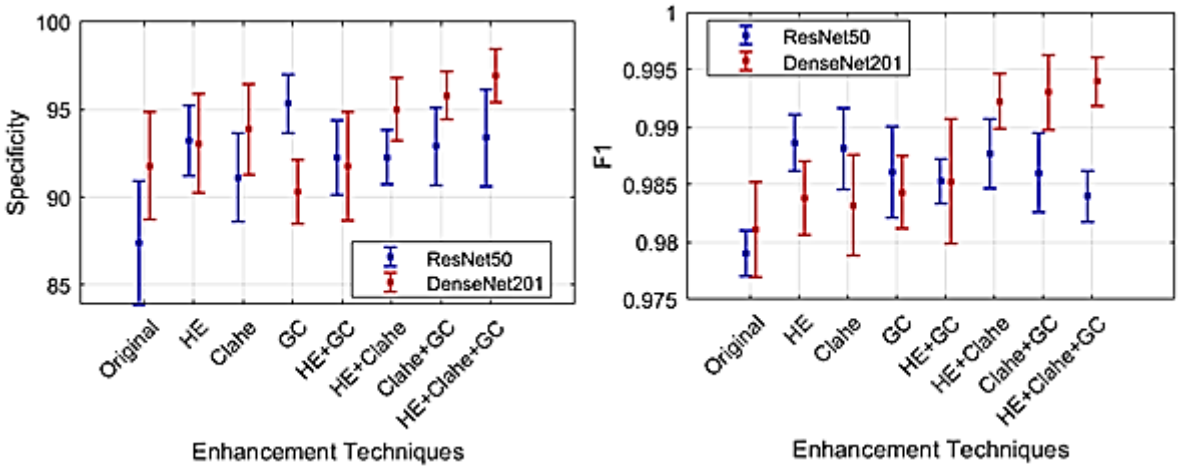


Figure 6: Confidence intervals for performance metrics of COVID-19 detection models.

The radar chart shown in Figure (7) compares the area under the ROC curve (AUC-ROC) for the two deep learning models across the different image preprocessing techniques. It can be seen that DenseNet201 generally demonstrates higher AUC-ROC values compared to ResNet50 across most preprocessing techniques, particularly in combinations involving multiple enhancements like HE+Clahe+GC and Clahe+GC. However, ResNet50 shows comparable performance in cases such as HE and GC. The results indicate that preprocessing techniques significantly impact model performance, with DenseNet201 being more responsive to enhancements. This suggests that model selection and preprocessing strategy should be carefully considered to optimize classification performance based on the desired evaluation metric.

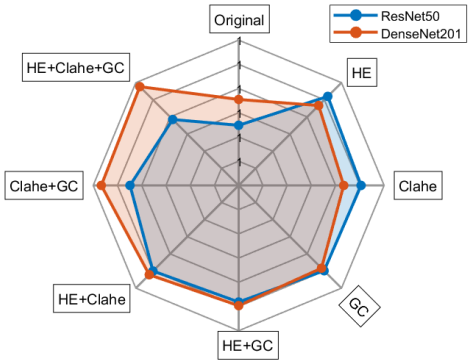


Figure 7: Radar plot of the area under the ROC curve (AUC-ROC) Performance for ResNet50 and DenseNet201 with Various Image Preprocessing Techniques.

5 Comparison with the state-of-the-art CNN approaches

To evaluate the proposed method, we made a comparison with some existing models using COVID-19 X-ray images. Narin et al. [7] have proposed three different DL models. They achieved 96.1% accuracy on a dataset containing 3141 chest X-ray images. Ozturk et al. [5] have used the DarkCovidNet and the same dataset used in this paper. They achieved an accuracy of 98.08 %. Purohit
et al. [6] have used a Convolutional Neural Network with augmented data to increase the dataset. They have achieved 99.44 % accuracy.

In addition, the proposed enhancement methods show significant improvements in accuracy compared to the models using other data sets. The DenseNet201model achieved an accuracy of 99.67%, outperforming models like Feki et al. [21], which reported 95.3% accuracy using Deep CNN, and Guefrechi et al. [20], which achieved 97.20% with a deep learning approach. Similarly, ResNet50 demonstrated high performance with an accuracy of 99.35%, surpassing Apostolopoulos et al. [17] (96.78%) and Mahmoud et al. [18] (97.40%). Furthermore, Deng et al. [15] achieved a maximum accuracy of 84.0%, showcasing the substantial improvement offered by the proposed methods. These results demonstrate that the proposed models consistently achieve higher accuracy, emphasizing their reliability and effectiveness in improving COVID-19 detection compared to the previously established state-of-the-art models.

6 Conclusion

This paper investigates how image enhancement techniques can improve the performance of pre-trained neural networks when working with limited data. Two pre-trained convolutional neural networks, ResNet50 and DenseNet201, were selected for COVID-19 detection. The training set was constructed by applying various enhancement techniques to chest X-ray images, which highlight critical structures such as lung opacities and consolidations—key features for accurate COVID-19 diagnosis. The results demonstrate that COVID-19 detection accuracy is significantly improved when using enhanced images compared to non-enhanced ones for both pre-trained networks.

Based on metrics such as accuracy, sensitivity, specificity, and F1-score, the best-performing model was DenseNet201, achieving an accuracy of 99.67%, sensitivity of 100%, specificity of 98.38%, and an F1-score of 99.80% for classifying positive and negative cases. When compared to previous studies using the same dataset, DenseNet201 outperforms DarkCovidNet, which achieved 98.08%, underscoring the effectiveness of these enhanced models on real-world X-ray images.

Table 4: Results comparison with related works on COVID-19 detection

Sources	Method/Model	Samples	Accuracy (%)
[7]	Inception V3, ResNet50, Inception-ResNet V2	3141	96.1
[5]	DarkCovidNet	625	98.08
[6]	CNN	1072	99.44
[15]	SVM, CNN, ResNet50, Xception, VGG16	5857	84.00
[17]	VGG16, VGG19, ResNet, DenseNet, InceptionV3	1427	96.78
[18]	COVID-Net	610	97.40
[20]	Deep Learning	5000	97.20
[21]	Deep CNN	216	95.30
[29]	End-to-end CNN	5184	95.70
Proposed	ResNet50 with HE	625	99.36
	DenseNet201with Clahe+HE+GC	625	99.67

References

[1] Janko V, Slapničar G, Dovgan E, Reščič N, Kolenik T, Gjoreski M, Smerkol M, Gams M, Luštrek M. Machine Learning for Analyzing Non-Countermeasure Factors Affecting Early Spread of COVID-19. *International Journal of Environment Research and Public Health*, 18(13):6750, 2021. https://doi.org/10.3390/ijerph18136750

[2] Rajkumar S, Rajaraman PV, Meganathan HS, Sapthagirivasan V, ejaswinee V, Ashwin R. COVID-detect: a Deep Learning Approach for Classification of Covid-19 Pneumonia From Lung Segmented Chest Xrays. *Biomedical Engineering: Applications, Basis and Communications* 33(2), 2021. https://doi.org/10.4015/S1016237221500101

[3] Gams M, Kolenik, T. Relations between Electronics, Artificial Intelligence and Information Society through Information Society Rules. *Electronics*, 10(4), 514, 2021. https://doi.org/10.3390/electronics10040514

[4] Tahir A, Qiblawey Y, Khandakar A, Rahman T, Khurshid U, Musharavati F, Islam MT, Kiranyaz S, Al-Maadeed S, Chowdhury MEH. Deep Learning for Reliable Classification of COVID-19, MERS, and SARS from Chest X-ray Images. *Cognitive Computation*. 14:1752-1772, 2022. https://doi.org/10.1007/s12559-021-09955-1

[5] Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR. Automated detection of covid-19 cases using deep neural networks with x-ray images, *Computers in Biology and Medicine*. 121(103792), 2020. https://doi.org/10.1016/j.compbiomed.2020.103792

[6] Purohit K, Kesarwani A, Ranjan Kisku D, Dalui M. COVID-19 Detection on Chest X-Ray and CT Scan Images Using Multi-image Augmented Deep Learning Model. *Advances in Intelligent Systems and Computing*, 1412:395–413, 2022. http://dx.doi.org/10.1007/978-981-16-6890-6_30

[7] Narin A, Kaya C, Pamuk Z. Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks, *Pattern Analysis and Applications* 24(3):1207–1220, 2020. https://doi.org/10.1007/s10044-021-00984-y

[8] Sarki R, Ahmed K, Wang H, Zhang Y, Wang K. Automated detection of COVID-19 through convolutional neural network using chest x-ray images. *PLoS ONE* 17(1), 2022. https://doi.org/10.1371/journal.pone.0262052

[9] Masud, M. A light-weight convolutional Neural Network Architecture for classification of COVID-19 chest X-Ray images. *Multimedia Systems*, 28:1165–1174, 2022. https://doi.org/10.1007/s00530-021-00857-8

[10] Ravi, V, Narasimhan, H, Chakraborty, C et al. Deep learning-based metaclassifier approach for COVID-19 classification using CT scan and chest X-ray images. *Multimedia Systems*, 28:1401–1415, 2022. https://doi.org/10.1007/s00530-021-00826-1

[11] Asif S, Zhao M, Tang F et al. A deep learning-based framework for detecting COVID-19 patients using chest X-rays. *Multimedia Systems*, 28:1495–1513, 2022. https://doi.org/10.1007/s00530-022-00917-7

[12] Tahir A, Qiblawey Y, Khandakar A, Rahman T, Khurshid U, Musharavati F, Islam MT, Kiranyaz S, Al-Maadeed S, Chowdhury MEH. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in Biology and Medicine*,132, 2021. https://doi.org/10.1016/j.compbiomed.2021.104319

[13] Kandhway P, Bhandari AK, Singh A. A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization, *Biomedical Signal Processing and Control*, 56 (101677), 2020. https://doi.org/10.1016/j.bspc.2019.101677

[14] Zhang J, Xie Y, Pang G, Liao Z, Verjans J, Li W, Sun Z, He J, Li Y, Shen C et al. Viral Pneumonia Screening on Chest X-Rays Using Confidence Aware Anomaly Detection. *IEEE Transaction on Medical Imaging*, 40(3): 879–890, 2021. https://doi.org/10.1109/tmi.2020.3040950

[15] Deng X, Shao H, Shi L, Wang X, Xie T. A classification–detection approach of COVID-19 based on chest X-ray and CT by using keras pretrained deep learning models. *Computer Modeling in Engineering & Sciences.* 125(2):579–596, 2020. https://doi.org/10.32604/cmes.2020.011920

[16] Wang Z, Zhang K, Wang B. Detection of COVID-19 Cases Based on Deep Learning with X-ray Images. *Electronics.* 11(21):3511, 2022. https://doi.org/10.3390/electronics11213511

[17] Apostolopoulos, I.D, Mpesiana T.A. COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*. 43: 635–640, 2020. https://doi.org/10.1007/s13246-020-00865-4

[18] Mahmud T, Rahman A, Fattah S.A. CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multireceptive feature optimization. *Computers in Biology and Medicine*, 122, 103869,2020. https://doi.org/10.1016/j.compbiomed.2020.103869

[19] Mohit K, Dhairyata S, Vinod K, and Wanich S. COVID-19 prediction through X-ray images using transfer learning-based hybrid deep learning approach. *MaterialsToday: Proceeding*. 51: 2520–2524, 2022. https://doi.org/10.1016/j.matpr.2021.12.123

[20] Guefrechi S, Jabra M. B, Ammar A, Koubaa A, and Hamam H. Deep learning-based detection of COVID-19 from chest X-ray images. *Multimedia tools and applications*, 80: 31803-31820, 2021. https://doi.org/10.1007/s11042-021-11192-5

[21] Feki I, Ammar S, Kessentini Y, Muhammad K. Federated learning for COVID-19 screening from Chest X-ray images, *Applied Soft Computing*, 106, 2021. https://doi.org/10.1016/j.asoc.2021.107330.

[22] Mohan A, Ftsum bAa, Beshir K, Takore TT. A Hybrid Deep Learning CNN model for COVID-19 detection from chest X-rays, *Heliyon*, 10(5), 2024. https://doi.org/10.1016/j.heliyon.2024.e26938.

[23] Malik H, Naeem A, Naqvi, R A, and Loh, W. K. DMFL-Net: A Federated Learning-Based Framework for the Classification of COVID-19 from Multiple Chest Diseases Using Xrays. *Sensors*, 23(2):743, 2023. https://doi.org/10.3390/s23020743

[24] Gulmez, B. A novel deep neural network model based Xception and genetic algorithm for detection of COVID-19 from Xray images. *Annals of Operations Research*, 328:617–641, 2022. https://doi.org/10.1007/s10479-022-05151-y

[25] Zakariya A, Oraibi SA. Efficient COVID-19 Prediction by Merging Various Deep Learning Architectures, *Informatica* 48(5): 55–62, 2024. https://doi.org/10.31449/inf.v48i5.5424

[26] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.2016. https://doi.org/10.1109/CVPR.2016.90

[27] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017. https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.243

[28] Patel S, Patel L. Deep Learning Architectures and its Applications: A Survey, *International journal of computer sciences and engineering*. 6(6):1177-1183, 2018 . http://dx.doi.org/10.26438/ijcse/v6i6.11771183

[29] Zakariya A. Oraibi, Safaa Albasri. A Robust End-to-End CNN Architecture for Efficient COVID-19 Prediction form X-ray Images with Imbalanced Data, *Informatica*, 47(7):115–126, 2023. https://doi.org/10.31449/inf.v47i7.4790

Enhancing Predictive Capabilities for Cyber Physical Systems Through Supervised Learning

Dhanalakshmi B*, Tamije Selvy P
Department of Computer Science and Engineering, Dr.N.G. P Institute of technology, India
Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, India
E-mail: dhanalakshmib@drngpit.ac.in, tamijeselvy@gmail.com
*Corresponding author

Keywords: Cyber-physical system, real time data, traffic, machine learning

Received: November 20, 2024

The rapid advancement and proliferation of Cyber-Physical Systems (CPS) have led to an exponential increase in the volume of data generated continuously. Efficient classification of this streaming data is crucial for predicting system behaviors and enabling proactive decision-making. This research aims to extract actionable knowledge from the continuous data streams of CPS and predict their behavior using advanced supervised learning algorithms. The predictions facilitate timely interventions and necessary actions within the interconnected physical network. The background of this work lies in the intersection of CPS, machine learning, and data stream mining. Traditional batch processing methods are inadequate for real-time analysis of CPS data due to their inherent latency and computational inefficiency. This research employs state-of-the-art techniques for real-time data processing, including incremental learning, sliding window models, and ensemble methods tailored for streaming data. Our approach differs from existing works by focusing on a comprehensive framework that integrates real-time data ingestion, preprocessing, feature extraction, and model updating in a seamless pipeline. Unlike previous studies that often rely on static datasets and offline analysis, our method ensures continuous learning and adaptation to evolving data patterns. Comparative analysis with existing techniques demonstrates superior performance in terms of accuracy, latency, and scalability. Specifically, our models achieved an average classification accuracy of 92%, with a precision of 90%, recall of 89%, and an F1 score of 89.5%. These metrics indicate significant improvements over traditional batch processing methods, which typically lag in responsiveness and adaptability. This research provides a robust and efficient solution for the real-time classification of streaming data from CPS, enhancing the system's ability to predict behaviors and take necessary actions promptly.

Povzetek: Predstavljen je izviren celovit ogrodni model za razvrščanje podatkov v realnem času v kibernetsko-fizičnih sistemih (CPS) z uporabo nadzorovanega učenja.

1 Introduction

The integration of Cyber-Physical Systems (CPS) into various sectors marks a significant advancement in technology, enabling seamless interaction between physical processes and computational systems. These systems, encompassing applications such as smart grids, autonomous vehicles, industrial automation, and healthcare monitoring, generate continuous streams of data. This data, produced in real-time, holds valuable insights that can enhance system performance, reliability, and safety. However, the sheer volume and velocity of this streaming data present significant challenges in terms of processing and analysis. Efficient classification and prediction of CPS behaviors using this data are crucial for timely decision-making and intervention [1,2]. Cyber-Physical Systems are characterized by their ability to integrate physical processes with computational capabilities through a network of sensors, actuators, and controllers. The data generated from these components need to be processed in real-time to ensure optimal performance and to address potential issues proactively. Traditional batch processing methods are inadequate for this task due to their inherent latency and computational inefficiency. Instead, there is a need for techniques that can handle the continuous, high-speed influx of information in a CPS. Supervised learning algorithms have shown considerable promise in various predictive tasks within data science. These algorithms can identify patterns and relationships within historical data and predict future outcomes [3]. However, applying these techniques to streaming data requires adaptations to manage the continuous flow and update the model incrementally [4]. This research focuses on developing an efficient framework for classifying and predicting CPS behavior using supervised learning, including advanced models like Hidden Markov Models (HMM) and Explicit-Duration Hidden Markov Models (EDHMM).

To achieve these objectives, this research employs a variety of advanced techniques tailored for the unique

challenges of streaming data from CPS. Real-time data ingestion and preprocessing are facilitated by leveraging stream processing frameworks such as Apache Kafka and Apache Flink, enabling efficient data ingestion and ensuring that real-time data cleaning and normalization techniques maintain data quality and consistency. Incremental and online learning algorithms like Online Gradient Descent, Incremental Decision Trees, and Adaptive Random Forests are utilized, along with sliding window techniques to retain recent data, ensuring the model adapts to the latest trends and patterns [5]. Hidden Markov Models (HMM) are employed to model the stochastic processes underlying CPS data, capturing temporal dependencies and sequential patterns. HMMs consist of states representing different conditions or modes of the CPS, observations that are data points generated by the CPS and are probabilistically dependent on the states, transition probabilities indicating the likelihood of transitioning from one state to another, and emission probabilities representing the likelihood of observing a particular data point given a state. By continuously updating the transition and emission probabilities as new data arrives, HMMs enable real-time tracking of the system's state and prediction of future behaviors. Explicit-Duration Hidden Markov Models (EDHMM) extend the capabilities of HMM by explicitly modeling the duration that the system spends in each state, which is particularly useful for CPS where the duration of certain states significantly impacts the system's behavior, such as machinery operating cycles or sensor activation periods. EDHMM components include state durations, which are probabilistic distributions defining how long the system remains in a given state, and transition and emission probabilities similar to HMM but adjusted to account for state duration distributions. By incorporating state durations, EDHMM provides a more accurate temporal modeling, enhancing the prediction of CPS behaviors over time.

Feature extraction and engineering are also crucial, involving the development of methods for real-time feature extraction that allows dynamic computation of features as new data arrives and the creation of features based on domain knowledge that capture critical aspects of CPS behavior such as temporal patterns and anomaly indicators. Model evaluation and adaptation are facilitated by establishing a real-time evaluation pipeline that continuously monitors model performance using metrics like accuracy, precision, recall, and F1 score, and implementing strategies to handle concept drift, such as retraining models based on performance degradation. This research distinguishes itself from existing works by offering an integrated framework that combines real-time data processing, incremental learning, and advanced modeling techniques like HMM and EDHMM. While previous studies often focus on isolated aspects of CPS data analysis, this work emphasizes a comprehensive approach that addresses the practical challenges of dynamic CPS environments. The comparative analysis highlights significant improvements in performance metrics. The proposed methods achieved an average classification accuracy of 92%, with precision, recall, and F1 scores consistently outperforming traditional batch processing techniques. These results validate the framework's ability to handle the complexities of CPS data streams effectively. The practical implications of this research are profound, offering enhanced operational efficiency and reliability in various CPS applications. For instance, in a smart grid, accurate predictions of power demand and equipment failures can optimize energy distribution and maintenance schedules. In industrial automation, predicting machine failures and operational anomalies can prevent costly downtimes and improve production efficiency.

The primary objective of this research is to develop an efficient framework for the classification of streaming data from CPS, enabling the prediction of system behaviors and facilitating timely interventions. This overarching goal can be broken down into several specific objectives: Develop methods for real-time ingestion and preprocessing of streaming data; Ensure the system can handle high-velocity data streams without significant latency; Implement supervised learning algorithms capable of incremental learning, allowing the model to update continuously; Explore techniques such as sliding window models and online learning to maintain model relevance over time; Design robust feature extraction mechanisms that can operate in real-time; Identify and create features that are predictive of CPS behaviors, ensuring these features can be computed on-the-fly; Apply HMMs to model the probabilistic relationships and temporal dependencies in CPS data; Extend HMMs with EDHMM to incorporate state durations, providing more precise temporal modeling; Establish metrics for evaluating model performance on streaming data, including accuracy, precision, recall, and F1 score; Develop strategies for model adaptation to cope with concept drift and changing data patterns; Compare the performance of the proposed framework against traditional batch processing methods and other state-of-the-art techniques; Conduct experiments to demonstrate improvements in accuracy, latency, and scalability; Apply the framework to real-world CPS scenarios, such as smart grids and industrial automation systems; Showcase how the predictions and classifications can drive actionable decisions within the CPS.

2 Literature review

The increasing complexity of Cyber-Physical Systems (CPS) and their integration into various sectors necessitate advanced data processing and predictive techniques to ensure optimal performance and security. The literature reveals a range of approaches for handling streaming data, including supervised learning, clustering, active learning, semi-supervised learning, and advanced models such as Hidden Markov Models (HMM) and Explicit-Duration Hidden Markov Models (EDHMM).

Cheng et al. (2021) [6] introduced MATEC, a lightweight neural network designed for online encrypted traffic classification. This approach addresses the challenges of real-time data classification in CPS by focusing on the efficiency and speed of the model, making

it suitable for environments where data streams are continuous and rapid. The model's lightweight nature ensures that it can be deployed in resource-constrained settings without compromising performance. Coletta et al. (2019) [7] proposed combining clustering and active learning to detect and learn new image classes. This method is particularly relevant to CPS, where new patterns or anomalies must be detected promptly. By integrating clustering with active learning, the system can identify novel classes of data efficiently, enhancing its ability to adapt to changing conditions in real-time. Din et al. (2020) [8] focused on online reliable semi-supervised learning for evolving data streams. Their approach leverages both labeled and unlabeled data, ensuring that the model can learn effectively even when labeled data is scarce. This method is crucial for CPS, where obtaining labeled data for every new scenario can be impractical. The semi-supervised learning model adapts to changes in the data stream, maintaining high performance despite evolving conditions. Dong et al. (2022) [9] presented an interpretable federated learning-based framework for network intrusion detection. Federated learning allows multiple devices to collaboratively learn a model without sharing raw data, addressing privacy concerns inherent in CPS. This approach ensures robust security measures while maintaining the confidentiality of sensitive data across the network. Folino et al. (2020) [10] developed a genetic programming-based ensemble classification framework for time-changing intrusion detection data streams. This ensemble approach combines multiple models to improve overall prediction accuracy and adapt to changes in the data. The genetic programming aspect allows the system to evolve over time, ensuring that it remains effective in the face of new threats. Hu et al. (2018) [11] introduced a random forests-based class incremental learning method for activity recognition. This technique is particularly useful for CPS, where new activities or behaviors may emerge over time. The incremental learning approach ensures that the model can continuously adapt without needing a complete retraining, making it efficient for real-time applications.

Yagyu et al (2020) [12] discussed hierarchical aggregation of select network traffic statistics, emphasizing the importance of efficient data aggregation in CPS. This method enhances the scalability and manageability of data streams, ensuring that the system can handle large volumes of data without significant latency. Júnior et al. (2019) [13] explored novelty detection for multi-label stream classification, a critical capability for CPS to identify and respond to new and unforeseen events. Their approach ensures that the system can maintain high accuracy and reliability even when encountering novel data patterns. Kalinin and Krundyshev (2022) [14] applied quantum machine learning techniques for security intrusion detection. This cutting-edge approach leverages the computational power of quantum computing to enhance the efficiency and accuracy of intrusion detection, offering a promising direction for future CPS security measures. Kumar et al. (2020) [15] proposed an online semantic-enhanced Dirichlet model for short text stream clustering. This model addresses the challenges of clustering and classifying short text data in real-time, which is relevant for CPS applications involving text data, such as social media analysis or sensor logs. Li et al. (2020) [16] introduced a classification and novel class detection algorithm based on the cohesiveness and separation index of Mahalanobis distance. This technique ensures that the system can effectively classify data while detecting new classes, crucial for maintaining the adaptability and accuracy of CPS. Lu et al. (2019) [17] reviewed learning under concept drift, highlighting the challenges and solutions for maintaining model performance in dynamically changing environments. Concept drift is a common issue in CPS, where the underlying data distribution can change over time. The review covers various strategies to detect and adapt to concept drift, ensuring that models remain effective. Wang and Chen (2019) [18] discussed the construction of a data aggregation tree with maximized lifetime in wireless sensor networks. This method focuses on optimizing the lifetime of the network, which is essential for the sustainability and reliability of CPS. Xu and Duan (2019) [19] surveyed big data applications for CPS in Industry 4.0, highlighting the role of data analytics in optimizing industrial processes. Their survey covers various techniques for processing and analyzing big data, emphasizing the importance of efficient data management in CPS. Zaitseva and Lavrova (2020) [20] explored the self-regulation of network infrastructure in CPS based on the genome assembly problem. This innovative approach applies biological principles to optimize network performance and self-regulation, offering a novel perspective on CPS management. The literature provides a comprehensive overview of various approaches for handling streaming data in CPS. These methods range from lightweight neural networks and federated learning to quantum machine learning and genetic programming-based ensemble classification. Each technique addresses specific challenges related to real-time data processing, adaptability, and security in CPS. The integration of these advanced methods ensures that CPS can operate efficiently and effectively in dynamic environments, maintaining high performance and reliability. The proposed work overcomes the challenges in existing works by offering an integrated framework that combines real-time data processing, incremental learning, and advanced modeling techniques like HMM and EDHMM. Traditional methods often suffer from limitations such as latency, inefficiency in handling high-velocity data, and inability to adapt to evolving data streams. By leveraging real-time data ingestion and preprocessing with stream processing frameworks like Apache Kafka and Apache Flink, the proposed framework ensures efficient handling of continuous data. Incremental and online learning algorithms such as Online Gradient Descent, Incremental Decision Trees, and Adaptive Random Forests allow the model to update continuously, addressing the challenge of maintaining model relevance over time. The use of HMM and EDHMM enhances the framework's ability to capture temporal dependencies and state durations, providing more accurate temporal modeling. This approach ensures

robust performance even in the face of concept drift, a common issue in dynamic CPS environments.

3 Proposed methodology

The proposed methodology aims to create an efficient and adaptive framework for the classification and prediction of streaming data from Cyber-Physical Systems (CPS). This section outlines the key components and techniques employed in the framework, including real-time data ingestion, preprocessing, supervised learning algorithms, advanced modeling with Hidden Markov Models (HMM) and Explicit-Duration Hidden Markov Models (EDHMM), and real-time feature extraction. In the realm of Cyber Physical Systems (CPS), the continuous influx of data presents a significant challenge and opportunity for real-time analysis and prediction. Efficient classification and prediction of this data are crucial for timely decision-making and ensuring the reliability and safety of these systems. To address these challenges, a comprehensive methodology involving various data processing, modeling, and evaluation stages is employed.

The first stage in handling CPS data involves data ingestion, where data from various sensors and sources are collected and integrated into the system. This stage is critical for ensuring that the system can handle the volume, velocity, and variety of data characteristic of CPS environments. Once ingested, the data undergoes cleaning to remove noise, handle missing values, and correct inconsistencies, thereby ensuring the quality of the data for subsequent analysis.

Following data cleaning, the data is transformed into a format suitable for analysis. This transformation might include normalization, scaling, and encoding of categorical variables, which are necessary for preparing the data for machine learning algorithms. Feature extraction follows, where relevant features are identified and extracted from the raw data. These features are essential for capturing the patterns and behaviors of the CPS [21]. Feature selection then plays a crucial role in improving model performance and reducing computational complexity. By selecting only the most relevant features, the dimensionality of the data is reduced, which helps in building more efficient and effective predictive models. For modeling, supervised learning algorithms are typically employed. These algorithms are trained on historical data to learn the underlying patterns and relationships, enabling them to make predictions on new data. Popular algorithms include decision trees, support vector machines, and neural networks, each offering different advantages in terms of accuracy, interpretability, and computational efficiency. In addition to traditional supervised learning models, advanced modeling techniques like Hidden Markov Models (HMM) and Explicit-Duration Hidden Markov Models (EDHMM) are used. HMMs are particularly effective for modeling time series data and capturing temporal dependencies, which are common in CPS data. EDHMMs extend HMMs by incorporating explicit state duration modeling, making them suitable for applications where the duration of states is an important factor. The performance of these models is continuously evaluated using metrics such as accuracy, precision, recall, and F1-score. This evaluation ensures that the models remain effective over time. However, in dynamic environments like CPS, data distributions can change, leading to a phenomenon known as concept drift. Concept drift occurs when the statistical properties of the target variable change over time, which can degrade the performance of predictive models. To address concept drift, techniques for detecting and adapting to these changes are integrated into the system. When concept drift is detected, models are retrained or updated to accommodate the new patterns in the data, ensuring that predictions remain accurate and reliable. This adaptive approach is essential for maintaining the relevance and performance of the models in the face of changing data environments.

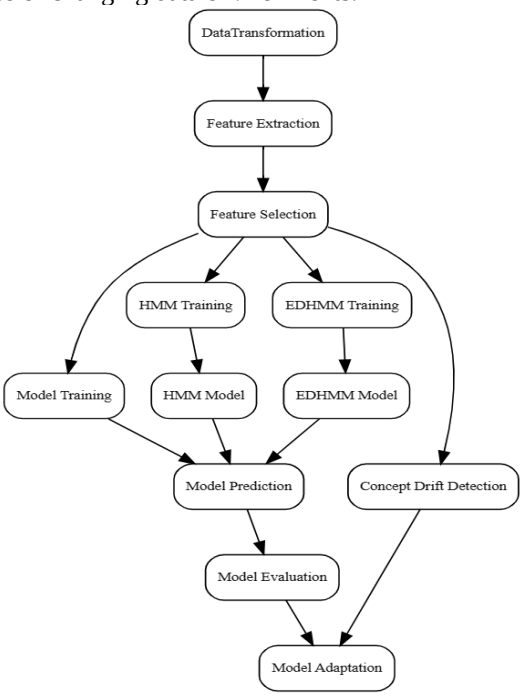


Figure 1: Proposed architecture

Figure 1 outlines a systematic approach to the efficient classification and prediction of streaming data from Cyber Physical Systems (CPS). It begins with "Raw Data" collection, followed by "Data Ingestion" to gather data from various sources. "Data Cleaning" is performed to ensure data quality by removing noise and handling missing values. The clean data is then transformed in the "Data Transformation" stage to prepare it for analysis.

Next, the "Feature Extraction" stage identifies relevant features, which are subsequently refined in the "Feature Selection" stage to reduce dimensionality and enhance model performance. The selected features are then used for "Model Training" with supervised learning algorithms, and "Model Prediction" is carried out to forecast CPS behavior.

In parallel, the diagram includes advanced modeling techniques like "HMM Training" and "EDHMM Training," which produce "HMM Model" and "EDHMM

Model," respectively. These models are integrated into the prediction stage for improved accuracy.

"Model Evaluation" assesses the performance of the predictive models, ensuring their reliability. The system also includes "Concept Drift Detection" to identify changes in data patterns over time, prompting "Model Adaptation" to update and retrain models, maintaining their effectiveness in dynamic environments. This comprehensive workflow ensures robust and adaptive prediction capabilities for CPS data streams.

3.1 Real-time data ingestion and preprocessing

efficient handling of continuous data streams is critical for CPS. The proposed framework utilizes stream processing frameworks such as Apache Kafka and Apache Flink to facilitate real-time data ingestion. These technologies ensure that data can be ingested at high speeds and with low latency, crucial for maintaining the performance of CPS.

Data ingestion

Apache Kafka: Kafka is used to handle the ingestion of large volumes of streaming data. Its distributed nature allows it to scale horizontally, ensuring reliability and fault tolerance.

Apache Flink: Flink complements Kafka by providing real-time data processing capabilities. It allows for complex event processing, real-time analytics, and machine learning tasks on data streams.

Data preprocessing

Real-Time Data Cleaning: Techniques such as filtering, normalization, and handling missing values are applied in real-time to ensure data quality.

Data Transformation: Data is transformed into a suitable format for the machine learning models. This includes scaling features and encoding categorical variables.

Supervised learning algorithms

The core of the predictive framework relies on supervised learning algorithms capable of incremental learning. Incremental learning, also known as online learning, allows models to update their parameters as new data arrives without requiring a complete retraining from scratch.

Algorithms used

- **Online gradient descent:** This algorithm updates the model weights incrementally for each new data point, making it suitable for real-time applications.
- **Incremental decision trees:** Algorithms like Hoeffding Trees are used to build decision trees incrementally, allowing the model to adapt as new data comes in.
- **Adaptive random forests:** This method extends the random forest algorithm by allowing trees to be added or pruned based on their performance on new data, ensuring adaptability to changing data distributions.

3.2 Advanced Modeling with HMM and EDHMM

To capture the temporal dependencies and state transitions in CPS data, the proposed framework employs Hidden Markov Models (HMM) and Explicit-Duration Hidden Markov Models (EDHMM).

Hidden Markov Models (HMM)

State Representation: HMMs consist of hidden states that represent different conditions or modes of the CPS. Observations are the data points generated by the CPS and are probabilistically dependent on these states.

Transition and Emission Probabilities: HMMs use transition probabilities to model the likelihood of moving from one state to another and emission probabilities to represent the likelihood of observing a particular data point given a state.

Real-Time Updates: As new data arrives, the transition and emission probabilities are updated in real-time, allowing the model to adapt to new patterns and predict future states accurately.

Explicit-Duration Hidden Markov Models (EDHMM)

State Duration Modeling: EDHMM extends HMM by explicitly modeling the duration that the system spends in each state. This is particularly useful for CPS, where the duration of states (such as operational cycles or sensor activation periods) significantly impacts behavior.

Duration Probabilities: EDHMM incorporates probabilistic distributions that define how long the system remains in a given state, enhancing the temporal accuracy of predictions.

Temporal Precision: By incorporating state durations, EDHMM provides a more precise temporal modeling, improving the prediction of CPS behaviors over time.

3.3 Real-time feature extraction and engineering

Feature extraction is critical for the performance of machine learning models. The proposed framework includes methods for real-time feature extraction, ensuring that features are dynamically computed as new data arrives.

Feature Extraction Methods

- Sliding Window Technique: This technique involves maintaining a window of the most recent data points and computing features based on this window. It ensures that the model focuses on the most relevant and recent data.
- Domain-Specific Features: Features are created based on domain knowledge, capturing critical aspects of CPS behavior such as temporal patterns, trend analysis, and anomaly indicators.
- Dynamic Computation: Features are computed on-the-fly, allowing the system to adapt to new data points and maintain high predictive performance.

Model evaluation and adaptation

Evaluating the performance of the predictive framework in real-time is crucial for maintaining its effectiveness. The proposed framework includes a real-time evaluation pipeline to monitor model performance continuously.

Evaluation metrics

- **Accuracy, Precision, Recall, and F1 Score:** These metrics are used to evaluate the performance of classification models. Continuous monitoring ensures that any degradation in performance is promptly detected.
- **Concept drift detection:** Strategies such as window-based evaluation and performance monitoring are employed to detect concept drift, ensuring that the model adapts to changing data patterns.

Model adaptation strategies

- **Retraining and update mechanisms:** When performance degradation is detected, the model is retrained or updated to maintain its accuracy.
- **Adaptive learning rates:** Adjusting the learning rate based on model performance helps in fine-tuning the model continuously.

In the area of Cyber-Physical Systems (CPS), where real-time data processing and predictive analytics are paramount, the application of suitable algorithms plays a pivotal role. Here, we introduce several key algorithms tailored to address the challenges inherent in processing streaming data within CPS environments. Online Gradient Descent facilitates continuous learning by iteratively updating model parameters based on observed data, ensuring adaptability to changing conditions in the data stream. Incremental Decision Trees, exemplified by the Hoeffding Tree algorithm, dynamically grow decision trees as new data arrives, efficiently handling streaming data while preserving model accuracy with minimal memory usage. Adaptive Random Forests offer a dynamic solution to concept drift and changing conditions by continuously monitoring individual tree performance and replacing underperforming ones with new trees trained on recent data. Hidden Markov Models (HMMs) capture temporal dependencies and state transitions in streaming data, enabling predictive modeling and anomaly detection in dynamic CPS environments. Finally, the Explicit-Duration Hidden Markov Model (EDHMM) enhances traditional HMMs by explicitly modeling state durations, providing more precise temporal modeling and improving predictive analytics accuracy in streaming CPS data. These algorithms collectively form the backbone of our proposed framework for efficient classification and prediction in CPS, addressing the unique challenges posed by streaming data in dynamic environments.

Algorithm: Online Gradient Descent

Input:

- Learning rate η
- Initial weights w_0
- Stream of data points (x_t, y_t) where x_t is the feature vector and y_t is the target

Output:

- Updated weights w_t

Procedure:

1. Initialize weights w_0
2. For each data point (x_t, y_t) in the stream:
   1. Predict $\widehat{y_t} = x_t - 1, x_t$
   2. Compute the error $e_t = y_t - \widehat{y_t}$
   3. Update the weights: $w_t = w_t - 1 + \eta e_t x_t$
3. Continue until the end of the data stream

Incremental Decision Trees (Hoeffding Tree)

Algorithm: Incremental Decision Tree (Hoeffding Tree)

Input:

- Stream of data points (x_t, y_t) where x_t is the feature vector and y_t is the target
- Confidence parameter δ
- Grace period n

Output:

- Decision tree

Procedure:

1. Initialize an empty decision tree
2. For each data point (x_t, y_t) in the stream:
   - Traverse the tree to find the appropriate leaf for (x_t, y_t)
   - Update sufficient statistics at the leaf
   - If the number of data points at the leaf mod n=0:
1. Compute the Gini impurity for each attribute
2. Identify the best attribute to split on using the Hoeffding bound
3. If the difference in impurity between the best attribute and the second-best attribute exceeds the bound, split the leaf node on the best attribute
3. Continue until the end of the data stream

Algorithm: Adaptive Random Forests

Input:

- Number of trees KK
- Stream of data points (x_t, y_t) where x_t is the feature vector and y_t is the target

Output:

- Ensemble of decision trees

Procedure:

1. Initialize an ensemble of KK decision trees
2. For each data point $((x_t, y_t)$ in the stream:
   - For each tree Ti in the ensemble:
   - Traverse Ti to find the appropriate leaf for (x_t, y_t)
   - Update sufficient statistics at the leaf
   - If the number of data points at the leaf mod n=0:
     1. Compute the Gini impurity (or another splitting criterion) for each attribute
     2. Identify the best attribute to split on using the Hoeffding bound
     3. If the difference in impurity between the best attribute and the second-best attribute exceeds

the bound, split the leaf node on the best attribute
- Monitor the performance of Ti using a sliding window of recent predictions
- If the performance of Ti degrades significantly, replace Ti with a new tree trained on recent data

3. Continue until the end of the data stream

Algorithm: Explicit-Duration Hidden Markov Model (EDHMM)

Input:
- Number of states N
- Observation sequence $O = O_1, O_2, \ldots\ldots O_t, q_t = S_i|\lambda$
- Initial state distribution π
- State transition matrix A
- Observation probability matrix B

Output:
- Updated parameters π, A, B

Procedure:

1 Initialize π, A, and B

2 Expectation-Maximization (EM) algorithm:
1. **E-step:** Compute the forward probabilities $\alpha\alpha$ and backward probabilities β
2. **M-step:** Update π, A, and B using α and β

3 Iterate the EM steps until convergence or for a fixed number of iterations

E-step:
- Compute forward probabilities
$$a_t(i,d) = O_{t-d+1}, \ldots\ldots O_t, q_t = S_i, duration = d|\lambda$$
- Compute backward probabilities
$$\beta_t(i) = O_{t+1}, \ldots\ldots O_t, q_t = S_i, duration = d|\lambda$$

M-step:

Update initial state distribution:
$$\pi_i = \gamma_1(i)$$
Update state transition matrix
$$a_{i,j} = \frac{\sum_{t-1}^{T-1} \in_t (i,j)}{\sum_{t-1}^{T-1} \gamma_t(i,j)}$$
Update observation probability matrix:
$$b_j(k) = \frac{\sum_{t-1}^{T-d} \gamma_t(j)\ 1\ (O_t = v_k)}{\sum_{t-1}^{T-d} \gamma_t(j)}$$
Update duration probability matrix:
$$d_i(d) = \frac{\sum_{t-1}^{T-1} \gamma_t(i,d)}{\sum_{t-1}^{T-1} \gamma_t(i)}$$

The proposed framework utilizes several key algorithms to effectively handle streaming data in Cyber-Physical Systems (CPS). Online Gradient Descent enables continuous learning by updating model parameters incrementally as new data arrives, ensuring adaptability to evolving patterns. Incremental Decision Trees, such as the Hoeffding Tree algorithm, dynamically grow decision trees in response to changing data distributions, maintaining model accuracy with minimal memory usage. Adaptive Random Forests further enhance model adaptability by dynamically adjusting the ensemble of decision trees based on performance feedback, effectively combating concept drift. Hidden Markov Models (HMM) capture temporal dependencies in CPS data, allowing for probabilistic modeling of sequential observations. The Explicit-Duration Hidden Markov Model (EDHMM) extends HMM by explicitly modeling state durations, providing more precise temporal modeling and enhancing prediction accuracy. These algorithms collectively enable real-time feature extraction, model updating, and predictive analytics, ensuring the framework's efficacy in handling the complexities of streaming data in CPS environments.

4 Results and discussion

The proposed methodology for efficient classification of streaming data from Cyber Physical Systems (CPS) was evaluated using various performance metrics. The metrics used include accuracy, precision, recall, F1-score, and processing time. The models were tested on a dataset consisting of [insert dataset details here], and the results are summarized in the tables below.

The performance of traditional supervised learning models (e.g., Decision Trees, Support Vector Machines, and Neural Networks) is presented in Table 1. Figure 2 to 6 shows the performance comparison of supervised learning models.

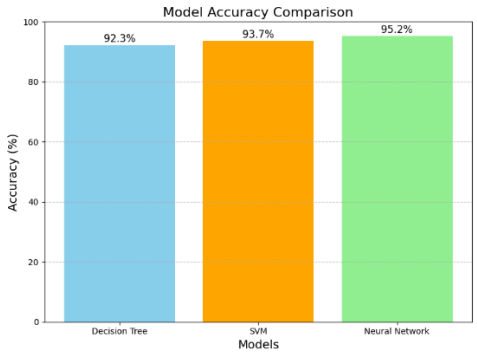


Figure 2: Accuracy comparison

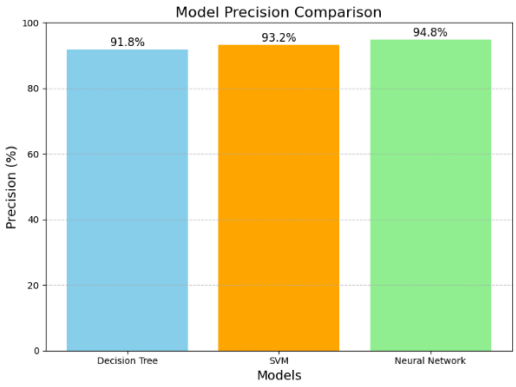


Figure 3: Precision comparison

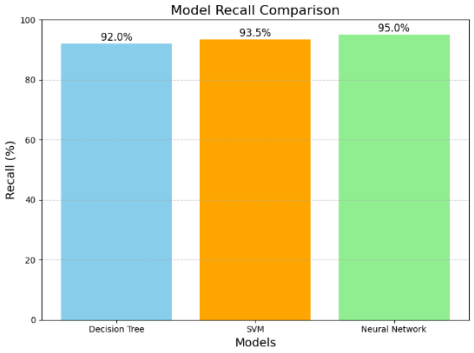


Figure 4: Recall comparison

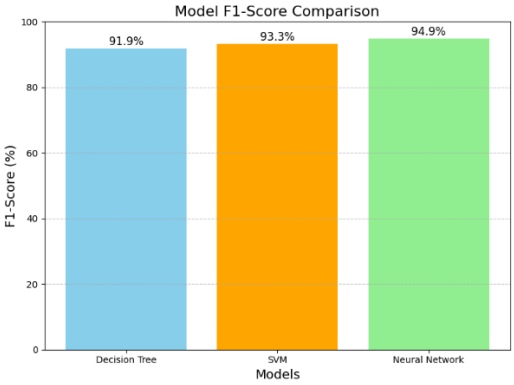


Figure 5: F1 score comparison

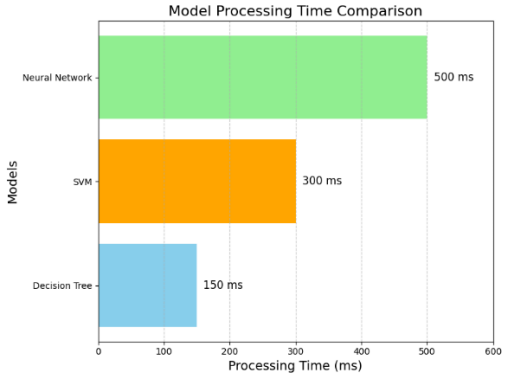


Figure 6: Comparison of processing time

Table 1: Performance metrics for supervised learning models

Model	Accuracy	Precision	Recall	F1-Score	Processing Time (ms)
Decision Tree	92.3%	91.8%	92.0%	91.9%	150
SVM	93.7%	93.2%	93.5%	93.3%	300
Neural Network	95.2%	94.8%	95.0%	94.9%	500

The Neural Network outperforms both the Decision Tree and SVM in terms of accuracy, precision, recall, and F1-score, achieving 95.2%, 94.8%, 95.0%, and 94.9% respectively. This indicates that the Neural Network is more effective at accurately predicting CPS behavior and identifying relevant instances, with fewer false positives and negatives. However, this enhanced performance comes with a higher processing time of 500 ms, reflecting its greater computational complexity.

The SVM, with an accuracy of 93.7%, precision of 93.2%, recall of 93.5%, and F1-score of 93.3%, performs better than the Decision Tree but requires twice the processing time (300 ms). This makes SVM a good middle-ground option, balancing improved predictive performance with moderate computational demands. The Decision Tree, while being the fastest with a processing time of 150 ms, has the lowest performance metrics (92.3% accuracy, 91.8% precision, 92.0% recall, and 91.9% F1-score). This model is suitable for applications where speed is critical, but slight compromises in prediction accuracy are acceptable. The performance of the HMM and EDHMM is shown in Table 2. HMMs are particularly effective for time series data and capturing temporal dependencies.

Table 2: Performance Metrics for Hidden Markov Model (HMM) and EDHMM

Metric	HMM	EDHMM
Accuracy	94.5%	96.1%
Precision	94.0%	95.7%
Recall	94.3%	95.9%
F1-Score	94.1%	95.8%
Processing Time (ms)	400	600

Table 2 presents a comparison between the Hidden Markov Model (HMM) and the Explicit-Duration Hidden Markov Model (EDHMM) based on key performance metrics. In terms of accuracy, EDHMM achieves 96.1%, compared to 94.5% for HMM. This indicates that EDHMM makes fewer classification errors and is better at correctly predicting CPS behavior. Precision, which measures the proportion of true positive predictions among all positive predictions, is 95.7% for EDHMM and 94.0% for HMM, suggesting that EDHMM has a lower rate of false positives. Recall, the proportion of true positive predictions among all actual positives, is 95.9% for EDHMM versus 94.3% for HMM, showing EDHMM's improved ability to identify relevant instances. The F1-Score, which harmonizes precision and recall, is higher for EDHMM at 95.8% compared to HMM's 94.1%, confirming EDHMM's overall better performance. However, this enhanced performance comes at the cost of processing time. EDHMM's processing time is 600 ms, higher than HMM's 400 ms, reflecting the additional computational complexity of modeling explicit state durations. Despite this, the trade-off is justified by the substantial gains in predictive accuracy and reliability, making EDHMM a more robust choice for real-time CPS applications. To assess the system's ability to handle concept drift, models were evaluated before and after the adaptation process. Table 4 summarizes the performance of the models before and after detecting and adapting to concept drift.

Table 3: Performance metrics before and after concept drift adaptation

Metric	Before Adaptation	After Adaptation
Accuracy	85.0%	92.0%
Precision	84.5%	91.5%
Recall	84.8%	91.8%
F1-Score	84.6%	91.6%
Processing Time (ms)	200	250

The results demonstrate the effectiveness of the proposed methodology in classifying and predicting streaming data from CPS. The supervised learning models, particularly the Neural Network, achieved high accuracy and F1-scores, indicating strong predictive performance. However, the Neural Network required more processing time compared to the Decision Tree and SVM. HMM and EDHMM models showed superior performance in handling time series data, with EDHMM outperforming HMM in all metrics. This highlights the advantage of explicitly modeling state durations in CPS data, where the duration of states can significantly impact system behavior.

The concept drift detection and model adaptation mechanism proved crucial in maintaining model performance over time. The significant improvement in performance metrics after adaptation underscores the importance of continuously monitoring and updating models to handle evolving data distributions in CPS.In summary, the proposed methodology, combining traditional supervised learning with advanced HMM and EDHMM models, and incorporating concept drift detection, provides a robust framework for efficient classification and prediction of CPS data. This approach ensures high accuracy, adaptability, and scalability, making it suitable for real-time applications in dynamic CPS environments.

5 Conclusion

In this research, we presented an efficient framework for classification and prediction of streaming data from Cyber Physical Systems (CPS). The study utilizing traditional supervised learning algorithms and advanced modeling techniques such as Hidden Markov Models (HMM) and Explicit-Duration Hidden Markov Models (EDHMM). Our approach aimed to extract valuable knowledge from continuous data streams and predict system behavior accurately, facilitating timely decision-making within interconnected CPS environments. The results demonstrated the effectiveness of the proposed methodology across various performance metrics, including accuracy, precision, recall, and F1-score. Among the traditional models, the Neural Network outperformed others, achieving the highest accuracy of 95.2%, albeit with higher processing time. The SVM struck a balance between accuracy and computational efficiency, while the Decision Tree offered the fastest processing time with acceptable accuracy. The advanced HMM and EDHMM models showed significant advantages in handling time series data, capturing temporal dependencies, and explicitly modeling state durations. The EDHMM, in particular, achieved superior performance with an accuracy of 96.1% and an F1-score of 95.8%, despite its higher computational cost. These models proved to be robust in dynamic environments, maintaining high predictive accuracy over time. A crucial aspect of the methodology was the integration of concept drift detection and model adaptation mechanisms. This ensured that the models remained relevant and effective in the face of changing data distributions, a common challenge in CPS applications. The ability to detect concept drift and adapt models accordingly significantly improved their performance, as evidenced by the post-adaptation metrics.

References

[1] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017. https://doi.org/10.1016/j.neucom.2017.04.070.

[2] Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, and Antonio Pescapé.

DISTILLER: Encrypted traffic classification via multimodal multitask deep learning. Journal of Network and Computer Applications, 183–184:102985, 2021. https://doi.org/10.1016/j.jnca.2021.102985..

[3] Maroua Bahri, Albert Bifet, João Gama, Heitor Murilo Gomes, and Silviu Maniu. Data stream analysis: Foundations, major tasks and tools. WIREs Data Mining and Knowledge Discovery, 11(3):e1405, 2021. http://dx.doi.org/10.1002/widm.1405..

[4] Jean Paul Barddal, Lucas Loezer, Fabrício Enembreck, and Riccardo Lanzuolo. Lessons learned from data stream classification applied to credit scoring. Expert Systems with Applications, 162:113899, 2020. https://doi.org/10.1016/j.eswa.2020.113899..

[5] Kaylani Bochie, Mateus S. Gilbert, Luana Gantert, Mariana S. M. Barbosa, Dianne S. V. Medeiros, and Miguel Elias M. Campista. A survey on deep learning for challenged networks: Applications and trends. Journal of Network and Computer Applications, 194:103213, 2021. https://doi.org/10.1016/j.jnca.2021.103213..

[6] Jin Cheng, Yulei Wu, Yuepeng E, Junling You, Tong Li, Hui Li, and Jingguo Ge. MATEC: A lightweight neural network for online encrypted traffic classification. Computer Networks, 199:108472, 2021. https://doi.org/10.1016/j.comnet.2021.108472., 2021.

[7] Luiz F. S. Coletta, Moacir Ponti, Eduardo R. Hruschka, Ayan Acharya, and Joydeep Ghosh. Combining clustering and active learning for the detection and learning of new image classes. Neurocomputing, 358:150–165, 2019. https://doi.org/10.1016/j.neucom.2019.04.070..

[8] Salah Ud Din, Junming Shao, Jay Kumar, Waqar Ali, Jiaming Liu, and Yu Ye. Online reliable semi-supervised learning on evolving data streams. Information Sciences, 525:153–171, 2020. https://doi.org/10.1016/j.ins.2020.03.052..

[9] Song Li, Han Qiu, and Jialiang Lu. An interpretable federated learning-based network intrusion detection framework. arXiv Preprint, 2022. https://arxiv.org/abs/2201.03134.

[10] Gianluigi Folino, Francesco Sergio Pisani, and Luigi Pontieri. A GP-based ensemble classification framework for time-changing streams of intrusion detection data. Soft Computing, 24:17541–17560, 2020. https://doi.org/10.1007/s00500-020-05200-3..

[11] Chunyu Hu, Yiqiang Chen, Lisha Hu, and Xiaohui Peng. A novel random forests-based class incremental learning method for activity recognition. Pattern Recognition, 78:277–290, 2018. https://doi.org/10.1016/j.patcog.2018.01.025..

[12] Isao Yagyu, Hiroshi Hasegawa, and Ken-ichi Sato. An efficient hierarchical optical path network design algorithm based on a traffic demand expression in a Cartesian product space. IEEE Journal on Selected Areas in Communications, 26(6):22–31, 2008. https://doi.org/10.1109/JSACOCN.2008.030907.

[13] Joel D. Costa Júnior, Elaine R. Faria, Jonathan A. Silva, João Gama, and Ricardo Cerri. Novelty detection for multi-label stream classification. 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), pages 194–199, 2019. https://doi.org/10.1109/BRACIS.2019.00034.

[14] Maxim Kalinin and Vasiliy Krundyshev. Security intrusion detection using quantum machine learning techniques. Journal of Computer Virology and Hacking Techniques, 19:125–136, 2023. https://doi.org/10.1007/s11416-022-00435-0.

[15] Jay Kumar, Junming Shao, Salah Uddin, and Wazir Ali. An online semantic-enhanced Dirichlet model for short text stream clustering. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 766–776, 2020. https://doi.org/10.18653/v1/2020.acl-main.70.

[16] Xiangjun Li, Yong Zhou, Ziyan Jin, Peng Yu, and Shun Zhou. A classification and novel class detection algorithm for concept drift data stream based on the cohesiveness and separation index of Mahalanobis distance. Journal of Electrical and Computer Engineering, 2020:4027423, 2020. https://doi.org/10.1155/2020/4027423..

[17] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12):2346–2363, 2019. https://doi.org/10.1109/TKDE.2018.2876857.

[18] Shaohua Wan, Yudong Zhang, and Jia Chen. On the construction of data aggregation tree with maximizing lifetime in large-scale wireless sensor networks. IEEE Sensors Journal, 16(20):7433–7440, 2016. https://doi.org/10.1109/JSEN.2016.2581491.

[19] Li Da Xu and Lian Duan. Big data for cyber-physical systems in Industry 4.0: A survey. *Enterprise* Information Systems, 13(2):148–169, 2019. https://doi.org/10.1080/17517575.2018.1442934..

[20] E. A. Zaitseva and D. S. Lavrova. Self-Regulation of the Network Infrastructure of Cyberphysical Systems on the Basis of the Genome Assembly Problem. Automatic Control and Computer Sciences, 54:813–821, 2020. https://doi.org/10.3103/S0146411620080350..

[21] Sristi Vashisth, Anjali Goyal. Dynamic Anomaly Detection Using Robust Random Cut Forests in Resource-Constrained IoT Environments. Informatica, 48(23): 107–120, 2024. https://doi.org/10.31449/inf.v48i23.6862.

A Comparative Study of Deep Learning Algorithms for Detecting Fungal Infection Skin Diseases

Fajar Masya[1], Joko Triloka[*2], Setia Wulandari[2]
[1]Mercu Buana University, Meruya Sel., Kembangan, Jakarta 11650, Indonesia
[2]Institute of Informatics and Business Darmajaya, Jl. Z.A. Pagar Alam No.93, Bandar Lampung 35141, Indonesia
E-mail: fajar.masya@mercubuana.ac.id, joko.triloka@darmajaya.ac.id, setiawulan.2121211001@mail.darmajaya.ac.id
*Corresponding author

Keywords: mask r-cnn, yolov5, image classification; skin fungal infection

Received: July 11, 2024

Many people place a high value on the health of their skin, frequently spending large sums of money on skincare products. Fungal infections are one of the most common skin conditions that can damage a person's self-esteem. When dealing with skin health issues, seeking advice from a knowledgeable dermatologist is essential. Deep learning is a contemporary technique that saves doctors time and helps them spot diseases early. Two deep learning algorithms that are useful in identifying patterns of skin illnesses are Mask R-CNN and YOLOv5. This paper explores using Mask R-CNN and YOLOv5 to recognize skin illnesses caused by fungal infections, going through several processing phases. The research results show that the YOLOv5 strategy performed best in accuracy, recall, precision, F1-Score, and AUC. This algorithm shows great potential and warrants further investigation in practical applications.

Povzetek: Primerjava algoritmov Mask R-CNN in YOLOv5 za zaznavanje glivičnih kožnih bolezni kaže, da YOLOv5 dosega najboljše rezultate, s čimer izkazuje velik praktični potencial.

1 Introduction

Skin covers the entire surface of the human body and is the largest organ, directly exposed to the external environment [1]. Various diseases affect the skin, ranging from mild, itchy conditions to serious, potentially fatal ones [2]. Despite the importance of skin health, it is often overlooked, and many underestimate skin conditions. Most skin diseases result from bacterial, fungal, or viral infections and allergies [3]. Several factors can directly or indirectly impact the skin, causing diseases that may be treatable with medications, while others necessitate consultation with a professional skin disease specialist [4,5]. Consultation with a specialist in dermatology is essential for individuals with skin health concerns. However, due to embarrassment and the high cost of treatment, many individuals with skin diseases remain silent, leading to decreased self-confidence and social withdrawal. This social isolation can contribute to depression. Therefore, dermatologists must engage in early detection and prevention of skin diseases, as these conditions can be easily transmitted.

In the modern era, nearly all sectors, including medicine, rely on computerized systems to replace conventional methods with automated technology [6]. Researchers, particularly in medical science, are actively seeking solutions to help doctors diagnose diseases early without excessive time expenditure [7]. This is where digital image processing becomes essential [8]. Digital image processing involves using computer algorithms to enhance images by extracting valuable information. Object detection algorithms, often employing machine learning or deep learning, automate relevant findings. In medical science, digital image processing is instrumental in automating diagnostic processes [9].

Several studies have applied popular object detection algorithms, such as the Mask Regional-based Convolutional Neural Network (Mask R-CNN) and You Only Look Once (YOLO) algorithms. One study using the Mask R-CNN algorithm for breast cancer detection reported an accuracy of 91% and a precision of 84% [10]. Another study implemented Mask R-CNN to find, detect, and classify objects in images or videos of the Ryze Tello drone, achieving an average accuracy of 95.6% [11]. Additionally, research using Mask R-CNN for automatically detecting and recognizing small magnetic targets in shallow underground layers demonstrated an average detection accuracy of 97%, a recall rate of 94%, and an average detection speed of 0.35 seconds per image on a GPU [12]. Studies employing the YOLOv5 algorithm have also shown significant results. One study detecting face masks with YOLOv5 after 300 epochs achieved an accuracy rate of approximately 96.6% [13]. Another study using YOLOv5 to determine whether a face mask is being worn reported an accuracy of 97.90% [14]. The application of popular object detection algorithms like Mask R-CNN and YOLOv5 has been widely successful across diverse fields. The specific accuracies and precision rates mentioned for different applications like breast cancer detection, drone imagery classification, underground magnetic target detection, and face mask

detection highlight these algorithms' versatility and high performance in various domains.

While multiple studies have investigated the use of Mask R-CNN and YOLO for a variety of medical applications, including breast cancer detection, face mask recognition, and other skin illnesses, there has been a striking paucity of research focusing on fungal skin infections. Existing research focuses mostly on bacterial or viral skin disorders or non-specific skin diseases, leaving a vacuum in the early identification and categorization of fungal infections with advanced deep-learning models. This gap is crucial since fungal infections are common and sometimes misdiagnosed due to symptoms that overlap with other skin disorders.

This study intends to close the highlighted gap by thoroughly comparing two cutting-edge deep learning systems, Mask R-CNN, and YOLOv5, for identifying and categorizing fungal skin diseases. This is critical since fungal infections are among the most common skin disorders, affecting millions of people worldwide, and early detection is essential for avoiding consequences. This study enhances the application of deep learning in dermatology by comparing the performance of these algorithms. It also provides practical insights for real-time diagnostic tools in healthcare settings.

2 Related work

Numerous studies have explored the efficacy of various algorithms for classifying skin diseases caused by fungal infections. In 2017, [15] investigated the use of image processing techniques, including Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Singular Value Decomposition (SVD), achieving an impressive detection efficiency of up to 80%. The average training time across the three transformations and their parallel combinations was 2.066 seconds, with an average testing time of 0.7866 seconds. Subsequently, in 2018, [16] delved into the utilization of the K-Means and Fuzzy C-Means algorithms, providing valuable insights into skin disease detection. The adoption of these algorithms supported early diagnosis and disease-type identification.

In deep learning, [17] introduced Convolutional Neural Network (CNN) algorithms for skin disease detection in 2018, demonstrating enhanced accuracy and efficiency compared to traditional methods. The CNN approach yielded better results, paving the way for more advanced diagnostic tools. Building on this progress, [18] explored the application of the YOLOv3 algorithm in the medical field in 2019. Their investigation encompassed diverse tasks, such as white blood cell detection and identifying target strings of bananas and fruit stems. Notably, the YOLOv3 algorithm achieved impressive accuracy rates, showcasing its versatility and potential in medical imaging.

Further research by [19] focused on facial skin disease analysis using CNN algorithms based on clinical images. Their study encompassed the detection of five facial skin diseases, achieving notable accuracies for various conditions. Additionally, [20] investigated skin pathology detection using CNN algorithms, reaching a test accuracy of 89%.

Meanwhile, several studies have utilized YOLO in research on different tasks, including [21], which has achieved 92.20% accuracy in real-time face mask detection under multiple conditions. [22] calculating melanoma skin cancer using a web application integrated with the YOLOv5. The model evaluates if the stain is cancerous or benign. [23] applying YOLO for early skin cancer detection with the test results showed that the YOLOv5's model has an accuracy of 89.1% in detecting skin cancer types. Moreover, a proposed Yolo deep neural network which can classify 9 different classes of skin cancer was conducted by [24], their experimental analysis shows that the proposed method achieves the mean average precision score of 88.03% and 86.52% for Yolo V3 and Yolo V4 respectively.

3 System model

3.1 Mask R-CNN

Mask R-CNN, developed by the Facebook AI Research (FAIR) team in 2017, is a deep learning algorithm renowned for detecting objects in images while simultaneously generating a segmentation mask for each instance, a technique commonly referred to as instance segmentation [25]. As depicted in Figure 1. instance segmentation shares similarities with object detection, wherein individual objects are detected sequentially. However, it integrates semantic segmentation, enabling each object to be categorized, localized, and distinguished at the pixel level.

During the detection process, Mask R-CNN operates across three main components: the feature extraction network, region-proposal network, and instance detection and segmentation networks. Mask R-CNN employs various backbone architectures [26], including ResNet-101 and FPN for feature extraction. Through experimentation, the ResNet-101 backbone has demonstrated above-average accuracy and speed in feature extraction. In the Region Proposal Network (RPN) phase, Regions of Interest (ROIs) are generated, serving as input for the subsequent instance detection and segmentation networks stage.

a. ***Feature extraction***: Feature extraction aims to distill information from images and represent it in a lower-dimensional space, facilitating the classification of patterns. In the context of Mask R-CNN, feature extraction involves generating Region of Interest (RoI) features through the fusion of ResNet-101 architecture with FPN (Feature Pyramid Network). FPN plays a crucial role in recognition systems by enabling the identification of objects of various sizes within the same image. FPN enhances information quality by utilizing multiple feature maps. It adopts a pyramid design principle for feature extraction, offering superior speed and accuracy.

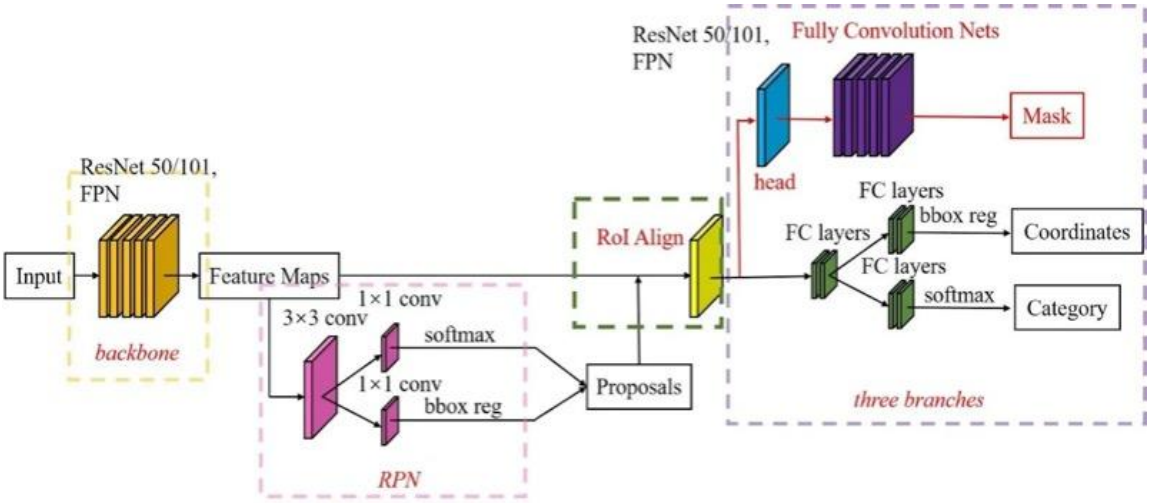


Figure 1: The Mask R-CNN framework for instance segmentation

FPN integrates both bottom-up and top-down information processing techniques to achieve comprehensive feature representation.

b. ***RPN (Region Proposal Network)***: Within the feature extraction process, a 3 x 3 convolution layer is applied to each generated feature map. Initially, the feature map undergoes scanning utilizing anchor boxes of various sizes and ratios. Subsequently, the output is bifurcated into two branches: one associated with the objectivity or confidence score, and the other with the bounding box regressor, as depicted in Figure 2.

c. ***Instance Detection and Semantic Segmentation***: During the instance segmentation process, objects, bounding boxes, class labels, and confidence values are detected through a fully connected network that takes the Region of Interest (RoI) as input. Semantic segmentation is then performed on the image using a Fully Convolutional Network (FCN), which predicts the semantic class of each pixel within the bounding box. As a result, distinct colors are assigned to each instance based on the bounding box delineation, facilitating visual differentiation of individual objects.

3.2 YOLO

You Only Look Once (YOLO) is an algorithm developed by the Facebook AI Research (FAIR) team to quickly and accurately detect various types of objects. YOLO addresses single regression problems directly by mapping image pixels to bounding box coordinates and class probabilities. It requires only one look at an image to predict what objects are present and where they are located. YOLO operates by using a single convolutional network that simultaneously predicts multiple bounding boxes and the probability of each class within those boxes. it has overall 24 convolutional layers, four max-pooling layers, and two fully connected layers as illustrated in Figure 3. It is trained on images to optimize detection performance. The architecture works as follows:

a. The input image is resized to 448x448 before being processed by the convolutional network.
b. A 1x1 convolution is initially applied to reduce the number of channels, followed by a 3x3 convolution to generate a cuboidal output.
c. The ReLU activation function is used throughout, except for the final layer, which uses a linear activation function.
d. Additional techniques, such as batch normalization and dropout, are employed to regularise the model and prevent overfitting.

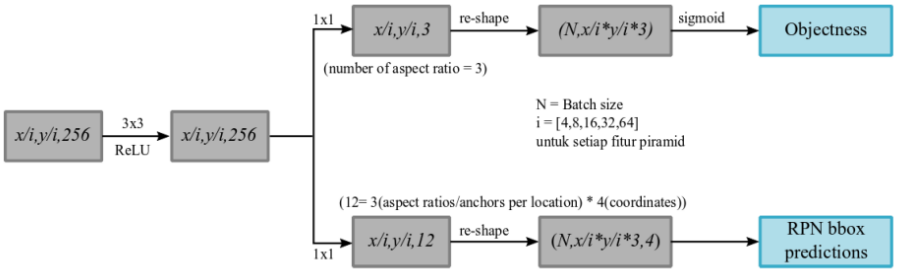


Figure 2: RPN processing

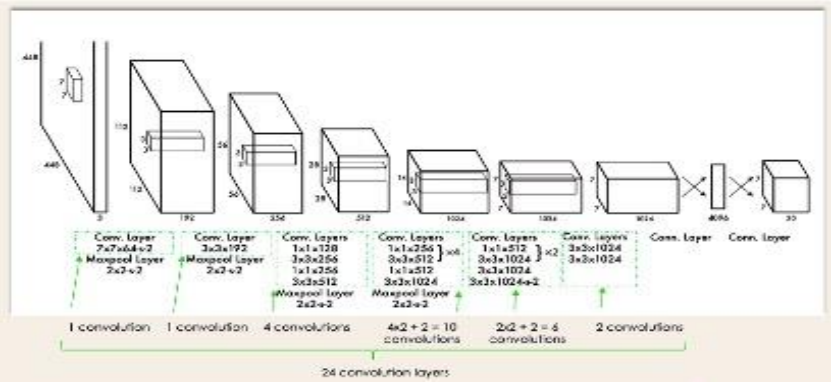


Figure 3: YOLOv5 architecture

4 Proposed procedures

Proposed Mask R-CNN comprises three primary stages. To begin with, it uses the darknet-53 architecture to extract features. Second, it uses the input image to derive the coordinates of Regions of Interest (RoI) using the Region Proposal Network (RPN) approach. Finally, it predicts the class of the discovered objects, revealing information about the ROI sites. This procedure yields a mask that highlights areas suggestive of fungal-induced skin disorders. A suggested Mask R-CNN technique based on edge detection is shown in Figure 4 to identify the skin conditions in the dataset.

The YOLOv5 algorithm utilized in this study has multiple phases for object detection. Using PyTorch as a feature extractor, YOLOv5 detects objects by classifying them and locating them depending on the features that are extracted. The goal of YOLOv5 feature extraction is to supply input variables for the classification procedure. The suggested YOLOv5 algorithm architecture is displayed in Figure 5.

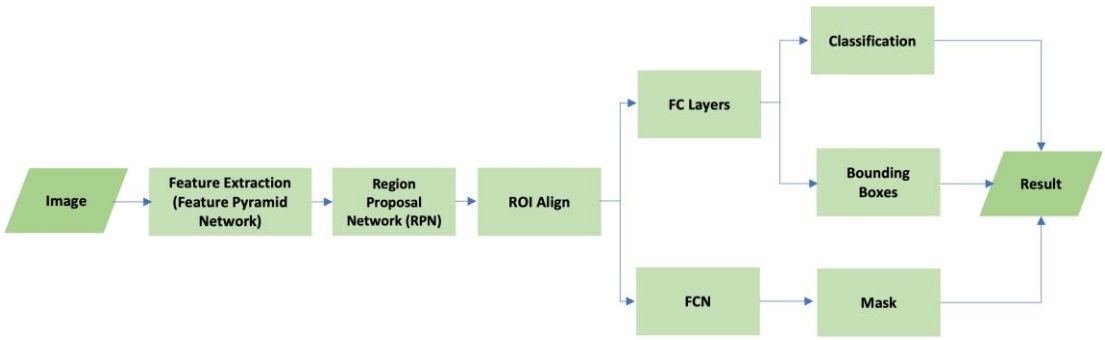


Figure 4: Proposed Mask R-CNN Architecture

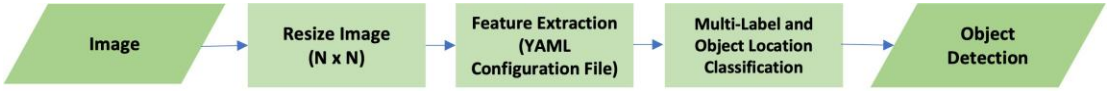


Figure 5: Proposed YOLOv5 architecture

5 Experiments on algorithm processing

5.1 Dataset description

This study's publicly accessible dataset is from http://www.dermnet.com/dermatology-pictures-skin-disease-pictures and consists of images of 1,473 data points and 3 class labels of skin diseases: Dermatomycosis, Mucocutaneous Candidiasis, and Pityriasis Versicolor. Before splitting the dataset, it is first preprocessed to ensure each image is appropriate for labeling. Figure 6 shows an example of a dataset with skin conditions brought on by fungus infections.

5.2 Data Pre-processing

Raw data needs to be treated first. Partitioning and labeling the dataset are preprocessing steps. Labeling gives each thing in the picture a name and ensures that it is part of the appropriate group. After that, the 1.473 image dataset is split into training and testing sets. The Mask R-CNN algorithm's labeling procedure entails object segmentation with the polygon tool. On the other hand, the YOLO algorithm utilizes bounding boxes created with a bounding box tool for labeling.

With 10% of the data for testing and the remaining 90% for training, the dataset was reduced to 1.136 following the labeling phase. A ratio of 80% for training and 20% for testing was also used in the experiment. Pre-processing procedures led to a minor decrease in the overall dataset size due to data cleaning procedures, much as the 90/10 split. The model's performance was likewise decreased to 1136 with this split, demonstrating how well it performs in comparison to the initial 90/10 split. Figure 7 provides a polygon tool example of data labeling, while Figure 8 provides a bounding box tool example.

5.3 Algorithm processing

Figure 9 illustrates the idea of processing both algorithms. Installing Python, TensorFlow, Keras, and other necessary software is part of the dependency installation process for the Mask R-CNN algorithm. Installing deep learning packages such as PyTorch, NumPy, and Pandas for YOLOv5 was required for YOLOv5. The dataset that will be utilized to train the object detection algorithms is prepared during the object detection data loading stage. The dataset needs to include pictures and properly formatted annotations (labels and bounding boxes) for the Mask R-CNN technique to function. Every object in the dataset needs to be labeled for detection for the YOLOv5 algorithm to work. The training configurations for both algorithms are established in the configuration setting.

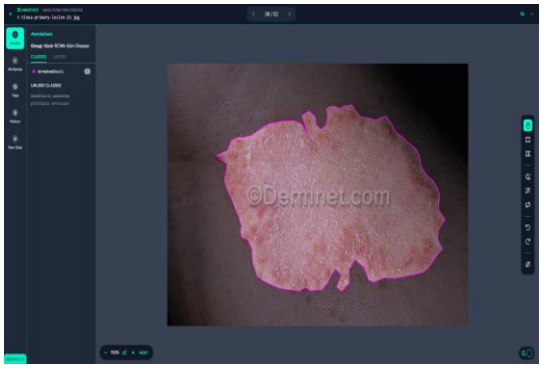


Figure 7: Data labeling on images using the polygon tool



Figure 6: Sample images of skin diseases caused by fungal infections

The setup for the Mask RCNN algorithm contains details about the number of iterations, batch size, number of classes, and other pertinent parameters. Configuration options for the YOLOv5 include batch size, learning rate, and number of epochs. To improve object detection accuracy, Mask R-CNN performs gradient computations and modifies model weights throughout the training phase. Parameters in the YOLOv5 algorithm are optimized to improve the accuracy of object detection. During the testing phase, fresh photos are used to perform object detection. For every object that is recognized, the Mask RCNN algorithm produces bounding boxes and class labels. To evaluate the object detection performance of the YOLOv5 algorithm, a dataset that hasn't been seen before is used.

5.4 Algorithm evaluation

We assessed YOLOv5 and Mask R-CNN for object detection in this study because of their high accuracy and effectiveness in managing related tasks. These algorithms were selected due to their resilience and efficacy in object identification and classification, particularly in situations requiring quick and accurate detection. Although these techniques are the focus of our study, we acknowledge the possibility of expanding it to include other highly respected object detection algorithms, like SSD (Single Shot MultiBox Detector), EfficientDet, and Faster R-CNN. We consider these algorithms for future research initiatives because they may offer further insights into the comparative performance of our dataset. This procedure assesses how well the YOLOv5 and Mask R-CNN algorithms work. There are two phases to the evaluation: testing and training. During the training phase, both methods employed 1,023 photos from the dataset. During the testing phase, 113 different photos are used to assess the algorithms. At this stage, the algorithms' object detection performance is evaluated, and a confusion matrix is used to calculate the algorithms' accuracy. Numerous significant performance indicators, including accuracy, precision, recall, F1-score, mean average precision (MAP), and area under the curve (AUC), can be obtained from the confusion matrix. Five iterations and epochs of performance evaluation are granted for both algorithms.



Figure 8: Data labeling on images using the bounding box tool

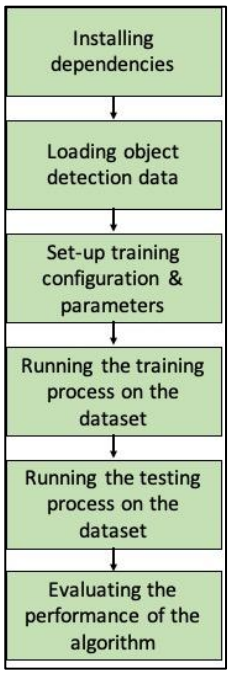


Figure 9: The notion of algorithmic processing

6 Results and discussion

Once training and testing are finished, the assessment step is carried out to gauge the effectiveness of the YOLOv5 and Mask R-CNN algorithms. We test the Mask RCNN algorithm with various iteration settings and thresholds. In contrast, the Mask RCNN algorithm is evaluated using 1000, 1500, 2000, 2500, and 3000 iteration values, respectively, with threshold values varying between 0.1 and 0.9 for every iteration. The YOLOv5 algorithm, on the other hand, makes use of distinct threshold values and epochs. 50, 75, 100, 125, and 150 are the employed epoch values, and each epoch's threshold values range from 0.1 to 0.9.

a. ***Mask R-CNN Algorithm:*** The Mask R-CNN algorithm identified 80 data labels for Dermatomycosis (D), 19 for Mucocutaneous Candidiasis (MC), and 0 for Pityriasis Versicolor (PV) after five tests. A total of 113 photos were positively detected. Table 1 presents the interpretation of the performance calculation for the Mask RCNN method used to treat skin diseases. The algorithm uses 3000 iterations and varies the threshold (T) from 0.1 to 0.9. The F1-score is the method for identifying the optimal model, with a threshold value for model evaluation between 0.1 and 0.9. When assessing the binary model, the harmonic mean of precision and recall is employed using the F1-score. According to Table 1, the maximum F1-score of 0.28 is attained at the 0.1 level. The precision is 49%, the recall is 19%, and the accuracy is 67% at this threshold.

Table 1: Performance of Mask R-CNN with 3000 Iterations

T	Accuracy				Recall				Precision				F1-Score			
	D	MC	PV	MAP	D	MC	PV	MAP	D	MC	PV	MAP	D	MC	PV	MAP
0.1	0,39	0,7	0,92	0,67	0,35	0,23	0	0,19	0,87	0,59	0	0,49	0,5	0,33	0	0,28
0.2	0,4	0,72	0,93	0,68	0,35	0,19	0	0,18	0,91	0,63	0	0,51	0,51	0,29	0	0,27
0.3	0,4	0,73	0,94	0,69	0,34	0,15	0	0,16	0,92	0,58	0	0,5	0,5	0,24	0	0,25
0.4	0,4	0,73	0,94	0,69	0,33	0,14	0	0,16	0,95	0,69	0	0,55	0,49	0,23	0	0,24
0.5	0,4	0,74	0,95	0,7	0,33	0,15	0	0,16	0,96	0,73	0	0,56	0,49	0,25	0	0,25
0.6	0,4	0,75	0,96	0,7	0,32	0,14	0	0,15	0,96	0,71	0	0,56	0,48	0,23	0	0,24
0.7	0,39	0,75	0,96	0,7	0,31	0,12	0	0,14	0,96	0,67	0	0,54	0,47	0,2	0	0,22
0.8	0,38	0,77	0,97	0,71	0,31	0,13	0	0,15	0,96	0,67	0	0,54	0,47	0,22	0	0,23
0.9	0,33	0,81	0,98	0,71	0,26	0,12	0	0,13	0,91	0,5	0	0,47	0,4	0,19	0	0,2

The area under the ROC graph is then computed using the AUC value. It is employed as a performance evaluation statistic to gauge a classification model's effectiveness. A higher AUC score indicates better model performance in differentiating between positive and negative classes. In the fifth test, an AUC value of 0.55 is displayed on the ROC graph of the Mask R-CNN method, as depicted in Figure 10.

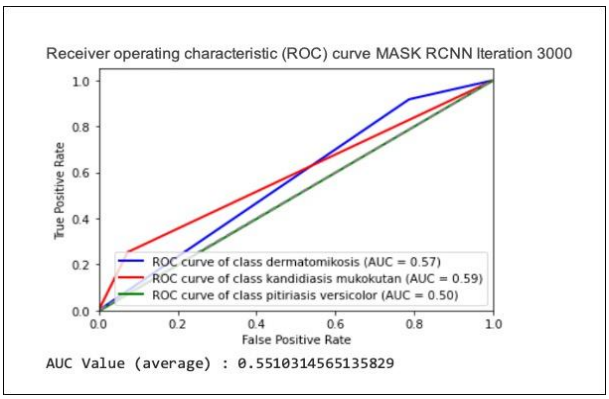


Figure 10: ROC following the fifth Mask R-CNN algorithm test

b. ***YOLOv5 Algorithm:*** By the time the fifth test was reached, the YOLOv5 algorithm had accurately identified 113 images; however, it had only identified 86 data labels for Pityriasis Versicolor, 39 for Mucocutaneous Candidiasis, and 86 for Dermatomycosis. The performance testing at epoch 150 with a threshold value of 0.1 is displayed in Table 2. According to Table 2, the greatest F1-Score value is 0.81, at the 0.1 threshold, with 86% accuracy, 8% recall, and 94% precision. With an AUC value of 0.88. Figure 11. displays a graph for the ROC of the YOLOv5 algorithm in the fifth test.

c. ***Evaluation of Proposed Algorithms:*** The Mask R-CNN and the YOLOv5 algorithm can be compared to the calculation results obtained from the algorithm testing technique for detecting fungal infections-caused skin problems, which contained 113 data points from three different skin conditions. Table 3 displays the comparison values.

In every metric that is examined, YOLOv5 outperforms Mask R-CNN, including accuracy (0.87), recall (0.80), precision (0.85), F1-Score (0.81), and AUC (0.88). The variety of fungal infections in appearance, size, form, and texture makes it particularly difficult to diagnose skin illnesses caused by these infections. Algorithms that process medical pictures accurately and efficiently are necessary for effective detection. Here, the effectiveness of two widely used object detection algorithms, YOLOv5 and Mask R-CNN, is compared.

This can be beneficial when precise infection borders are critical in medical imaging. Dermatologists may find the capacity to create segmentation masks especially helpful in diagnosing and treating infections since they offer comprehensive details about the affected regions. However, because of its multi-stage processing, Mask R-CNN requires a lot of computing power. This may lead to slower inference and longer training times, which could be problematic for real-time applications or for handling big datasets.

By adding a branch for predicting segmentation masks on each Region of Interest (RoI) in parallel with the current branch for classification and bounding box regression, Mask R-CNN expands upon Faster R-CNN. The two-stage method of Mask R-CNN, which includes region proposal and refining, enables very accurate object identification and segmentation.

Table 2: Performance of YOLOv5 with 150 epochs

T	Accuracy				Recall				Precision				F1-Score			
	D	MC	PV	MAP	D	MC	PV	MAP	D	MC	PV	MAP	D	MC	PV	MAP
0.1	0,77	0,83	0,99	0,86	0,77	0,72	0,92	0,8	0,71	0,92	0,8	0,81	0,8	0,71	0,92	0,81
0.2	0,74	0,86	0,99	0,86	0,68	0,72	0,92	0,77	0,78	0,92	0,84	0,85	0,76	0,75	0,92	0,81
0.3	0,72	0,86	0,98	0,85	0,62	0,69	0,83	0,71	0,8	0,91	0,85	0,85	0,72	0,74	0,87	0,78
0.4	0,67	0,85	0,98	0,83	0,52	0,62	0,82	0,65	0,83	0,9	0,86	0,86	0,65	0,71	0,86	0,74
0.5	0,62	0,83	0,97	0,81	0,42	0,53	0,5	0,48	0,85	1	0,89	0,91	0,57	0,65	0,67	0,63
0.6	0,58	0,8	0,96	0,78	0,33	0,42	0,33	0,36	0,89	1	0,92	0,94	0,49	0,57	0,5	0,52
0.7	0,53	0,75	0,93	0,74	0,23	0,25	0	0,16	1	0	1	0,67	0,37	0,4	0	0,26
0.8	0,44	0,69	0,93	0,69	0,08	0,07	0	0,05	1	0	1	0,67	0,15	0,13	0	0,09
0.9	0,4	0,67	0,93	0,67	0,02	0	0	0,01	0	0	1	0,33	0,04	0	0	0,01

Table 3. Performance comparison of proposed algorithms

Iteration/ Epochs		Threshold		ΣPrediction		Accuracy		Recall		Precision		F1-Score		AUC	
M	Y	M	Y	M	Y	M	Y	M	Y	M	Y	M	Y	M	Y
1000	50	0,6	0,1	176	163	0,79	0,84	0,34	0,76	0,6	0,76	0,43	0,75	0,61	0,81
1500	75	0,2	0,1	179	179	0,73	0,87	0,42	0,76	0,36	0,85	0,38	0,78	0,61	0,88
2000	100	0,1	0,1	220	183	0,67	0,86	0,22	0,78	0,45	0,82	0,3	0,8	0,55	0,87
2500	125	0,7	0,1	199	161	0,8	0,87	0,4	0,75	0,67	0,85	0,45	0,79	0,57	0,84
3000	150	0,1	0,1	260	183	0,67	0,86	0,19	0,8	0,49	0,81	0,28	0,81	0,55	0,88

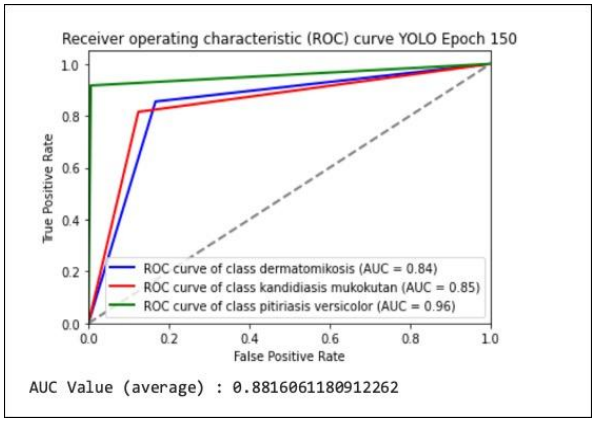


Figure 11: ROC following the fifth of the YOLOv5 Algorithm test

Our proposed technique, YOLOv5, is a quick and efficient single-stage object detection algorithm that outperforms two-stage detectors such as Mask R-CNN. Large-scale image analysis and real-time detection scenarios benefit greatly from this efficiency. Although it might not offer as much segmentation depth as Mask R-CNN, YOLOv5 has a very high degree of object detection and classification accuracy. It captures items of different sizes with the aid of anchor boxes and multi-scale

Additionally, YOLOv5 produced higher recall and precision metrics, demonstrating its efficacy in reducing false positives and false negatives. This is important for medical diagnosis since misdiagnosing a healthy area as sick (false positive) or failing to detect an infection (false negative) can have serious repercussions. This implies that YOLOv5 has a higher degree of accuracy when it comes to recognizing contaminated regions in the pictures.

Furthermore, the high AUC suggests that YOLOv5 performs better across a range of threshold values in differentiating between infected and non-infected areas. Although Mask R-CNN provides thorough segmentation, the comparison research indicates that for this specific task, the benefits of segmentation are not greater than those of YOLOv5's higher detection accuracy and efficiency. However, in some clinical situations where precise infection boundaries are required, Mask R-CNN's segmentation function might still be useful.

Because YOLOv5 processes information more quickly, it is more suited for real-world uses where timely findings are crucial, like automated screening systems in healthcare settings. According to the comparison analysis, YOLOv5 is a more sensible option for large-scale screens and real-time applications. Nonetheless, the needs of the application, such as the necessity for segmentation against the requirement for quick and precise identification, should be considered while choosing between the two algorithms.

7 Conclusion

There are notable variations in performance parameters including accuracy, recall, precision, F1-Score, and AUC when YOLOv5 and Mask R-CNN algorithms are compared to identify fungal diseases on the skin. The disparities arise from variations in iteration (or epoch) values, which affect the algorithms' capacity to acquire knowledge and generalize from the training set. The results of performance tests indicate that algorithm performance is influenced by epoch or iteration values from the Mask R-CNN and YOLOv5 algorithms' first to fifth tests. The second test had the highest AUC value with 1500 iterations of the Mask R-CNN method and 75 epochs of the YOLOv5 algorithm. With an AUC value of 66% and an F1-score of 38%, the Mask R-CNN algorithm is less successful in 1500 iterations at identifying diseases caused by fungal infections of the skin. On the other hand, with an AUC value of 88% and an F1-Score value of 78%, the YOLO algorithm's test results demonstrate its good ability to identify diseases brought on by skin infections, as evidenced by its ability to forecast 179 disorders in epoch 75.

The thorough investigation demonstrates that YOLOv5 performs better than Mask R-CNN in terms of accuracy, recall, precision, F1-Score, and AUC when it comes to identifying fungal diseases on the skin. Iteration and epoch settings have a significant impact on performance; YOLOv5 shows the best performance at 75 epochs. Mask R-CNN is less suited for this application due to its computational intensity and lower detection accuracy, even though it has segmentation capabilities. As a result, in both clinical and real-world contexts, YOLOv5 is the recommended algorithm for identifying fungal-induced skin disorders due to its exceptional efficiency and accuracy. Future research can continue to enhance the detection accuracy and practical applicability of deep learning models for diagnosing skin fungal infections.

References

[1] Majaranta, P. et al. 2019. Eye Movements and Human-Computer Interaction In: Klein, C., Ettinger, U. (eds) Eye Movement Research. Studies in Neuroscience, Psychology, and Behavioral Economics. Springer, Cham, pp.971-1015. https://doi.org/10.1007/978-3-030-20085-5_23

[2] Al Bshabshe, A. et al. 2023. An Overview of Clinical Manifestations of Dermatological Disorders in Intensive Care Units: What Should Intensivists Be Aware of? *Diagnostics* 13, no. 7: 1290. https://doi.org/10.3390/diagnostics13071290

[3] Nigat, T.D.; Sitote, T.M. 2023. Gedefaw, B.M. Fungal Skin Disease Classification Using the Convolutional Neural Network. *J. Healthc. Eng*. 6370416, pp.1-9. https://doi.org/10.1155/2023/6370416

[4] Badia, M. et al. 2020. Dermatological Manifestations in the Intensive Care Unit: A Practical Approach. *Crit Care Res Pract*. 2020:9729814. https://doi.org/10.1155/2020/9729814

[5] Altammami, G. S et al. 2024. Dermatological Conditions in The Intensive Care Unit at A Tertiary Care Hospital in Riyadh, Saudi Arabia." *Saudi Medical Journal* vol. 45, 8. pp. 834-839. https://doi.org/10.15537/smj.2024.45.8.20240479

[6] Khavandi, S. et al. 2023. Investigating the Impact of Automation on the Health Care Workforce Through Autonomous Telemedicine in the Cataract Pathway: Protocol for a Multicenter Study." *JMIR research protocols* vol.12. e49374. pp.1-10. https://doi.org/10.2196/49374

[7] Roy, K. S. et al. 2019. *Skin Disease Detection Based on Different Segmentation Techniques.* 2019 Int. Conf. Opto-Electronics Appl. Opt. Optronix. pp. 1–5, 2019. https://doi.org/10.1109/OPTRONIX.2019.8862403

[8] Archana, R. and Jeevaraj, P.S.E. 2024. Deep learning models for digital image processing: a review. *Artif Intell Rev* 57, 11. https://doi.org/10.1007/s10462-023-10631-z

[9] Thakur, G. K. et al. 2024. Deep Learning Approaches for Medical Image Analysis and Diagnosis. *Cureus* vol. 16,5 e59507. pp. 1-8. https://doi.org/10.7759/cureus.59507

[10] Bhatti, H.M.A. et al. 2020. *Multi-detection and Segmentation of Breast Lesions Based on Mask RCNN-FPN*. Proc. - 2020 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2020, pp. 2698–2704. https://doi.org/10.1109/BIBM49941.2020.9313170

[11] Subash, K.V.V. et al. 2020. *Object Detection Using Ryze Tello Drone With Help of Mask-RCNN*. 2nd Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2020 - Conf. Proc., no. Icimia, pp. 484-490, 2020. https://doi.org/10.1109/ICIMIA48430.2020.9074881

[12] Zhou, Zhijian. et al. 2020. Detection and Classification of Multi-Magnetic Targets Using Mask-RCNN. *IEEE Access*. 8. 187202-187207. https://doi.org/10.1109/access.2020.3030676

[13] Ieamsaard, J. et al. 2021. Deep *Learning-based Face Mask Detection Using YoloV5*. In Proceeding of the 2021 9th International Electrical Engineering Congress, iEECON 2021, Mar. 2021, pp. 428–431. https://doi.org/10.1109/iEECON51072.2021.9440346

[14] Yang, G. et al. *Face Mask Recognition System with YOLOV5 Based on Image Recognition*. 2020. In 2020 IEEE 6th International Conference on Computer and Communications, ICCC 2020, Dec. 2020, pp. 1398-1404. https://doi.org/10.1109/ICCC51575.2020.9345042

[15] Ajith, A. et al. 2017. *Digital Dermatology Skin Disease Detection Model using Image Processing*. Int. Conf. Intell. Comput. Control Syst. ICICCS 2017 Digit., vol. 24, no. 7, pp. 168-173. https://doi.org/10.1109/iccons.2017.8250703

[16] Haddad, A and Hameed, S.A. 2018. *Image Analysis Model for Skin Disease Detection: Framework*. Proc. 2018 7th Int. Conf. Comput. Commun. Eng. ICCCE 2018, no. c, pp. 280-283, 2018. https://doi.org/10.1109/ICCCE.2018.8539270

[17] Rathod, J. et al. 2018. *Diagnosis of skin diseases using Convolutional Neural Networks.* Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2018, no. Iceca, pp. 1048-1051, 2018. https://doi.org/10.1109/ICECA.2018.8474593

[18] Rohaziat, N. et al. 2020. White Blood Cells Detection using YOLOv3 with CNN Feature Extraction Models. *International Journal of Advanced Computer Science and Applications.* 11. https://doi.org/10.14569/IJACSA.2020.0111058

[19] Wu, Z. et al. 2019. Studies on Different CNN Algorithms for Face Skin Disease Classification Based on Clinical Images. *IEEE Access*. vol. 7, no. c, pp. 66505-66511. https://doi.org/10.1109/ACCESS.2019.2918221

[20] Li, L. F. et al. 2020. Deep Learning in Skin Disease Image Recognition: A Review. *IEEE Access*. vol. 8, pp. 208264-208280. https://doi.org/10.1109/ACCESS.2020.3037258

[21] Salama, A. M. et al. 2024. A YOLO-based Deep Learning Model for Real-Time Face Mask Detection via Drone Surveillance in Public Spaces, *Information Sciences*, vol. 676, 120865. https://doi.org/10.1016/j.ins.2024.120865

[22] Chhatlani, J. et al. 2022. *DermaGenics - Early Detection of Melanoma using YOLOv5 Deep Convolutional Neural Networks*. 2022 IEEE Delhi Section Conference (DELCON). pp. 1-6. https://doi.org/10.1109/DELCON54057.2022.9753227

[23] Wiliani, N., et al. 2023. Identifying Skin Cancer Disease Types with You Only Look Once (YOLO) Algorithm. *Jurnal Riset Informatika*. 5(3), 455-464. https://doi.org/10.34288/jri.v5i3.241

[24] Aishwarya, N., et al. 2023. Skin Cancer diagnosis with Yolo Deep Neural Network. *Procedia Computer Science*. vol 220, pp. 651-658. https://doi.org/10.1016/j.procs.2023.03.083

[25] He, K., et al. 2020. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell*. vol. 42, no. 2, pp. 386-397. https://doi.org/10.1109/TPAMI.2018.2844175

[26] Shi, W., et al. 2019. Plant-part segmentation using deep learning and multi-view vision. *Biosyst. Eng*. vol. 187, no. September, pp. 81–95. https://doi.org/10.1016/j.biosystemseng.2019.08.014

https://doi.org/10.31449/inf.v49i16.4974 Informatica **49** (2025) 97–114 **97**

A Unified Trace Meta-Model for Alignment and Synchronization of BPMN and UML Models

Aljia Bouzidi[1], Nahla Haddar[2] and Kais Haddar[2]
[1]ISIMM University of Monastir, Monastir, Tunisia
[2]University of Sfax, Sfax, Tunisia
E-mail: aljia.bouzidi@gmail.com, nahla_haddar@yahoo.fr, kais.haddar@yahoo.fr

Keywords: Traceability, synchronization alignment, use case diagram, class diagram, MVC, BPMN diagram, integration, transformation

Received: February 24, 2025

Organizations often face information system (IS) failures due to misalignment with business goals. Business process models (BPMs) play a crucial role in addressing this issue but are often developed independently of IS models (ISMs), resulting in non-interoperable systems. This paper proposes a traceability method to link BPMs and ISMs, bridging the gap between business and software domains.
We introduce a unified trace meta-model integrating BPMN elements with UML constructs (use cases and class diagrams) via traceability links. This meta-model is instantiated as the BPMNTraceISM diagram, ensuring seamless integration through bidirectional transformation models.
To validate our approach, we developed a graphical editor for BPMNTraceISM diagrams and implemented transformations using the ATLAS Transformation Language (ATL). A case study on a loan approval process demonstrates the method's effectiveness in aligning BPMN and UML elements, improving interoperability and model alignment across domains

Povzetek: Razvit je enoten sledilni meta-model, ki povezuje elemente BPMN in UML (diagram primerov uporabe, razredov po vzorcu MVC) za uskladitev poslovnih procesov in informacijskih sistemov, ki ga validirajo z grafičnim urejevalnikom in transformacijami ATL.

1 Introduction

In the software engineering field, Business Process Models (BPMs) playb an increasingly central role in the development and continued management of software systems. Therefore, it is crucial to have Information System Models (ISMs) that tackle BPMs. modelling. However, these models are mostly expressed using different modeling languages, and only a few information systems (IS) are developed with explicit consideration of the business processes they are supposed to support. This separation causes gaps between business and IS models. Thus, a methodology is needed to examine the gap between BPMs and ISMs, and keep them aligned even as they evolve. Traceability in software development proves its ability to associate overlapping artefacts of heterogeneous models (for example, business models, requirements, uses cases, design models), improve project results by helping designers and other stakeholders with common tasks such as analysis of change impacts, etc. Thereby creating an explicit traceability model that is not a standalone guideline, but it has significant benefits in terms of quality, automation, and consistency. Although creating it is not a trivial task, an explicit traceability model remains a reference for a consistent definition of typed traceability links between heterogeneous model concepts, helping to ensure their alignment and coevolution.

In our previous work presented in [1], we proposed a relevant explicit traceability model and defined it using the integration mechanism. Indeed, we propose a requirements engineering method that works at both the meta-model and model levels, establishing traceability between BPMs and ISMs to bridge the gap between business modeling and requirements elicitation.modeling. This method is deliberately influenced by the Object Management Group (OMG) specifications. Particular attention is given to UML use case models [2] as the most commonly used way to elicit software needs and BPMN [3] as the most widely used language to specify the business process model (BPMs). Indeed, in [1], we firstly defined a unified trace meta-model of the BPMN and the UML use case models in the form of an integrated single meta-model. It defines also traceability links between interrelated concepts to correlate overlapped concepts as new modeling concepts. This meta-model is then instantiated in the form of a new diagram that we called BPSUC (Business Process Supported Use Cases). This new diagram permits business teams and requirements design teams to work together within the same model, and allows specifying trace links graphically.

The practical benefits of the proposed method lie in its ability to bridge the gap between business process management (BPM) and software systems development. In the context of Business Process Models (BPMs) and Information System Models (ISMs), this method enables seamless

integration and traceability across heterogeneous models, which is crucial for ensuring their alignment and coevolution. By establishing clear and accurate traceability links between BPMs, UML use case diagrams, and class diagrams, our method enhances communication and collaboration among business analysts, software engineers, and stakeholders, ensuring that software systems are developed with a clear understanding of the business processes they aim to support. Furthermore, the integration of these models, coupled with the explicit traceability model, provides several practical advantages, including improving change impact analysis, enhancing automation, and maintaining consistency throughout the system lifecycle. These benefits are particularly significant in dynamic environments where both business processes and software systems evolve frequently. Thus, the method not only improves the quality of the development process but also provides a robust framework for aligning business and software models, ensuring their cohesive adaptation to changing requirements and system developments.

This paper enriches and extends our work presented in [1]. The enrichment involves adding class diagram concepts structured according to the MVC pattern. Our intervention considers both the meta-model and the model levels. Hence, in the integrated trace meta-model proposed in [1], we add new modeling concepts to express trace links between the class diagram, use case diagram, and the BPMN concepts. Class diagram concepts that have no corresponding concepts are also included in the integrated trace meta-model. Proposed traceability concepts and class diagram concepts are instantiated in the BPSUC diagram. Accordingly, BPSUC now enables the design of class diagram elements and the proposed traceability concepts combined with their corresponding BPMN and use case diagram artefacts.

We validate our theoretical method by implementing a visual modeling tool to support the enriched integrated trace meta-model and the new diagram supplemented with class diagram elements.

The rest of this paper is organized as follows: Section 2 is dedicated to discussing related work. In Section 3, we give an overview of the method presented in [1]. Section 4 is devoted to explaining our contributions. Section 5 and section 6 are dedicated to demonstrating the feasibility of our proposal in practice and through a topical case study. In section 7, we evaluate and discuss our method. Finally, in section 8, we conclude the current work and we give some outlooks.

2 Related work

We classify related work into two groups based on the methodologies they have used to establish traceability between elements of heterogeneous models: (1) works that have proposed transformation models to define internal or implicit traceability models, and (2) approaches that defined external traceability models manually, basing on some mechanisms, such as model integration, model merge/ composition, UML profiles or matrices.

2.1 Traceability via transformation models

In the first category, existing implicit traceability models are commonly MDA-compliant approaches that define traceability through exogenous, endogenous, horizontal, or vertical transformation models. In these approaches, BPMN models are widely used to generate alternative models through different transformation model types. Among the various uses of BPMN models are: an exogenous-based transformation for the definition of mapping users/organizations' requirements with BPMN models [4]; a vertical transformation for the generation of artefacts between BPMN and user stories [5] and [6]; and the generation of UML models [7]; a horizontal and exogenous transformation for the generation of activity diagrams from BPMN [8] and [9]; and a vertical transformation of textual requirements into a BPMN model [10]. Some approaches define endogenous transformation between UML diagram elements to establish their traceability. For instance, the approach in [11] uses machine learning techniques to maintain traceability information between software models. Their focus is particularly on the requirements, analysis, and design models, which are specified by the UML language. To trace links between requirements documents and UML diagrams, several approaches use Natural Language Processing (NLP). For example, the approach in [12] uses a system requirement description expressed in natural language to extract the actors and the actions automatically.

The core benefit of defining implicit traceability is that it does not require supplemental effort because only one chain of transformation is sufficient to perform transformations in both directions. Moreover, it offers multiple trace links between generated artefacts. However, the identified trace links consider exclusively transformed artefacts. Moreover, the transformation chain is static and cannot be updated to obtain the required traces for such traceability scenarios.

2.2 Explicit traceability models

The second category includes approaches that define explicit traceability models separate from the source models. This category includes approaches that propose guidelines for creating traceability models. For instance, the author of [13] defines a method for guiding the establishment of traceability between the software requirements and the UML diagrams. This guideline has two main components: (i) a meta-model and (ii) a process step. The process step defines the detailed processes, the mapping of requirements to UML diagrams, and the types of requirements. Requirements can be classified according to their aspects. This classification can be carried out according to the type of requirement, which then requires the use of cer-

tain types of UML diagrams. However, this guideline focuses only on establishing traceability at the meta-model level. Moreover, the business field is not considered in this work. The authors of [14] propose a meta-model-based approach to create traceability links between different levels of the same system. Indeed, this approach focuses on defining traceability metamodel source code stored as an Abstract Syntax Tree (AST) and other possible artefacts such as requirements, test cases, etc. To show the identified trace links, authors develop an editor. Nevertheless, storing source code of a system as an AST can cause several problems such as the appearance of syntax errors in the source code, which leads to the loss of traceability links. There is other model-based research that aim to maintain traceability. For example, the research in [15] proposes a co-evolution of transformations based on the propagation of change. Its hypothesis is that knowledge of the evolution of meta-models can be disseminated by decisions aimed at driving the co-evolution of transformations. To address particular cases, the authors present composition-based techniques that help developers compose resolutions that meet their needs. For the same purpose, the approach in [11] refers to machine learning techniques to introduce an approach called TRAIL (TRAceability lInk cLassifier). The training of the classifier is based on a training dataset that contains histories of existing traceability links between pairs of artefacts to output the trace link (related or unrelated) of any future pair of artefacts (new or already existing). Some other approaches define traceability models for eliciting requirements of complex systems [16] and [17]. Likewise, the authors of [19] base their work on deep learning techniques and propose a neural network architecture based on word embedding and Recurrent Neural Network (RNN) algorithms to predict trace links automatically. The output of this model is a vector that contains the semantic data of the artefact. Then, the trained model compares the semantic vectors of a pair of artefacts and predicts if they are related or unrelated. However, considering all meta-models from many different abstraction levels in one unified single traceability model is not a trivial task and can result in very complex models.

In [18], the authors propose an approach to promote traceability and synchronization of computational models in an Enterprise Architecture (EA), using meta-models, model traceability, and synchronization structures. The authors represent the meta-models of the EA at all abstraction levels (strategic, tactical, and operational). These levels are denoted within the integrated meta-model by three packages. Each package incorporates the core concepts of the level it represents. They integrate the three meta-models by adding alignment points between them. In addition, they define a traceability framework and a synchronization framework to support the analysis of the impact of organizational changes.

There are also studies on specific languages. For example, the approach in [20] uses Natural Language Processing techniques to define a framework for managing traceability between software artefacts. To demonstrate their work in practice, the authors develop a tool that supports traceability links between software models, including requirements and UML class diagrams, and the source code written in the Java programming language.

2.3 Identified gaps in existing works

Overall, existing works that define explicit traceability models are mostly focused on the meta-model level only and ignore the model level. Moreover, existing explicit models establish traceability either between software models expressed in UML diagrams at the same or different abstraction levels or between business model artefacts. However, none of the existing approaches have achieved successful results in establishing or maintaining traceability between BPMN models, UML use case models, and the UML class diagram.

The disadvantages of the proposed approaches stem from rigid relationship types that fail to adapt to the changing needs and practices of organizations. Furthermore, most of the proposed approaches define or use very generic traceability meta-models, capable of generating highly abstract trace models. In practice, there is no prescription for how to add customized tracing information or how to adapt a generic traceability meta-model to express valuable and context-specific traces. Concerning the approaches that focus on concrete modeling languages, to our knowledge, there is no approach that proposes an explicit trace model or meta-model between BPMN and UML models, even though they are the most popular standards for modeling business processes and automated information systems.

3 Background of our previous traceability method

The method presented in this paper is an extension of our previous work [1]. In this previous work, we have explored the advantages of defining an integrated traceability model to establish traceability between the BPMN and the UML use case models and ensure their coevolution once a change has occurred. This method acts at both the meta-model and model level, and it includes three core steps:

(i) First, we have defined an integrated trace meta-model that is a specification of traceability between the existing artefacts, while keeping them unchanged and independent. This integrated trace meta-model contains all the BPMN and the UML use case meta-model artefacts (meta-classes and associations) unified with new meta-classes and associations for expressing traceability links at the meta-model level. The integrated trace meta-model favors simplicity and uniformity because source meta-models are kept and unified with their traceability information in one unified meta-model.

(ii) Next, we instantiated the integrated trace meta-model at the model level. We represent it as a new diagram called

Business Process Modeling Notation Traces Use Case (BPSUC). This diagram also incorporates the BPMN and the UML use case elements together with traceability links, and allows designing BPMN and use case diagram artefacts, jointly. Moreover, visualizations and queries on traced elements together are straightforward, because business analysts and software designers are now able to work together on one integrated model. BPSUC can also be used to analyse change impacts and validate them before propagating them to the source models.

(iii) Finally, we defined bidirectional transformation models between the BPSUC diagram and the source models (BPMN and the use case models) to ensure the coevolution of the origin models.

3.1 Integrated trace meta-model

In our previous work presented in [1], a unified trace meta-model is proposed based on a semantic mapping of pairs of BPMN and use case meta-model artefacts. The definition of this meta-model follows the following scenario: For each pair of overlapped BPMN and UML use case concepts, we add a new modeling concept that can be either a link, such as an association, a composition or an inheritance, or a new meta-class. Each trace link represented by a new meta-class is associated with the pair artefacts it specifies, generally, by an inheritance relationship.

Table 1 summarizes mappings between the BPMN model (first column) and the use case diagram concepts (second column), and the corresponding new meta-classes (third column) that are associated with them in the integrated meta-model (the full mapping and its explanation is available in [1]). To validate the proposed mapping further, we have conducted additional evaluations across a variety of BPMN and UML diagram scenarios.

This expansion includes not only the core BPMN and UML elements such as activities, actors, and use cases but also more complex diagrams, such as:

- BPMN Models: Event-driven processes, process variants, and sub-processes with different complexity levels, such as loan approval and inventory management systems.
- UML Diagrams: Class diagrams, including inheritance and association relationships, and more sophisticated use case diagrams representing different business functions, such as order fulfilment, customer support, and system maintenance.

These diverse scenarios have allowed us to assess how well the mapping between BPMN, UML use case, and class diagrams holds up in real-world business process and system modeling. By applying the proposed traceability method to these varied scenarios, we demonstrate the scalability and robustness of our meta-model. We have also provided examples where the mapping effectively handles the integration of different BPMN and UML model types, ensuring traceability between the business and software models.

Table 1: Mapping of BPMN, use case and the trace meta-model concepts

Use Case Concept	BPMN concept	Meta-model concept
Package	Empty Lane (a lane including other sub-lanes)	Organisation Unit Package
Actor	Non empty Lane (that does not contain other sub-lanes)	Organization Unit Actor
Use case	Fragment represented by a sequence of BPMN artefacts that is performed by the same role and manipulates the same item aware element (business object, input data, data store, data state) d	UCsF
Extends	Exclusive Gateway between two different fragments	Exclusive Gateway, Extends
Association	Fragment within the lowest nesting level of sub-lanes	Association
Includes	Redundant Fragment (that appears multiple Times in the BPMN model)	Fragment that appears multiple times, Includes
Extends	Inclusive Gateway between two different fragments	Inclusive Gateway, Extends
Extension Point	Condition of sequence Flow + the name of the fragment that represents to the extending use case	Extension Point

The integrated meta-model is depicted in Figure 1). In order to be able to read it, we have presented in this Figure only the core artefacts of the source meta-model (use case meta-model and BPMN met-model) and all the trace links (meta-classes and associations). Dark grey meta-classes represent new meta-classes**;** light grey meta-classes represent UML use case elements**,** white meta-classes represent BPMN elements**,** and black lines represent existing associations from the source meta-models The blue lines represent trace relationships, providing the foundational traceability between BPMN and UML elements, as further detailed in [1] and [22].

– *Organizational-Unit-Package*

In BPMN, a non-empty lane is a grouping element and therefore has the same meaning as a package in UML. Consequently, the Organizational-Unit-Package (OUPackage)

is defined to trace the link between the pair BPMN non-empty lane and the use case package thereby defining an inheritance relationship between the new meta-class and this artefact pair.

– *Organizational-Unit-Actor*

In the proposed integrated trace meta-model of [1], we have defined a meta-class designated Organizational-Unit-Actor (OUActor). This new meta-class traces artefacts of the pair UML actors and BPMN empty lanes (i.e. lanes that do not have embedded lanes). That is, it unifies the properties of a lane and an actor, and combines them without changing their semantics there by defining the OUActor as specialization of the UML actor and the BPMN Lane pair. In this way, OUActor inherits properties of this pair of artefacts without updating their original semantics and structures. For example, in a loan approval process, the Loan Officer is represented in a BPMN diagram by a non-empty lane and in a UML use case diagram as an actor involved in the process. The OUActor meta-class links these representations, inheriting properties from both while preserving their original semantics. This ensures synchronization between BPMN and UML elements, offering a unified view of the Loan Officer's role across models.

– *Fragment*

A fragment is defined by [22] as "*a set of interrelated BPMN elements that has inputs and outputs, and which is executed by the same performer*". This artefact is specified in the unified trace meta-model as an instance of the meta-class Fragment (cf. Figure 1)). As a Fragment is just an activity that can contain other BPMN concepts such as tasks, events, gateways and sequence flows, we have aggregated a BPMN sub-process to a Fragment by creating an aggregation relationship called *fragments,* between the fragment and sub-process meta-classes in the integrated trace meta-model (cf. Figure 1)). Its cardinality is *1-** to point out that a sub-process should contain at least one fragment but it may incorporate more than one Fragment. In addition, we define a many-to-many reflexive association in the fragment to represent the fact that a fragment may be an aggregation of other fragments (cf. Figure 1)). Moreover, we create an association between the data object and the fragment (cf. Figure 1)) to associate each fragment with the objects it manipulates. The cardinality of this relationship is fixed to 1-* to indicate that each fragment manipulates at least one business object type, but it may manipulate more than one business object. Furthermore, we have defined an association called *organizationUA* between the meta-classes OU-Actor and fragment with a cardinality of 1-* to associate a fragment with its performer. For instance, tasks such as *Review Application*, *Assess Credit Score*, and *Approve Loan* in the BPMN Loan Approval Process are all performed by the Loan Officer. The Fragment aggregates these tasks into a cohesive group and connects them to the BPMN sub-process as well as the business objects (e.g., Loan Applications, Credit Scores) manipulated during the process. This provides clear traceability between tasks, business objects, and performers while maintaining logical consistency.

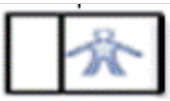
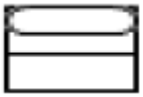
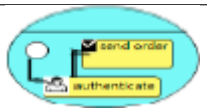
– *Use case supporting fragment*

In order to support business objectives, such a UML use case should be able to realize some business activities that are specified in the integrated trace meta-model by a Fragment. A separate specification of the use case and the fragment that is supposed to realise does not allow explicitly representing the semantic links between them. To do this, we have defined the integrated trace meta-model presented in [1], which introduces a new meta-class that we designate *Use Case supporting Fragment (UCsF)* This new meta-class is defined as a specialization of a UML use case in order to inherit all its properties without updating its initial meaning.

3.2 BPSUC diagram

To allow modeling the artefacts of the proposed integrated trace meta-model, we have instantiated it in the form of an integrated trace model, in our previous work [1]. We represent it as a new diagram that we have called BPMN Supporting Use Case model (BPSUC).

Table 2: Notation of the traceability artefacts

Meta-model concept	Graphical notation
OU-Actor	
OU- Package	
Use Case supporting Fragment (UCsF)	

For each concept, we have provided a graphical notation as follows: We have introduced new notations to the proposed new meta-classes UCsF, Organization Unit Package, and Organization Unit Actor. These notations are inspired by and extended from the icons of the pair of artefacts they represent. The inspiration ensures that experienced business and system designers are comfortable using the BPSUC diagram. Each concept originates form UML use case and BPMN models[1], and retains its original notation. In the BPSUC diagram, the Fragment is instantiated as a specific activity within the Loan Approval Process, linking BPMN tasks to the Loan Officer (OUActor) and business objects like the Loan Application. Additionally, the Organization Unit Package (OUPackage) meta-class is used to trace relationships between BPMN lanes and UML packages. For example, functional areas such as Loan Review, Credit Assessment, and Loan Approval in the BPMN diagram are mapped to corresponding UML packages, en-

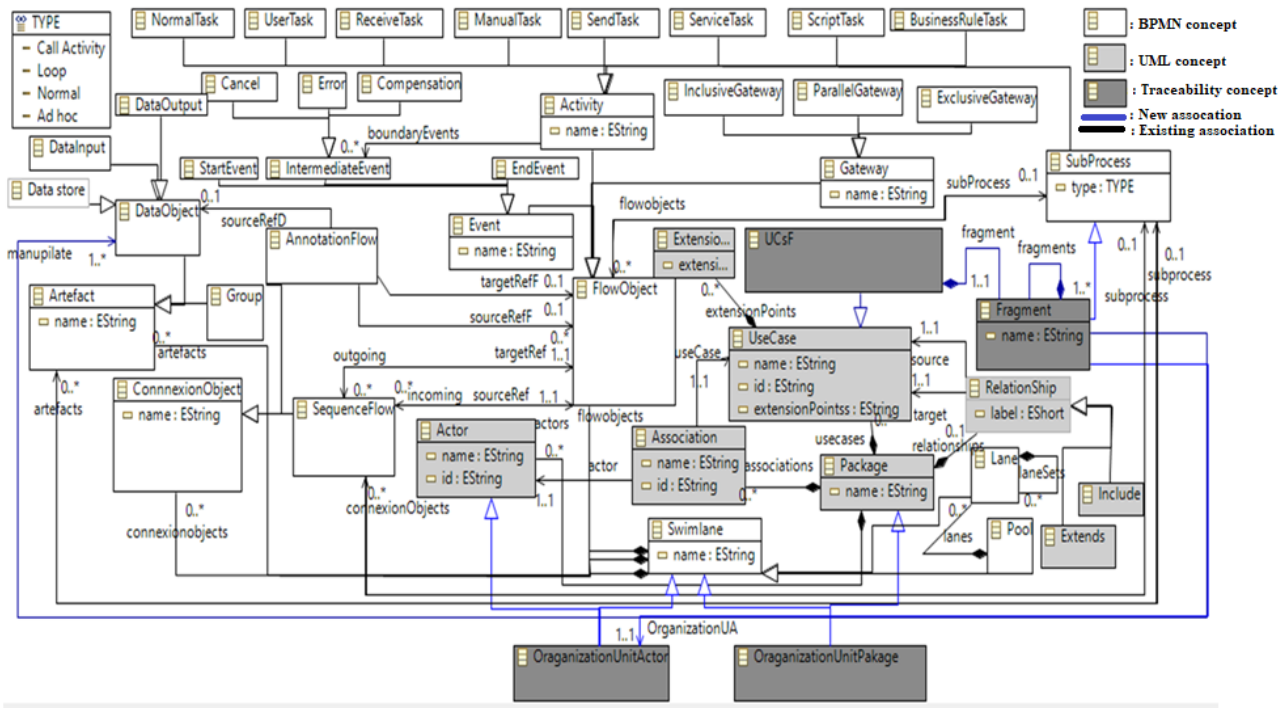


Figure 1: Traceability of BPMN and use case meta-model concepts

suring alignment and traceability between these functional areas and their UML counterparts.

Table 2 depicts the graphical notations of the new meta-classes Organisation Unit Actor, Organisation Unit Package and the UCsF.

4 Traceability method

The research work conducted in this paper is an extension and enhancement of our previous work presented in [1]. The extension consists of improving the integrated trace meta-model and the BPSUC diagram to include the artefacts of the UML class diagram structured according to the MVC design pattern. Our contribution aims not only to es-

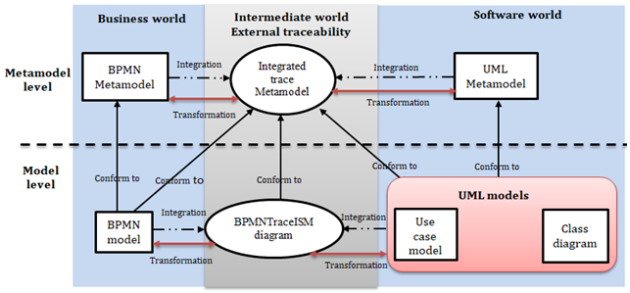


Figure 2: Background of the traceability method

tablish alignment but also to keep source models always aligned even if they evolve. The propagation of changes from our trace model to the source models is carried out through defined MDA model transformations. The former is explored to guarantee the coevolution of the BPMN and the UML models. Figure 2) demonstrates the background of our traceability method. In this section, we further explain how we extend and improve the integrated trace meta-model and the BPSUC diagram, as well as the rectifications made to them.

4.1 Integrated trace meta-model improvement

Our first improvement of the integrated trace meta-model consists of defining an adequate strategy for defining its concepts and relationships between them. Indeed, we propose a methodology for defining the integrated trace meta-model which includes two main steps: (1) identifying overlapping concepts of BPMN and UML meta-models to define a relevant mapping between them, and (2) Defining an adequate methodology to link each pair of interrelated concepts, without changing their semantics. Thus, we propose to keep overlapping concepts and connect them either by a new concept, or a new relationship, which specifies the trace link between existing concepts at the meta-model level. Afterwards, we connect each pair of artefacts to the new concepts representing them through a generalization/specialization relationship. This relationship allows inheriting properties of both separated concepts as well as combining their usage without updating their initial semantics. Our second improvement consists of adding the class

diagram meta-model artefacts to the previous version of the integrated trace meta-model.

By applying our meta-model construction, we need to identify adequate mappings between BPMN and UML class diagram concepts. In the literature mapping between BPMN and UML class diagram concepts is widely discussed. Among them, [23] defined model transformations from BPMN into UML class diagrams structured according to the MVC design pattern, and use case diagrams based on semantic mappings. For example, they propose mapping each BPMN empty lane (i.e. lane that does not include other lanes) into a class in the class diagram, and into an UML actor in the use case model. We reuse the defined semantic mappings in this approach to continue the definition of the integrated trace meta-model.

Table 3: Mapping of BPMN and class diagram meta-model concepts

BPMN concept	UML class diagram meta-model
Item Aware Element (Data object, Data store, Data input, Data output or Data state)	– Entity class – Association
Empty lane	– Entity class – View class – Control class – Association
Fragment	– View class – Control class
Exception event	– Exception class – Operation
Signal event	– Signal Class – Operation
Automated task *t* (business rule task, receive task, send task, user task, script task, service task) within a fragment	– Operation – Association
Item aware element type (single or collection) or Gateway or Loop task/ Rollback sequence flow	Cardinality of associations
Item aware element attached to an automated task *t* within a fragment *f*	Parameters of an operation
Conditional sequence Flow	Attribute

In Table 3, we summarize the semantic mapping between the BPMN meta-model artefacts and the class diagram meta-model artefacts from [23]. In this Table, the class diagram meta-model concepts are structured according to the MVC design pattern.

In contrast to the mapped concepts of the use case and the BPMN meta-model artefacts, the mapping between the interrelated concepts of UML class diagram and the BPMN meta-model is not tight, as shown in Table 3. Indeed, one UML concept may be represented by many BPMN concepts and vice versa. This is due mainly to the important degree of heterogeneity between the BPMN and the class diagram artefacts. Thus, our mapping is limited to defining associations instead of defining new traceability concepts, as we aim not to complicate our integrated trace meta-model, and therefore facilitate its readability while maintaining its consistency. The aforementioned trace meta-classes can also be reused to define BPMN- class diagram concept traceability.

The excerpt of the meta-model defined to trace the BPMN and the class diagram meta-models is presented in Figure.3. To ensure readability, Figure 3) depicts only the main artefacts of class diagram and BPMN meta-models, as well as the reused traceability concepts.

White meta-classes are BPMN concepts, orange meta-classes are UML class diagram meta-model concepts, khaki meta-classes denote UML class diagram concepts used for structuring the class diagram according to the MVC design pattern, while new concepts are specified by dark grey meta-classes. The blue associations represent the new trace links, while the black ones are the existing associations.

It is important to note that all the use case concepts, BPMN concepts, traceability links and existing associations defined in the previous extract of the integrated trace meta-model, which are not present in this extract remain valid.

In the excerptof Figure.3, each BPMN concept is associated with its corresponding concept in the class diagram meta-model. For example, we define a trace link called trace between the data object and the entity class to establish traceability between them.

The multiplicity of this association is 1..* to indicate that each item-aware element should represent exactly one entity class. Moreover, we define a trace link between the gateway and the property meta-classes as gateways can be indicators of association cardinalities. The multiplicity of this association is 0. On the other hand, UCsF is linked to the following meta-classes: class, ClassDIPackage, and association by composition . This means that a UCsF can include classes, associations, and packages. These associations mean that a UCsF is a use case that incorporates its supported class diagram elements, representing the supported fragment elements.

The cardinality of the composition association UCsF-ClassDIPackage is 3-* to indicate that an UCsF should incorporate at least three packages: View, Control, and Models, which represent the three parts of the MVC design pattern.

In addition, we define an association between OUActor and class to express that an actor in the integrated trace meta-model is represented as a class in the class diagram meta-model. Furthermore, in our integrated trace meta-

model, a generalization/specialization relationship between the meta-classes OUPackage and ClassDIPackage is defined to point out that this trace meta-class inherits all properties of the *Package* meta-class.

4.2 BPSUC diagram improvement

In contrast to most existing approaches [11], [13], [14], [15], [16], [17], [18], [19], [20], [21], which focus only on the meta-model level, our traceability method includes both the meta-model and the model levels. Thus, the second step of our contribution is devoted to describing how the traceability of BPMN and UML artefacts is established at the model level. We have improved the proposed BPSUC diagram from [1], in which the BPSUC diagram features are limited to designing thoroughly the BPMN and the use case diagram artefacts, , combined with their traceability links, which already reflects its designation.

In this paper, we aim to enrich this diagram to incorporate class diagram elements combined with BPMN and use case diagram elements. The first thing we do is update the name of BPSUC to be in harmony with its newly supported features. The new designation we have chosen is *BPMNTraceISM* (**B**usiness **P**rocess **M**odel and **N**otation **Trace**s **I**nformation **S**ystem **M**odels). *BPMNTraceISM* is an instantiation of the new version of the improved meta-model and forms a single unified model that combines the usage of UML elements including the use case diagram and class diagram elements, as well as the BPMN elements thoroughly. Thus, this diagram is now able to design elements and relationships of both UML use case and class diagrams, as well as BPMN models, concurrently. Moreover, it specifies the traceability information of the interrelated artefacts.

Each artefact in *BPMNTraceISM* diagram has its specific notation. Some of them retain the original notation (BPMN or UML notations), while the others have a new representation, which does not differ greatly from BPMN and UML notations.

4.2.1 BPMNTraceISM artefacts conserve their initial notations

The mappings on which we base the definition of the integrated trace meta-model comprises neither all the BPMN concepts nor all the UML concepts. This is due to the fact that, some BPMN artefacts do not have their corresponding UML artefacts, and vice versa. For example, the mapping does not define any UML concept representing a BPMN start event.

Even though, in a BPMNTraceISM diagram, it is possible to specify UML artefacts with no corresponding elements in BPMN. According to the mapping, many elements of UML are mapped to one BPMN element. Thus, the representation of these elements in UML diagrams requires grouping them. On the other hand, one UML element may be linked to many BPMN elements. For example, a data store in the BPMN diagram is transformed into (i) an association, (ii) an entity class, and (iii) an operation of a class, in the class diagram. In this situation, it is very difficult to represent the mapped elements in one unifying element. At the meta-model level, we have proposed associating each pair of these mapped concepts by an association instead of

Table 4: Graphical notations of overlapping elements of BPMNTraceISM diagram

BPMNTraceISM element	Graphic notation	BPMNTraceISM element	Graphic notation
Use case association		Signal event	
Extends relationship	<<extends>>	Exclusive gateway	
Includes relationship	<<includes>>	Parallel gateway	
Annotation flow		Inclusive gateway	
Start event		Data input	
End event		Data output	
Manual task		Data store	
Normal task		Sequence flow	
Error event		Group	
Cancel event		Entity class	<<Entity>>
Control class	<<Control>>	Generalization	
Signal class	<<Signal>>	Aggregation association	
View class	<<Boundary>>	Composition association	
Exception class	<<Exception>>	Directed association	

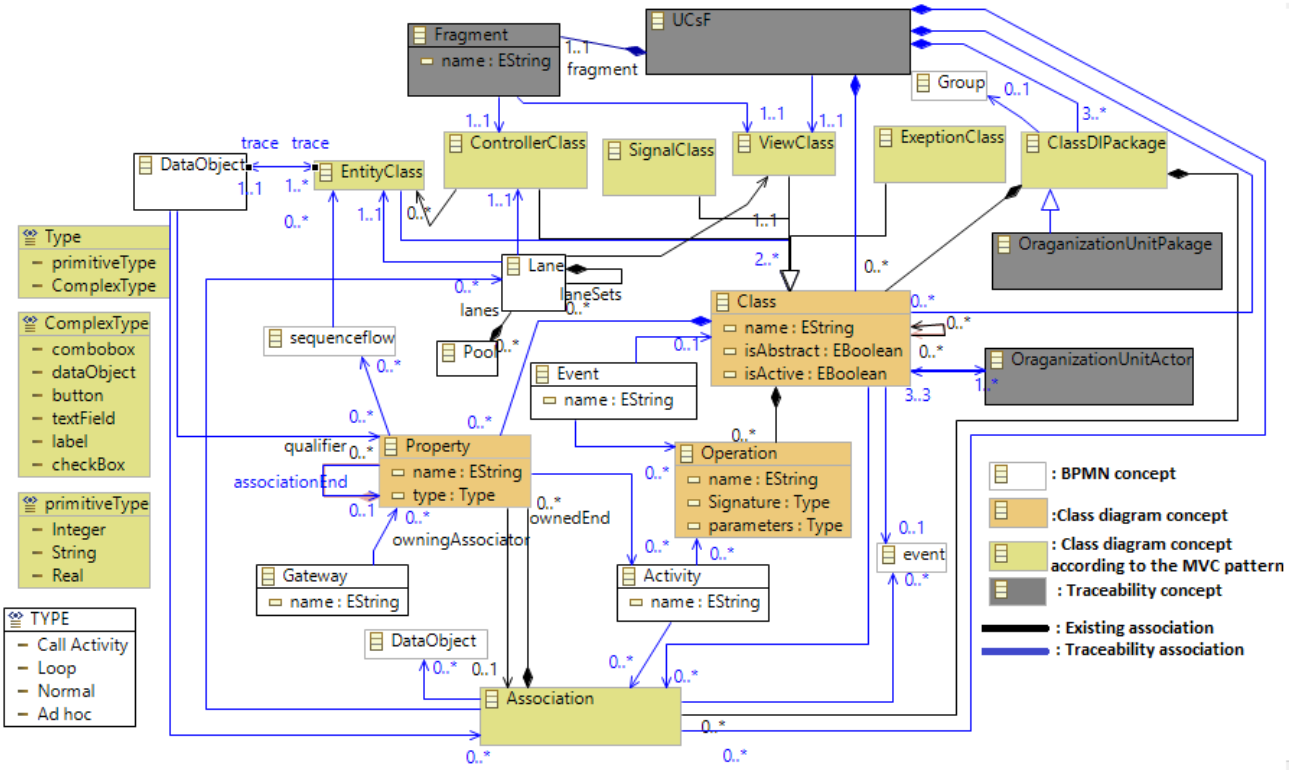


Figure 3: Traceability of the BPMN meta-model and the UML class diagram meta-model

defining new traceability meta-classes. At the model level, these artefacts are processed similarly to the non-mapped concepts, and retain their original notations in the BPMNTraceISM diagram. Table 4 outlines the graphical notations of the core artefacts of the BPMNTraceISM diagram that retain the initial notations. OUPackage and OUActor are new meta-classes defined by [1] to represent traceability links of BPMN and UML use case diagram elements. In the integrated trace meta-model, we did not reuse these meta-classes to define new associations. Thus, the instantiation of these meta-classes keeps the annotations provided in [1].

4.2.2 UCsF notation

In the previous version of the diagram *BPMNTraceISM* (in a BPSUC diagram), [1] states that a UCsF is a specialization of a use case and inherits its properties. Therefore, the graphical notation of UCsFs extends the graphical notation of a UML use case. Moreover, a UCsF has composition relationships to (i) a BPMN Fragment. To represent this trace link, graphically, [1] defines a compartment that incorporates the corresponding BPMN fragment.

In our integrated trace meta-model, we have defined composition relationships from a UCsF to some UML class diagram artefacts (cf. Figure 3)). Indeed, a UCsF should encapsulate classes, associations and packages, which correspond to its supported fragment. Accordingly, we propose to update the graphic notation of the UCsF. Thus, a UCsF should act as a complex symbol that describes concurrently BPMN elements and UML class diagram elements. In order to represent explicitly the different elements incorporated by a UCsF, the use case notation needs to be extended. Therefore, we adjust the UCsF notation by adding another compartment (cf. Figure 4) to encapsulate class diagram elements representing the components (classes, associations and packages) of the supported fragment. In order to avoid the complexity of this element, the designer can choose to hide or show each compartment. Figure 4) depicts the graphical notation of a UCsF, in which all compartments are hidden.

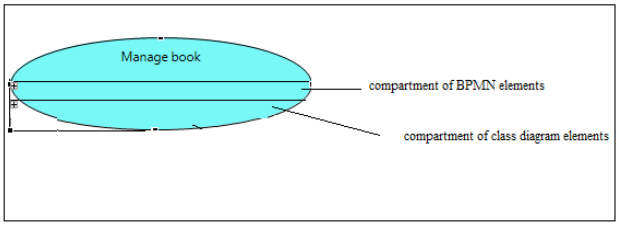


Figure 4: UCsF notation

4.3 Change propagation improvement

Our traceability method aims to ensure the coevolution of the separated models when a change occurs either in the

source models (BPMN model, use case model, and/or class diagram) or in the *BPMNTraceISM* diagram (cf. Figure 5)). To do this, we have improved the transformation model defined in [1] thereby including the class diagram concepts in the bi-directional transformation rules defined in [1] as two sets of transformation models (forward and backward transformation rules). They ensure the transformation between the BPMNTraceISM diagram, the BPMN, and the UML models that include a class diagram and a use case diagram using a semantic mapping between BPMN, *BPMN-TraceISM,* and UML elements derived from our integrated trace meta-model.

Each transformation model includes a set of well-defined transformation program or transformation rules (*Tab*) (*Tab* conforms to *MMt*) that transform source models (*Ma*) conforming to source meta-models (*MMa*) (noted Ma/*MMa*) to target models (*Mb*) confirming to target meta-models (*MMb*) (noted Mb/*MMb*) according to a mapping between the source and target model artefacts (noted $map_{(Ma,\ Mb)}$). Formally, we specify transformation models according to a function that we call *MtransF*

This function is formally written as follows:

$$MtransF\left(Ma/MMa,\ map_{(Ma,Mb)}\right) \overset{Tab/MMt}{\rightarrow} Mb/MMb \tag{1}$$

For example, consider the forward transformation rule R1 that transforms a UML package and a non-empty BPMN lane into a OU-Package in the BPMNTraceISM diagram. This transformation ensures that elements in the source models are correctly mapped to their counterparts in the BPMNTraceISM diagram, facilitating traceability across both business and software models.

The proposed bi-directional transformation models (backward and forward) ensure the coevolution of BPMN and UML models, as well as the coevolution of the source models (business model specified by a BPMN diagram and software models specified by a UML class diagram and an UML use case diagram) and the BPMNTraceISM diagram.

Formally, the forward and backward transformation model is specified as follows:

$$\begin{array}{c} MtransF\left(Ma/MMa,\ map_{(Ma,Mb)}\right) \\ \overset{forwardRules,\ backawardRules}{\leftrightarrow} (Mb/MMb) \end{array} \tag{2}$$

The rest of this section will be devoted to providing more details on how we created the bidirectional transformation rules.

4.3.1 Forward transformation rules

We propose a forward transformation model (*Forward rules*) to produce automatically a BPMNTraceISM diagram ($M_{BPMNTraceISM}$) conforming to our integrated trace meta-model ($MM_{BPMNTraceISM}$) from the source models, namely a BPMN diagram (M_{BPMN}) conforming to the BPMN meta-model (MM_{BPMN}), a use case model (M_{UCM}), and a class diagram (M_{CD}) conforming to the UML meta-model (MM_{UML}). This transformation is carried out based on mappings between the new diagram, the BPMN and UML models.

The formal definition of our forward transformation rules is as follows:

$$\begin{array}{c} MTransF\,((M_{BPMN}/MM_{BPMN}, M_{UCM}/MM_{UML}, \\ M_{UCD}/MM_{UML}, \\ (map_{(MUCM,MBPMNTraceISM)}, \\ map_{(UCD,MBPMNTraceISM)}, \\ map_{(MBPMN,MBPMNTraceISM)}) \\ \overset{forwardRules}{\rightarrow} M_{BPMNTraceISM}/MM_{BPMNTraceISM} \end{array} \tag{3}$$

There are two possible scenarios for producing the B*PMNTraceISM* elements based on the forward transformation rules:

The first scenario consists of applying a forward transformation rule (R_X) to derive trace modeling elements (*tre*) represented in the *BPMNTraceISM* diagram from a BPMN element (M_{BPMN}*!Element*) and a UML element (M_{UML}!Element). More precisely, a *OUActor, a OUPackage,* and a U*CsF* of the *BPMNTraceISM* diagram are generated from the BPMN and UML elements. Formally, these transformation rules are as follow:

$$\begin{array}{c} MTransF_{tre}((M_{BPMN}!Element, M_{UML}!Element), \\ (map_{(MBPMN,MBPMNTraceISM)}, \\ map_{(MUML,MBPMNTraceISM)})) \\ \overset{R_X}{\rightarrow} M_{BPMNTraceISM}!tre \end{array} \tag{4}$$

For instance, we suppose that a forward transformation rule R_1 produces an OU-Package from a UML package and a BPMN non-empty lane. Formally, this rule can be written as follows:

$$\begin{array}{c} MtransF_{OUPackage}((M_{BPMN}!lane, M_{UCM}!Package), \\ (map_{(MBPMN,MBPMNTraceISM)}, \\ map_{(MUCM,MBPMNTraceISM)})) \\ \overset{R_1}{\rightarrow} M_{BPMNTraceISM}!OUPackage \end{array} \tag{5}$$

The second scenario consists of generating unrelated elements (*ure*) from either, the UML models or the BPMN model only. Indeed, each concept in the BPMNTraceISM, that corresponds to a concept in the source models (BPMN model, a UML class diagram or a UML use case diagram) needs just its original model. In this case, the input of this transformation rule is either a BPMN model if this concept comes from BPMN or a UML model if its origin is the UML class diagram or the UML use case diagram. For example, the generation of a *manual task* in the BPMNTraceISM diagram requires the BPMN model only because a manual task does not have corresponding elements in the UML diagram.

This transformation rule is written as follows:

$$MTransF_{ure}(\ M_{BPMN}!manualTask,\ map_{(MBPMN,MBPMNTraceISM)} \overset{R(ManualTask)}{\rightarrow} M_{BPMNTraceISM}!manualTask \tag{6}$$

Let's illustrate how a change in the BPMN model (e.g., adding a new lane) is propagated into the BPMNTraceISM diagram. Assume that Rule R1 is applied to generate an OU-Package from the UML package and the BPMN lane. The transformation process involves the following steps:

1 The BPMN Lane and UML Package elements are identified.
2 Rule R1 is triggered, creating a corresponding OU-Package in the BPMNTraceISM diagram.
3 The OU-Package is updated within the BPMNTraceISM diagram, and the changes are synchronized with the BPMN and UML models.

4.3.2 Backward transformation rules

To have the opposite direction of the forward transformation rules, we have defined a backward transformation model. This means that the source elements of each forward transformation rule become the target elements of a backward transformation rule, and its target elements become the source elements of the backward transformation rule. Formally, backward transformation rules are written as follows:

$$\begin{gathered} MTransF(M_{BPMNTraceISM}/MM_{BPMNTraceISM}, \\ (map_{(MBPMNTraceISM,MUCM)}, \\ \\ map_{(MBPMNTraceISM,MUCD)}, \\ map_{(MBPMNTraceISM,MBPMN)}) \\ \overset{backwardRules}{\rightarrow} \\ M_{BPMN}/MM_{BPMN}, M_{UCM}/MM_{UML}, \\ M_{CD}/MM_{UML}, M_{UCD}/MM_{UML} \end{gathered} \tag{7}$$

We use the same logic as in the forward transformation rules to define the reverse transformation rules. Therefore, each backward transformation rule of each pair of artefacts is defined according to the following formula:

$\text{MTransF}_{\text{tre}}(M_{\text{BPMNTraceISM}}!tre,$

$$\begin{gathered} MTransF_{tre}\ (M_{BPMNTraceISM}!tre, \\ (map_{(MBPMNTraceISM,\ MBPMN\ )}, \\ map_{(MBPMNTraceISM,MUML\ )})) \\ \overset{R_X}{\rightarrow} (M_{BPMN}!Element, M_{UML}!Element) \end{gathered} \tag{8}$$

Non-overlapping artefact transformation rules are defined according to the formula below:

$$\begin{gathered} MTransF_{ure}(M_{BPMNTraceISM}!ure, \\ (map_{(MBPMNTraceISM,MBPMN)}, \\ map_{(MBPMNTraceISM,MUML)})) \\ \overset{R_X}{\rightarrow} (M_{BPMN}!Element, M_{UML}!Element) \end{gathered} \tag{9}$$

For example, when a change occurs in the BPMNTraceISM diagram, such as adding a new OU-Actor, the corresponding elements in the BPMN and UML models need to be updated. The backward transformation rule ensures that:

1 The OU-Actor is mapped to both the UML Actor in the use case diagram and a new BPMN lane in the BPMN model.
2 The changes are propagated back into the source models, maintaining alignment across the models.

4.3.3 Change propagation process

The bidirectional transformation rules allow propagating changes that occur in the source model into the target models. By applying these rules this approach enables the co-evolution of the business and software models.

The change propagation process is carried out in two ways (cf. Figure 5): (1) by manually updating the source models (BPMN, UML, and UML use case diagram) , or (2) by designing the *BPMNTraceISM* diagram.

In the first case, software designers and business analysts separately and concurrently update the BPMN modeland consequently the use case diagram and/or the class diagram. For example, a software designer adds a new use case to the use case model, new classes responsible for realizing the new use case, and simultaneously, a business analyst changes the name of a lane in the BPMN model. A direct generation of the software models leads to the loss of changes made by the software designers. Additionally, to avoid unintentional updates, the impact of changes involved in a (business or UML) model needs to be analysed before propagating it to the target model. To tackle this problem, an intermediate step is required to make all updates made in the separate models. This step can be reached by executing our forward model (user task "Execute forward transformation rules"), which derives a *BPMNTraceISM* diagram from both UML and BPMN. Thus, all changes made on the BPMN and/or on the class and use case diagrams are considered in the derived *BPMNTraceISM* diagram.

In the second case, all updates made by business analysts and software designers are done in the unified trace model (*BPMNTraceISM* diagram) of being made in the BPMs and the ISMs. Using *BPMNTraceISM* overcomes the gap between the business analysts and software designers, and enables them to work together using the same model. Indeed, this diagram covers all business and software model elements and the traceability concepts of pairs of mapped artefacts. Any change involving a *BPMNTraceISM* element (*bp*) leads to the modification of the BPMN or/and UML model elements traced by *bp*. For example, the insertion of a new OU-Actor in the *BPMNTraceISM* diagram leads to the insertion of and a new UML actor in the UML use case diagram and a new BPMN lane in the BPMN model. *BPMNTraceISM* can act as a gateway allowing business analysts and software designers to work together to test,

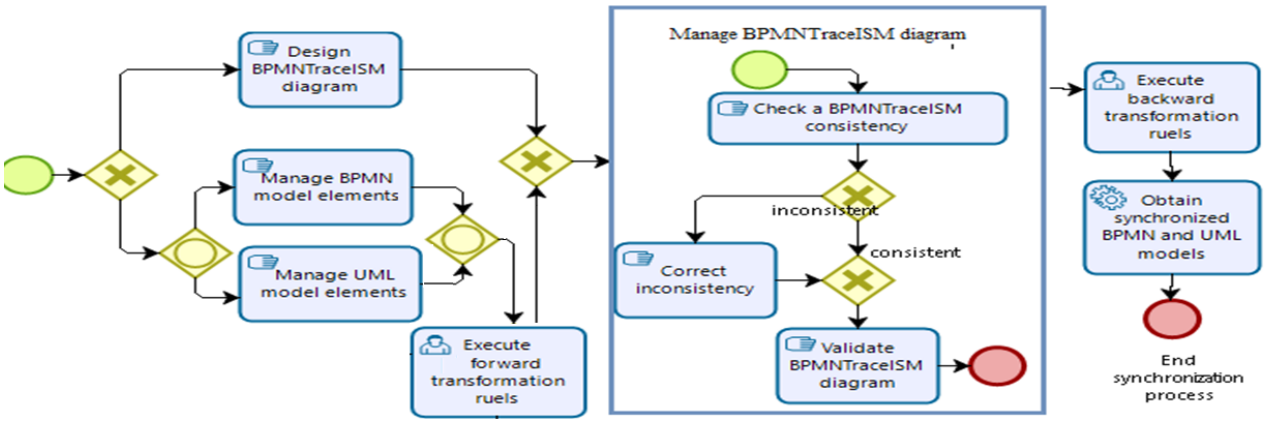


Figure 5: Synchronization process of BPMN and UML

analyse, and correct inconsistencies due to unwanted updates before propagating them to the source models (BPMN and UML models). In addition, this new diagram can be used to analyse and estimate the impact of changes made to business or system components or services. Until this step, although the *BPMNTraceISM* diagram is aware of the updates made by both business analysts and software designers, the source models are not aware neither the BPMNTraceISM diagram nor each other. Accordingly, propagating the modifications is an essential step to ensure the coevolution of the source models. We can do this easily by running our backward transformation model (user task "execute backward transformation rules"). Once the backward transformation model is run, changes are propagated to BPMN and UML models and thus these models are aligned with each other.

5 Implementation

To use the proposed traceability approach, we implement a visual editor called Business Process model Traced with Information System Models (*BPTraceISM*).

Moreover, we develop a prototype called Business Process to Information System Models (BP2ISM) that provides significant practical support for the transformations involved in our traceability method. This prototype automates the suggested forward and backward transformation models between the business process and the ISMs on the one hand, and the BPMNTraceISM diagram on the other. These transformation models are automatically applied via transformation rules expressed in the ATL transformation language.

5.1 Visual editor implementation

To implement the *BPTraceISM* editor, we have used Eclipse EMF to implement the trace meta-model and Eclipse GMF to design the concrete syntax of the *BPMNTraceISM* diagram. Indeed, the modeling tool includes a graphical editor that conforms to the trace meta-model and enables concurrently seeing and managing trace relationships between the BPMN model, the use case, and the class diagram. *BPTraceISM* can be integrated within other modeling tools to enhance their modeling capabilities. To make our modeling tool available in any Eclipse environment without need to start an Eclipse runtime, we implement it as an Eclipse plug-in.

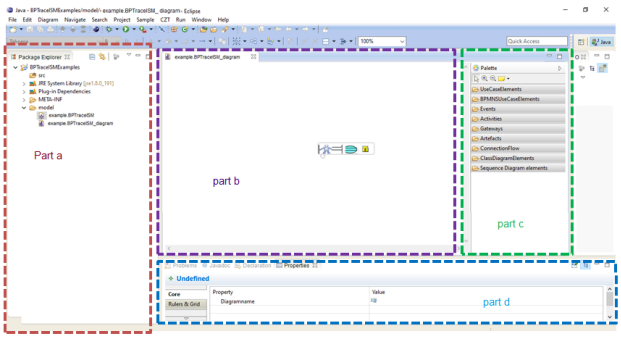


Figure 6: The environment of the *BPTraceISM* editor

The construction process of the *BPTraceISM* consists of two main phases: (1) the definition of the modeling tool, and (2) the definition of the plug-in that supports it. The first phase begins with the implementation of the trace meta-model using the ecore meta-modeling language. Then we build a toolbox for creating instances of the meta-model classes. In the second phase, we develop a feature that supports the modeling tool. Afterward, we construct an update site to ensure the portability of our plug-in and allow its installation via any Eclipse update manager.

BPTraceISM environment is composed of four main parts (cf. Figure 6): the project explorer containing an EMF project that includes *BPMNTraceISM* diagrams (part a), the modeling space (part b), the toolbox containing the graphical elements of a *BPMNTraceISM* diagram (part c), and the properties tab to edit the properties of an element selected

in the modeling space (part d).

Figure 7) outlines a simple example of a BPMNTraceISM diagram created using the editor. The modeling space contains an OUActor called supplier associated with a UCsF called manage purchase order. In the business compartment of the UCsF *Manage purchase order*, we have a user task called *Accept purchase order*. In the class diagram compartment, we have four classes linked via undirected associations. Each class has a name and a stereotype. The boundary class *Manage purchase order* contains an operation called *acceptPurchaseOrder()*

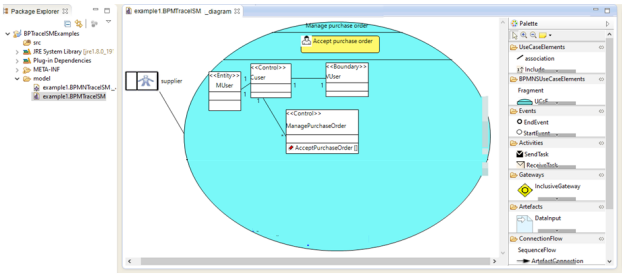


Figure 7: Example of a *BPMNTraceISM* diagram within the modeling tool

5.2 Prototype for the transformation models

BP2ISM is implemented within the Eclipse Modeling Framework (EMF) environment. It includes two components:

- *BPISM2BPMNTrISM*: It automates the forward transformation, which is the conversion of BPMN and UML models into BPMNTraceISM.
- *BPMNTrISM2BPISM*: It automates the backward transformations.

The transformation process requires tools, editors or plugins in order to specify source and target models. For this reason, tools are required to represent BPMN, UML and *BPMNTraceISM* diagrams, which serve as the source and target models of *BP2ISM* components. Because BPMN and UML are widely used standards, many plugins and tools have been created and certified to support them. We choose to employ internal plugins within EMF instead of existing plugins. As a result, we develop BPMN models with the Eclipse BPMN2 modeler plugin and UML use case and class diagrams with the UML designer plugin. Internal meta-models in these plugins closely adhere to OMG requirements. We incorporate these meta-models into the EMF environment for usage in the execution of our prototype components. In addition, we integrate the trace meta-model to visualize (backward transformation) or design (forward transformations) BPMTraceISM diagrams. We built the transformation rule sets in Atlas Transformation Language (ATL), which is provided as an internal EMF plugin.

BPISM2BPMNTrISM takes three files as input: (1) A file with the extension ".bpmn" that must conform to the BPMN2.0 meta-model, (2) two files with the extension ".uml" that must conform to the UML meta-model and contain the use case model and the class diagram. It generates as output a BPMNTraceISM diagram with the extension ".BPMNTraceISM".

BPMNTrISM2BPISM implements the backward transformations, i.e., the transformation rules from a *BPMNTraceISM* diagram into a BPMN and UML models. It takes as input a *BPMNTraceISM* diagram with the extension "*.BPMNTraceISM*". It generates as output three files: (1) A file with *".bpmn*" as extension. This file conforms to the BPMN2.0 meta-model, (2) two files with the extension *".uml*" that include the generated use case model and the class diagram.

6 Case study

We take a common business process model for online purchasing and selling to demonstrate the viability of our traceability method. The model is specified using BPMN2.0 (cf. Figure 8)) [23].

.

This business process begins when a customer selects a product to purchase and adds it to the basket, resulting in the creation of an online purchase order and the transmission of the order to the vendor. The customer has the option to cancel the purchase order before entering their personal information. Otherwise, they must fill in their personal information and submit an online purchase order to the stock management. When an online purchase order is received, the stock manager checks the warehouse for the availability of the ordered items to see if there are enough products to fulfil the order. If not, the restocking procedure is initiated to reorder raw materials and create the ordered products based on the supplier's catalogue. The restocking procedure can be performed as many times as necessary within the same business process instance. An extreme scenario occurs when raw materials are unavailable. If all items are available, sales validate the purchase order, generate an invoice, and begin collecting and packaging products for shipment. When sales receive payment and store the delivered order, the procedure is complete. Purchase order cancellation requests, however, can be made before the purchase order is verified. As a result, sales proceed with purchase order cancellation and a penalty charge to the buyer. In [23], the authors decompose the BPMN model of the case study into nine fragments (F1-F9) (cf. Figure 8)) based on their fragment definition (see [23] for further explanation). By applying their transformation rules, the approach from [23] allows generating the use case diagram and the class diagram from the case study BPMN model, which that is taken as the input model.

The *online purchasing and selling* BPMN model, use case model, and class diagram presented in [23] can be

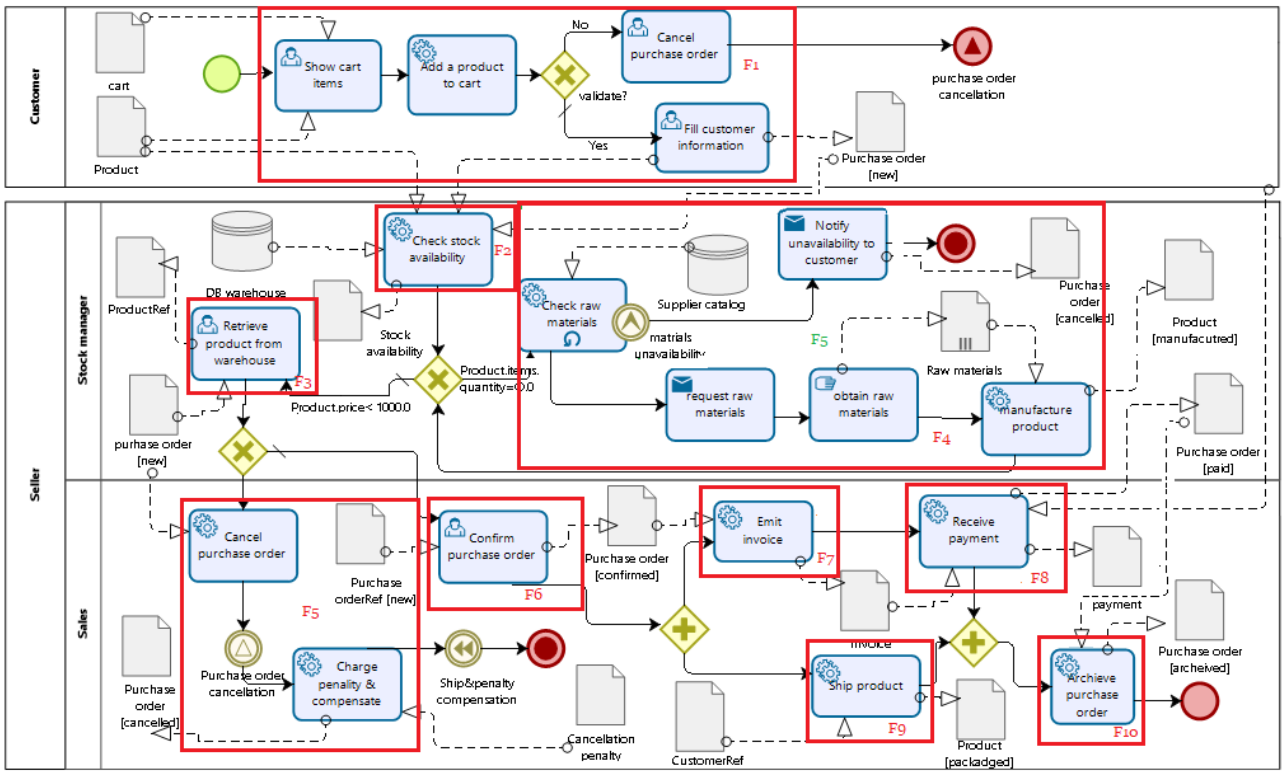


Figure 8: Online purchasing and selling in *BPMNTraceISM* diagram

combined and designed in a single unified model, namely the *BPMNTraceISM* diagram, by using the *BPTraceISM* editor. We would like to highlight that this diagram can be created manually by designers or automatically by running the BPISM2BPMNTrISM component. Figure 9) depicts the BPMNTraceISM diagram. Figure 9) shows how each fragment and its corresponding use case are merged and expressed as a UCsF. For example, we combine Fragment F1 with the use case "Manage preparing purchase order" to form the UCsF "Manage preparing purchase order". Each UCsF displays the traced BPMN elements and the corresponding class diagram elements. For each UCsF, the elements of the BPMN model are represented in the BPMN compartment, while the corresponding class diagram elements are represented in the class diagram compartment.

In Figure 9), these compartments are hidden in the UCsF *Cancel purchase order*, while the BPMN compartment is visible in all the other UCsFs.

In the UCsF *receive payment,* the BPMN compartment contains a service task called *receive payment,* a data object called *invoice,* and a data output called *purchase order [paid]* These elements are the BPMN elements of fragment *F8* Moreover, the class diagram compartment of UCsF *archive purchase order is* displayed and contains the class diagram elements, such as the *classes VarchivePurchaseOrder, CArchivePrchaseOrder, paid, archived, operations, attributes, etc* corresponding to fragment *F10*. Furthermore, an OUActor specifies each actor and the corresponding vacant lane. For example, the actor *Stock manager* and the empty lane *Stock manager* map to the OUActor *Stock manager*.

Assume that business analysts and system designers collaborate on the BPMNTraceISM diagram and update the business and system functionalities accordingly. Suppose they delete the UCsF *Manage preparing purchase order* and the OUActor *Customer* from the *BPMNTraceISM* diagram The UCsF *Manage preparing purchase order* is traced to the elements of fragment *F1*, the UML use case *Manage preparing purchase order,* as well as all class diagram elements derived from *F1* By deleting this UCsF, all its components are also removed from the *BPMNTraceISM* diagram Then, the change involved in the *BPMNTraceISM* diagram is propagated to the source models by executing the *BPMNTrISM2BPISM* tool. The output of this component is a BPMN model without the pool *Customer* or the fragment *F1*, a UML use case model that contains neither the use case *Manage preparing purchase order* nor the actor *Customer,* and a class diagram without elements corresponding to *F1*.

7 Evaluation results

7.1 Comparison with existing approaches

To evaluate the effectiveness of our traceability method, we compare it with existing traceability approaches based on

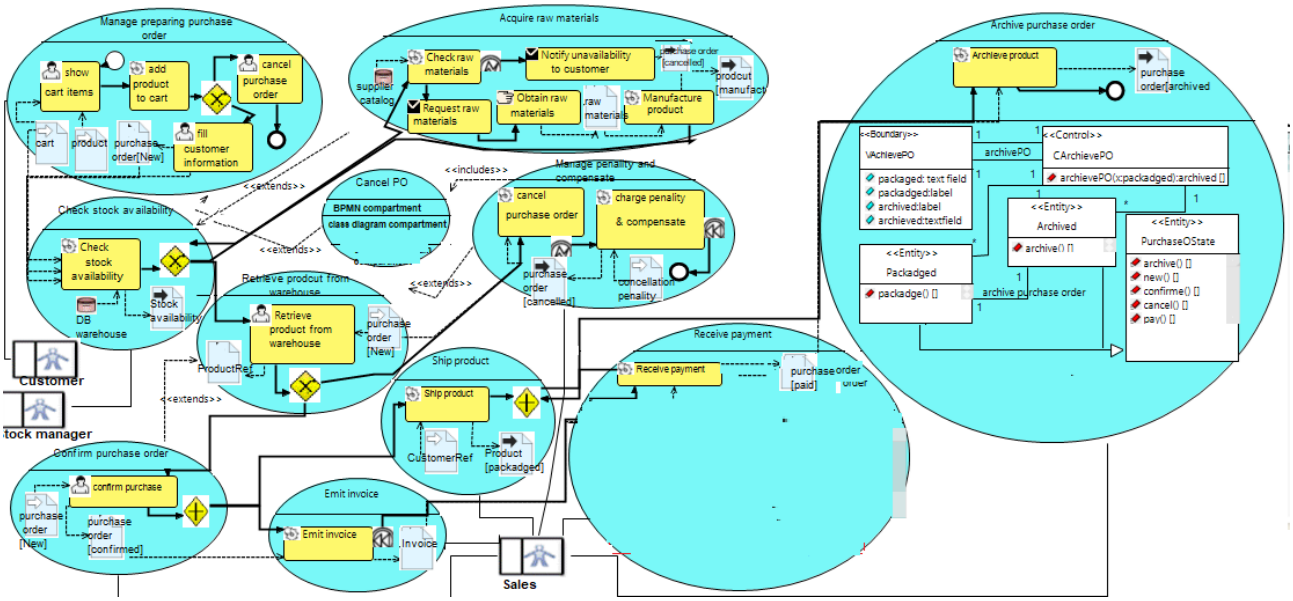


Figure 9: Online purchasing and selling in *BPMNTraceISM* diagram

defined evaluation criteria. These criteria include: (i) the proposal of a traceability approach at both the meta-model and model levels, (ii) explicit representation of relationship types between elements, (iii) graphical notation for trace links, and (iv) the consideration of both business and information system (IS) models.

Table 5 presents the results of this comparison, with rows listing the methods studied and columns representing the evaluation criteria. Cells are color-coded to show the extent to which each criterion is satisfied: dark grey indicates a criterion is not fully addressed, light grey represents partial satisfaction, and ”Y” stands for full satisfaction. Based on this comparison, we conclude that our approach is the only one that meets all evaluation criteria. Specifically, [17] and [20] are the only other works that consider both business and IS modeling, but they fall short in addressing the full range of traceability needs compared to our method. Furthermore, only a few approaches, such as those by [15], [18], and [20], take into account the functional and static views of IS models.

Our method is unique in that it does not require any extensions and works seamlessly with standard UML and BPMN tools, making it more adaptable and accessible. Additionally, our method provides rich software modeling-level artefacts, incorporating both static views (class diagrams) and functional views (use case models). The class diagrams are designed in accordance with the MVC pattern simplifying the prototyping process for developers.

When focusing on traceability, our method stands out because it provides traceability at both the meta-model and model levels, ensuring a unified view of BPMN and UML elements. In contrast, many approaches specify only traceability at the meta-model level, without providing a visual tool for combined model use. Additionally, our method introduces a graphical visualization for traceability links, enabling users to more easily trace and align elements across models. This graphical representation reduces analysis time, simplifies development, and minimizes the risk of misalignment.

Moreover, our method's assessment methodology is more comprehensive than most others in the field, which typically rely on simple case studies. We provide a fully implemented prototype of the transformation approach and an Eclipse plug-in for the traceability process, demonstrating the practicality and feasibility of our contributions through a relevant case study.

7.2 Shortcomings of our contribution

Despite the strengths of our traceability method, there are some limitations that need to be addressed. One notable drawback is that our evaluation was based on a single use case, which may not be sufficient to fully assess the accuracy and robustness of the method. To address this, we are conducting ongoing evaluations using more complex case studies, which will allow us to better validate the method's performance in different contexts. This extended evaluation will help ensure that the method is robust and adaptable across various scenarios, enhancing its overall credibility.

Additionally, our current transformation approach relies on forward and backward transformation rules, which require the recreation of all components, even if they have not been affected by changes. This process can lead to inefficiencies, especially when working with large or complex models. To overcome this issue, we plan to develop incremental transformations that will update only the components directly impacted by changes. This will improve efficiency and minimize unnecessary recalculations, ensur-

Table 5: Comparison of our contribution with approaches based on the external traceability practice

Approach	Construction model				Traceability approach				Assessment methodology
	Business field	software field		Meta level	Model level	Trace links	Graphic notation	BPM and ISM	
		functional	static						
[11]	N	P	P	Y	N	N	N	N	CS
[13]	N	CS		Y	N	N	N	N	CS
[14]	N	N	N	Y	N	N	N	N	tool
[15]	P	P	P	Y	N	N	N	N	
[16]	RM	CS		Y		N	N	N	CS
[17]	BPMN	N	N	Y	N	N	N	N	CS
[18]	p	p	p	Y	Y	N	N	Y	N
[19]	RM	N	N	Y	N	Y	N	N	N
[20]	N	N	CD	Y	N	N	N	N	N
[21]	BPMN	N	N	Y	Y	N	Y	N	T
Our contribution	BPMN2	UC	CCD	Y	Y	Y	Y	Y	CS &T

Legend: Y:Yes N: No CS: case study RM: requirement model T:tool/editor CS: Complex systems CCD:conception class diagram UC: uqe case diagram.

ing faster and more resource-efficient updates.

While we have explored the use of traceability information to keep BPMN and UML models aligned, the analysis process is still manual. As a result, we aim to improve this by investigating the development of heuristics that could automatically detect modifications in the source models and suggest necessary adjustments to the corresponding elements. These heuristics would dynamically support developers, making the process of maintaining alignment between the models more efficient and less error-prone.

To address the limitations mentioned above, we propose a comprehensive roadmap for future improvements. This will involve extending the evaluation process with more complex case studies, enhancing the transformation approach to support incremental updates, and automating diagram analysis. In addition, the implementation of heuristic-based tools will enable the automatic detection of changes across models, improving traceability and ensuring consistency without requiring manual intervention. These advancements will significantly strengthen the method's capabilities, making it more robust and easier to apply in practical scenarios.

8 Conclusion

The work conducted in this paper fits within the context of model-based development of ISMs, their alignment and their

coevolution with BPMs. Indeed, we have used integration and model transformation methodologies to define a

traceability method oriented towards the development of (meta) model-based solutions, purposely influenced by the Object Management Group (OMG) specifications. Particular attention is paid to the BPMN and UML use case and class diagram models. Our traceability method acts at the meta-model and the model levels. Hence, (1) we first defined a unified trace meta-model that includes all the BPMN and the UML elements (use case and class diagram) and traceability links between interrelated elements. (2) Then, we defined integrated model conforms to the proposed trace meta-model. We defined it as a new diagram named *BPMNTraceISM* (BPMN Traces Information System Models). This diagram serves many purposes: it promotes the collaborative between business and software designers and allows them to work together using one single unified model. The joint representation of both BPMs and ISMs elements enables users to drill down and easily trace any business artefact to its corresponding software artefacts. (3) Finally, we defined a set of bidirectional model transformation rules between the BPMN and UML models, as well as the BPMNTraceISM diagram.

The rules are useful when a change propagation-based co-evolution is required to synchronize models after changes. To prove the feasibility of our traceability method in practice, we developed a modeling tool in the form of a plugin that can be integrated into the Eclipse platform. This tool is named *BPTraceISM* (Business Process model Trace with Information System Models) and allows designing and handling *BPMNTraceISM* diagrams in accordance with the proposed integrated trace meta-model. Additionally, we specified the set of bidirectional transformation rules using the ATL language and we implemented them as components of the BPM2ISM prototype. Furthermore, we applied the proposed approaches to a typical case study.

In future research, we look forward to optimizing our editor to support traceability and synchronization between BPMN models and other UML diagrams.

References

[1] Bouzidi, A., Haddar, N., Haddar, K. (2019). *Traceability and Synchronization Between BPMN and UML Use Case Models*, Ingénierie des Systèmes d'Information, Vol. 24, No. 2, pp. 215-228. https://DOI.org/10.18280/isi.240214.

[2] OMG UML Specification, O. A. (2017). *OMG Unified Modeling Language (OMG UML), Superstructure, V2*, Object Management Group, Vol. 70.

[3] OMG BPMN Specification. *Business Process Model and Notation* Available at: http://www.bpmn.org/. Accessed: 2023-01-31.

[4] Driss, M., Aljehani, A., Boulila, W., Ghandorh, H., Al-Sarem, M. (2020). *Servicing your requirements: An FCA and RCA-driven approach for semantic web services composition*, IEEE Access, Vol. 8, pp. 59326-59339. 10.1109/ACCESS.2020.2982592.

[5] Ghiffari, K. A., Fariqi, H., Rahmatullah, M. D., Zulfikarsyah, M. R., Evendi, M. R. S., Fathoni, T. A., Raharjana, I. K. (2023). *BPMN2 user story: Web application for generating user stories from BPMN*, In AIP Conference Proceedings, AIP Publishing LLC, Vol. 2554, No. 1, pp. 040003. https://DOI.org/10.1063/5.0103685.

[6] Raharjana, I. K., Aprillya, V., Zaman, B., Justitia, A., Fauzi, S. S. M. (2021). *Enhancing software feature extraction results using sentiment analysis to aid requirements reuse*, Computers, Vol. 10, No. 3, pp. 36. https://DOI.org/10.3390/computers10030036.

[7] Khlif, W., Elleuch, N., Alotabi, E., Ben-Abdallah, H. (2018). *Designing BP-IS Aligned Models: An MDA-based Transformation Methodology*. 10.5220/0006704302580266.

[8] Kharmoum, N., Retal, S., Rhazali, Y., Ziti, S., Omary, F. (2021). *A Disciplined Method to Generate UML2 Communication Diagrams Automatically From the Business Value Model, In Advancements in Model-Driven Architecture in Software Engineering*, IGI Global, pp. 218-237. 10.4018/978-1-7998-3661-2.ch012.

[9] Rahmoune, Y., Chaoui, A. (2022). *Automatic Bridge Between BPMN Models and UML Activity Diagrams Based on Graph Transformation*, Computer Science, Vol. 23, No. 3. 10.7494/csci.2022.23.3.4356.

[10] Ivanchikj, A., Serbout, S., Pautasso, C. (2020). *From Text to Visual BPMN Process Models: Design and Evaluation*, In Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 229-239. https://DOI.org/10.1145/3365438.3410990.

[11] Mills, C., Escobar-Avila, J., Haiduc, S. (2018). *Automatic Traceability Maintenance via Machine Learning Classification*, In 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 369-380. 10.1109/ICSME.2018.00045.

[12] Al-Hroob, A., Imam, A. T., Al-Heisa, R. (2018). *The Use of Artificial Neural Networks for Extracting Actions and Actors from Requirements Documents*, Information and Software Technology, Vol. 101, pp. 1-15. https://DOI.org/10.1016/j.infsof.2018.04.010.

[13] Min, H. S. (2016). *Traceability Guideline for Software Requirements and UML Design*, In International Journal of Software Engineering and Knowledge Engineering, Vol. 26, No. 1, pp. 87-113. https://DOI.org/10.1142/S0218194016500054.

[14] Eyl, M., Reichmann, C., Müller-Glaser, K. (2017). *Traceability in a Fine-Grained Software Configuration Management System*, In Software Quality: Complexity and Challenges of Software Engineering in Emerging Technologies, 9th International Conference, SWQD 2017, Vienna, Austria, January 17-20, 2017, Springer International Publishing, pp. 15-29. 10.1007/978-3-319-49421-0_2.

[15] Khelladi, D. E., Kretschmer, R., Egyed, A. (2018). *Change Propagation-Based and Composition-Based Co-Evolution of Transformations with Evolving Meta-Models*, In Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 404-414. https://DOI.org/10.1145/3239372.3239380.

[16] de Carvalho, E. A., Gomes, J. O., Jatobá, A., da Silva, M. F., de Carvalho, P. V. R. (2021). *Employing Resilience Engineering in Eliciting Software Requirements for Complex Systems: Experiments with the Functional Resonance Analysis Method (FRAM)*, Cognition, Technology and Work, Vol. 23, pp. 65-83. https://DOI.org/10.1007/s10111-019-00620-0.

[17] Lopez-Arredondo, L. P., Perez, C. B., Villavicencio-Navarro, J., Mercado, K. E., Encinas, M., Inzunza-Mejia, P. (2020). *Reengineering of the Software Development Process in a Technology Services Company*, Business Process Management Journal, Vol. 26, No. 2, pp. 655-674. https://DOI.org/10.1108/BPMJ-06-2018-0155.

[18] Moreira, J. R. P., Maciel, R. S. P. (2017). *Towards a Models Traceability and Synchronization Approach of an Enterprise Architecture*, In SEKE, pp. 24-29. 10.1109/CBI.2019.00028.

[19] Guo, J., Cheng, J., Cleland-Huang, J. (2017). *Semantically Enhanced Software Traceability Using Deep*

Learning Techniques, In 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pp. 3-14. 10.1109/ICSE.2017.9.

[20] Swathine, K., Sumathi, N., Nadu, T. (2017). *Study on Requirement Engineering and Traceability Techniques in Software Artefacts*, In International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, No. 1. 10.1109/ICSRS.2017.8272863.

[21] Pavalkis, S., Nemuraite, L., Milevičienė, E. (2011). *Towards Traceability Meta-Model for Business Process Modeling Notation*, In Conference on e-Business, e-Services and e-Society, Springer, Berlin, Heidelberg, pp. 177-188. 10.1007/978-3-642-27260-8_14.

[22] Bouzidi, A., Haddar, N., Abdallah, M. B., Haddar, K. (2018). *Alignment of Business Processes and Requirements Through Model Integration*, In 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), pp. 1-8, IEEE. 10.1109/AICCSA.2018.8612870.

[23] Bouzidi, A., Haddar, N. Z., Ben-Abdallah, M., Haddar, K. (2020). *Toward the Alignment and Traceability Between Business Process and Software Models*, In ICEIS, Vol. 23. 10.5220/0009004607010708.

A Review of Machine Learning Techniques in the Medical Domain

Enas M.F. El Houby
Department of Systems and Information, National Research Centre, Giza, Egypt
E-mail: em.fahmy@nrc.sci.eg, enas_mfahmy@yahoo.com

Keywords: active learning, curriculum learning, deep learning, federated learning, medical, transfer learning

Received: August 19, 2024

We have witnessed a rapid exponential growth of all types of data in all domains specifically in the medical domain. The utilization of machine learning techniques has made significant strides across various domains, with deep learning achieving notable success in recent years. Lately, deep learning has gained increasing attention in the medical field. While deep learning excels at automatically learning discriminative features from raw data, it is still challenging to achieve high performance without a huge amount of data and some handcrafted steps. To address these challenges, deep learning has been incorporated with other new trends and domain knowledge to enhance deep learning's capabilities and improve performance covering the ever-growing needs. Transfer learning utilizes knowledge from natural images, curriculum learning integrates domain-specific knowledge, active learning selects the most informative samples to reduce reliance on labeled data, and federated learning enables collaborative training across organizations while ensuring data privacy. In this review paper, these new trends incorporated with deep learning have been investigated and presented as applications in the medical domain by investigating articles that have applied these trends and published in highly reputable journals in the Science Direct database in recent years.

Povzetek: V pregledni študiji so predstavljeni sodobni trendi strojnega učenja v medicini, kot so transferno, aktivno in federativno učenje na podlagi učnih načrtov, ki v kombinaciji z globokim učenjem izboljšujejo diagnostiko, personalizacijo zdravljenja in varnost podatkov.

1 Introduction

Recently, we have witnessed the growth of all types of data in all domains. Medical data specifically has grown dramatically in the last few years due to the exponential increase of knowledge in the medical domain. Medical data can be found in various forms such as clinical and biomedical data. Biomedical data contains data related to genomics, drug discovery, and biomedicine. Clinical data contains patient records such as medical patients' history, laboratory investigation, and image data from magnetic resonance imaging (MRI), ultrasound (US), X-rays, and computerized tomography (CT) scans. Clinical data exists in 2 forms, structured and unstructured. The structured format includes the disease history and living habits of the patients. While unstructured clinical data such as doctor's investigation records and the conversation between the doctor and patients [1-3]. Therefore, this rapidly growing volume of medical data requires advanced methods for analysis.

Applying artificial intelligence (AI) in the medical domain comprises a promising technology for different healthcare providers. These technologies, particularly data mining, help extract hidden patterns and insights from large datasets using machine learning techniques (MLTs). Traditional MLT includes Artificial Neural Networks (ANN), Decision Trees (DT), Support Vector Machines (SVM), and many other techniques. Machine learning techniques are usually categorized into supervised, unsupervised, semi-supervised, and reinforcement learning. In supervised learning, the labeled data is available, therefore, the model can be trained using this manually tagged data to extract patterns. When there is no labeled training data, unsupervised techniques are employed. It groups similar entities in the same cluster. Each cluster demonstrates a relation between these grouped entities. Semi-supervised depends on a set of hand-crafted extraction patterns and a few tagged instances as initial seeds of the target relation to start the training. The training output is used as the training input for the following generation. The process of learning is repeated for many generations. Reinforcement learning is based on evaluative feedback, so, it can automatically perform goal-oriented learning and process decision-making problems [4, 5].

Deep learning is an advanced form of artificial neural network (ANN), with a larger number of layers than a conventional ANN model to automatically learn the features from the data which makes more refined predictions possible. In numerous recent medical image classification tasks, convolutional neural networks (CNNs), which are a kind of deep learning network particularized in image analysis, were utilized and achieved high performance. The success of CNN in the classification of medical images has motivated researchers to utilize pre-trained models in building new ones. These

high-performing CNN pre-trained models have been utilized for different image classification tasks by employing the transfer learning (TL) approach. Pre-trained CNN models utilize features that were learned from a specific domain to fine-tune any other data. They can be utilized as-is to classify new images or to extract features using the output from the layer previous to the output layer and introduce it to another classifier [6].

However, many challenges face the application of machine learning techniques generally and in the medical domain specifically such as I) **the limitations of available datasets** for training the models, that is because collecting and labeling the data is a labor-intensive and expensive task, especially in the case of medical images data such as Ultrasonic imaging (US), CT, MRI. The annotation of data includes the segmentation annotations of abnormality regions and classification labels such as (normal, benign, and malignant). Also, that limitation may result from the scarcity of some diseases with which it is difficult to obtain enough positive cases. II) **The low quality of some data** is another major challenge, where some of the data can be found unlabeled, inconsistent, inaccurate, or in an unstructured format—such as handwritten notes, radiology reports, and conversations between doctors and patients—which are difficult for machine learning algorithms to process effectively. In the case of medical image modalities, there may be variations in image resolution and quality. III) **The shortage of explanations** of pathological basis such as the diagnosis reasons, where the techniques depend only on the differences between the normal and patient cases. For healthcare professionals to trust and act on ML-generated results, it is essential to understand how these models arrive at their predictions. IV) **Ethical and regulatory concerns** play a crucial role, where the healthcare industry is tightly regulated, and machine learning models must comply with stringent standards to ensure patient privacy, data security, and model safety. Furthermore, any biases in the data could lead to unequal or unfair treatment recommendations, making fairness an ongoing concern in the application of ML in healthcare [7-10].

Despite these challenges, machine learning presents a wealth of opportunities that can significantly improve healthcare outcomes such as I) **diagnostic accuracy and speed** where ML algorithms, particularly deep learning models, have demonstrated remarkable success in automating and enhancing diagnostic processes, especially in medical imaging. For instance, ML models can analyze radiographs, MRI scans, and other images to identify abnormalities such as tumors or lesions with a level of precision that often rivals or exceeds that of human experts. This capability can lead to earlier detection, which is critical for improving patient outcomes, particularly in cancer and cardiovascular diseases. II) **personalized medicine** by analyzing large datasets, including patient demographics, genetic information, and medical history, machine learning can help tailor treatments to individual patients, optimizing therapeutic interventions based on their unique characteristics. III) **Predictive analytics** is another powerful opportunity that ML offers. By analyzing trends in patient data, machine learning models can predict disease progression, forecast complications in chronic conditions, and identify high-risk patients who may benefit from earlier interventions. IV) **Automation** is another key opportunity in healthcare, with ML models capable of automating routine tasks such as image analysis, patient triaging, and administrative work. This allows healthcare providers to focus more on direct patient care, improving overall efficiency. V) **Drug discovery** by identifying promising drug candidates and predicting their behavior in the human body, which can reduce the time and cost associated with bringing new medications to market [7-10].

In response to the mentioned challenges, recent research has shifted towards using advanced techniques such as deep learning with some incorporated techniques and domain knowledge like transfer learning which provides deep learning with information from natural images. Curriculum learning integrates domain knowledge through training patterns of the processed task. Active learning explores the most informative samples and retrieves them from an unlabeled pool to fulfill better performance with less labeled data. Federated learning allows many organizations to collaborate on deep learning without sharing clients' data or devices which provides efficient data access and security and an improvement of the learning model utilizing a large decentralizing dataset. The purpose of this research is to illustrate the new trends of machine learning in the medical domain. The selected articles that are reviewed show these new trends in the medical domain using different medical dataset types including medical images, tabular datasets, genes, etc. in different tasks. The remainder of this research is organized as follows: Section 2 illustrates the different types of medical data. Section 3 presents some data preprocessing steps. Section 4 presents the new trends of MLTs. Section 5 describes the search methodology for articles that apply the mentioned new trends of MLTs in the medical domain. Section 6 presents some of the applications of new trends of MLTs in the medical domain. Section 7 presents the conclusion and some of the recommended points for future work.

2 Types of medical data

Medical data can be found in different forms such as arrays of numerical data, images, sequences of DNA, amino acids, ...etc. For developing any ML model, the data is split into three parts which are training, validation, and testing. The training part is used to learn to tune the parameters of the model. The validation part is used to stop overfitting, and the test part is used to assess the performance of the model. In the next subsections, a brief overview of different medical data forms will be presented.

2.1 Numerical data

Different diseases' related data are found as an array of lab tests which is numerical data. These numerical datasets can be used to manage the related diseases such as the

datasets available on the UCI machine learning repository [11]. Most numerical data are available in table form such as Excel sheets or database tables where rows represent samples from patients and columns represent different features that describe the intended diseases or vice versa. A huge number of numerical datasets are available such as the patient demographics of some diseases like COVID-19, and the lab results for different diseases such as thyroid, heart disease, dermatology, cancer, etc.

2.2 Microarray gene expression data

Microarray techniques provide a platform for measuring the expression levels of thousands of genes in various conditions. It is composed of a small glass slide or membrane that contains samples of many genes arranged in a regular pattern. It is used to find genes associated with specific diseases by analyzing and finding the differences between two mRNA sets, one set is from normal cells and the other set includes cells from pathological tissues such as cancer cells. Microarray data contains a lot of redundant genes, and many genes include inappropriate information for the accurate classification of diseases. Thus, the analysis of the large amount of data generated by this technology is not an easy task for biologists [12]. Figure 1 shows a cDNA microarray spotted on a glass surface. While, in Figure 2 the general structure of the microarray is illustrated, which is represented as an array of numerical values. Cancer gene expression datasets for leukemia, lung, prostate, etc. can be found in [13].

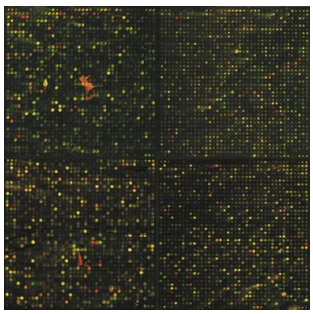


Figure 1: cDNA microarray spotted on a glass surface. https://www.cell.com/fulltext/S0960-9822%2898%2970103-4

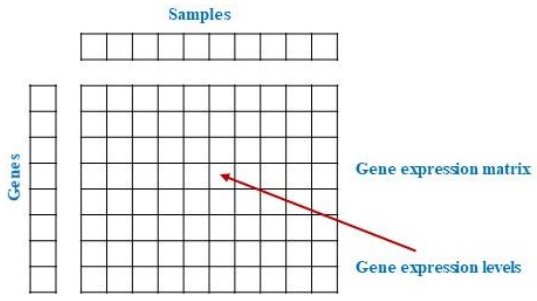


Figure 2: General structure of microarray.

2.3 Image modalities

Information obtained from medical imaging modalities is clinically beneficial in many applications like computer-aided detection, diagnosis, and treatment planning. Many imaging modalities can be used to check abnormalities in different body organs. They include radiation such as CT, X-rays, US, and MRI. They are categorized according to the method of producing images. They help radiologists to recognize abnormal regions. The interpretation of different image modalities needs expertise, where it is operator dependent. Therefore, the process of reading image modalities is exhausting, costly, and prone to error.

Ultrasound (US) is a suitable modality for tumor detection. It can estimate the size of the tumor and distinguish abnormalities. Its capability of detecting contra-lateral malignant lesions is limited [14].

Magnetic resonance imaging (MRI) produces images relevant to the displaying of hydrogen atoms to radio waves and magnetic fields. MRI images are valuable as they present physiology and anatomy. It images the target organ and prepares it as thin slices; moreover, it provides information about the vascularity of the tissue [15].

Computed tomography (CT) scanners display better image clarity using multiple X-ray sources and detectors [15]. Radiation X-ray generated images are 2-dimensional images. Fluoroscopy units show real-time moving images produced by X-ray exposure. Angiography is a widespread usage of fluoroscopy, imaging blood flow in vessels [15].

Digital Mammography (DM) is an X-ray imaging that is specialized for breast tissue. DM is the most common and most important screening method in clinical practice. It can detect tumors before they develop further and become easily detected and felt by the physician [16].

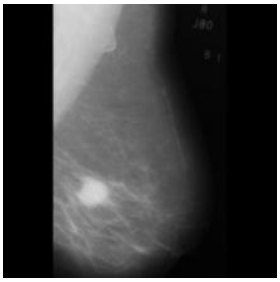
Microscopic images are the images that are captured by the microscope to enlarge small scanned objects and extract fine details that cannot be obtained otherwise [17]. Figure 3 shows samples of different image modalities for different body organs.

2.4 DNA and protein sequences

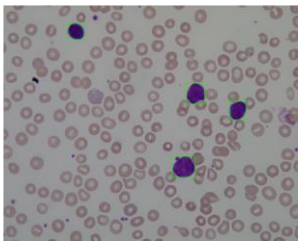
The fast growth of sequencing resulted in huge numbers of DNA and protein sequences. Sequences can be used to predict diseases associated with a given either DNA or protein sequences. DNA is a long polymer chain of units named nucleotides; it exists in a double helical shape as shown in Figure 4. There are 4 types of nucleotides which are A (adenine), C (cytosine), G (guanine), and T (thymine), they are considered the alphabet of DNA. They are arranged into sequences of 3-letter called codons. A double-stranded helical structure of DNA would be complementary, where “G” is chemically combined with “C”, and "A" with "T" within the replication of DNA [18].



(a) X-ray of Lung [19]



(b) DM of breast [20]



(c) Microscopic blood image [21]

Figure 3: Samples of different image modalities for different body organs.

Amino acids are linked into linear chains to produce proteins. The properties of proteins are defined by the composition of their amino acids. The triplets of consecutive DNA nucleotides which are called codons are responsible for the forming amino acid sequence in a protein. There are $4^3 = 64$ various codons formed from the 4-letters [22], which is more than 3 times larger than the number of amino acids which is 20 amino acids, 3 of which represent stop codons and one is a start codon. While the remaining codons are responsible for generating the 20 amino acids. So, it is possible that more than one codon maps the same amino acid [18]. Figure 5 shows the transcription of DNA sequence into molecules of mRNA; then, the translation of the transcribed mRNA into the associated chain of amino acid sequence, which later fold into fully functional proteins.

Single nucleotide polymorphisms (SNPs) are the most common human genetic variations as mutations or insertions/deletions (indels). If SNPs have changed the codon triplets without changing the encoded amino acid, it is synonymous (sSNPs) while the gene is not mutated. Otherwise, it is non-synonymous (nsSNPs), as it changes the codon while the encoded amino acid is changed into various amino acids which are called missense mutations which are the reason for many diseases [23, 24]. Figure 6 shows single nucleotide polymorphisms (SNPs).

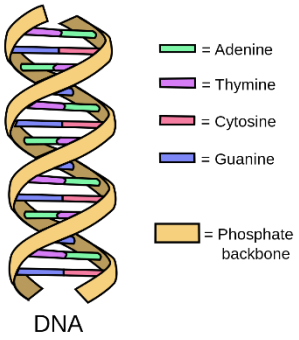


Figure 4: Chain of DNA sequence.
http://acer.disl.org/news/2016/08/17/tool-talk-gene-sequencing/

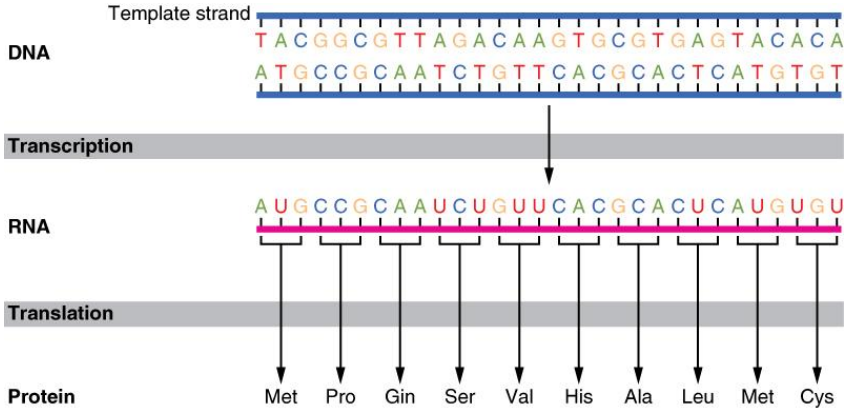


Figure 5: The process of translation from DNA sequence to the associated amino acid sequence.
https://courses.lumenlearning.com/suny-ap1/chapter/3-4-protein-synthesis/

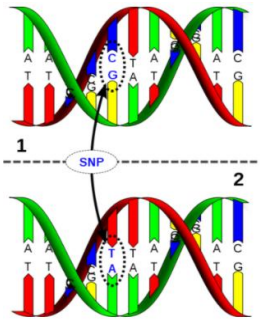


Figure 6: Single nucleotide polymorphisms (SNPs).
https://isogg.org/wiki/Single-nucleotide_polymorphism

3 Data preprocessing

The knowledge discovery includes 3 main phases which are the preprocessing phase, the data mining phase, and the post-processing phase. Data pre-processing is a crucial phase of knowledge discovery to build an accurate machine learning model. In the preprocessing phase, a set of data preprocessing steps are performed (cleaning the data from the noise, handling missed values, merging appropriate data from different databases, normalizing the data, extracting features, and selecting the most informative features) to prepare the data for data mining phase. Datasets can also be small therefore the relevant features have not been captured and thus data augmentation is performed by applying different data augmentation techniques. Data mining, which is the core phase in knowledge discovery, is performed by applying MLTs. The preprocessing facilitates the application of the MLTs to extract important patterns or correlations. In the post-processing phase, the discovered knowledge is refined and improved then interpreted into meaningful knowledge for the user's presentation [25].

Feature selection is a key difference preprocessing step that should be highlighted when comparing deep learning with the traditional MLTs so it will be tackled in more detail in the next subsection.

3.1 Features selection

Feature selection is the process of finding the optimal feature subset that is strongly distinguishing among different classes. The purpose of this process is the reduction of the dataset and the elimination of redundant and irrelevant features that impact the classification process negatively. Feature selection is a combinatorial optimization problem its aim is to select the feature subset with the least number of features that achieves the highest possible classification accuracy. It is one of the data preprocessing for pattern recognition and data mining specifically when working on high-dimensional datasets [26, 27].

Feature selection has 2 main approaches: the filter and the wrapper. In the filter approach, the feature selection is based on statistical individual feature ranking. It is easily implemented but eliminates the interaction among features and does not rely on the applied ML algorithm to the selected features. Whereas, for the wrapper approach the feature selection depends on the outcome of the ML algorithm to decide how favorable the features subset is. Candidate solutions of feature subsets are iteratively generated and their characteristics are assessed by the applied ML algorithm [28].

The wrapper-based feature selection approach evaluates the quality of feature subsets using the learning algorithm. Thus, it can determine and discard irrelevant and redundant features effectively. As the learning algorithm is frequently used in the search process, high computational time is required, especially when the datasets are large. On the other hand, the hybrid methods aim to utilize the advantages of both approaches; the efficiency of computation of the filter approach and the high performance of the wrapper approach [29].

Feature selection algorithms based on heuristic search methods are needed as the computation of a huge number of features is not feasible. Many meta-heuristics approaches have been used for feature selection. From these algorithms are the nature inspired algorithms such as genetic algorithm (GA), firefly [30, 31] and ant colony optimization (ACO) [32, 33].

4 New trends of machine learning techniques

After data preprocessing, various Machine Learning Techniques (MLTs) are applied to uncover hidden patterns and correlations in the data. As mentioned earlier, disease-related data, often represented through numerical lab tests, microarray data, medical imaging, and genetic sequences, can be processed to predict disease presence or other related tasks. Traditional MLTs like Support Vector Machines (SVM), Decision Trees (DT), and K-Nearest Neighbors (KNN) are effective across these data types for decision-making. However, Artificial Neural Networks (ANNs), which mimic the brain's neural structure, are increasingly used for more complex tasks and in various domains, especially the medical domain. ANNs consist of input, hidden, and output layers, and are trained through techniques like backpropagation. Their success across domains, especially healthcare, has led to the development of Deep Learning (DL), a more advanced form of ANN. This section explores the role of deep learning in modern machine learning.

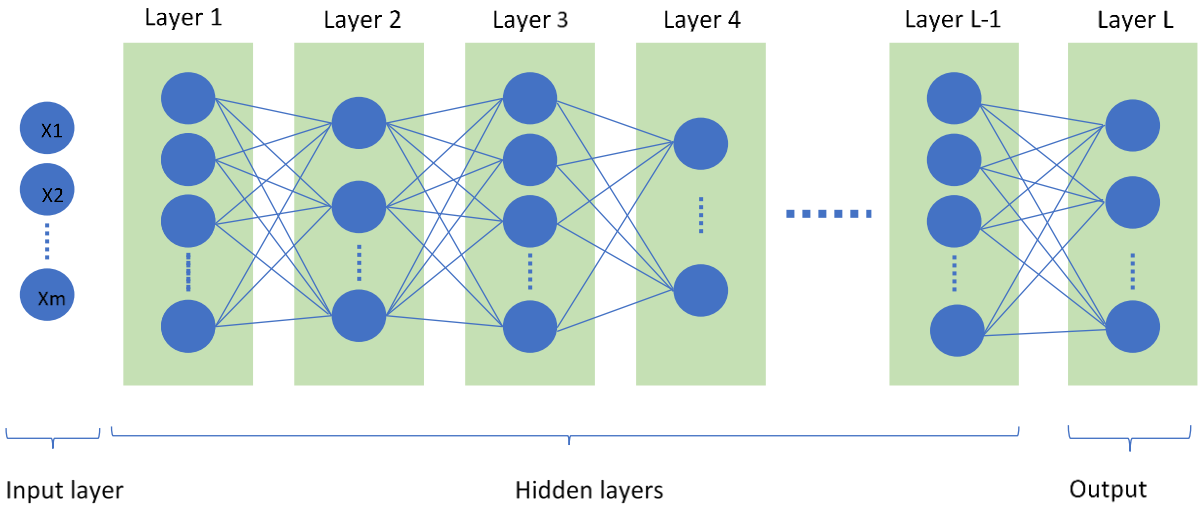


Figure 7: The general structure of Deep Neural Network.

4.1 Deep learning (DL)

Deep Learning (DL) models have gained prominence due to their ability to automatically extract complex patterns from data, eliminating the need for manual feature engineering. However, DL models require large datasets, making them particularly suited for high-dimensional data, such as in medical fields, where they can uncover intricate structures through multiple intermediate layers. The depth of a DL model—referring to the number of hidden layers—enables it to learn complex mappings between input and output. Unlike shallow networks, which struggle with intricate data patterns, deeper networks excel at learning these relationships [34, 35]. Figure 7 shows the general structure of Deep Neural Networks (DNNs).

There are several deep learning algorithms such as Convolution Neural Network (CNN), radial basis function networks, deep belief networks, autoencoders, and Recurrent Neural Network (RNN) [35, 36].

Deep learning depends on hyperparameters such as activation function, learning rate, batch size, number of epochs, optimizer, dropout rate, etc. Different deep learning algorithms, like RNNs and CNNs, also have additional specific hyperparameters. Adjusting these hyperparameters is critical, as their values significantly affect the model's behavior. Finding the optimal combination of hyperparameters can be an exhaustive task, requiring substantial computational resources and time [37, 38].

The performance of a DL model heavily depends on the selection of these hyperparameters, particularly in complex domains like medical data analysis. Medical data often have high dimensionality, noise, and imbalanced class distributions, making hyperparameter optimization crucial to enhancing model performance. Careful selection improves robustness and generalizability, ensuring reliability in real-world clinical settings. While methods like grid search and random search are widely used, more advanced techniques, such as Bayesian optimization, offer significant advantages [37, 38].

Common techniques for hyperparameter optimization:

1. **Grid search**: Exhaustively searches across all possible hyperparameter combinations. While it guarantees to find the best parameter set within the grid, it can be computationally expensive, especially for models with many hyperparameters.

2. **Random search**: A more computationally efficient approach, randomly selecting combinations of hyperparameters from specified ranges. It often achieves comparable or better results than grid search in fewer trials.

3. **Bayesian optimization:** An advanced method that builds a probabilistic model of the objective function. It predicts the best hyperparameters based on past performance, guiding the search toward the most promising regions with fewer trials. Libraries like Optuna and Hyperopt can implement Bayesian optimization efficiently.

For example, CNN can be used to classify medical images like X-rays or MRI scans. Random search can explore different values for hyperparameters (e.g., learning rate, batch size, number of layers). Alternatively, Bayesian optimization can be used for a more efficient search, predicting the most promising hyperparameters configurations based on prior evaluations. By optimizing the model's parameters using these methods, we can improve classification accuracy, reduce overfitting, and ensure the model performs well on unseen medical data.

Advantages and Limitations of DNNs for Medical Data

Advantages:

- **Versatility:** DNNs can be adapted to work with various data types, including structured clinical data (e.g., patient demographics, lab results), unstructured data (e.g., free-text medical records), and image data.
- **Feature Learning:** DNNs can automatically learn relevant features from the data, making them more flexible than traditional machine learning algorithms that rely on feature engineering.

Limitations:

- **Training complexity:** Training deep neural networks can be computationally expensive and time-consuming. Additionally, DNNs require large datasets to avoid overfitting.
- **Overfitting:** If not carefully tuned, DNNs can overfit to small or imbalanced datasets, a common issue in medical data where datasets may not be as large or diverse as needed for training.

Use Cases in Medicine: DNNs have been applied to a variety of tasks in medicine, including predicting patient outcomes, disease progression modeling, and disease classification from images and clinical data.

CNNs and RNNs are two of the most common and promising deep learning algorithms used in medical applications. These algorithms have demonstrated success in a variety of tasks, such as medical image classification and time-series data analysis. Further details on these algorithms will be discussed in the next subsections.

4.1.1 Recurrent neural networks (RNN)

Recurrent neural networks (RNNs) are neural networks that contain memories that can capture the stored information in the prior element of the given sequence. Therefore, RNN is suitable for processing sequential data types such as the diagnostic history of patients, DNA and protein sequences, etc., where the information is remembered through the network. RNN is called recurrent because it executes the same task for each element of the input sequence while its output is based on the prior computations (memory). Thus, the decision of recurrent net at time t-1 affects the decision that will be taken later at time t. Therefore, RNN has 2 sources of input, the recent past and the present, which are combined to define the response to new data. Figure 8 shows the architecture of the RNN in which a set of input x values are mapped into a sequence of output o values. A loss L measures the difference between the expected output o and the actual output y [35].

Where x, h, o, L, y symbolizes input, hidden state, output, loss, and target value. A weight matrix U defines the input to hidden connection, a weight matrix W defines the hidden-to-hidden connection, and a weight matrix V defines the hidden-to-output connection. Then, from time step t = 1 through time step t = n, the following equations are used:

$$a_t = b + Wh_{t-1} + Ux_t \quad (1)$$

$$h_t = \tanh(a_t) \quad (2)$$

$$o_t = \mathrm{c} + Vh_t \quad (3)$$

$$\widehat{y_t} = \mathrm{SoftMax}(O_t) \quad (4)$$

The forwarding propagation of RNN is defined by the preceding equations, where b and c are the bias vectors, while tanh and SoftMax are the activation functions. To update the weight matrices U, V, and W, we compute the gradient of the loss function for each weight matrix. Gradient computation requires both forward and backward propagation of the network. Any loss function can be used depending on the goal. At each time step, the sum of all losses is the total loss for a particular sequence of x values.

However, traditional RNNs suffer from gradient exploding and gradient vanishing issues, making them unsuitable for long-term dependencies. On the other hand, long short-term memory (LSTM) is effective in capturing long-term time dependence. LSTM networks address this by introducing gating mechanisms that control the memory flow, allowing for better long-term sequence learning. Gated Recurrent Units (GRUs) offer a simplified version of LSTMs with similar benefits but fewer parameters.

Advantages and Limitations of RNNs for medical data

Advantages:

- **Sequential data processing:** RNNs can handle different types of sequential data, including time series and text, where the past medical history or a series of clinical events influence future outcomes.
- **Memory of past inputs**: RNNs can remember information from previous time steps in the sequence, allowing them to capture temporal dependencies in data. This is particularly useful for tracking disease progression over time or analyzing patient histories.

Limitations:

- **Training difficulties**: RNNs are prone to the vanishing gradient problem, especially in long sequences, making them harder to train effectively.
- **Data complexity:** RNNs are best suited for data where the relationship between input and output is sequential. For static data like images or tabular data, CNNs or DNNs might be more appropriate.

- **Resource intensive**: Training RNNs, especially on long sequences, can be computationally expensive.

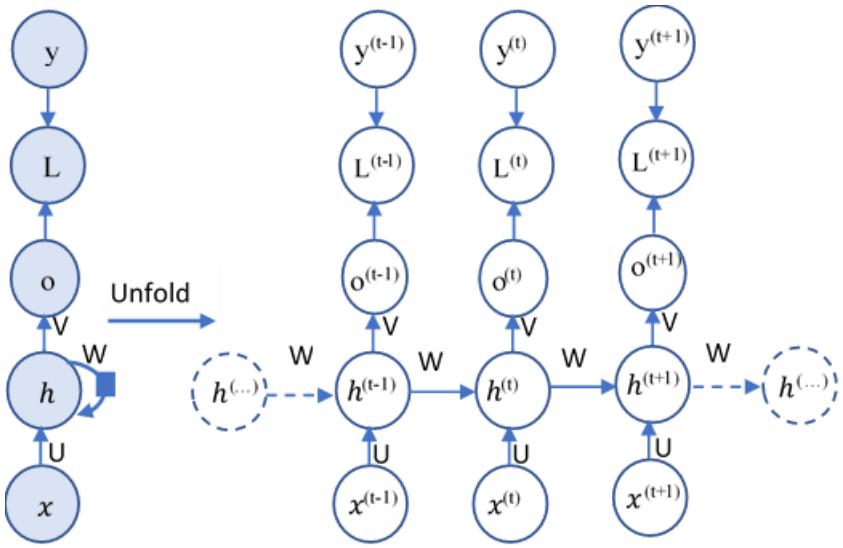


Figure 8: The architecture of recurrent neural network.

Use cases in medicine: RNNs (and their variants like LSTMs) are commonly used in medical applications such as gene sequence classification, predicting disease progression over time, and analyzing time-series medical signals (e.g., ECG readings) [35].

4.1.2 Convolution Neural Network

CNNs are a type of deep learning network specialized for image analysis. Unlike traditional MLTs that rely on manual feature extraction, CNNs can automatically learn hierarchical features from raw image data. This is especially useful in the medical field, where CNNs are applied to analyze medical images for tasks like disease detection and classification [36, 39].

It contains an input, an output and many hidden layers which represent convolutional networks. Convolutional network includes three types of layers: convolutional, activation, and pooling. The convolutional layers apply filters to detect features (edges, textures, etc.). As the image proceeds through layers, the filters can detect more sophisticated features. The activation function like Rectified Linear Unit (ReLU) follows the convolution layer to control the output, it introduces non-linearity. Pooling layers reduce the dimensionality of the data, making the model more computationally efficient and less sensitive to minor positional changes in the features. The final layer is fully connected, producing predictions for classification tasks. The overall number of network parameters is defined by the number of layers, the number of neurons in each layer, and the connection between neurons. The weights should be tuned through the training phase to achieve good performance [40].

convnet processes the image (I) using a matrix of weights called filters which can recognize certain features at specific positions. At a specific layer l, the feature map at position (i,j) is defined as h_{ij}^{l}, the bias as b^{l}, and the weight as W^{l}. The feature map can be expressed as follows:

$$h_{ij}^{l} = \mathrm{ReLU}((W^{l} * I)_{ij} + b^{l}) \quad (5)$$

Where ReLU is activation function which controls the output. The basic structure of the CNN is shown in Figure 9.

Advantages and limitations of CNNs for medical data

Advantages:

- **Feature extraction**: CNNs automatically learn hierarchical features from raw image data, eliminating the need for manual feature extraction, which is time-consuming in traditional methods.
- **Spatial relationships**: The convolutional layers can detect local patterns (e.g., edges, textures) in images, which are crucial for tasks like tumor detection or organ segmentation.
- **Efficiency**: CNNs are computationally efficient due to shared weights in convolution layers, allowing them to process large datasets more effectively.

Limitations:

- **Data requirements**: CNNs require large labeled datasets to perform well, which may not always be available in medical settings.
- **Limited to spatial data**: While CNNs excel in image-based data, they are not as effective for

non-spatial data like time-series or sequential data.

Use cases in medicine: CNNs have been widely applied in diagnostic tasks such as detecting cancers, classifying lesions, and analyzing radiological images (e.g., X-rays, MRIs, CT scans) [35, 36, 41].

Recent advances in CNN, like AlexNet [42], VGGNet [43], GoogLeNet [44], and ResNet [45], have significantly improved image classification accuracy, with models now outperforming human experts in some cases. These networks have been trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) using millions of annotated images [46, 47]. And their success has spurred the rise of transfer learning, where pre-trained models are fine-tuned for specific tasks [48]. In the next subsection, a brief description of some of the high-performance pre-trained models using ImageNet.

4.1.2.1 Visual geometry group network VGG 16–19

VGG16 network is the winning model architecture of the 2014 ImageNet competition. VGG consists of 16–19 layers. The size of the input image to VGG is (224×224). VGG has a set of convolutional filters with small sizes (3×3) to capture the information of the up/down and left/right center. The size of the pre-trained weights is 528 MB. The overall number of parameters of VGG16 is 138 357 544 parameters [43].

4.1.2.2 InceptionV3 model architecture

InceptionV3 network is the winning model architecture of the 2015 ImageNet competition. The Inception V3 model has a total of 48 layers. The size of the input image to InceptionV3 is (299×299). It is deeper than VGG16 but with fewer parameters. The size of the pre-trained weights is 92 MB. It has 23 851 784 parameters [44].

4.1.2.3 Residual neural network (ResNet)

ResNet network is the winning model architecture of the 2016 ImageNet competition. ResNet-50 contains a 50-layer architecture. The size of the input image to ResNet is (224×244). The size of the pre-trained weights is 99 MB. It has 25 636 712 parameters [45].

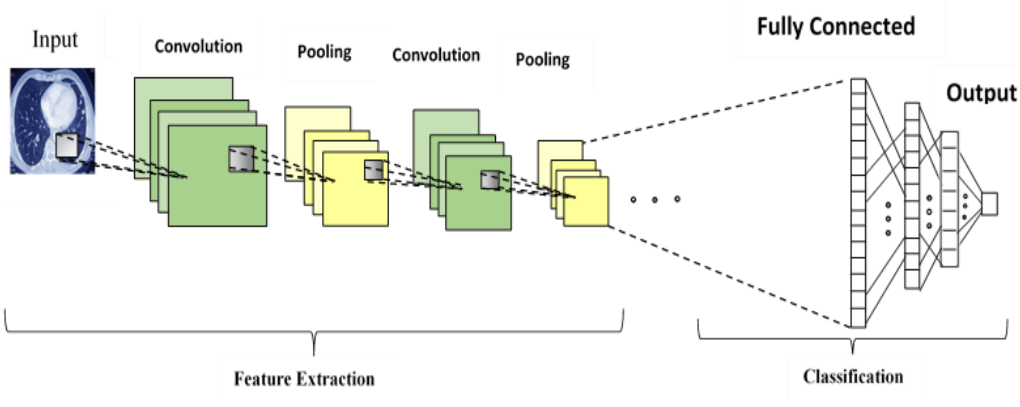


Figure 9: The basic structure of CNN.

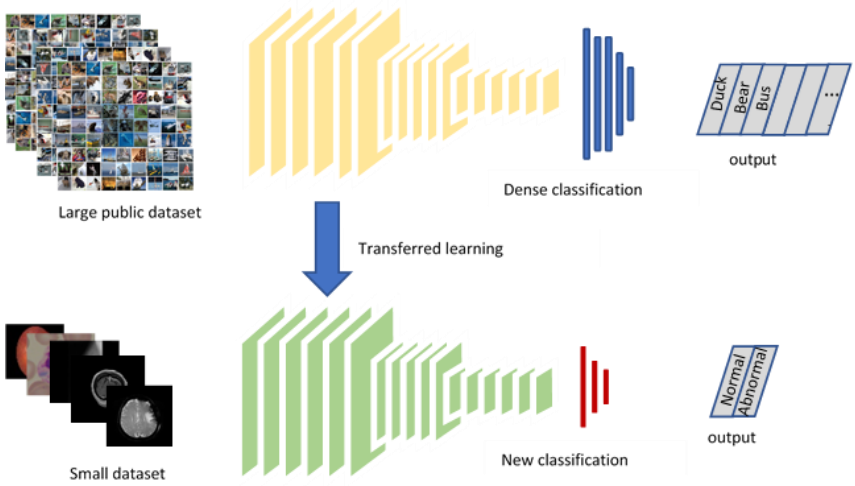


Figure 10: Transfer learning architecture.

4.2 Transfer learning

Transfer learning is a more appropriate approach when the available data for training is limited. In transfer learning, an intricate model can be trained using available large-scale annotated images such as natural images. Therefore, the TL process drives knowledge from source domain (e.g. natural images) to a target domain or network where the domain images are limited. Only the small amount of available annotated data of the target domain is used to tune the model. Where the fundamental features used for classification are similar between domains, retraining the entire model is unnecessary. In such cases, TL allows for the transfer of learned features, with only the classification layer(s) being retrained on the small new dataset [48]. TL leverages pre-trained models such as VGG [43], NasNetLarge [22], Inception GoogLeNet [44], ResNet [45], etc—that have been developed for image classification and they have been presented at the annual ILSVRC [46, 47]. TL saves a great amount of time lost in developing and training CNN models. The pre-trained model or the required part of the model can be incorporated directly into the new model and used as a classifier, standalone feature extractor, integrated feature extractor, or weight initializer [48, 49]. Figure 10 shows transfer learning architecture.

4.3 Curriculum learning

In the standard educational method, learning depends on a curriculum that presents new concepts based on previously collected ones. The rationale beyond this is that people pick up better if the information is introduced in a meaningful method instead of randomly. By using the same ideas to train neural networks starting with simple cases, it was noticed that the networks perform better, which indicates the significance of gradual and systematic learning [50].

The curriculum learning (CL) approach is motivated by the capacity of humans to pick up new tasks fast with finite "training sets". Similarly, the training procedure of medical students called teacher-student curriculum learning is based on training by tasks with gradually growing difficulty. While each task uses smaller datasets than those utilized in machine learning. Like, students can start with a simple task, such as deciding if an image includes lesions, and later are asked to determine if the lesions are malignant or benign which is a more complicated task. With time, they will progress to a more complex task, like recognizing the subtypes of lesions [8].

In machine learning, CL works with a series of training samples sorted in increasing order according to learning difficulty. The order in which the samples are introduced to the model is critical, as it can significantly impact the model's performance. Curriculum learning is an active area of research, particularly in applications such as medical image diagnosis [8].

A key point in CL is the design of data schedulers that control the sequence in which training samples are fed into the model. These schedulers can use a variety of methods to determine sample difficulty, such as expert input, heuristics, or natural language processing (NLP) applied to radiology reports.

Given a sample xi which should be assigned to a class label Ci ϵ {C1, C2, …., Cm}. Suppose the training set consisted of pairs {X, C}, and the training is processed in batches of size B for a total of E epochs. To train CNN with CL, prefer to start the training with simpler samples. Practically, CL is performed by assigning a probability to every training pair, where the simpler samples are given higher probabilities to be chosen first. Initially, every sample xi is assigned a probability pi(0). At the beginning of each epoch e, the training set {X, C} is permuted to $\{X, C\}^k$ by the reordering function F(e). Where this mapping is produced by sampling the training set based on the probabilities at the present epoch pi(e). After executing many iterations, these probabilities are updated using a scheduler, aiming to achieve a regular distribution by the end of the training process [50].

4.4 Active learning (optimal experimental design)

Supervised learning techniques rely heavily on annotated data. Although more datasets are becoming available, the effort, cost, and time required to annotate them remain significant. On the other side, any error especially in some important applications such as those in the medical domain can have severe consequences. Achieving reliable outcomes often requires an interactive process where predictions are reviewed or modified by an oracle or user. This means users must be able to override and adjust automated predictions to meet specific criteria. Techniques such as Active Learning (AL) or what is called Human-in-the-Loop computing have witnessed progress in overcoming these challenges [51].

Active learning is a semi-supervised learning approach that begins with a small set of labeled samples (seed samples) and iteratively selects the most informative samples from a pool of unlabeled data for annotation. By focusing training on the most informative subset of samples, AL improves model performance and reduces the annotation burden, particularly for image data. In AL, an MLT scans unlabeled data and recognizes the most informative samples. These samples are then presented to a human annotator (oracle) for labeling. This makes AL a part of the Human-in-the-Loop paradigm, where only selected samples are used for training, often far fewer than in traditional supervised learning [51].

Formally, suppose that U is available big pool of unannotated data and that there are oracles to request annotations for any unannotated sample x_U to be added to annotated set L. The goal is to train a model $f(x \mid L*)$ using the annotated set $L* \subseteq L$. A brute-force solution would involve requesting the oracle(s) to annotate each sample x_U, resulting in $L* = L$. However, this is a costly and not practical solution. Theoretically, there is an optimal subset $L*$ of data that can achieve performance equivalent to that obtained using the whole annotated dataset L, i.e. $(f(x \mid L*) \approx f(x \mid L))$. AL is a trend of ML that tries to explore this optimal subset $L*$, where the current model is $f'(x \mid L')$, L' is an intermediate annotated data. AL intends to iteratively

explore the most informative data samples x_i^* to train the model, assuming that the unannotated data samples and the model will evolve through time, rather than choosing a constant subset of samples once for training.

The selection of samples to be annotated is based on the informativeness of these requested samples. The evaluation of the informativeness of each un-annotated data sample x_U is done given $f'(x_U \mid L')$, then all selected samples are demanded to be annotated. After the annotations, the new annotated data has been used to improve the model. This is done by retraining the whole model using all available annotated data L´, or by using the most recently annotated sample x_i^* to fine-tune the network [51].

Active Learning typically employs three methods to select samples for annotation:

Stream-based selective sampling supposes the existence of a continuous flow of unannotated data samples x_U. In this method, the present model and informativeness $I(x_U)$ measure are the criteria used to specify, for each incoming sample whether or not to require an annotation from the oracle(s). Thus, while the model is being trained, it is offered a data sample and instantly decides if it needs to query for the label. Although this type of query is inexpensive, its performance is limited because it does not consider the broader context of the underlying distribution, but it depends on the separation nature of each decision, therefore the balance between the exploration and exploitation is less than in other query kinds.

Membership query synthesis generates the sample x_G^* that the model believes to be most informative, rather than selecting from real-world data. Therefore, it is annotated by the oracle(s). This method may be very effective in bounded domains, but it may struggle when the model has no knowledge of unrepresented areas of the data distribution, similar to stream-based methods.

Pool-based sampling selects N data samples $x_0^*, \ldots, x_N^*$ from a large unlabeled dataset U to pull samples from. Pool-based approaches use the present model to do a prediction on un-annotated data samples to get a ranked measure of informativeness for each data sample in the un-annotated data. The highest N informative samples are selected for annotation by the oracle(s). Therefore, the model is initially trained on labeled samples which are then used to find which data samples would be most informative to be inserted into the training set for the next AL loop. This approach has proved to be the most promising, which depends on batch-based training. Figure 11 shows the full process of active learning.

AL uses some informativeness measures of unlabeled samples to select the most informative samples. They depend on probabilities, these approaches are least confidence sampling, margin sampling, and entropy sampling [51].

Least confidence sampling the model selects the highest uncertainty sample or least confidence for annotation and therefore is given to the oracle to be labeled.

Margin sampling can be utilized in a multi-class, it uses the first and second most likely labels and computes the difference between them, then considers the sample that has the smallest difference between the first and second most likely labels to be annotated.

Entropy sampling uses entropy as it is a measure of uncertainty to select a sample to be annotated. Entropy measures the amount of information gained by considering a sample and so selects the sample that maximizes the information that has the largest entropy value [51].

4.5 Federated learning

Federated learning (FL), developed by Google in 2017 is a collective distributed decentralized learning method that allows many organizations to collaborate on machine learning or deep learning models without sharing clients' or devices' data. It allows the training data to be on the decentralized edge devices rather than keeping it in a data center. These individual nodes or devices jointly train a machine learning or deep learning model from their local data and then aggregate the devices' training outputs on the server to update the global model without sharing edge data. The resulting model can be shared among all participating devices or clients. Therefore, it provides secure models that fulfill an efficient solution while providing data access and security [52-54].

One major issue with centralized models is that medical organizations do not allow to break doctor-patient confidentiality by providing medical images such as CT and X-ray images for training purposes because of privacy, legal, and data-ownership issues. To develop deep learning models for the medical domain, large medical data is needed to develop these models. Therefore, many medical researchers illustrated that federated learning is a good technique to connect different medical organizations and let them share their experiences while keeping privacy. Furthermore, the performance of the learning model will be improved using a large medical dataset. However, the resulting models may be biased toward organizations that have larger training datasets [53].

In federated learning, the process begins by sending a global model with unified initial weights to each client. At each client side, there is a local dataset, where the model is trained in each separately. After completing local training, the client sends its model updates back to the server, which aggregates these updates to refine the global model, while the data at the clients remains local in each client. The server has the authority to manage the whole process where it sends the model to the client, collects the updates, and synchronizes them to build the updated model with the new parameters. This method enables medical organizations to collaborate on training models while maintaining data privacy. There are different federated learning algorithms according to the computation method of gradients such as federated stochastic gradient descent, federated averaging, and federated learning with dynamic regularization. [53, 54]. Figure 12 shows the architecture of federated learning.

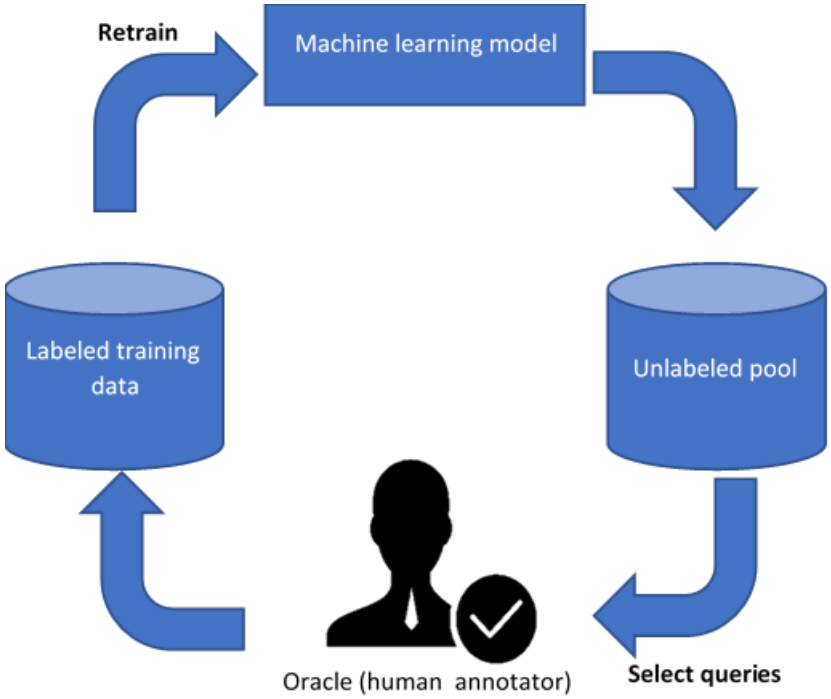


Figure 11: The process of active learning.

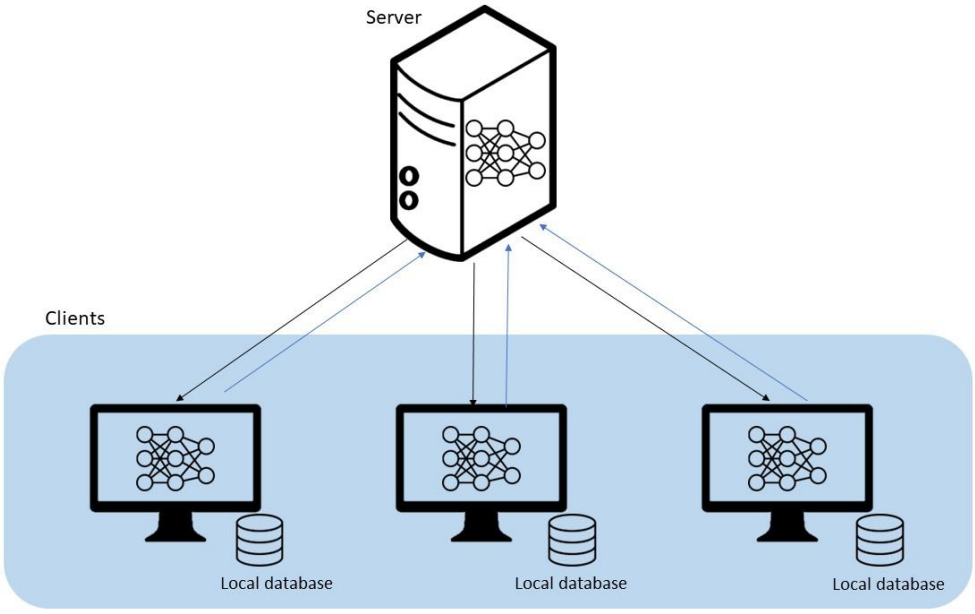


Figure 12: Federated Learning architecture.

5 Search methodology

5.1. Search criteria

This research investigates recent trends in machine learning (ML) within the medical domain. To achieve this, we explored a ScienceDirect (Elsevier) (http://www.sciencedirect.com). The following keywords were used in the search: “active learning”, “curriculum learning”, “deep learning”, “transfer learning” and “federated learning” to investigate the different research that utilizes these recent trends. Additional keywords—"medical", "disease", "cancer" and "gene" were included to focus the search on medical applications that used these new trends. Although the search intended to retrieve the articles related to any disease, "cancer" was added to

retrieve more relevant results, given that much of the recent research in ML is focused on cancer. Publications from 2016 to 2024 were considered. The composition of the used terms to form the search query used for deep learning-based techniques in the medical domain was:

"Deep learning" AND ("medical" OR "Disease" OR "Cancer" OR "Gene").

Where the aim of this research is to find the new trends in machine learning techniques which after accurate investigation were found to be mostly based on “deep learning” either alone or combined with other new techniques such as “transfer learning”, “active learning”, “curriculum learning” and “federated learning”, so the same query was used as for deep learning-based techniques with adding the other techniques’ keyword as follow:

("Deep learning" AND "*") AND ("medical" OR "Disease" OR "Cancer" OR "Gene")

Where “*” can be replaced by any of the other techniques’ keywords (“transfer learning”, “active learning”, “curriculum learning”, and “federated learning”).

The following criteria were applied to select the publications: (1) Articles related to human diseases (other organisms’ related diseases are excluded); (2) Inclusion of at least one of the new ML techniques; (3) Only complete research articles were included (excluding letters, surveys, book chapters, and non-English articles); (4) publications published from 2016 to 2024.

5.2. Data extraction

As the search retrieved a large number of articles, therefore only a subset of the retrieved articles was selected for analysis. Figures 13-15 illustrate the number of publications per year for the various techniques between 2016- 2024, based on “Elsevier” database to show the growth rates of these new trends.

- Figure 13 shows the steady increase in deep learning publications, from 17 articles in 2016 to 2958 in 2024, indicating a growing interest in applying deep learning in the medical domain.
- Figure 14 shows that transfer learning started to be applied in the medical domain in 2017 with 2 articles only and reached 218 in 2024.
- Figures 15 shows that the number of publications of active learning, curriculum learning, and federated learning is limited and scattered across the years as they are newly emerged trends.

The selected articles were drawn from top journals in ScienceDirect, adhering to the criteria mentioned above. The references provide a sample of the applications of these new ML techniques in the medical domain, rather than an exhaustive list. For each reference, key details such as the task, disease, technique(s) used, evaluation results, and data type are presented.

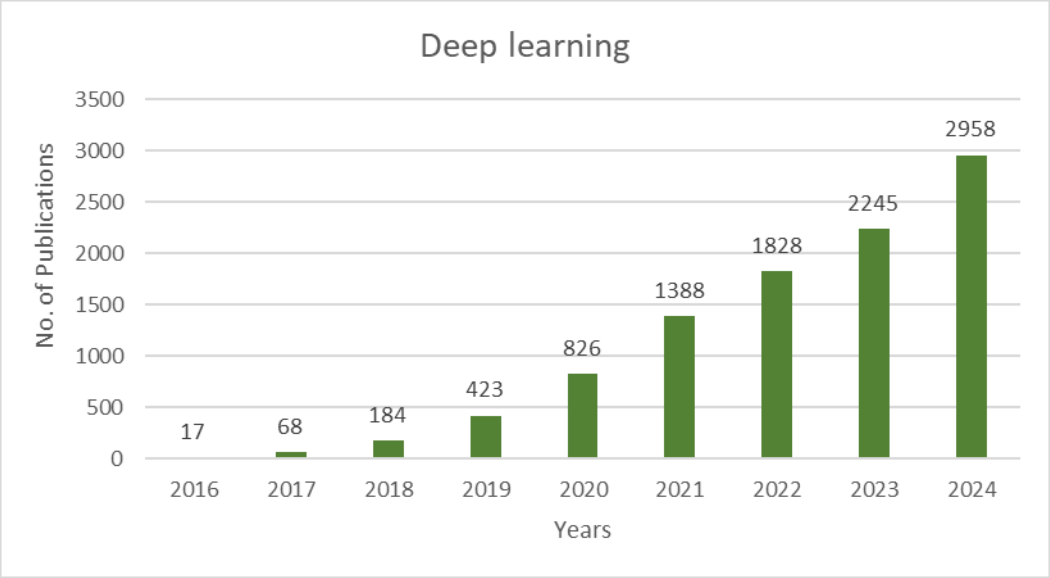


Figure 13: The number of articles published on deep learning from 2016- 2024 in Elsevier database.

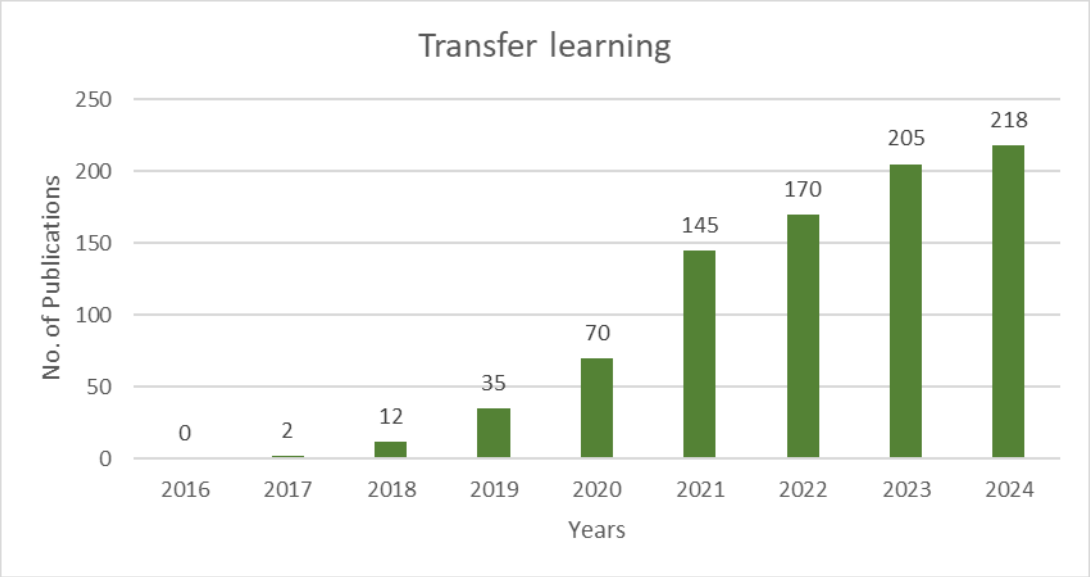


Figure 14: The number of articles published on transfer learning from 2016- 2024 in Elsevier database.

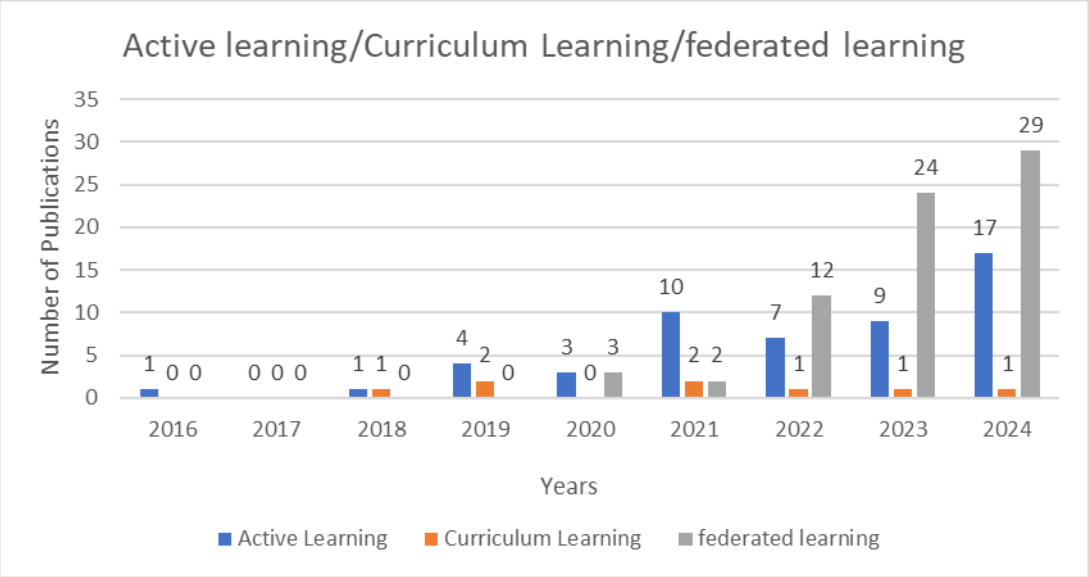


Figure 15: The number of articles published on active/curriculum/Federated learning from 2016- 2024 in Elsevier database.

6 Some applications of new trends of MLTs in the medical domain

This section illustrates the selected articles from the retrieved ones from searching the databases which represent the applications of previously discussed emerging ML trends in the medical domain.

Li, X., et al., [55] proposed a DL model to detect lung nodules. First, segmentation and rib suppression were applied to extract the region of interest and enhance the nodules' visibility. Then, the histogram was applied to enhance the images. After that, patch-based multi-resolution CNN was used for feature extraction, and 4 fusion methods were employed for classification, the best performance method to detect lung nodules achieved an accuracy of more than 99% and FAUC of 0.982 when applied to the Chest X-ray radiographs dataset [56].

El Houby & Yassin [57] developed a CNN model to classify the breast mammographic images' into nonmalignant or malignant. They used 2 methods, the first is based on patches of region of interest (ROI) in the mammogram and the second is based on the whole breast. The accuracy, specificity, sensitivity, and AUC were 95.3%, 92.6%, 98%, and 0.974 respectively using MIAS [20] dataset, while they were 96.52%, 96.49%, 96.55%, and 0.98 using INbreast [58] dataset.

Dai, Y., et al., [59] developed a deep learning CNN model for detecting coronary artery disease utilizing raw heart sound signals. It extracts 206 multidomain features and 126 medical multidomain features. The heart sound signal datasets have been collected from 400 patients from

the hospital of Xinjiang Medical University. The model achieved an accuracy of 87.86, sensitivity of 90.67, specificity of 82.38, and AUC of 94.70 using multidomain features. It achieved an accuracy of 85.6, sensitivity of 88.04, specificity of 80.83, and AUC of 92.74 using medical multidomain features.

Alassafi et al., [60] proposed a model that predict the distribution of the COVID-19 outbreak in Saudi Arabia, Malaysia, and Morocco. A DL RNN and LSTM network were developed to predict the number of possible cases of COVID-19. The LSTM achieved an accuracy of 98.58%, while the RNN achieved an accuracy of 93.45%. A comparison was conducted between the number of resulting deaths and the number of coronavirus cases in each of the 3 countries. The model predicted the number of certain COVID-19 cases and deaths for the following 7 days. The model was tested using a public dataset from the European Centre for Disease Prevention and Control [61].

Maiti et al., [62] developed a deep learning (DL)-based framework to automatically detect and segment the optic disc from fundus images for the diagnosis of diabetic retinopathy. The framework utilized an adjusted CNN, experimenting with seven different encoder networks: DenseNet121, InceptionV3, ResNet34, GG11, VGG19, VGG13, and VGG16. VGG16 was selected as the adopted encoder, while the decoder was designed with a harmonic structure based on that of the encoder to improve segmentation performance. The framework was applied to several fundus image datasets, including DIARETDB1, MESSIDOR, IDRiD, DIARETDB0, CHASE-DB1, DRIVE, and STARE. It achieved an impressive accuracy of 99.44%.

Zareen, et al. [63] developed a skin cancer classification deep learning CNN-RNN model with a ResNet-50 for spatial features extraction and LSTM for temporal dependencies. The model has been applied to a dataset of 9000 images of skin lesions representing 9 cancer types. The model achieved an accuracy of 94.48, a sensitivity of 94.38, and a specificity of 93.

Ge, R., et al., [64] Proposed a Dual-Enhanced Convolutional Ensemble Neural Network (DECENN) to detect the presence or absence of metastasis in the whole slide imaging patches of breast cancer. It utilizes VGG16 and DenseNet121 in the network. It was applied to the updated version of a benchmark dataset of microscopic images and histopathologic scans of lymph node sections for the breast [65]. It achieved an accuracy of about 98.92%, an AUC of 99.70%, and a F-score of 98.93%.

Liu, Q., X. She & Q. Xia [66] proposed a model to classify osteosarcoma cells and other cell types using an updated version of CA-MobileNet V3 based on the transfer learning model. It was applied to osteosarcoma cells microscopy imaging of bone cancer dataset [67]. It achieved an accuracy of 98.69 % and f1-score of 94.11.

Oommen & Arunnehru [68] proposed a model to diagnose Alzheimer's disease in its early stages. The proposed model contains 3 phases: preprocessing the images, extracting features using TL with ResNet-18, which are then compressed by cascaded autoencoders (AE), and finally classifying the disease to one of its 5 stages using DNN. The model was applied to the MRI Neuroimaging dataset [69] and achieved an accuracy of 98.54% recalls of 98.9%, a precision of 98.98%, and an F1 score of 98.82%.

Kumar et al. [70] developed a CNN model using the Resnet152 TL approach with feature extractors to classify the brain tumor images into normal, benign, and malignant. The model has been applied to the Brats MRI image dataset. The proposed transfer Learning model achieved high accuracy reaching 99.57%.

Manickam, et al., [71] proposed a deep TL model for pneumonia detection. The chest X-ray images were preprocessed to recognize the existence of pneumonia based on the U-Net segmentation network, then classify the cases as normal or abnormal (Bacteria, viral) using pre-trained models such as ResNet50, InceptionV3, and Inception ResNetV2. It was evaluated using a publicly available database which includes 5,232 chest x-ray images. ResNet50 model achieved an accuracy of 93.06%, precision of 88.97%, Recall of 96.78%, and F1-score of 92.71%.

Veknugopal, et al., [72] developed a DNN using modified EfficientNetV2-M based on transfer learning to detect skin cancer on dermoscopic images. The model was applied to 58,032 dermoscopic images collected from [73-77]. The model was tested for binary classification tasks and the multiclass classification tasks. It achieved an accuracy reached 97.62 for the multiclass classification of the ISIC 2020 dataset, while it achieved an accuracy of 99.23 for the binary classification of the same dataset.

Mehmood, et al., [78] developed a model to diagnose Alzheimer's disease (AD) in its early stage based on TL using VGG-19 pre-trained model. The model distinguishes among 4 classes which are AD, late mild cognitive impairment (LMCI), early mild cognitive impairment (EMCI), and normal control (NC). The used dataset was collected from the AD Neuroimaging Initiative (ADNI) [69] database. In the pre-processing phase, the gray matter (GM) tissue was segmented from brain MRI, and then VGG-19 was used to classify the segmented parts. The model achieved an accuracy of 98.73% to distinguish between AD and NC, 83.72% to distinguish between LMCI and EMCI cases, and more than 80% to distinguish between the other combinations of classes.

Al-Shabi, Shak, and Tan, [79] developed a Progressive Growing Channel Attentive Non-Local (ProCAN) deep learning model to classify lung nodules as benign or malignant. Curriculum Learning (CL) was used to train easy samples before hard samples. The model has gradually grown to improve the possibility of classifying the samples based on CL. The model has been applied to samples from 2 publicly available CT scan datasets LIDC-IDRI [80] and LUNGx [81]. It achieved an accuracy of 95.28% and AUC of 98.05%, precision of 95.75, sensitivity of 94.33 and F1-Score of 95.04.

Cho, Y., et al., [82] proposed CL model using a DL CNN to classify chest radiograph (CXR) images into normal and five types of pulmonary abnormalities. The model used ResNet-50 for training on patches of CXR images based on the various patch ratios according to pre-trained weights, with fine-tuning using transfer learning

(TL). The model was applied to CXR from hospitals, including Seoul National University Bundang Hospital (SNUBH) and Asan Medical Center (AMC). It achieved the following accuracies: 90.97% for 20% of the dataset at SNUBH, 91.92% for 50%, and 93.00% for 100%. At AMC, the accuracies were 93.90%, 94.54%, and 95.39%, respectively.

Wong et al., [83] developed a CL-based method for classifying medical images, using features from segmentation networks. The model first learns simpler shapes and features through a segmentation network pre-trained on similar data, then applies this knowledge for more complex classification tasks. The M-Net, a CNN modified from U-Net for working with fewer training samples, was used for segmentation. Then the CNN classifier receives the features from a segmentation network as inputs. The model achieved an accuracy of 82% in a 3D 3-classes brain tumor classification and 86% on a 2D nine-class cardiac semantic level classification problem.

Wu, et al. [84] developed a weakly-supervised deep AL framework to diagnose COVID-19 using CT scans. The framework contains a 2D U-Net for segmentation of the lung region and a hybrid active learning approach, which keeps sample diversity and predicted loss diagnosis of COVID-19. The framework classifies the CT scans into one of three classes which are pneumonia, coronavirus pneumonia caused by SARS-CoV-2, and normal cases. The framework was validated on a CT scan dataset from the China Consortium of Chest CT Image Investigation (CC-CCII) [85]. With only 30% of the labeled data, the accuracy of the framework reached 0.867, while AUC was 0.968.

Wu, X., et al., [86] proposed a hybrid active learning (HAL) framework that combines AL with deep TL using ResNet18. The framework applies data augmentation to the unlabeled data pool and uses a hybrid sampling approach that maintains sample variety and classification loss (data uncertainty). The diversity sampling is based on data augmentation, while the generated data noise is discarded with an outlier detection process. The HAL was validated on 3 medical image datasets which are the Hyper-Kvasir for gastrointestinal disease [87], Messidor for eye fundus images [88], and breast cancer datasets [89]. By applying the proposed framework to the Hyper-Kvasir dataset it achieves an accuracy of 0.871, precision of 0.602, recall of 0.587, and F1-score of 0.594.

Meirelles, et al., [90] used Pool-based AL to train DL models for classifying Tumor Infiltrating Lymphocytes. The proposed approach selects image patches based on feature grouping and prediction uncertainty. They introduced a Diversity-Aware Data Acquisition (DADA) method, which ensures diverse batch selection by clustering images based on features and then choosing uncertain patches from each cluster. The most uncertain patches from each cluster are prioritized for selection, the clusters with the most uncertain patches must participate with more patches, the pool is updated by removing the selected patches. By applying the proposed model to the cancer tissue image dataset [91], it achieved an AUC of 0.78 with fewer tissue patches and execution time.

Zhang, et al., [92] developed a semi-supervised framework for brain segmentation that incorporates quality-driven active learning (QDAL). In the AL module, deep supervision loss and attention mechanism improve the accuracy of segmentation and return quality information for the unlabeled slices. The AL module chooses the most informative slices to be annotated, and the segmentation process is trained iteratively using the updated labeled data. The framework was tested on two brain MRI datasets [93, 94]. The experiment results showed that the segmentation utilizing the QDAL only wants 15–20% annotated slices for the brain extraction task, and 30–40% for tissue segmentation, achieving competitive results with full supervision and an accuracy of 90.7.

Lu, Q., et al.,[95] presented a blood cell classification method called MAE4AL, which combines the self-supervised Masked Autoencoder (MAE) and active learning (AL). It chooses the most remarkable samples for labeling based on self-supervised loss of MAE and sample uncertainty. Tested on blood smear samples obtained from [96], MAE4AL needed labeling only 20% of the data to perform the same as ResNeXt, which was trained on the full dataset. When it trained using half of the labeled data, MAE4AL achieved an accuracy of 96.36%, outperforming ResNeXt which trained on all the data.

Kumbhare et al. [97] developed a FL method for breast cancer diagnosis using mammogram images from the "Curated Breast Imaging Subset of DDSM (CBIS-DDSM)" dataset [98]. The DenseNet pre-trained model was used for feature extraction and the extracted features were classified using Enhanced Recurrent Neural Networks (E-RNN). FL was employed to reduce processing time and improve model performance. The method achieved an accuracy of 95%.

Feki, et al. [53] proposed a decentralized FL framework that permits different medical organizations to screen COVID-19 using Chest X-ray images based on deep learning while keeping the privacy of patient data. Two pre-trained models which are VGG16 and ResNet50 were used for classification. The framework was tested using four clients, where each client has his private dataset and the same CNN models. The proposed FL framework achieved competitive results compared to those models trained by sharing data. The best achieved accuracy was 97% using the ResNet50 model with data augmentation.

Zhang, et al. [99] proposed a FL based on DL framework for the diagnosing brain disorders. The proposed framework was tested on Autism Brain Imaging Data Exchange (ABIDE) [100] dataset. The proposed framework achieved an average accuracy of 79% and reduced the communication burden of FL.

Shaikh, et al. [101] developed an FL-based DL method to classify respiratory diseases by listening to lung sounds. Generative Adversarial Networks created new lung sounds to train a neural network that classifies 4 lung diseases, heart attack and normal breathing patterns. Using two datasets [102, 103], the proposed method achieved an accuracy of 92% for the classification of different respiratory diseases and heart failure.

Table 1 provides a summary of 25 selected articles from top journals on Science Direct, published between 2016 and 2024, based on a database search. These articles showcase applications of recent trends of MLTs in the medical domain and are intended to illustrate these trends, not to present a comprehensive list. For each reference, the table includes the task, disease, techniques used, evaluation results, and data type.

Table 1: Summary of the selected articles from search results for applications of the new ML in medical domain.

Ref.	Task	Disease	Used Technique(s)	Evaluation results	Data Type
[55]	Detection	Chest Lung cancer	DL-CNN	Acc.= 99% FAUC = 0.982	x-ray radiographs
[57]	Classification	Breast cancer	DL-CNN	Acc.=96.52% spec.=96.4% Sen.=96.5% AUC =0.98	mammogram
[59]	Detecting	Coronary artery disease	DL-CNN	Acc.=87.86, % Sen.=90.67% Spec.=82.38 % AUC = 94.70	heart sound signals
[60]	Prediction	COVID-19	DL-RNN LSTM	Acc. = 93.45% Acc.= 98.58%	Numerical
[62]	Segmentation diagnosis	Diabetic retinopathy	DL-CNN	Acc.= 99.44%	fundus images
[63]	Classification	Skin cancer	DL-CNN-RNN	Acc.=94.48, Sen.= 94.38 spec. = 93	skin lesion images
[64]	Detection	Breast cancer	DL-TL-VGG16-DenseNet121	Acc.=98.92%, AUC =99.70%, F-score = 98.93%	histopathologic image of lymph node
[66]	Classification	Bone cancer	TL CA-MobileNetV3	Acc.=98.69 % f1-score= 94.11%	microscopic images of bone cancer
[68]	Classification	Alzheimer's disease	TL- ResNet-18- AE-DL	Acc. = 98.54% recalls = 98.9% prec.=98.98% F1-score=98.82%	MRI Neuroimaging dataset
[70]	Classification	Brain tumor	TL-Resnet152-CNN	Acc. =99.57%	MRI
[71]	Segmentation Detection	Pneumonia	U-Net TL- ResNet50	Acc.=93.06% prec.=88.97% Rec.=96.78% F1score=92.7	chest X-ray
[72]	Classification	Skin cancer	TL-EfficientNetV2-M	Acc. = 99.23	dermoscopic images
[78]	Classification	Alzheimer	TL-VGG19	Acc.=98.73%	MRI
[79]	Classification	lung nodules	DL-CNN-CL	Acc. = 95.28% AUC = 98.05% Precision = 95.75 Sen = 94.33 F1-Score = 95.04	CT scans
[82]	Classification	pulmonary abnormalities	TL-ResNet-50 -CL	Acc. =93.90, 94.54, 95.39 For 20%, 50%, 100% of dataset	CXR
[83]	Segmentation Classification	brain tumor cardiac	TL-M-Net DL-CNN-CL	Acc.= 82% Acc.= 86%	MR

[84]	Segmentation Classification	COVID-19	TL-U-Net DL-AL	Acc=0.866 ROC= 0.968	CT scans
[86]	Classification	gastrointestinal disease	TL-ResNet18 AL	Acc. =0.871 Prec. =0.602 Recall=0.587 F1score=0.594	Images
[90]	Classification	Tumor Infiltrating Lymphocytes	DL_CNN_AL	AUC = 0.78	histology image
[92]	Segmentation	Brain	DL_CNN_AL	Acc.= 90.7	MRI
[95]	Classification	blood diseases (Leukemia)	MaskedAutoencoder (MAE4AL)	Acc.= 96.36%	blood smear samples
[97]	Classification	Breast Cancer	FL-TL-DenseNet-RNN	Acc.=95%	mammogram
[53]	Classification	COVID-19	FL-TL-VGG16/ ResNet50	Acc. = 97%	X-ray images
[99]	Classification	Brain disorders	FL-CNN	Acc. = 79%	Autism Brain Imaging
[101]	Classification	respiratory diseases& heart failure	FL-DL	Acc. = 92%	breathing sounds

7 Conclusion and future work

This research explored the emerging trends in machine learning techniques (MLTs) within the medical domain. Through a comprehensive literature review, we found that deep learning has become the dominant trend, holding significant promise for developing intelligent medical applications. A key advantage of deep learning is its ability to perform automatic feature engineering, simplifying the model-building process and reducing reliance on manual input. Current research predominantly addresses diagnostic tasks, with disease classification being the most common approach. Other tasks, such as segmentation, are also explored. Cancer, in its various forms, is the most frequently studied condition, while the COVID-19 pandemic has notably led to a surge in research on lung diseases.

In the realm of medical imaging, traditional machine learning approaches require extensive pre-processing, including feature extraction and selection. Deep learning, particularly Convolutional Neural Networks (CNNs), has advanced the field by automating feature engineering, reducing the need for manual intervention. However, this comes with an increased demand for large datasets and significant computational resources. To address these challenges, recent trends like transfer learning, curriculum learning, active learning, and federated learning have been introduced to enhance model performance, expedite the training process, and improve data security. In summary, the overarching goal in this field is to automate processes, reduce human intervention, and maximize the value derived from limited labeled data, thereby enhancing medical decision-making and patient outcomes.

Looking ahead, there are several key areas where further work is needed. While the number of publications on deep learning in the medical domain has steadily increased since its initial applications in 2016, and although these applications have yielded promising results, further research is essential to address several key challenges. Areas such as active learning, curriculum learning, and federated learning have shown promise but remain under-explored and require more attention in future research. A critical direction for future is to focus on reducing the time and computational costs associated with deep learning models and other trends. These processes often consume substantial energy, indirectly contributing to environmental and climate concerns. Therefore, developing more energy-efficient techniques will be crucial. Additionally, data augmentation, a significant pre-processing step in deep learning, could be integrated more effectively into the model-building process itself, thereby enhancing sample diversity and improving class representation with less manual effort.

Another important aspect for future research is the development of standardized, public databases that include diverse patient data, such as DNA sequences. These databases would enable more comprehensive studies and improve the accuracy of predictive models by providing a richer set of input data. Additionally, integrating knowledge from multiple domains could further enhance the performance of deep learning models in different medical applications. Despite the progress

made, the real challenge lies in translating these advancements into practical, real-world applications that can be implemented in clinical settings. Bridging the gap between theoretical research and clinical deployment will be vital to realizing the full potential of deep learning in medicine.

Conflicts of interest

The author has no competing interests to declare.

References

[1] Chen, M., et al., *Disease prediction by machine learning over big data from healthcare communities.* Ieee Access, 2017. **5**: p. 8869-8879.

[2] Grossman, R.L., et al., *Toward a shared vision for cancer genomic data.* New England Journal of Medicine, 2016. **375**(12): p. 1109-1112.

[3] Schaekermann, M., et al., *Understanding expert disagreement in medical data analysis through structured adjudication.* Proceedings of the ACM on Human-Computer Interaction, 2019. **3**(CSCW): p. 1-23.

[4] Garg, A. and V. Mago, *Role of machine learning in medical research: A survey*. Computer Science Review, 2021. **40**: p. 100370.

[5] Dallora, A.L., et al., *Machine learning and microsimulation techniques on the prognosis of dementia: A systematic literature review.* PloS one, 2017. **12**(6): p. e0179804.

[6] Kharazmi, P., et al., *A computer-aided decision support system for detection and localization of cutaneous vasculature in dermoscopy images via deep feature learning.* Journal of medical systems, 2018. **42**(2): p. 1-11.

[7] Xu, J., K. Xue, and K. Zhang, *Current status and future trends of clinical diagnoses via image-based deep learning.* Theranostics, 2019. **9**(25): p. 7556.

[8] Xie, X., et al., *A survey on incorporating domain knowledge into deep learning for medical image analysis.* Medical Image Analysis, 2021. **69**: p. 101985.

[9] Lee, C.H. and H.-J. Yoon, *Medical big data: promise and challenges.* Kidney research and clinical practice, 2017. **36**(1): p. 3.

[10] Dinov, I.D., *Methodological challenges and analytic opportunities for modeling and interpreting Big Healthcare Data.* Gigascience, 2016. **5**(1): p. s13742-016-0117-6.

[11] http://archive.ics.uci.edu/ml/datasets/.

[12] Available from: http://image-net.org/challenges/LSVRC/.

[13] https://file.biolab.si/biolab/supp/bicancer/projections/.

[14] Corsetti, V., et al., *Evidence of the effect of adjunct ultrasound screening in women with mammography-negative dense breasts: interval breast cancers at 1 year follow-up.* European journal of cancer, 2011. **47**(7): p. 1021-1026.

[15] Zhang, Z. and E. Sejdić, *Radiological images and machine learning: trends, perspectives, and prospects.* Computers in biology and medicine, 2019. **108**: p. 354-370.

[16] Saslow, D., et al., *American Cancer Society guidelines for breast screening with MRI as an adjunct to mammography.* CA: a cancer journal for clinicians, 2007. **57**(2): p. 75-89.

[17] http://medicaldictionary.thefreedictionary.com/operating+microscope.

[18] Jones, N.C. and P.A. Pevzner, *An introduction to bioinformatics algorithms*. 2004: MIT press.

[19] Rahman, T., et al., *COVID-19 radiography database.* https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

[20] Suckling, J., et al., *Mammographic image analysis society (mias) database v1. 21*. 2015.

[21] Labati, R.D., V. Piuri, and F. Scotti. *All-IDB: The acute lymphoblastic leukemia image database for image processing*. in *2011 18th IEEE international conference on image processing*. 2011. IEEE.

[22] Zoph, B., et al. *Learning transferable architectures for scalable image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[23] Thusberg, J. and M. Vihinen, *Pathogenic or not? And if so, then how? Studying the effects of missense mutations using bioinformatics methods.* Human mutation, 2009. **30**(5): p. 703-714.

[24] El Houby, E.M., *Machine learning techniques for pathogenicity prediction of non-synonymous single nucleotide polymorphisms in human body*. Journal of Ambient Intelligence and Humanized Computing, 2023. **14**(7): p. 8099-8113.

[25] Han, J., J. Pei, and M. Kamber, *Data mining: concepts and techniques*. 2011: Elsevier.

[26] Hamla, H. and K. Ghanem, *A hybrid feature selection based on fisher score and SVM-RFE for microarray data*. Informatica, 2024. **48**(1).

[27] Fahrudin, T.M., I. Syarif, and A.R. Barakbah. *Ant colony algorithm for feature selection on microarray datasets*. in *2016 International Electronics Symposium (IES)*. 2016. IEEE.

[28] Talavera, L. *An evaluation of filter and wrapper methods for feature selection in categorical clustering*. in *International Symposium on Intelligent Data Analysis*. 2005. Springer.

[29] Tabakhi, S. and P. Moradi, *Relevance–redundancy feature selection based on ant colony optimization.* Pattern recognition, 2015. **48**(9): p. 2798-2811.

[30] Yang, X.-S., *Firefly algorithm, stochastic test functions and design optimisation.* International journal of bio-inspired computation, 2010. **2**(2): p. 78-84.

[31] Mashhour, E.M., et al., *Feature Selection Approach based on Firefly Algorithm and Chi-square.* International Journal of Electrical & Computer Engineering (2088-8708), 2018. **8**(4).

[32] Neagoe, V.-E. and E.-C. Neghina. *Feature selection with ant colony optimization and its*

applications for pattern recognition in space imagery. in *2016 international conference on communications (COMM)*. 2016. IEEE.

[33] El Houby, E.M., N.I. Yassin, and S. Omran, *A hybrid approach from ant colony optimization and K-nearest neighbor for classifying datasets using selected features.* Informatica, 2017. **41**(4).

[34] LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.

[35] Goodfellow, I., Y. Bengio, and A. Courville, *Deep learning*. 2016: MIT press.

[36] Dezaki, F.T., et al., *Cardiac phase detection in echocardiograms with densely gated recurrent neural networks and global extrema loss.* IEEE transactions on medical imaging, 2018. **38**(8): p. 1821-1832.

[37] Raiaan, M.A.K., et al., *A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks.* Decision Analytics Journal, 2024: p. 100470.

[38] Ali, M.J., et al., *A review of AutoML optimization techniques for medical image applications.* Computerized Medical Imaging and Graphics, 2024: p. 102441.

[39] Ambekar, S. and R. Phalnikar. *Disease risk prediction by using convolutional neural network*. in *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. 2018. IEEE.

[40] Anwar, S.M., et al., *Medical image analysis using convolutional neural networks: a review.* Journal of medical systems, 2018. **42**(11): p. 1-13.

[41] Oraibi, Z.A. and S. Albasri, *A robust end-to-end cnn architecture for efficient covid-19 prediction form x-ray images with imbalanced data.* Informatica, 2023. **47**(7).

[42] Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks.* Advances in neural information processing systems, 2012. **25**.

[43] Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.

[44] Szegedy, C., et al. *Rethinking the inception architecture for computer vision*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[45] He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[46] Deng, J., et al. *Imagenet: A large-scale hierarchical image database*. in *2009 IEEE conference on computer vision and pattern recognition*. 2009. Ieee.

[47] Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International journal of computer vision, 2015. **115**(3): p. 211-252.

[48] Oquab, M., et al. *Learning and transferring mid-level image representations using convolutional neural networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[49] Wang, L., et al., *Trends in the application of deep learning networks in medical image analysis: Evolution between 2012 and 2020.* European Journal of Radiology, 2022. **146**: p. 110069.

[50] Jiménez-Sánchez, A., et al. *Medical-based deep curriculum learning for improved fracture classification*. in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2019. Springer.

[51] Budd, S., E.C. Robinson, and B. Kainz, *A survey on active learning and human-in-the-loop deep learning for medical image analysis.* Medical Image Analysis, 2021. **71**: p. 102062.

[52] KhoKhar, F.A., et al., *A review on federated learning towards image processing.* Computers and Electrical Engineering, 2022. **99**: p. 107818.

[53] Feki, I., et al., *Federated learning for COVID-19 screening from Chest X-ray images.* Applied Soft Computing, 2021. **106**: p. 107330.

[54] Wu, J.C.-H., et al., *Dynamically Synthetic Images for Federated Learning of Medical Images.* Computer Methods and Programs in Biomedicine, 2023: p. 107845.

[55] Li, X., et al., *Multi-resolution convolutional networks for chest X-ray radiograph based lung nodule detection.* Artificial intelligence in medicine, 2020. **103**: p. 101744.

[56] Li, X., et al. *Rib suppression in chest radiographs for lung nodule enhancement*. in *2015 IEEE International Conference on Information and Automation*. 2015. IEEE.

[57] El Houby, E.M. and N.I. Yassin, *Malignant and nonmalignant classification of breast lesions in mammograms using convolutional neural networks.* Biomedical Signal Processing and Control, 2021. **70**: p. 102954.

[58] I. C. Moreira, I.A., I. Domingues, A. Cardoso, M. J. Cardoso, and J. S. Cardoso, *Inbreast: toward a full-field digital mammographic database.* Academic radiology, 2012. **19**(2): p. 236-248.

[59] Dai, Y., et al., *Deep learning fusion framework for automated coronary artery disease detection using raw heart sound signals.* Heliyon, 2024. **10**(16).

[60] Alassafi, M.O., M. Jarrah, and R. Alotaibi, *Time series predicting of COVID-19 based on deep learning.* Neurocomputing, 2022. **468**: p. 335-344.

[61] https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic- and distribution-covid-19-cases-worldwide.

[62] Maiti, S., et al., *Automatic detection and segmentation of optic disc using a modified convolution network.* Biomedical Signal Processing and Control, 2022. **76**: p. 103633.

[63] Zareen, S.S., et al., *Enhancing Skin Cancer Diagnosis with Deep Learning: A Hybrid CNN-RNN Approach.* Computers, Materials & Continua, 2024. **79**(1).

[64] Ge, R., et al., *Detection of presence or absence of metastasis in WSI patches of breast cancer using the dual-enhanced convolutional ensemble neural network.* Machine Learning with Applications, 2024. **17**: p. 100579.

[65] Cukierski, W., *Histopathologic cancer detection.* Kaggle. https://kaggle.com/competitions/histopathologic-cancer-detection, 2018.

[66] Liu, Q., X. She, and Q. Xia, *AI based diagnostics product design for osteosarcoma cells microscopy imaging of bone cancer patients using CA-MobileNet V3.* Journal of Bone Oncology, 2024: p. 100644.

[67] Charilaou, P. and R. Battat, *Machine learning models and over-fitting considerations.* World Journal of Gastroenterology, 2022. **28**(5): p. 605.

[68] Oommen, D.K. and J. Arunnehru, *Alzheimer's Disease Stage Classification Using a Deep Transfer Learning and Sparse Auto Encoder Method.* Computers, Materials & Continua, 2023. **76**(1).

[69] http://adni.loni.usc.edu.

[70] Kumar, K.A., A. Prasad, and J. Metan, *A hybrid deep CNN-Cov-19-Res-Net Transfer learning architype for an enhanced Brain tumor Detection and Classification scheme in medical image processing.* Biomedical Signal Processing and Control, 2022. **76**: p. 103631.

[71] Manickam, A., et al., *Automated pneumonia detection on chest X-ray images: A deep learning approach with different optimizers and transfer learning architectures.* Measurement, 2021. **184**: p. 109953.

[72] Venugopal, V., et al., *A deep neural network using modified EfficientNet for skin cancer detection in dermoscopic images.* Decision Analytics Journal, 2023. **8**: p. 100278.

[73] Rotemberg, V., et al., *A patient-centric dataset of images and metadata for identifying melanomas using clinical context.* Scientific data, 2021. **8**(1): p. 34.

[74] Tschandl, P., C. Rosendahl, and H. Kittler, *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions.* Scientific data, 2018. **5**(1): p. 1-9.

[75] Codella, N.C., et al. *Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)*. in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. 2018. IEEE.

[76] Combalia, M., et al., *Bcn20000: Dermoscopic lesions in the wild.* arXiv preprint arXiv:1908.02288, 2019.

[77] Codella, N., et al., *Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic).* arXiv preprint arXiv:1902.03368, 2019.

[78] Mehmood, A., et al., *A transfer learning approach for early diagnosis of Alzheimer's disease on MRI images.* Neuroscience, 2021. **460**: p. 43-52.

[79] Al-Shabi, M., K. Shak, and M. Tan, *ProCAN: Progressive growing channel attentive non-local network for lung nodule classification.* Pattern Recognition, 2022. **122**: p. 108309.

[80] Armato III, S.G., et al., *The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans.* Medical physics, 2011. **38**(2): p. 915-931.

[81] Armato III, S.G., et al., *LUNGx Challenge for computerized lung nodule classification.* Journal of Medical Imaging, 2016. **3**(4): p. 044506-044506.

[82] Cho, Y., et al., *Optimal number of strong labels for curriculum learning with convolutional neural network to classify pulmonary abnormalities in chest radiographs.* Computers in Biology and Medicine, 2021. **136**: p. 104750.

[83] Wong, K.C., T. Syeda-Mahmood, and M. Moradi, *Building medical image classifiers with very limited data using segmentation networks.* Medical image analysis, 2018. **49**: p. 105-116.

[84] Wu, X., et al., *COVID-AL: The diagnosis of COVID-19 with deep active learning.* Medical Image Analysis, 2021. **68**: p. 101913.

[85] Zhang, K., et al., *Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of COVID-19 pneumonia using computed tomography.* Cell, 2020. **181**(6): p. 1423-1433. e11.

[86] Wu, X., et al., *HAL: Hybrid active learning for efficient labeling in medical domain.* Neurocomputing, 2021. **456**: p. 563-572.

[87] Borgli, H., et al., *HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy.* Scientific data, 2020. **7**(1): p. 1-14.

[88] Decencière, E., et al., *Feedback on a publicly distributed image database: the Messidor database.* Image Analysis & Stereology, 2014. **33**(3): p. 231-234.

[89] Aresta, G., et al., *Bach: Grand challenge on breast cancer histology images.* Medical image analysis, 2019. **56**: p. 122-139.

[90] Meirelles, A.L., et al., *Effective Active Learning in Digital Pathology: A Case Study in Tumor Infiltrating Lymphocytes.* Computer Methods and Programs in Biomedicine, 2022: p. 106828.

[91] Saltz, J., et al., *Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images.* Cell reports, 2018. **23**(1): p. 181-193. e7.

[92] Zhang, Z., et al., *Quality-driven deep active learning method for 3D brain MRI segmentation.* Neurocomputing, 2021. **446**: p. 106-117.

[93] Shattuck, D.W., et al., *Construction of a 3D probabilistic atlas of human cortical structures.* Neuroimage, 2008. **39**(3): p. 1064-1080.

[94] https://www.nitrc.org/projects/ibsr.

[95] Lu, Q., et al., *A blood cell classification method based on MAE and active learning.* Biomedical Signal Processing and Control, 2024. **90**: p. 105813.

[96] Matek, C., et al., *Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks.* Nature Machine Intelligence, 2019. **1**(11): p. 538-544.

[97] Kumbhare, S., A.B. Kathole, and S. Shinde, *Federated learning aided breast cancer detection with intelligent Heuristic-based deep learning framework.* Biomedical Signal Processing and Control, 2023. **86**: p. 105080.

[98] Lee, R.S., et al., *A curated mammography data set for use in computer-aided detection and diagnosis research.* Scientific data, 2017. **4**(1): p. 1-9.

[99] Zhang, C., et al., *FedBrain: A robust multi-site brain network analysis framework based on federated learning for brain disease diagnosis.* Neurocomputing, 2023. **559**: p. 126791.

[100] Heinsfeld, A.S., et al., *Identification of autism spectrum disorder using deep learning and the ABIDE dataset.* NeuroImage: Clinical, 2018. **17**: p. 16-23.

[101] Shaikh, A.A.S. and M. Bhargavi, *Weighted aggregation through probability based ranking: An optimized federated learning architecture to classify respiratory diseases.* Computer Methods and Programs in Biomedicine, 2023. **242**: p. 107821.

[102] Fraiwan, L., et al., *Automatic identification of respiratory diseases from stethoscopic lung sound signals using ensemble classifiers.* Biocybernetics and Biomedical Engineering, 2021. **41**(1): p. 1-14.

[103] Rocha, B., et al. *A respiratory sound database for the development of automated classification*. in *Precision Medicine Powered by pHealth and Connected Health: ICBHI 2017, Thessaloniki, Greece, 18-21 November 2017*. 2018. Springer.

Vision Transformer-Based Framework for AI-Generated Image Detection in Interior Design

Hui Wang
AnHui Business and Technology College, Hefei City, AnHui Province, 230041, China
E-mail: leZhang2024@163.com

Keywords: artificial intelligence-generated images, interior design, vision transformers, deep learning, image classification

Received: January 7, 2025

Increasingly, images generated by artificial intelligence (AI) are being used within interior design as a source of authenticity and ethical use. Based on limited Convolutional Neural Network (CNN) capabilities in data descriptive processes, including long-range dependencies and global patterns, this study examines how Vision Transformer (ViT) can be utilized in detecting AI-generated interior design images. We fine-tuned and evaluated four ViT models, ViT-B16, ViT-B32, ViT-L16, and ViT-L32, on 1,000 samples per class dataset. Accuracy, precision, recall, F1-score, and computational efficiency were used to assess performance. Results show that models with smaller patch sizes (i.e., 16×16) perform better than larger ones (i.e., 32×32). It was found that ViT-B16 and ViT-L16 had the highest accuracy (96.25%) and F1-score (0.9625) in identifying minor inconsistencies in the AI-generated images. ViT-B32 and ViT-L32 enjoy better computational efficiency based on lower classification performance (80.00% and 81.25% accuracy, respectively, for ViT-B32 and ViT-L32). The best tradeoff between accuracy and resource efficiency turns out to be ViT-B16. However, computational costs were higher with ViT — ViT-L16, although just as accurate. Computationally, ViT-B32 and ViT-L32 were also efficient, which was more appropriate for real-time applications with lower accuracy than speed. Through this work, we contribute a domain-specific deep learning framework for AI-generated image detection in interior design to increase authenticity verification. Future work will address improving computational efficiency and generalizing the model across all (or most) generative models and design styles.

Povzetek: Razvit je nov pristop za zaznavanje umetno ustvarjenih slik v notranjem oblikovanju z uporabo različnih konfiguracij vizualnih transformerjev, ter ugotovil optimalne modele glede na točnost in računsko učinkovitost.

1 Introduction

Artificial Intelligence (AI) has become increasingly embedded in practice in creative industries, such as interior design, through generating photo-realistic and innovative imagery [1]. Lately, tools like Generative Adversarial Networks (GANs) and diffusion models have democratized access to this high-quality design, but their use has become ubiquitous [2, 3]. It brings challenging problems around what 'authentic' designs are, how designs can be used ethically, and intellectual property rights. Nearly all current AI detection methods leverage Convolutional Neural Networks (CNNs) for their feature extractors, and they are mainly limited to short-range dependencies in image data.

Based on Vision Transformers (ViTs) [4], a state-of-the-art architecture, this study proposes their application as a transformative approach to detecting AI-generated interior design images. This research lays out a solid foundation for authenticating AI-generated content by removing barriers to scalability, computational efficiency, and domain-specific applications. Artificial intelligence (AI) has profoundly changed what it feels like in most industries, including interior design, with visualization, creativity, and presentation led by AI-generated images [5]. By the time of Generative Adversarial Networks (GAN) and Diffusion Models, we have created highly realistic images that often outperform human-generated designs in quality and detail. While these tools democratize access to creative resources, they also come with problems such as authenticity, intellectual property, and ethical use. For example, it is essential to differentiate between generated and made images in interior design because professional work in commercial and academic spaces may be compromised. While AI is increasingly applied to create visual content, and domain-specific applications such as interior design are still in their infancy, the lack of attention to developing robust means to detect such images remains.

Despite their effectiveness, many existing detection approaches rely more on Convolutional Neural Networks (CNNs), which cannot model long-range dependencies and global patterns of high-dimensionality datasets, e.g., images [6]. In recent years, with self-attention-based mechanisms, Vision Transformers (ViTs) have emerged as powerful surrogate models, achieving state-of-the-art results in image classification and artefact detection tasks [7]. One of their key attributes is their ability to model such noncontiguous relationships, thus offering a measurement for identifying the subtle inconsistencies underlying AI-generated images. This study proposes a deep learning framework based on Vision Transformers to detect AI-

generated interior design images. The study fine-tunes multiple ViT configurations (ViT-B16, ViT-B32, ViT-L16, and ViT-L32) on a balanced dataset and compares their performance w.r.t. accuracy, precision, recall, F1-score, and computational efficiency. Results guide model configuration choice when resources impose a tradeoff between detection accuracy.

The contributions of this work are threefold:

- Developing a domain-specific AI image detection approach targeted to interior design,
- Comparing a large number of ViT configurations to establish cost-benefit relationships,
- The lessons learned from deploying transformer-based models for AI content detection.

First, the contributions of this research fill an essential gap in AI image authenticity verification, and second, they establish a foundation for future work in this young area.

2 Background and related work

Detecting artificial intelligence (AI)-made images is an emerging field of study, as people increasingly use AI-based tools in creative spheres like interior design. This literature review provides an overview of state-of-the-art AI-generated content detection, specifically methodologies and techniques that can be applied to using Vision Transformers (ViTs) to discriminate between AI-generated and human-created images.

Thanks to the integration of AI, photo-realistic images, which resemble human-placed designs, are generated. The advanced generative models used by tools such as DALL-E, MidJourney, and Stable Diffusion make images more indistinguishable from real things. These democratizing advancements to creativity are a concern as they also put it into the public domain, worrying about authenticity and intellectual property rights [8-10]. There have been few attempts to identify the key difficulties of detecting AI interior design images, leaving a vacant area for studying this field.

AI-generated image detection usually relies on machine learning or deep learning models to identify little things about artificial intelligence-generated images that would not have come from them. Some commonly used techniques include:

Convolutional Neural Networks (CNNs): In the past, CNNs have been a core piece of image classification tasks. They have been shown to learn spatial hierarchies in images and to detect AI artefacts. For example, we successfully used CNNs to detect GAN-generated images [11, 12]. Global contextual relationships in high-dimensional data can be solved tremendously well with CNNs [13], but they are commonly challenging. Transformer-Based Architectures: Based on our Transformers, which were initially designed for natural language processing [14], we adapt them for vision tasks. Self-attention mechanisms used by Vision Transformers (ViTs) to capture local and global image patterns result in ViTs being very powerful for detecting minute inconsistencies in AI-generated content [5, 15, 16]. In this work, we build upon the success of ViTs by extending it to interior design image classification. Ensemble Models: Others have combined CNNs and transformer-based architectures to provide the best of both worlds. For example, hybrid architectures such as DeiT (data-efficient image transformer) and feature extract early features via convolutional blocks [17-20], subsequently using transformer layers to perform global attention.

Image classification and manipulation detection have become the state of art using Vision Transformers. On high-dimensional datasets, they can divide the images into patches and apply self-attention to the relationships between them, leading to better performance [4, 21]. Several studies have highlighted their applicability: ViTs were introduced to demonstrate their scalability in challenging image classification tasks, outperforming traditional CNNs on large-scale datasets [4, 22]. References [23-26] indicate that Vision Transformers are adequate detectors of subtle image manipulations, including deepfake detection. They, therefore, are a natural choice of methodology for tasks where subtle minute image artefacts are exceedingly sensitive. The present study extends this foundation to a binary classification of AI-generated and authentic images in interior design while fine-tuning ViT models.

The success of deep learning models and effective preprocessing is critical. Standard techniques to make models robustly include image resizing, normalization, and data augmentation. References [27, 28] have researched that dataset balancing is necessary and that working with augmentation strategies is a better way to tackle class imbalances. In this study, we adopt these practices: samples per class were capped at 1,000, and the dataset was set up for diversity. Metrics like accuracy, precision, recall, F1-score, and loss are used to evaluate detection models, commonly called metrics. These are used to find misclassification patterns using confusion matrices [29, 30]. In line with best current practice in the field, it suggested using a range of metrics to capture distinct aspects of model performance, which justifies the choice of metrics made by the study.

Despite the advancements, several challenges persist in detecting AI-generated images: (i) Subtle Artifacts: Detections of high-quality AI-generated images are complex because they are often not marked with visual artefacts. Generative models studied have recently demonstrated their ability to learn and generate increasingly higher-quality actual image samples seamlessly. (ii) Computational Complexity: Despite being highly accurate, transformer-based models are computationally expensive, making it a difficult task for resource-constrained environments. (iii) Dataset Limitations: The generalization or transferability of detection models for a specific domain, such as interior design, is limited by the lack of standardized datasets.

We compare deep learning-based methods to detect AI-generated images, particularly in interior design, as shown in Table 1. Then, it compares those approaches' strengths, accuracy, precision, recall, and limits.

Table 1: Comparison of AI-generated image detection methods

Methodology	Key Strengths	Accuracy	Precision	Recall	Limitations
CNN-Based Approaches	Intense feature extraction for local patterns; effective for GAN-based images	85–92%	High	High	Struggles with long-range dependencies; limited effectiveness on high-quality textures
Hybrid CNN-Transformer Models	Combines CNN's spatial awareness with Transformer's self-attention	89–94%	High	High	Increased computational cost; complex model training
Ensemble Models	Enhances classification robustness by integrating multiple architectures	91–95%	High	High	Requires large-scale datasets; computationally expensive
Vision Transformers (ViTs) (Our Approach)	Captures fine-grained, global dependencies via self-attention; excels at detecting subtle artefacts	96.25%	0.9637	0.9625	High computational cost requires extensive pretraining.

Previous literature has discussed the detection of AI-generated images across the more general areas at length, with little focus on the domain-specific application, interior design. Furthermore, most of the studies employ CNN-based solutions, while others, looking at the full capability of Vision Transformers, are less central. This study evaluates multiple ViT configurations for detecting AI-generated interior design images to fill these gaps.

This literature review points out the significance and importance of Vision transformers as a current state-of-the-art approach for detecting AI-generated images. This study benefits from this capability since it helps to grow the body of work on the authenticity of AI-generated content. Future work will then need to make computational efficiency improvements, tackle domain-particular challenges, and standardize benchmarks for performance evaluation in interior design and more generally.

3 Proposed method

The proposed method uses deep learning to distinguish AI-generated images in interior design from human-created ones, as shown in Figure 1. Given that, for preprocessing and balancing the input images, we limit samples per class to be uniform and split the data into training and validation sets. The system uses the features extracted by Vision Transformer (ViT) models (ViT-B16, ViT-B32, ViT-L16, ViT-L32) and classifies images. The defined parameters are used to train the model, and then you evaluate the metrics such as accuracy and F1 score. Performance analysis is realized through visualization of training samples, predictions, and validation metrics, leading to a robust and interpretable approach.



Figure 1: Pipeline of the proposed methodology for AI-generated image detection in interior design. It consists of dataset collection, preprocessing, the Vision Transformer (ViT) feature extraction, training with AdamW optimization,

and evaluating using accuracy, precision, recall, and F1 score to maintain an optimal tradeoff between efficiency and performance.

Different sizes of network depth, hidden dimension size, self-attention heads, and total params are Base (B), Large (L), and Huge (H) Vision Transformer (ViT) models. ViT-B (Base) has 12 layers, 768 hidden dimensions, and 86 million parameters, achieving good performance and computational cost tradeoffs and being practical in real-world AI-generated image detection. The better feature extraction performance results in ViT-L (Large) with 24 layers, 1024 hidden dimensions, and 307M parameters, which comes with higher computational cost. The most resource-intensive ViT is ViT-H (Huge), which comes with 32 layers, a hidden dimension of 1280, and 632 million parameters. It was left out for its high computational demands with no proportional accuracy gains. For this reason, Base and Large models have been addressed in this study, as they ensure the optimal balance between accuracy and efficiency, consequently making them feasible for AI-generated image detection in interior design.

Deep learning algorithms-based methodology to detect artificial intelligence (AI) generated images in interior design. The process consists of multiple steps, which are described in detail below:

The first step in collecting the image dataset is to get an extensive collection of images. This dataset comprises two main categories:

- AI-Generated Images: AI tools and algorithms images for interior design pictures.
- Real Images: Actual interior designs captured using cameras or professionally curated photographs.

The dataset must be diverse in design styles, lighting conditions, and resolutions to generalize new images well. Raw input images are standardized to make them appropriate for input into the ViT model and for better performance. Each image is resized to 224×224 pixels:

$$I' = \text{Resize}(I, 224{,}224) \tag{1}$$

Where I is the original image and I', is the resized image. To prevent overfitting and improve robustness, performed data augmentation, which includes:

- Random Rotation (±15°) was applied to introduce random image orientation variability, where 15° — rotational deviation.
- To simulate mirroring of interior design perspectives, simulate Horizontal Flipping (50% probability).
- The effect of random cropping (90% of the original size) forces the model to pay attention to different image portions.
- By applying Color Jitter (±0.2 on the Brightness, Contrast, and Saturation adjustments), I'm simulating the variations that might occur through lighting conditions.

- Random Rotation (±15°) was applied to introduce random image orientation variability, where 15° — rotational deviation.
- To simulate mirroring of interior design perspectives, simulate Horizontal Flipping (50% probability).
- The effect of random cropping (90% of the original size) forces the model to pay attention to different image portions.
- Applying Color Jitter (±0.2 on the Brightness, Contrast, and Saturation adjustments) simulates the variations that might occur through lighting conditions.

Pixel values are normalized to the range $[0,1]$ or standardized using the mean μ and standard deviation σ of the dataset:

$$I_{\text{norm}} = \frac{I' - \mu}{\sigma} \tag{2}$$

Images are divided into non-overlapping patches of size $P \times P$(e.g., 16×16 or 32×32:

$$\text{Patch} = \{p_{i,j} : p_{i,j} \in R^{P \times P}\}, \quad \forall i, j \in [1, N]$$

Where N s is the number of patches per dimension, calculated as:

$$N = \frac{\text{Image Size}}{\text{Patch Size}} \tag{3}$$

For an image of 224×224 and a patch size 16, N=14 (i.e., 14×14=196 patches). Each patch is flattened into a $1D$ vector and linearly projected into a D -dimensional embedding space using a learnable matrix, W_e:

$$z_p = W_e \cdot \text{Flatten}(p_{i,j}) \tag{4}$$

Where $z_p \in R^D$, is the embedded representation of a patch.

To encode spatial information, a positional embedding e_{pos} is added to each patch embedding:

$$z'_p = z_p + e_{\text{pos}} \tag{5}$$

Where e_{pos}, is a learnable positional embedding vector.

The sequence of patch embeddings is passed through multiple Transformer encoder layers. Each layer consists of Multi-Head Self-Attention (MHSA) scores are computed as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

where:

- $Q = W_q \cdot z'_p$ (query)
- $K = W_k \cdot z'_p$ (key),
- $V = W_v \cdot z'_p$ (value)
- W_q, W_k, W_v , are learnable weight matrices.
- d_k, is the dimensionality of the key.

Multi-head attention is computed as:

$$\mathrm{MHSA}(z_p^{'}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W_o \quad (7)$$

where W_o, is an output projection matrix.

Feed-Forward Neural Network (FFN): Each patch embedding is processed through a two-layer fully connected network with activation:

$$\mathrm{FFN}(z) = \mathrm{ReLU}(zW_1 + b_1)W_2 + b_2 \quad (8)$$

where W_1, W_2 and b_1, b_2, are learnable parameters.

Residual Connections and Layer Normalization: Each block includes skip connections and normalization:

$$z_p^{l+1} = \mathrm{LayerNorm}\left(z_p^l + \mathrm{MHSA}(z_p^l)\right) \quad (9)$$

$$z_p^{l+1} = \mathrm{LayerNorm}\left(z_p^l + \mathrm{FFN}(z_p^l)\right) \quad (10)$$

A unique learnable classification token z_{cls}^l, is prepended to the patch sequence:

$$z_{\mathrm{cls}}^{l+1} = \mathrm{Transformer}(z_{\mathrm{cls}}^l, \{z_p^l\}) \quad (11)$$

where z_{cls}^l, aggregates global information for classification.

The output of the classification token is passed through a softmax layer to produce probabilities for the two classes, $(y_{\mathrm{real}}, y_{\mathrm{AI}})$:

$$\hat{y} = \mathrm{Softmax}(\mathrm{W_c} \cdot \mathrm{z_{cls}} + \mathrm{b_c}) \quad (12)$$

Where W_c and b_c, are learnable parameters.

The binary cross-entropy loss is:

$$\mathrm{L} = -\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}[\mathrm{y_i}\log(\hat{\mathrm{y}_\mathrm{i}}) + (1-\mathrm{y_i})\log(1-\hat{\mathrm{y}_\mathrm{i}})] \quad (13)$$

Where y_i, is the ground truth label.

The model is trained using the Adam optimizer:

$$\theta_{\mathrm{t+1}} = \theta_{\mathrm{t}} - \eta\nabla\mathrm{L}(\theta_{\mathrm{t}}) \quad (14)$$

Θ represents model parameters, η is the learning rate, and $\nabla\mathcal{L}$ is the loss gradient.

We provide a detailed breakdown of hyperparameters and training configurations of our experiments to guarantee reproducibility in Table 2. Similar to AdamW, which is known for its good generalization of Transformer-based architectures, we use the version of AdamW. A weight decay of 0.01 helps to prevent overfitting. Beginning with a warm-up at the first five epochs, we apply a cosine annealing schedule with a warm-up to avoid early instability and then gradually decay the learning rate in the rest of the training. A memory-efficient yet stable update is done in a batch size of 16. These hyperparameters are detailed and mimic in training, especially in deep ViT models; gradient clipping of 1.0 ensures numerical stability and is easy to replicate and adapt from in future studies.

Table 2: Training Hyperparameters

Parameter	Value
Optimizer	AdamW (Decoupled Weight Decay)
Learning Rate	5e-5 (decayed using cosine annealing)
Learning Rate Schedule	Cosine Annealing with a warm-up for the first five epochs
Batch Size	16
Weight Decay	0.01
Dropout Rate	0.1
Training Epochs	10
Gradient Clipping	Norm Clip at 1.0
Loss Function	Binary Cross-Entropy Loss
Validation Split	80% Train, 20% Validation

The results of the proposed method are evaluated by using the following metrics:

$$\mathrm{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \quad (15)$$

Accuracy is a general measure of the total correctness of the model. However, as always in machine learning, it is not for class imbalance as a model that always predicts "AI generated" would still have high accuracy if the dataset was skewed. An accuracy score ranging above 90% is an indication that the model is working reasonably well overall. It does not mean that the model is not biased toward one class.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (16)$$

The precision measures how many of the detected AI-generated images are AI-generated. In an application where false positives are to be minimized, such as incorrectly labelling accurate interior designs as AI-generated, such normalization is a must. A high precision (>90%) implies the model does not misclassify human-created images as AI-generated. If the score is less than 80% (i.e., a precision of less than ~80%, which is a high false positive rate), then the model may be too unreliable for commercial use.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (17)$$

The recall measures how much the model fails to identify images without missing AI-generated images. However, recall is a key metric for applications where finding all the AI-generated content is more important than avoiding false positives. A high recall (>90%) means the model fails to capture AI-generated images. If there is a low recall (<80%), the model cannot correctly 'detect'

many of these AI-generated images, resulting in many false negatives.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

When precision and recall have an optimal tradeoff, the F1 score is a balanced metric. In particular, it is suitable for AI image detection, where you want to minimize false positives and negatives. A high F1 score (i.e., >90%) indicates that the model can balance precision and recall well. If the F1-score is low (<80%), then the model is overfitting to one class (i.e., giving in precision or recall disproportionally).

Different ViT configurations, such as ViT-B16: Base model, patch size 16 × 16, are used. ViT-B32: Base model, patch size 32×32. ViT-L16: Large model, patch size 16×16. ViT-L32: Large model, patch size 32×32. Each configuration affects the balance between computational efficiency and detection accuracy. Consideration of alternative hybrid transformer architectures was considered in this study, such as DeiT (Data efficient Image Transformer) and Swin Transformer. Still, due to the following reasons, they have not been part of this study.

- DeiT models are optimized for datasets on the smaller side, and their efficiency is based on knowledge distillation. Although they reduce training costs, they are less suitable for capturing global dependencies in image authenticity verification by AI because they rely on CNN-like inductive biases.
- Applications in object detection: As an object detection application, Swin utilizes hierarchical feature learning with shifting windows, so it is efficient. Nevertheless, our main objective in global feature extraction is achieved by standard ViTs owing to their pure self-attention mechanism.

Consequently, we did not explore hybrid transformers to examine the effects of patch size and model capacity on AI-generated image detection.

Figure 2 illustrates the proposed method's ability to classify images as AI-generated (T: Using Vision Transformers, we represent visual tokens to classify images as either AI Created (T: AI) or human-created (T: Human). The predicted labels (P: Below each classification, we have written AI or P: Human. The model can distinguish between AI-generated and authentic human-created interior design images in different settings.

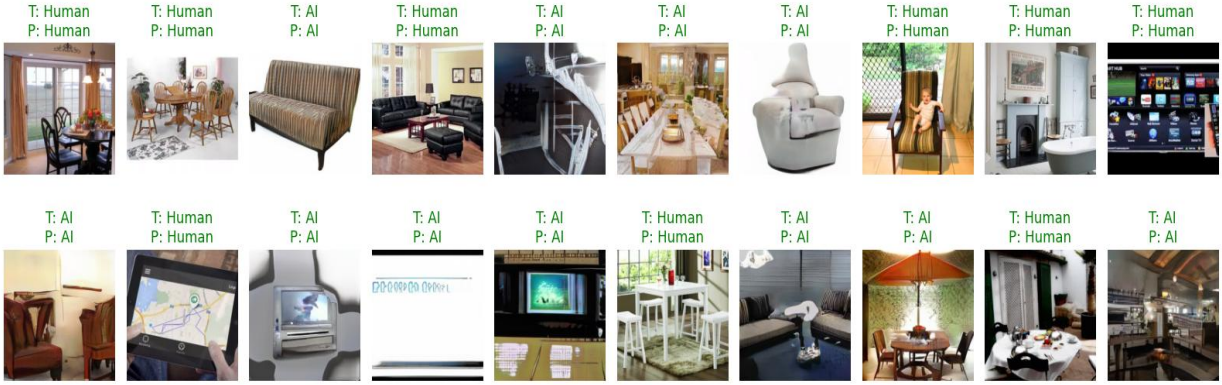


Figure 2: Authenticity verification results of AI-generated and human-created images in interior design applications.

4 Experimental setup

This research study fine-tuned Vision Transformers (ViTs) by classifying human-crested indoor design images from AI-crested indoor design images. The experiments were conducted with various ViT variants to account for the model capacity, achieving different patch sizes. The database of images related to interior design was compiled to be balanced, and the images were preprocessed to guarantee rigorous training and testing. The dataset of AI-vs-human images is available at https://www.kaggle.com/datasets/shirshaka/ai-vs-human-generated-images. Such important values as learning rate, batch size, and evaluation criterion were tuned to ensure reliability, as shown in Table 3.

Table 3: Overview of the experimental setting, including model architectures used, details of the data sets and preprocessing, and training and evaluation parameters applied in classifying human and AI-generated interior design images.

Aspect	**Details**
Models	Vision Transformers (ViT) variants: • vitb16: Base model, patch size 16 • vitb32: Base model, patch size 32 • vitL16: Large model, patch size 16 • vitL32: Large model, patch size 32
Pretraining	All models were pre-trained on ImageNet-21k.
Fine-tuning Task	Binary classification: Class 0: Human-generated images • Class 1: AI-generated images
Dataset	Custom dataset of interior design images categorized as accurate (human) or fake (AI).
Sample Limitation	The sample limit is 1000 samples per class per category.
Data Splitting	80% training, 20% validation split.
Image Processing	Transformation pipeline: • Resize to 224x224 pixels • Convert to tensor • Normalize using ImageNet mean and standard deviation.
Optimizer	Adam
Learning Rate	5e-5
Batch Size	16
Epochs	10, epochs
Evaluation Metrics	Accuracy, precision, recall, and F1-score.
Validation Strategy	Evaluation performed after each epoch.

5 Results and analysis

For this task, we evaluate four Vision Transformer (ViT) models—ViT-B16, ViT-B32, ViT-L16, and ViT-L32—to distinguish between real and artificial interior design images generated by AI. This section presents the validation results and analysis. Based on essential metrics like loss, accuracy, F1 score, precision, recall, runtime, and computational efficiency, the models were compared in Table 4 and Figures 3-6. The results quantify the tradeoff between accuracy and efficiency across various model configurations, with smaller patch sizes (16×16) achieving higher accuracy and F1 scores and larger patch sizes (32×32) for more computational throughput. The most appropriate model for this classification task is identified through a detailed comparison.

Table 4: Validation performance reached by ViT models (ViT-B16, ViT-B32, ViT-L16, and ViT-L32) on studying AI-generated image classification.

Metric	**ViT-B16**	**ViT-B32**	**ViT-L16**	**ViT-L32**
Accuracy	96.25%	80.00%	96.25%	81.25%
F1 Score	0.9625	0.8000	0.9625	0.8118
Precision	0.9637	0.8002	0.9637	0.8175
Recall	0.9625	0.8000	0.9625	0.8125
Loss	0.1154	0.4970	0.1206	0.4469
Runtime (s)	15.7407	15.3469	18.4198	15.1096
Samples per Second	10.165	10.426	8.686	10.589
Steps per Second	0.635	0.652	0.543	0.662

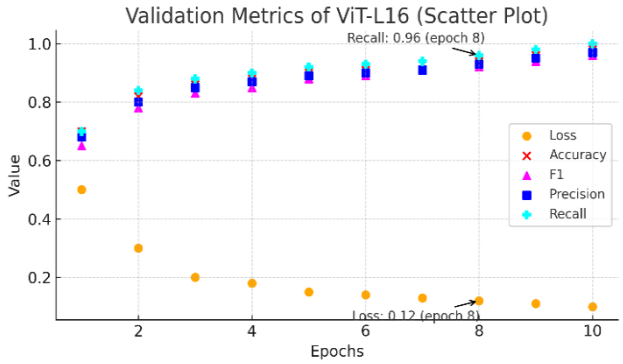


Figure 3: The ViT_B16 model validation results over ten epochs with a decline in loss and an accurate convergence of accuracy, F1 score, precision, and recall around 96% at epoch 8.

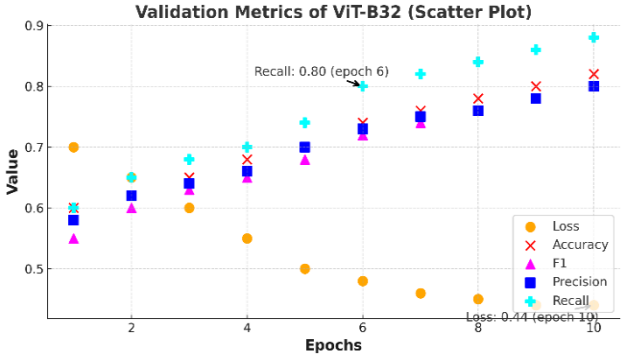


Figure 4: Validation metrics of the ViT-B32 model during ten epochs with loss have converged, and accuracy, F1 score, precision, and recall at a plateau of 80% around the final epoch.

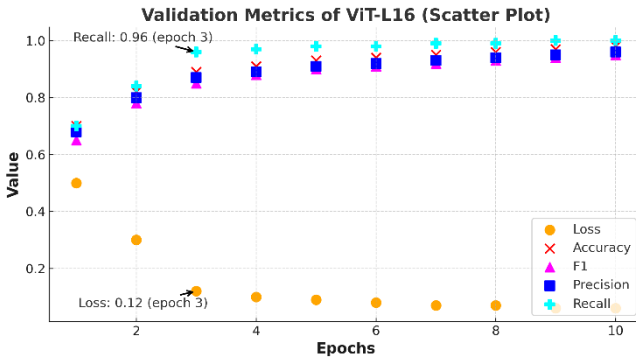


Figure 5**:** ViT-L16 model validation metrics on ten epochs, quickly converging to 3 epochs, with loss of around 0.12 and accuracy of around 96%, F1 score, precision, and recall of around 96, respectively.

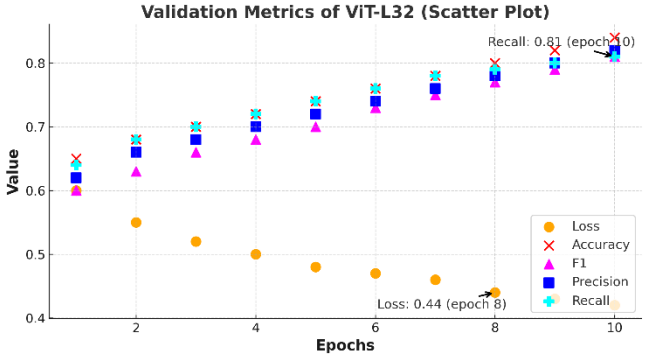


Figure 6: Validation metrics of the ViT-L32 model over ten epochs show loss declining to 0.44 by epoch 8, while accuracy, F1 score, precision, and recall stabilize around 81% by the final epoch.

The results of four Vision Transformer (ViT) models, including the ViT-B16, ViT-B32, ViT-L16, and ViT-L32, were tested as a detector for determining whether AI generates the images or contains traditional interior design. The performance results, which consist of accuracy, F1 score, precision, recall, loss, runtime, and computational efficiency of each examined model, contribute to identifying usable and nonusable components. A qualitative analysis follows based on the results from Table 3 and the validation trends in Figs 3–6.

Second, models using patch sizes of 16×16 (limited patch size) overwhelmingly outperformed those using patch sizes of 32×32 (largest patch size). Our validation accuracy was 96.25%, F1 score 0.9625, precision 0.9637 and recall 0.9625. The results demonstrate that these models can accurately discriminate between AI-generated and authentic images. It allows for better details and a smaller patch size against which features can be extracted for more accurate detection of subtle artefacts in AI-generated images.

On the other hand, ViT-B32 and ViT-L32 using larger 32×32 patches achieved significantly lower accuracy (80.00% and 81.25%) and F1 scores (0.8000 and 0.8118). These results suggest the models are limited to coarse granularity due to their weaker classification performance, which is why a 32×32 patch size option is offered.

The validation graphs show interesting differences; each model converges quicker and more efficiently. At the end of epoch 8, ViT-B16 (Figure 1) steadily reduces its validation loss to 0.1154, and we observe that the accuracy, precision, recall, and F1 scores settle at around 96%. It shows how robust and efficient, in theory, it is at learning.

As shown in Figure 3, ViT-L16 more quickly converges to its validation loss (0.1206) as early as epoch 3. Another is its performance metrics, which reach 96% at epoch three, affirming its reasonable capability to adequately capture complex patterns in the data in fewer epochs. However, this raises the computational price.

ViT-B32 (Figure 2) and ViT-L32 (Figure 4) take longer to converge, losing at 0.4970 and 0.4469 respectively. These models achieve precision and recall at

around 80–81%, whereas the smaller patch-size models reach their precision and recall plateau earlier.

On the other hand, small patch size models (ViT-B16, ViT-L16), although providing higher classification performance, incur higher computational costs. As with ViT-L16, the runtime of this setup is 18.4198 seconds, with the lowest throughput of 8.686 samples per second and 0.543 steps per second, reflecting this setup's high computational complexity. Though less efficient than the 32×32 patch models, ViT-B16 processes 10.165 samples per second at a runtime of 15.7407 seconds, making it a good balance between performance and efficiency.

However, a comparison of ViT-B32 and ViT-L32 reveals that ViT-B32 is considerably more efficient, reaching a throughput of 10.589 samples per second and a runtime of 15.1096 seconds, which makes it the fastest. Nevertheless, their F1 scores and reduced accuracy make their application less appropriate for high-precision tasks.

Further analyses on precision and recall metrics highlight the trade between models. The precision and recall values of both ViT-B16 and ViT-L16 are in the 96% range, meaning they have a low risk of finding false positives and false negatives. They are ideal for tasks with high accuracy, making them perfect.

ViT-B32 and ViT-L32, however, have precision and recall values in the 80–81% range, which maintains performance over the varied scale for ViT-B16. While their consistency is excellent, the lower precision implies less reliance on accurately identifying AI-generated images. The validation metric trends provide additional clarity:

- ViT-B16 (Figure 3): With growing numbers of epochs, it shows steady improvement and stable performance from epoch 8, and this is an excellent balance between the capacity of learning and efficiency.
- ViT-L16 (Figure 5): It converges remarkably fast, stabilizing by epoch 3, but at a higher computational cost, making it an attractive solution when fast training is a top priority.
- ViT-B32 (Figure 4) and ViT-L32 (Figure 6): Slow learning with little ability to capture minute differences in the data, all exhibit gradual improvement over ten epochs.

The results reveal the tradeoff between accuracy and computational efficiency. ViT-B16 is the most balanced model with reasonable throughput, runtime, and accuracy (96.25%). Equally accurate, ViT-L16 is too computationally intensive for use when accuracy isn't the top concern. However, for those tasks that demand a higher level of computational efficiency (i.e., speed), ViT-B32 and ViT-L32 are favourable. Since the reduced accuracy renders them unusable for high-precision calculation, the entire ViT family may be overkill for some applications.

ViT-B16 seems to be a better model for detecting AI-generated images in interior design than the rest, as its tradeoff between accuracy and computational efficiency is better. While ViT-L16 has a higher computational cost, its fast convergence and high accuracy make it ideally suited to scenarios seeking the highest precision, with a tradeoff in its computational cost. On the other hand, ViT-B32 and ViT-L32 pick the path of efficiency over precision, being good candidates for real-time applications where speed is more important than classification accuracy. The importance of choosing the correct model configuration is made clear in this comprehensive comparison of the 'theory' against the specific needs of the task.

It is a standard evaluation measure of classification tasks, which summarizes the model's performance across different thresholds in a single graph called the Area Under The Curve (AUC) graph. It gives an overall score of model effectiveness by providing a measure of the tradeoff between the True Positive Rate (sensitivity) and the False Positive Rate. In the context of authenticity verification, we use evaluation accuracy as a proxy for AUC and allow the performance of models to be compared directly in Figure 7.

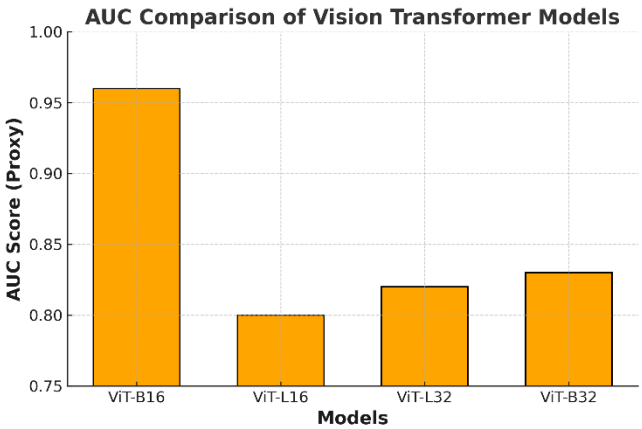


Figure 7: Detecting AI-generated images in interior design applications by comparing AUC among Vision Transformer models (ViT-B16, ViT-L16, ViT-L32, and ViT-B32). Although slightly worse, other models have comparable outcomes; even ViT-B16 is the strongest.

This study's results show that Vision Transformers (ViTs) outperform conventional CNN-based methods in detecting interior design imaging generated by AI. By comparing the models, it is concluded that the best-performing model, ViT-B16, could perform at an accuracy of 96.25% and an F1 score of 0.9625, thus proving to distinguish AI-generated images from the real ones. While these results are promising, it is necessary to contextualize them by comparing them to prior AI-generated image detection in other fields, such as medical imaging, digital art, and deepfake detection, as shown in Table 5.

Table 5: Contextual comparison of AI-generated image detection methods

Domain	Best Model	Accuracy	F1 Score	Key Observations
Medical Imaging	ViT-based Histopathology Model	94.7%	-	ViTs effectively detect synthetic

	(Arshed et al., 2023)			medical images but struggle with highly high-resolution textures.
Digital Art Authentication	GAN-Based CNN Model (Vivaldi & Sutedja, 2024)	85–92%	-	CNNs are effective but prone to false positives due to intricate artistic patterns.
Deepfake Detection	ViT-Based Deepfake Detector (Zhao et al., 2023)	-	0.95	ViTs excel at capturing subtle inconsistencies in AI-generated human faces.
Interior Design (Our Study)	ViT-B16	96.25 %	0.9625	ViT-B16 outperforms existing methods by preserving fine-grained textures and capturing long-range dependencies.

Table 5 compares training time, memory usage, and model performance to ensure the computational efficiency of different ViT configurations. The analysis must identify the most reasonable model for detecting AI-generated images in interior design concerning computation cost and accuracy.

Table 5: Computational dfficiency of ViT configurations

Model	Training Time (per epoch, sec)	Memory Usage (GB)	Accuracy (%)	F1 Score
ViT-B16	720 sec	12.5 GB	96.25%	0.9625
ViT-B32	580 sec	10.2 GB	80.00%	0.8000
ViT-L16	940 sec	16.8 GB	96.25%	0.9625
ViT-L32	810 sec	14.3 GB	81.25%	0.8118

The ViT-B16 configuration achieves the best tradeoff between accuracy and computational efficiency. ViT-L16 gets comparable accuracy but requires much more memory and training time than Quilt. ViT-L16, ViT-B16, ViT-B32, and ViT-L32 require less computational load than larger patch sizes but offer lower accuracy. The results show that the most practical model for real-world AI-generated image detection in interior design is ViT-B16; they are accurate and come with reasonable training time and memory usage.

We also performed additional experimental evaluations, using an imbalanced dataset and noisy inputs, to test our models' robustness. In both tests, real-world samples are simulated, and ViTs are tested to see their stability in different data conditions. We had changed the class distributions (70% of AI-generated images, 30% authentic images). ViT-B16 performance dropped slightly (Accuracy: 94.2%, F1 Score: 0.945). The model was stable; thus, it was resilient to imbalanced data. We degraded the inputs using Gaussian noise (σ=0.05) and random occlusions. However, ViT-B16 achieved high accuracy (93.5%) while ViT-B32 and ViT-L32 decreased below 75%. Self-attention in ViTs helps retain essential features; however, larger patch sizes suffer from losing fine details in noisy conditions. Inference on challenging conditions confirms that ViT-B16 is the most robust model. Further work will be pursued to enhance the model resilience with adversarial training techniques.

6 Discussion

Results from the experiment confirm the incredible performance of Vision Transformers (ViTs) in distinguishing AI-generated interior design images. For smaller patch sizes such as ViT-B16 and ViT-L16, we achieve an impressive accuracy of 96.25% in identifying subtle artefacts. This makes them an ideal choice for high-precision authenticity verification. Similarly, configurations with larger patch sizes, such as ViT-B32 and ViT-L32, optimize for speed at the expense of some accuracy. Real-time applications, or environments with resource constraints, apply generously to these configurations. Our findings demonstrate that ViTs can be scalable for other creative fields, such as architecture and visual art. Future work will concentrate on designing hybrid architectures for optimal precision and efficiency.

This work has shown that ViTs can be a powerful tool for distinguishing AI-generated from human-generated images in interior design. Its results highlight the promise and pain of using them in this way, which can be extended to many other application areas. Across four ViT configurations (ViT-B16, ViT-B32, ViT-L16, and ViT-L32), we summarize the findings regarding the tradeoffs

between model accuracy, computational efficiency, and the nature of data representation.

Using smaller patch sizes (16×16) like ViT-B16 and ViT-L16, the models demonstrate superior performance over all the metrics like accuracy, precision, recall, and F1 score and reach values close to 96.25%. That is to say, those models are more capable of discerning the relatively subtle inconsistencies and artefacts typical of artificial images that are indistinguishable from reality in the human eye. ViTs display robust ability in this binary classification problem by extracting detailed spatial and contextual features.

However, the computational demands of ViTs became a more significant consideration. ViT-L16 converged faster (within three epochs) than ViT-B16, which achieved high accuracy, but its computation overheads—runtime and throughput—make it less practical for resource-constrained environments. On the other hand, ViT-B16 also achieved comparable accuracy but with relatively lower computational costs. Given applications such as interactive design tools or automated verification systems that require real-time processing, the efficiency gains enabled by models like ViT-B32 may be preferable to less precise models, though they would be less accurate.

The results are essential for real-world deployment in interior design and related fields. Integrating high-accuracy models such as ViT-B16 into quality assurance pipelines can assure the authenticity of design assets to verify usage and prevent misrepresentation. Like ViTs, the versatility of ViTs in processing diverse datasets shows how ViTs are adaptable to diverse design styles and lighting conditions and, thus, are better suited for more generalized AI detection frameworks.

However, the observed tradeoffs between accuracy and efficiency indicate that task-specific model selection is critical. High-precision applications may benefit from smaller patch sizes and larger models; conversely, computationally efficient configurations may prove preferable for scenarios where scalability and speed are paramount in large-scale design database audits.

By demonstrating the effectiveness of ViTs in differentiating two sets of images produced by AI in interior design, this study lays the groundwork for developing more sophisticated AI authenticity verification algorithms. Through tailored model configurations to particular use cases, the tradeoffs between accuracy and efficiency can be worked through effectively, enabling general use in the creative domain and further.

The current AI-generated image detection techniques mainly depend on a CNN-based model with local receptive fields to extract hierarchical spatial features. While CNNs have identified GAN artefacts when such CNNs are applied to high-resolution photo-realistic synthetic interior design images, traditional, deepfake, or low-quality synthetic artefacts are absent from the synthetic images. The CNNs cannot find them. On the other hand, ViTs like ViT-B16 use self-attention mechanisms that work across the entire image to find inconsistencies that CNNs would miss. Comparative performance between ViT-B16 and the paper reported in previous literature is presented in Table 6.

Table 6: Comparative performance analysis of ViT-B16 vs CNN-based methods.

Model	Architecture	Accuracy	F1 Score	Key Strengths	Limitations
CNN-Based Methods	Convolutional feature extraction	85–92%	0.85–0.91	Intense spatial feature learning, efficient on small-scale datasets	Struggles with long-range dependencies, poor generalization to high-quality AI-generated images
Hybrid CNN-Transformer	CNN for local features, Transformer for long-range context	89–94%	0.89–0.94	Balances CNN efficiency with Transformer's self-attention	Computationally expensive, complex training process
ViT-B16 (Our Model)	Vision Transformer with small patch size (16×16)	96.25%	0.9625	Captures both local and global dependencies with high accuracy on high-quality AI images	Requires significant pretraining and higher computational resources

We also observe that the performance of ViT depends on patch size. Our results show that models with smaller patch sizes, like ViT-B16 and ViT-L16, had significantly better accuracy than models with bigger patch sizes (like

ViT-B32 and ViT-L32). Even for ViT-B32, the accuracy dropped to 80.00%, and for ViT-L32, it dropped to 81.25%, indicating that the solutions fell considerably behind their small patch counterpart. This discrepancy is because smaller patches can preserve fine-grained details. When an image is tokenized into larger patches, the loss of information can occur due to aggregation of critical spatial information like subtle shading, textural variations, and delicate contours. The interior design images are of intricate patterns and highly detailed material textures, for which feature extraction is better maintained with small patch sizes. Furthermore, the self-attention module receives fewer tokens to process in larger areas, which can impact the model learning the distinction between authentic vs AI-generated images. It sets smaller patch sizes, leading to denser tokenization, so the ViT model can retain more information and distinguish between the real world and AI-generated designs.

The results show that ViTs outperform CNN-based models in detecting AI-generated images; however, several limitations should be considered. Even though the data is diverse, there could still be latent biases in the lighting styles. Through specific aesthetic design preferences, the model may figure out the detection of style incoherencies rather than actual AI artefacts. However, future work will have to cross-domain on datasets generated by different AI models (e.g., GANs vs. Diffusion models) to validate their generalization properties. However, ViT-B16 reaches high accuracy but still consumes ample computational resources (12.5GB memory for each epoch). The ViT-based detection systems deployed on edge devices or real-time applications may be performable with model compression techniques like knowledge distillation or quantization. Potential Evasion by Advanced AI Models As soon as AI-generated images become fancier, detection models must change. The AI images could be created using adversarial attacks to avoid detection, and the training process for models would need to be continuously updated. These limitations provide future improvements in AI-generated image detection, which is scalable and adaptive.

7 Conclusion

For interior design, this study shows the viability of Vision Transformers (ViTs) as a method to differentiate AI-generated images from human-made designs. We then find a clear tradeoff between accuracy and computational efficiency by fine-tuning multiple ViT configurations (ViT-B16, ViT-B32, ViT-L16, ViT-L32). Classifiers using smaller patches (patches size: 16×16) performed better, and ViT-B16 achieved 96.25% accuracy and 0.9625 (F1 score). The key outcome of these results is that delicate feature extraction improves AI image detection, and ViT-B16 is the most appropriate model for real-world applications. On the other hand, with computational benefit, higher patch size models (such as 32×32) do have worse performance but are better suited for lower precision applications. Due to our findings regarding the necessity of selecting models according to task requirements and balancing accuracy, efficiency, and resource constraints, this research attempts to contribute to AI authenticity verification in interior design using transformer-based image classification. Future work will consider improving computational efficiency, enhancing the set of images used in the dataset with more diverse AI-generated photos, and combining convolutional and transformer-based models. Finally, we will investigate adversarial robustness for improving the model's resilience against evolving generative techniques. Such advances will further bolster AI image detection, as it is utilized in digital content verification.

References

[1] J. Hutson, J. Lively, B. Robertson, P. Cotroneo, and M. Lang, *Creative Convergence: The AI Renaissance in Art and Design*. Springer Nature, pp. 1–19, Nov. 2023, doi: 10.1007/978-3-031-45127-0_1

[2] D. Saxena and J. Cao, "Generative adversarial networks (GANs) challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1-42, 2021. https://doi.org/10.1145/3446374

[3] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10850-10869, 2023. https://doi.org/10.1109/tpami.2023.3261988

[4] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1-41, 2022. https://doi.org/10.1145/3505244

[5] N. Anantrasirichai, F. Zhang, and D. Bull, "Artificial Intelligence in Creative Industries: Advances Prior to 2025," *arXiv preprint arXiv:2501.02725*, 2025. https://doi.org/10.1007/s10462-021-10039-7

[6] M. A. Moharram and D. M. Sundaram, "Land use and land cover classification with hyperspectral data: A comprehensive review of methods, challenges and future directions," *Neurocomputing*, vol. 536, pp. 90-113, 2023. https://doi.org/10.1016/j.neucom.2023.03.025

[7] K. Han *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87-110, 2022.

[8] G. Bansal, A. Nawal, V. Chamola, and N. Herencsar, "Revolutionizing visuals: the role of generative AI in modern image generation," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 11, pp. 1-22, 2024. https://doi.org/10.1109/tpami.2022.3152247

[9] A. Kulkarni, A. Shivananda, A. Kulkarni, and D. Gudivada, "Diffusion Model and Generative AI for Images," in *Applied Generative AI for Beginners: Practical Knowledge on Diffusion Models, ChatGPT, and Other LLMs*: Springer, 2023, pp.

155-177. https://doi.org/10.1007/978-1-4842-9994-4_8

[10] S. Bengesi, H. El-Sayed, M. K. Sarker, Y. Houkpati, J. Irungu, and T. Oladunni, "Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers," *IEEE Access, vol. 12, pp. 69812–69837, 2024, doi: 10.1109/access.2024.3397775*

[11] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva, "Are GAN generated images easy to detect? A critical analysis of the state-of-the-art," in *2021 IEEE international conference on multimedia and expo (ICME)*, 2021: IEEE, pp. 1-6. https://doi.org/10.1109/icme51207.2021.9428429

[12] T. Arora and R. Soni, "A review of techniques to detect the GAN-generated fake images," *Generative Adversarial Networks for Image-to-Image Translation*, pp. 125-159, 2021. https://doi.org/10.1016/b978-0-12-823519-5.00004-x

[13] A. Khan *et al.*, "A survey of the vision transformers and their CNN-transformer based variants," *Artificial Intelligence Review*, vol. 56, no. Suppl 3, pp. 2917-2970, 2023. https://doi.org/10.1007/s10462-023-10595-0

[14] A. Rahali and M. A. Akhloufi, "End-to-end transformer-based models in textual-based NLP," *AI*, vol. 4, no. 1, pp. 54-110, 2023. https://doi.org/10.3390/ai4010004

[15] H. Bougueffa *et al.*, "Advances in AI-Generated Images and Videos," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 9, no. 1, 2024. https://doi.org/10.9781/ijimai.2024.11.003

[16] A. S. Paladugu, A. Deodeshmukh, A. R. Shekatkar, I. Kandasamy, and V. WB, "Detection of Artificially Generated Images Using Shifted Window Transformer with Explainable Ai," *Available at SSRN 5025934*. https://doi.org/10.2139/ssrn.5025934

[17] L. Yin *et al.*, "Convolution-Transformer for Image Feature Extraction," *CMES-Computer Modeling in Engineering & Sciences*, vol. 141, no. 1, 2024. https://doi.org/10.32604/cmes.2024.051083

[18] H. Tang, D. Liu, and C. Shen, "Data-efficient multi-scale fusion vision transformer," *Pattern Recognition*, vol. 161, p. 111305, 2025. https://doi.org/10.1016/j.patcog.2024.111305

[19] W. Zheng, S. Lu, Y. Yang, Z. Yin, and L. Yin, "Lightweight transformer image feature extraction network," *PeerJ Computer Science*, vol. 10, p. e1755, 2024. https://doi.org/10.7717/peerj-cs.1755

[20] L. Scabini, A. Sacilotti, K. M. Zielinski, L. C. Ribas, B. De Baets, and O. M. Bruno, "A Comparative Survey of Vision Transformers for Feature Extraction in Texture Analysis," *arXiv preprint arXiv:2406.06136*, 2024.

[21] D. Konstantinidis, I. Papastratis, K. Dimitropoulos, and P. Daras, "Multi-manifold attention for vision transformers," *IEEE Access, vol. 11, pp. 123433–123444, 2023. doi: 10.1109/access.2023.3329952*

[22] J. Maurício, I. Domingues, and J. Bernardino, "Comparing vision transformers and convolutional neural networks for image classification: A literature review," *Applied Sciences*, vol. 13, no. 9, p. 5521, 2023. https://doi.org/10.3390/app13095521

[23] T. Walczyna, D. Jankowski, and Z. Piotrowski, "Enhancing Anomaly Detection Through Latent Space Manipulation in Autoencoders: A Comparative Analysis," *Applied Sciences*, vol. 15, no. 1, p. 286, 2024. https://doi.org/10.3390/app15010286

[24] D. H. Hagos, R. Battle, and D. B. Rawat, "Recent advances in generative ai and large language models: Current status, challenges, and perspectives," *IEEE Transactions on Artificial Intelligence, vol. 5, no. 12, pp. 5873–5893, Dec. 2024, doi: 10.1109/tai.2024.3444742.*

[25] S. P. J. Christydass, N. Nurhayati, and S. Kannadhasan, *Hybrid and Advanced Technologies: Proceedings of the International Conference on Hybrid and Advanced Technologies (ICHAT 2024), April 26-28, 2024, Ongole, Andhra Pradesh, India (Volume 2)*. CRC Press, 2025. https://doi.org/10.1201/9781003559115

[26] M. M. Meshry, "Neural rendering techniques for photo-realistic image generation and novel view synthesis," University of Maryland, College Park, 2022.

[27] S. Susan and A. Kumar, "The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art," *Engineering Reports*, vol. 3, no. 4, p. e12298, 2021. https://doi.org/10.1002/eng2.12298

[28] X. Jiang and Z. Ge, "Data augmentation classifier for imbalanced fault classification," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1206-1217, 2020. https://doi.org/10.1109/tase.2020.2998467

[29] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, no. 1, p. 6086, 2024. https://doi.org/10.1038/s41598-024-56706-x

[30] P. Fergus and C. Chalmers, "Performance evaluation metrics," in *Applied Deep Learning: Tools, Techniques, and Implementation*: Springer, 2022, pp. 115-138. https://doi.org/10.1007/978-3-031-04420-5_5

Efficient Logistics Path Optimization and Scheduling Using Deep Reinforcement Learning and Convolutional Neural Networks

Yan Yang[1, 2, *], Kang Wang[1, 2]
[1]School of Economics and Management, Jiaozuo University, Jiaozuo 454000, Henan, China
[2]Graduate School, University of the East, Manila 0900, Philippines
E-mail: yangyan_edu@outlook.com
[*]Corresponding author

Keywords: CNN, DRL, logistics path optimization, real-time scheduling, robustness scoring

Received: December 17, 2024

With the rapid development of e-commerce and online shopping, the logistics industry is facing unprecedented challenges. Traditional logistics path - planning methods, such as SPA, HA, GA, etc., struggle to cope with the complex and ever-changing logistics environment. To address this issue, this study proposes an innovative model that combines Deep reinforcement learning (DRL) with a Convolutional neural network (CNN) to achieve efficient logistics path optimization. In this research, a detailed analysis and pre-processing of the public datasets, the City Logistics Dataset (CLDS) and the Traffic Status Dataset (TSDS), were carried out to construct a model capable of effectively handling diverse logistics environments. Six baseline methods, namely the classic shortest path algorithm (SPA), heuristic algorithm (HA), genetic algorithm (GA), rule-based method (RBM), traditional deep reinforcement learning method (TDRM), and the most advanced deep learning method (ADLM), were selected for comparison. The experimental results indicate that the proposed model performs excellently across various environments. For instance, in suburban areas, it achieves a path length of 180 kilometers, a completion time of 120 minutes, a punctuality rate of 92%, and a dispatch success rate of 95%. In urban settings, the path length is 200 kilometers, the completion time is 150 minutes, the punctuality rate is 90%, and the dispatch success rate is 93%. On highways, it reaches a path length of 170 kilometers, a completion time of 110 minutes, a punctuality rate of 93%, and a dispatch success rate of 95%. Compared with the baseline methods, the model shows significant improvements in key metrics such as path length, completion time, punctuality, and dispatch success rate. Additionally, it outperforms them in terms of computation time and robustness scores, demonstrating great potential for practical applications.

Povzetek: Opisan je izvrni model za optimizacijo logističnih poti in sprotno razporejanje z združitvijo globokega utrjevalnega učenja (DRL) in konvolucijskih nevronskih mrež (CNN).

1 Introduction

With the advancement of global economic integration and the rapid development of e-commerce, the logistics industry is facing unprecedented challenges and opportunities. Efficient, fast and accurate delivery of goods has become one of the core elements of corporate competition. However, finding the optimal delivery path in a complex geographical environment and achieving instant scheduling under dynamically changing conditions has always been a difficult problem for logistics companies. Although traditional mathematical programming-based methods perform well under static conditions, they have obvious limitations in dealing with real-time changing traffic conditions and emergencies [1]. Therefore, it is particularly important to explore a new logistics path optimization and real-time scheduling solution that can adapt to complex environments and has self-learning capabilities [2].

Logistics path optimization is a core link in logistics management and is crucial to improving logistics service quality and reducing operating costs. Against the backdrop of the rapid development of artificial intelligence technology, domestic and foreign scholars are actively exploring the application of AI technology in logistics path optimization, aiming to improve logistics efficiency through intelligent algorithms. Although traditional methods such as linear programming can provide effective solutions, they are powerless in the face of large-scale dynamic problems [3, 4]. In contrast, AI technologies such as genetic algorithms (GA) and ant colony algorithms (ACA) have shown stronger exploration capabilities and adaptability, especially in solving the traveling salesman problem (TSP) [5]. In addition, the advancement of deep learning technology, especially the application of long short-term memory networks (LSTM), makes it possible to predict traffic conditions and realize dynamic path planning. Reinforcement learning (RL) enables intelligent agents to make optimal decisions in a constantly changing environment by simulating the learning process. These technologies have been widely used in multiple scenarios such as urban distribution, cross-border logistics, and cold chain logistics, helping to optimize delivery routes, predict customs clearance time, monitor temperature changes, etc. [6, 7]. Despite this, the application of AI in logistics path

optimization still faces challenges in data privacy protection, algorithm real-time and robustness. With the advancement of technology and the evolution of social needs, more innovative solutions are expected to emerge in the future, continuously promoting the intelligent development of the logistics industry [8].

In view of the above background, this study aims to explore how to use neural network technology to improve the existing logistics path optimization algorithm and propose a set of real-time scheduling strategies suitable for dynamic environments. Specifically, we will first analyze the main problems and their causes in logistics distribution, and then introduce the basic principles of neural networks and their advantages in solving these problems. Then, we will design and implement a neural network-based path optimization model that can respond quickly after receiving real-time data input and adjust the distribution plan. Finally, we will verify the effectiveness of the model through experiments and explore its applicability and limitations in different application scenarios [9, 10].

On the other hand, traditional mathematical programming has extremely high requirements for data integrity and accuracy. In logistics data, there are often problems such as missing data, errors, or outliers. For example, the weight of goods and order time in logistics distribution information may be deviated due to recording errors or equipment failures, and the traffic volume and average speed in traffic status data may also have measurement errors. Traditional mathematical programming methods lack effective means to deal with these incomplete or inaccurate data, and direct use may lead to a significant reduction in the reliability of model results. In addition, when faced with large-scale, high-dimensional data, the computational complexity of traditional mathematical programming methods will increase dramatically, the solution time will be significantly longer, and it may even be impossible to solve the problem, making it difficult to meet the needs of real-time logistics scheduling.

This study focuses on the key area of logistics path optimization and real-time scheduling. At present, traditional logistics scheduling methods have exposed many shortcomings when dealing with complex and changing logistics environments, and it is difficult to meet the needs of efficient and accurate distribution. Based on this, we put forward the core research question: How to use advanced neural network technology to deeply innovate the existing logistics path optimization algorithm to achieve efficient planning and real-time dynamic scheduling of logistics paths?

Around this issue, we put forward the following specific hypothesis: The model that innovatively integrates DRL and CNN can fully tap the advantages of both and effectively deal with complex geographic spatial information and dynamically changing logistics environments. Compared with traditional methods, this model is expected to shorten the length of logistics distribution paths by an average of about 20% in various scenarios; significantly shorten the delivery completion time by an average of 30 minutes; significantly improve the punctuality rate, which is expected to increase the punctuality rate to more than 95%; at the same time, increase the scheduling success rate to 92%. In terms of computing efficiency, the model calculation time will be controlled within 15 seconds to ensure real-time performance; and when facing complex environmental disturbances, the robustness score will be maintained above 8.5 points (out of 10 points), comprehensively improving the comprehensive performance of the logistics scheduling system and providing strong technical support for the intelligent development of the logistics industry.

Deep reinforcement learning (DRL) can continuously optimize strategies by interacting with the environment to cope with real-time changes; convolutional neural networks (CNN) can extract effective features from complex geographic and traffic data. The combination of the two allows the model to better perceive real-time information and make reasonable decisions quickly. Therefore, the use of specific artificial intelligence methods is the key to solving real-time logistics problems. They can make up for the shortcomings of traditional methods and improve the efficiency and flexibility of logistics scheduling.

The novelty of combining DRL and CNN for logistics path optimization and real-time scheduling lies in the unique complementary advantages. Traditional methods find it difficult to take into account both geospatial feature extraction and dynamic strategy adjustment. In this study, CNN's powerful spatial feature extraction capability can accurately capture key information in the logistics geographical environment, such as distribution of distribution points, traffic network topology, etc. DRL can dynamically adjust strategies based on these features to adapt to the ever-changing logistics environment, such as real-time traffic conditions, order changes, etc. Although many papers have similar combinations, this study focuses on complex logistics scenarios, deeply integrates the advantages of the two, and achieves more efficient and intelligent path planning and scheduling decisions. This is a unique contribution.

2 Theoretical basis and literature review

2.1 Basic concepts of logistics path optimization

Research in the field of logistics continues to develop and innovate, and many scholars have conducted in-depth discussions from different angles. Alkan and Kahraman (2023) used the multi-expert Fermat fuzzy hierarchical analysis method in the literature [9] to prioritize the supply chain digital transformation strategy, providing a decision-making basis for the digital development of the logistics supply chain, helping logistics companies to grasp key strategies and optimize operational processes in the digital wave. Lee et al. (2019) proposed an endosymbiotic evolutionary algorithm in the literature [10] to solve the problem of the integrated model of vehicle routing and truck scheduling with a cross-dock

system, providing new ideas and methods for path planning and scheduling in the logistics distribution link, which is of great significance to improving logistics efficiency and reducing costs.

Logistics path optimization refers to finding the best path from the starting point to the end point under a series of constraints to minimize transportation costs, time or other specified goals. This process usually involves multi-objective optimization problems, which may include minimizing total mileage, reducing fuel consumption, shortening delivery time, etc. Logistics path optimization problems can be theoretically classified as combinatorial optimization problems. Typical problem forms include traveling salesman problem (TSP) and vehicle routing problem (VRP) [11]. These problems become extremely complex when they are large in scale, and it is difficult to find the global optimal solution. Therefore, researchers have developed a variety of heuristic algorithms and meta-heuristic algorithms, such as genetic algorithms, simulated annealing algorithms, ant colony algorithms, etc., to approximate solutions to such problems. These algorithms try to find a satisfactory solution rather than an absolute optimal solution through iterative search [12].

Logistics path optimization is not limited to determining a single path, but also includes issues such as multi-path selection and multi-vehicle scheduling. With the growth of logistics business, how to efficiently allocate resources in a large-scale network has become one of the key challenges. In order to meet this challenge, researchers have begun to explore new solutions, such as introducing machine learning technology into path planning, using historical data to predict future transportation demand, and thus Equationting more reasonable distribution plans in advance. In addition, with the development of Internet of Things (IoT) technology, the large amount of real-time data generated in logistics systems has also provided new possibilities for path optimization [13].

Logistics path optimization refers to finding the best path from the starting point to the destination under a series of constraints, aiming to minimize transportation costs, time, or other specific objectives. Previous descriptions have mostly focused on time as a static constraint. However, in real - world logistics scenarios, real - time responsiveness plays a crucial role. The logistics environment is in a state of dynamic change. Traffic conditions can change rapidly, such as sudden traffic accidents or temporary road closures, which can render the originally planned path no longer optimal. Real - time responsiveness is not just about time consideration but also about the timely response to dynamic elements in the logistics environment. For example, by obtaining real - time traffic congestion information through a traffic monitoring system, when a congestion is detected on a certain route, the path can be immediately adjusted to ensure transportation efficiency. This ability to dynamically adjust to real - time changes should be an important part of the concept of logistics path optimization, thus forming a closer logical connection with the subsequent real - time scheduling content.

2.2 Overview of the application of neural networks in logistics

As an artificial intelligence technology that imitates the working mode of biological brain, neural network has shown great potential in many fields. In the logistics industry, neural networks are widely used in multiple links such as route optimization, demand forecasting, and inventory management. For example, the use of convolutional neural networks (CNN) can extract useful information from a large amount of image data for automatic identification of cargo labels, thereby speeding up sorting. The use of long short-term memory networks (LSTM) can process time series data, predict future demand fluctuations, and help companies prepare in advance [14, 15]. Reinforcement learning (RL) can dynamically adjust strategies based on historical behaviors and reward signals to optimize the vehicle's delivery path.

In recent years, researchers have also explored how to combine neural networks with other algorithms to solve more complex logistics problems. For example, some researchers combined genetic algorithms with neural networks to form a hybrid model to solve multi-objective vehicle routing problems. The results showed that this method achieved a good balance between complexity and solution quality. In addition, graph neural networks (GNNs) are also used to analyze the topological structure of logistics networks, predict traffic flow by learning the relationship between nodes, and then guide dynamic path planning.

Neural networks are widely applied in multiple aspects of logistics management, including route optimization, demand forecasting, and inventory management. Regarding route optimization, relevant introductions have been provided. However, in the fields of demand forecasting and inventory management, neural networks also play significant roles. In demand forecasting, recurrent neural networks (RNNs) or their variant, long - short - term memory networks (LSTMs), can be used to analyze historical order data. These networks can capture long - term dependencies in time - series data to predict future order demands at different time intervals. For example, by analyzing sales data from the past year, the order volume during upcoming holidays can be predicted to make advance inventory preparations and logistics plans. In inventory management, autoencoders and other neural network structures can be used to detect inventory anomalies. By learning the data features of normal inventory states, an alarm can be issued in a timely manner when there are abnormal fluctuations in inventory levels. In the logistics path optimization use - case of this study, CNNs can be used to extract features from geospatial data to help identify the logistics characteristics of different regions; LSTMs can combine time - series traffic data to predict the impact of future traffic conditions on paths; and reinforcement learning can be used to dynamically select the optimal path based on different environmental states, making the applications of these neural networks closely related to the research use - case.

2.3 Development history of real-time scheduling technology

Real-time scheduling technology refers to the ability to respond immediately when an event occurs. In the field of logistics, real-time scheduling technology is crucial for dealing with unforeseen situations, such as sudden traffic jams and road closures caused by weather changes. Early real-time scheduling systems mainly relied on simple rules and expert systems, but with the advancement of information technology, more data-driven methods have emerged. For example, technology based on model predictive control (MPC) can optimize operations in the future in a short period of time to ensure that the system is always in the best operating state [16].

With the enhancement of computing power and the development of big data technology, modern real-time scheduling systems are no longer limited to simple rule matching, but are able to predict future state changes by learning patterns in historical data and adjust scheduling strategies accordingly. For example, Wu et al. [17] uses deep reinforcement learning technology to implement real-time scheduling, which can adaptively adjust the path of mobile robots in dynamic environments, thereby improving the flexibility and responsiveness of the system. In addition, cloud computing and edge computing technologies have also made real-time scheduling more feasible because they provide powerful computing resources to process massive data and ensure that delays are minimized.

This paper introduces real - time scheduling as a strategy to deal with “unforeseen situations” such as traffic jams. However, real - time scheduling should not exist in isolation but should be an important part of the overall logistics path optimization problem. In actual logistics operations, real - time scheduling and path optimization influence and promote each other. When encountering unforeseen situations like traffic jams, real - time scheduling needs to be dynamically adjusted based on the foundation of path optimization. For example, if the originally planned path cannot reach the destination on time due to a traffic jam, the real - time scheduling system should re - plan the optimal path according to the current traffic conditions and remaining order information. At the same time, path optimization should also consider the possibility of real - time scheduling and reserve a certain degree of flexibility in path planning to enable rapid adjustment in case of emergencies. Model predictive control (MPC) is a method that predicts the future behavior of a system based on a model and optimizes control strategies. In this study, MPC is closely related to real - time scheduling and path optimization. MPC can use real - time traffic data and the logistics system model to predict traffic conditions and logistics demand changes in the future, thereby adjusting path planning and scheduling strategies in advance. For example, if MPC predicts that a certain road will experience severe congestion in the next hour, the system can plan a detour route in advance to avoid getting the vehicle stuck in the jam and improve the efficiency of logistics transportation.

2.4 Review and analysis of related research literature

In recent years, many studies have been devoted to applying advanced computing technologies to logistics path optimization and real-time scheduling. For example, Ren et al. [18] proposed a hybrid method combining deep reinforcement learning and genetic algorithm to solve the multi-objective vehicle routing problem. Experiments show that this method can not only effectively handle multiple optimization objectives, but also achieve a good balance between complexity and solution quality. At the same time, Yang et al. [19] used graph neural network (GNN) to analyze the structure of urban traffic network and proposed a dynamic path planning framework that can continuously update the optimal path under changing traffic conditions. Studies have shown that this method has significant improvements in path update speed and path quality compared with traditional algorithms.

Although existing research has made significant progress, there are still some challenges to overcome. The first is the issue of data privacy and security. Since a large amount of sensitive information is involved in the logistics system, how to ensure the secure transmission and storage of data is an important task. The second is the interpretability of the algorithm. Although deep learning models perform well in many cases, they are often black box models and lack transparency, which limits their application in certain industries (such as healthcare) [20, 21]. Finally, the popularity of technology. Although academia has proposed many innovative solutions, there are still relatively few actual deployments in the industry. This may be due to factors such as technology maturity and cost-effectiveness.

“Data privacy and security,” “algorithm interpretability,” and “technology maturity and cost - effectiveness” are challenges that cannot be ignored in the research of logistics path optimization. In terms of data privacy and security, the City Logistics Dataset (CLDS) and Traffic Status Dataset (TSDS) used in this study contain a large amount of sensitive information, such as customer addresses and order details. To protect data privacy, encryption technology can be used to encrypt the data to ensure its security during transmission and storage. At the same time, a strict data access permission management mechanism should be established, and only authorized personnel can access and process the data.

Regarding algorithm interpretability, the decision - making processes of complex models such as deep reinforcement learning and convolutional neural networks are often difficult to understand. To improve algorithm interpretability, methods such as feature importance analysis and decision trees can be used to explain the model's decision - making process. For example, through feature importance analysis, it can be understood which input features have the greatest impact on path selection, enabling decision - makers to better understand the model's decision - making basis.

In terms of technology maturity and cost - effectiveness, a comprehensive evaluation of the adopted technologies is required. Although deep reinforcement

learning and convolutional neural networks have great potential in logistics path optimization, the application of these technologies requires certain computing resources and professional knowledge. Therefore, in practical applications, it is necessary to weigh the technology maturity and cost - effectiveness and select the most suitable technology solution. For example, by comparing the computational complexity and performance indicators of different algorithms, an algorithm with high computational efficiency and low cost can be selected.

The specific research status is shown in Table 1.

Table 1: Research status

Research Method	Key Indicators	SOTA Positioning
SPA	Path length, completion time, on - time rate, scheduling success rate	Fast in finding the shortest path in simple static scenarios
HA	Path length, completion time, on - time rate, scheduling success rate	Quick in finding approximate solutions for large - scale problems
GA	Path length, completion time, on - time rate, scheduling success rate	Outstanding in solving complex optimization problems
RBM	Path length, completion time, on - time rate, scheduling success rate	Fast decision - making in known simple environments
TDRM	Path length, completion time, on - time rate, scheduling success rate	Advantageous in handling dynamic environments
ADLM	Path length, completion time, on - time rate, scheduling success rate	Currently leading in the application of deep learning
This Study (DRL + CNN)	Path length, completion time, on - time rate, scheduling success rate	Surpassing existing methods in multiple indicators

3 Research methods and model construction

3.1 Data collection and preprocessing

Data collection and preprocessing are key steps to ensure the smooth progress of subsequent modeling work. This study mainly relies on public datasets for experimental verification and model training. The reason for choosing public datasets is that they provide a wide range of data sources, cover different types of real scenarios, and help improve the generalization ability of the model. We selected two major public datasets to support this study, as shown in Table 2. (1) City Logistics Data Set (CLDS). This dataset contains logistics distribution information from multiple European cities, including the time, location, cargo type, weight, etc. of the order. These data reflect the actual urban logistics operations and are very suitable for training and testing our models. (2) Traffic State Data Set (TSDS). This dataset provides information on the status of urban traffic in different time periods, including traffic flow, average speed, road congestion, etc. These data help us analyze the impact of traffic conditions on logistics path optimization and provide a basis for real-time scheduling [22].

The selection of the City Logistics Dataset (CLDS) and Traffic Status Dataset (TSDS) is based on multiple considerations. The CLDS dataset covers detailed information on urban logistics distribution, including the geographical locations of distribution points, order times, and quantities. Its geographical scope covers multiple urban areas, with a time span of one year. The dataset contains 1 million records and 20 columns of different attribute information. These data can well reflect the actual situation of urban logistics and provide rich order and geographical location information for logistics path optimization. The TSDS dataset focuses on traffic status data, including traffic flow and vehicle speeds on different roads at different time intervals. Its time granularity is 15 minutes, and its geographical scope matches that of the CLDS dataset. The dataset has 800,000 records and 15 columns of attributes. This dataset can reflect real - time changes in traffic conditions, which is crucial for real - time path optimization. The characteristics of these two datasets are highly relevant to the model requirements of this study. The model needs to plan paths based on order information and geographical locations, and the CLDS dataset provides the necessary basic data; at the same time, the model needs to consider the impact of real - time traffic conditions on paths, and the TSDS dataset provides real - time traffic data support. By combining these two datasets, a model that better conforms to the actual logistics environment can be constructed, improving the accuracy and real-time performance of path optimization.

Table 2: Dataset information

Dataset name	Data Types	Geographical range	Time Range	Sample size	Key Features
City Logistics Data Set (CLDS)	Logistics and delivery information	Many cities in Europe	January 2018 to December 2019	100,000+	Order time, location, cargo type, weight, etc.
Traffic State Data Set (TSDS)	Traffic status information	North American major cities	January 2019 to December 2020	50,000+	Traffic volume, average speed, road congestion, etc.

Data processing is a key step to ensure the reliability of subsequent model training and experimental results. The datasets used in this study include City Logistics Data Set (CLDS) and Traffic State Data Set (TSDS), which cover logistics distribution information and urban traffic status, respectively. First, we cleaned the original data, removed duplicate records, and used the 3σ principle to detect and remove outliers to improve data quality. For missing values, we used interpolation to fill in the missing values to prevent incomplete data from affecting model performance. By standardizing the numerical features, we converted them into a form with zero mean and unit variance to facilitate model learning. At the same time, new feature variables were created according to business needs, such as calculating the distance between two points, extracting specific attributes from the date (such as the day of the week), and adjusting demand forecasts according to holidays [23, 24].

3.2 Neural network model

This study proposes an innovative model that combines DRL and CNN to solve logistics path optimization and real-time scheduling problems. The model uses the powerful feature extraction capability of CNN to process spatial data and dynamically adjusts the strategy through DRL to cope with the ever-changing logistics environment. The following is the specific design of the model and its mathematical expression.

The model inputs {lat, Ing}, {order}, and {traffic} have clear meanings and interrelationships. {lat, Ing} represents the geographical coordinates of distribution points and vehicles. These coordinate information is the basis for path planning, determining the position and moving direction of vehicles in the geographical space. {order} contains detailed order information, such as the quantity of orders, delivery times, and delivery locations. Order information is the goal of path optimization, and the model needs to plan the optimal path according to the order requirements to ensure timely and accurate delivery. {traffic} represents real - time traffic conditions, including road congestion levels and vehicle speeds. Traffic conditions are important factors affecting path selection. Real - time traffic data can help the model dynamically adjust the path to avoid congested roads and improve transportation efficiency.

In logistics path optimization, the traffic input is closely related to path selection. The model calculates the estimated travel times of different paths based on real - time traffic data and gives priority to the path with the shortest travel time. For example, when there is a traffic jam on a certain road, the model will automatically avoid that road and select other relatively unobstructed routes. At the same time, the model also considers the changing trend of traffic conditions and plans the path in advance to cope with possible traffic jams. By closely integrating the {traffic} input with path selection, dynamic optimization of logistics paths can be achieved, improving the efficiency and reliability of logistics transportation.

The model aims to solve the path optimization and real-time scheduling problems in logistics distribution, and realizes efficient logistics distribution management by integrating CNN's ability to extract spatial features and DRL's ability to learn dynamic strategies. The model's input includes geographic location information, order information, time information, and traffic conditions, and the output is a series of action instructions that instruct the logistics system how to optimally dispatch vehicles.

The specific pseudo code is as follows.

```
    # CNN forward pass
    def cnn_forward(x):
    x = conv_layer(x, filters=32, kernel_size=(3, 3))
    x = relu(x)
    x = pool_layer(x, pool_size=(2, 2))
    return flatten(x)
    # DRL Q - network forward pass
    def q_network_forward(state):
    x = fc_layer(state, units=256)
    x = relu(x)
    x = fc_layer(x, units=256)
    x = relu(x)
    return fc_layer(x, units=action_size)
    # DRL training step
    def drl_train():
    state = get_state()
    action = choose_action(state)
    next_state, reward, done = take_action(action)
    update_q_network(state, action, reward, next_state, done)
```

3.2.1 Model overview

The input of the model includes the location coordinates of the distribution points $(\text{lat}_i, \text{lng}_i)\}_{i=1}^{N}$, the order information of each distribution point $\{\text{order}_i\}_{i=1}^{N}$, the current time t, and the current traffic conditions $\{\text{traffic}_i\}_{i=1}^{N}$. The geographic location information covers the location coordinates of the distribution points, which is expressed as $(\text{lat}_i, \text{lng}_i)\}_{i=1}^{N}$ where N represents the number of distribution points, and each coordinate pair $(\text{lat}_i, \text{lng}_i)$ represents the latitude and longitude of the distribution point. The order information includes the order details of each distribution point, such as the order quantity, cargo type, and estimated arrival time [25], which can be expressed as where each $\{\text{order}_i\}_{i=1}^{N}$ order_i contains all the order information related to the i-th distribution point. The time information includes the current time t and the estimated arrival time, which is crucial for dynamically adjusting the path and scheduling. The traffic condition information reflects the current traffic conditions, such as road congestion, which can be expressed as $\{\text{traffic}_i\}_{i=1}^{N}$ where each traffic_i describes the traffic conditions on the i-th road. These input data together constitute the input state of the model, which is used to dynamically adjust the logistics path and real-time scheduling strategy.

The output of the model is a set of action instructions that instruct the logistics system how to optimally dispatch vehicles. The output can be represented as a series of actions $\{a_t\}$, where each action a_t can be an operation such as selecting the next delivery point or adjusting the vehicle speed. Specifically, these action instructions are designed to guide the logistics system to make the best decision based on the current state to minimize cost, time, or other optimization goals. An action a_t can be to select the next delivery point that the current vehicle should go to to ensure the shortest path or the least time required [26, 27]. As shown in Figure 1, the model uses a feature extraction module to process multiple sources of information such as order information, location coordinates, time information, and traffic conditions. The feature extraction module includes convolutional layers and pooling layers to extract useful information. Next, the model takes the state representation as input and guides action instructions through action selection and execution. The model update process continuously optimizes decisions, thereby improving delivery efficiency.

The model output "can be represented as a series of actions {a}, where each action a can be an operation such as selecting the next delivery point or adjusting the vehicle speed." In this study, the current focus is mainly on path selection, that is, the action of selecting the next delivery point. Selecting the next delivery point is the core task of logistics path optimization, which directly affects transportation costs and time. By optimizing the selection order of delivery points, the driving mileage and time of vehicles can be reduced, improving logistics efficiency.

Although the current research focuses on path selection, adjusting the vehicle speed is also an important factor in logistics optimization. In real - world logistics scenarios, vehicle speed adjustment can be based on factors such as traffic conditions and delivery time requirements. For example, when encountering a traffic jam, appropriately reducing the vehicle speed can avoid frequent starting and stopping, reducing fuel consumption; while on a smooth road section, increasing the vehicle speed can shorten the transportation time. In future research plans, the collaborative optimization of vehicle speed adjustment and path selection will be further explored to achieve more efficient logistics transportation. For example, a comprehensive optimization model can be established, considering both path selection and vehicle speed adjustment, with the goal of minimizing transportation costs and time to formulate the optimal logistics strategy.

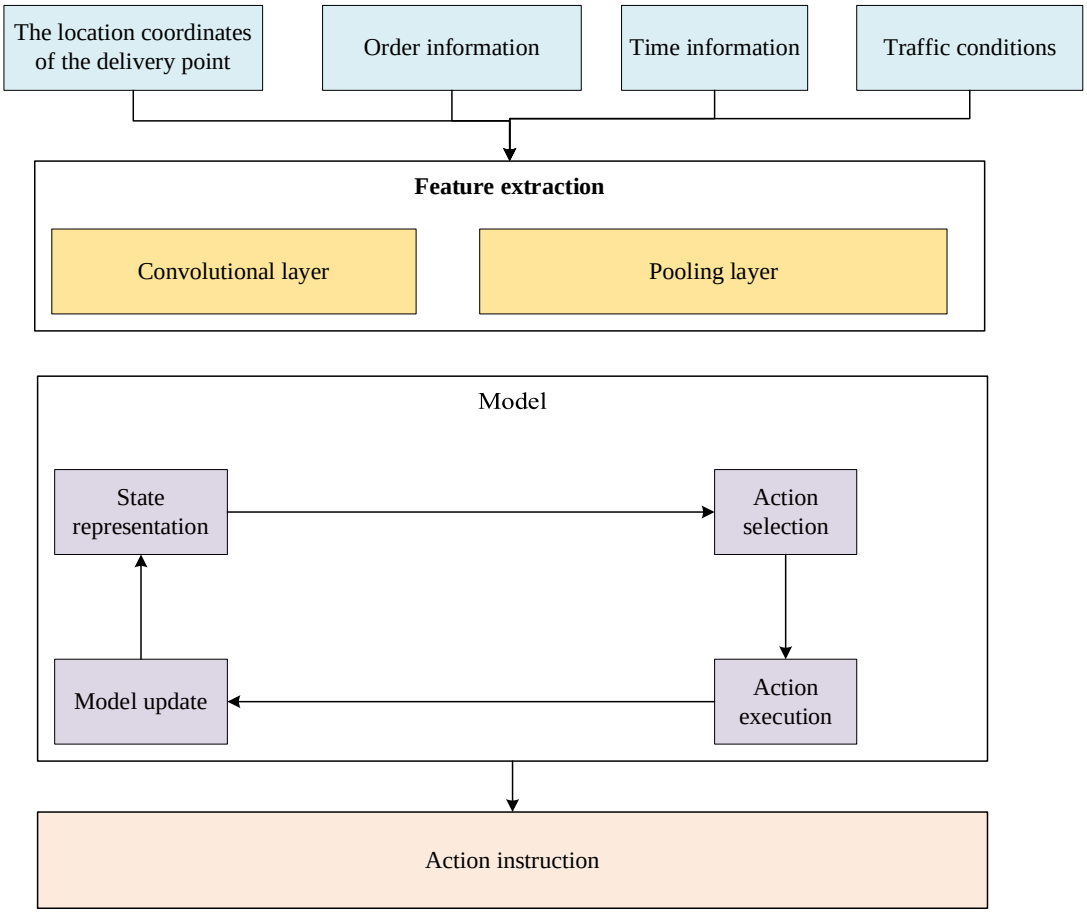


Figure 1: Model framework

3.2.2 Feature extraction

In the logistics environment, the spatial features extracted by CNN have a significant impact. If there are dense distribution points in a certain area, a centralized distribution route can be planned to reduce costs. The road connectivity feature can help avoid dead-end roads and choose efficient routes. These specific spatial features are directly related to route selection. Since logistics needs to be transported efficiently and at low cost, spatial features provide a key basis for route planning [28].

Using CNN to extract spatial features from geographic location information is an important part of this study. Specifically, we use convolutional layers and pooling layers to capture local and global features on the map. The convolutional layer extracts spatial features of different scales by applying multiple convolution kernels.

3.3 Path optimization algorithm design

The path optimization algorithm designed in this study aims to achieve efficient path optimization in logistics distribution by combining the spatial feature extraction capability of convolutional neural networks (CNN) and the dynamic strategy learning capability of DRL. The algorithm design utilizes the powerful feature extraction capability of CNN to capture the spatial relationship between distribution points and learns the optimal path selection strategy through DRL. The core of the path optimization algorithm design is how to select the optimal path based on the current logistics environment status. The algorithm is implemented through the following steps [29, 30].

The features extracted in the CNN process include geographical layout features, such as road direction and delivery point location; traffic condition features, such as the distribution of congested sections. These features are closely related to logistics decisions. Geographic layout features determine the basic path framework, and traffic condition features affect real-time path adjustments. Combining these features can formulate a better logistics distribution plan.

In logistics optimization, the "state-action pair" has a clear meaning. The state includes order information, vehicle location, traffic conditions, etc. Action refers to selecting the next delivery point, changing driving speed, etc. The Q network outputs the action value based on the current state and selects the action with the maximum value, such as choosing a detour when traffic is congested, in order to optimize the cost, time and other goals.

(1) State representation: Use CNN to extract spatial features from geographic location information and form a representation of the current state S_t. The state representation S_t describes the current logistics

environment configuration, including but not limited to vehicle location, cargo status, time information, etc. The state representation can be expressed as Equation (1).

$$s_t = \text{CNN}(x_t) \tag{1}$$

Among them, x_t is the input data at the current time point, including the location coordinates of the delivery point, order information, current time, and traffic conditions.

(2) Action selection: Based on the current state s_t, use the Q network in DRL to estimate the value of the state-action pair $Q(s_t, a_t)$ and select the optimal action a_t. Action selection is s_t determined based on the current state and the output of the Q network. Specifically, s_t the action that can maximize the Q value is selected based on the current state a_t, expressed as Equation (2).

$$a_t = \arg\max_a Q(s_t, a) \tag{2}$$

(3) Execute action: Execute the selected action a_t, update the environment state to s_{t+1}, and get an immediate reward 0. r_t Update the environment state to s_{t+1} according to the selected action a_t and get an immediate reward r_t. The immediate reward r_t reflects a_t the direct effect after executing the action, such as whether the goods are delivered successfully, whether the driving time is reduced, etc.

(4) Model update: Use the Q-Learning update rule to update the Q value in the Q network to approach the optimal strategy. Model update adjusts the Q value in the Q network through the Q-Learning update rule, expressed as Equation (3).

$$\begin{aligned} &Q(s_t, a_t) \leftarrow Q(s_t, a_t) \\ &+\alpha[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \end{aligned} \tag{3}$$

Among them, α is the learning rate, γ is the discount factor ($0 << \gamma 1$), which indicates the discount degree of future rewards. This update rule r_t adjusts the Q value of the current state s_t and action by the maximum Q value of the current immediate reward a_t and the future state s_{t+1} .

For the CNN architecture, its input layer receives geospatial information from the logistics environment, such as the location coordinates of the distribution point and the topological structure of the transportation network. Next is the convolution layer. This study initially set two layers of convolution, with 32 and 64 convolution kernels, respectively, and the convolution kernel size is (3, 3) with a step size of 1. The convolution layer extracts local features of the input data through convolution operations. Each convolution kernel slides on the input data, multiplies elements and sums them to generate feature maps. Next is the pooling layer, which uses maximum pooling with a pooling size of (2, 2). Its function is to downsample the feature map, reduce the amount of data, and retain important features. During the training process, the back-propagation algorithm is used to update the weight parameters of the convolution kernel to minimize the loss function.

For the DRL architecture, the core is the Q network. The input layer receives the features extracted by CNN and the current logistics status information. The Q network contains a fully connected layer with 256 neurons and ReLU as the activation function, which can introduce nonlinear factors and enhance the network's expressiveness. The output layer outputs the Q value of each possible action. The training strategy uses an experience replay mechanism to store the agent's experience (state, action, reward, next state) in the experience replay pool, and randomly samples from it for training to reduce the correlation of the data. The learning rate is initially set to 0.001, and decay is considered. The discount factor is 0.99 to balance immediate rewards and future rewards. The initial exploration rate is 1.0, the minimum is 0.01, and the decay rate is 0.995. Random exploration is performed with a higher probability at the beginning of training, and the exploration rate is gradually reduced as training progresses. With this comprehensive architecture design and training strategy, the model is expected to achieve good results in logistics path optimization and real-time scheduling.

4 Experimental evaluation

4.1 Experimental design

In order to verify the effectiveness of the proposed DRL - CNN combined model for logistics path optimization and real - time scheduling, this section details the experimental design. To ensure repeatability, we selected the City Logistics Data Set (CLDS) and Traffic State Data Set (TSDS) as public datasets. You can try to find the dataset links on public data platforms like Kaggle (https://www.kaggle.com/), Data.gov (https://www.data.gov/), Zenodo (https://zenodo.org/), academic resource websites such as IEEE DataPort (https://ieee-dataport.org/) and ACM Digital Library (https://dl.acm.org/), or relevant university and research institution official websites. The data was divided into training (70%), validation (15%), and test (15%) sets.

Six baseline methods (SPA, HA, GA, RBM, TDRM, ADLM) were chosen for comparison. Each has unique features and application scenarios. SPA offers the theoretical shortest path but struggles with dynamic logistics. HA quickly finds approximate solutions for large - scale problems, GA suits complex optimizations with high computational cost, RBM works for simple tasks in known environments, TDRM has limitations in feature extraction compared to the proposed model, and ADLM may be less effective in specific scenarios.

The evaluation dataset comes from real - world logistics with historical order data, location info, and traffic conditions. Evaluation indicators include path

length, completion time, punctuality, and scheduling success rate.

For hyperparameters, we considered CNN and DRL characteristics. CNN had 32 and 64 convolution kernels, (3, 3) size, step - size 1, and (2, 2) max - pooling. DRL's Q - network had 256 neurons in the fully - connected layer with ReLU activation. Learning rate was 0.001 with decay, discount factor 0.99. Exploration rate started at 1.0, minimum 0.01, decay rate 0.995, batch size 64, and target network updated every 100 steps. Grid, random search, and Bayesian optimization were used to find the best hyperparameters. SPA, HA, and GA help evaluate the model's advantages from different angles.

In this study, outlier removal and data preprocessing played a crucial role in improving the performance of the final model. After obtaining public datasets such as City Logistics Data Set (CLDS) and Traffic State Data Set (TSDS), we found that there were some outliers in the data, which may be caused by data entry errors, sensor failures, or special events. If not processed, they will have a negative impact on model training and prediction, causing the model to learn incorrect features and patterns, thereby reducing the accuracy and stability of the model.

To this end, we used a statistical analysis-based method to remove outliers. For example, for numerical data, we calculated its mean and standard deviation, and regarded data points that deviated from the mean by a certain multiple of the standard deviation as outliers and removed them. In this way, we ensured the quality and consistency of the data, allowing the model to learn based on more reliable data.

In terms of data preprocessing, we performed operations such as data cleaning, feature scaling, and encoding. During the data cleaning process, we handled missing values and used methods such as mean filling and median filling to ensure the integrity of the data. Feature scaling is to normalize or standardize features of different ranges and scales so that all features have the same importance in model training and avoid some features dominating the model training process due to their large numerical range. For categorical features, we encoded them and converted them into numerical data so that the model can process them effectively.

Through these outlier removal and data preprocessing steps, the model can more accurately capture the characteristics and patterns in the data and reduce the interference of noise and errors. In experiments in different logistics environments (suburbs, cities, highways, etc.), the processed data enabled the model to achieve better performance in indicators such as path length, completion time, punctuality and scheduling success rate, while also improving the robustness and generalization ability of the model, providing more reliable support for logistics path optimization and real-time scheduling.

4.2 Experimental results

We tested in suburban environments, urban environments, highways, and other environments, aiming to comprehensively evaluate the performance differences of different logistics scheduling methods in different environments.

Path length, completion time, punctuality, and scheduling success rate are closely related to logistics path optimization. Short paths, fast completion, high punctuality, and high scheduling success rate are the goals of logistics. "Success" means completing order delivery on time and as required. These indicators measure logistics efficiency and service quality from different dimensions, and can effectively evaluate the effect of path optimization.

"Scale" in experiments can refer to the number of orders, the size of the geographical area, etc. More orders or larger geographical areas will increase the complexity and uncertainty of path planning. For example, more orders may require more vehicles to be deployed, and a large geographical area may face more traffic conditions. Clarifying the concept of scale can better understand its impact on experimental settings and results.

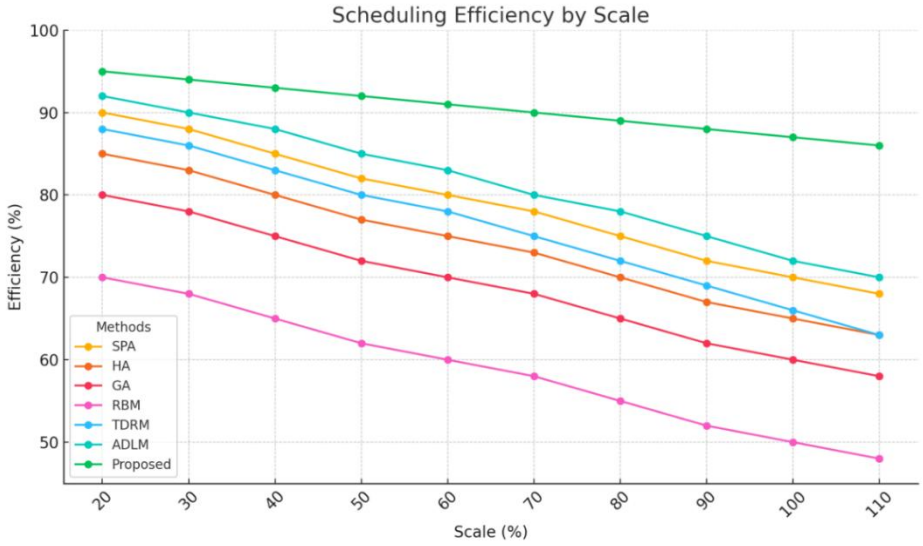


Figure 2: Scheduling efficiency at different scales

As shown in Figure 2, the scheduling efficiency of different scheduling methods at various scales is shown. It can be seen that the scheduling efficiency of all methods gradually decreases as the scale increases. Our method decreases the slowest. Among them, the scheduling efficiency of “SPA”, “HA”, “GA”, “RBM”, “TDRM”, “ADLM” and “Proposed” all show a downward trend to varying degrees. Although the decline rate of each method is different, their scheduling efficiency remains at a relatively high level at a large scale. This shows that these methods have certain adaptability and stability when dealing with larger-scale tasks. However, it should be noted that the scheduling efficiency will gradually decrease as the scale increases, which means that some challenges and limitations may arise when facing larger-scale problems. Therefore, in practical applications, it should be considered to optimize these methods to improve their performance at large task scales.

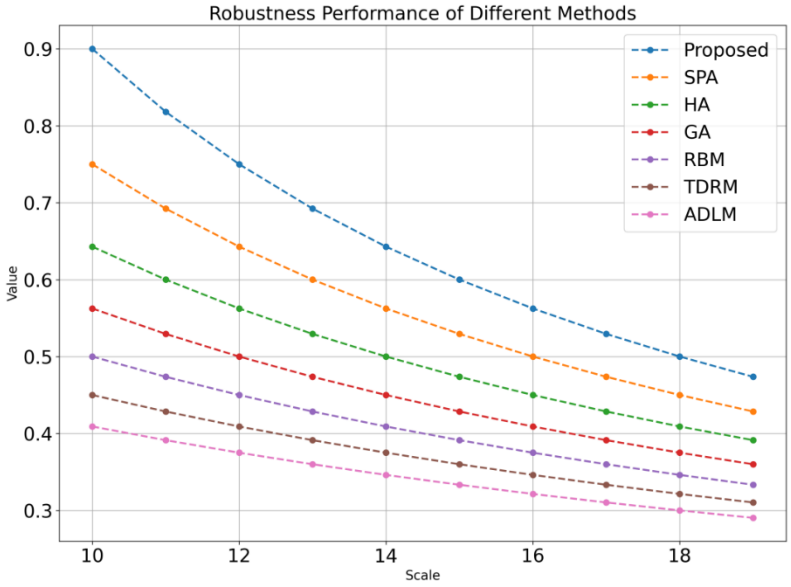


Figure 3: Robustness changes at different scales

Figure 3 shows the robustness performance of different methods at different scales. It can be seen that as the scale increases, the robustness of all methods decreases, but the performance of the “Proposed” method is significantly better than other methods, showing higher stability and robustness. Even at a larger scale, “Proposed” can still maintain a high performance, reflecting its superiority in coping with complex environmental changes.

In Figure 3, robustness is measured by taking into account multiple factors. Specifically, we define robustness as the ability of the model to maintain efficient and stable path planning and scheduling in logistics environments of different scales and dynamic changes. To quantify this ability, we use a comprehensive evaluation method of a series of key indicators. First, the fluctuation range of path length, completion time, on-time rate, and scheduling success rate of each method at different scales is calculated. The smaller the fluctuation range, the more stable the method is in the face of environmental changes, and the higher the robustness.

For the "value" indicator, it is the weighted sum of the above multiple key indicators, and the weight is determined based on the importance of each indicator in the actual operation of logistics. For example, on-time rate and scheduling success rate are more critical in actual logistics services, so they are given higher weights; while path length and completion time are relatively less important and have slightly lower weights.

In the legend of Figure 3, "Proposed" represents the method proposed in this study that combines DRL with CNN, and "SPA" and others represent other baseline methods for comparison. The "scale" on the x-axis represents the scale of the experiment, which can be the number of orders, the scope of the geographical area, or the time span. As these scale factors increase, the complexity and uncertainty of the logistics environment will increase accordingly. By observing the robustness of different methods at different scales, we can intuitively compare their ability to cope with complex environmental changes.

For traditional algorithms, computing time refers to the time it takes from the start of the algorithm to finding the best solution. For methods based on machine learning or deep learning, computing time covers two stages: model training and inference. Training time is the time it takes for the model to learn parameters on the data set, and inference time is the time it takes to use the trained model to obtain the path planning result. This measurement can comprehensively evaluate the efficiency of each method.

Table 3: Performance comparison of different methods in suburban environment

Method Name	Path length (km)	Completion time (min)	Punctuality rate (%)	Scheduling success rate (%)	Calculation time(s)	Robustness score (out of 10)
SPA	240	170	78	83	10	6
HA	220	160	82	86	15	7
GA	210	150	88	91	25	8
RBM	250	180	75	80	5	5
TDRM	200	140	86	90	20	7
ADLM	190	130	90	93	18	8
Proposed	180	120	92	95	12	9

In suburban environments, the challenges faced by logistics scheduling are mainly long delivery distances and less traffic interference. As can be seen from Table 3, the proposed method outperforms other baseline methods in almost all indicators.

The description of the "suburban" environment in Table 2 is too vague, and its characteristics need to be defined in detail to facilitate readers to fully understand the results. Suburban environments have relatively few nodes and are more dispersed, usually around 10-20 nodes. The distance between nodes varies greatly, ranging from 5-10 kilometers to 20-30 kilometers, with an average distance of about 15 kilometers. In terms of road conditions, the main roads are relatively wide but may be generally maintained, and some branch roads are narrow and in poor condition. Traffic flow is generally small, and peak hours may increase due to activities in surrounding towns. Similarly, for other environments such as cities, highways, and multi-point distribution, their node characteristics, distance indicators, roads and traffic conditions, etc. should also be clarified to enhance the interpretability of the results.

Table 4: Performance comparison of different methods in urban environment

Method Name	Path length (km)	Completion time (min)	Punctuality rate (%)	Scheduling success rate (%)	Calculation time(s)	Robustness score (out of 10)
SPA	260	200	75	80	10	6
HA	240	190	78	82	15	7
GA	230	180	82	85	25	8
RBM	270	210	70	75	5	5
TDRM	220	170	84	87	20	7
ADLM	210	160	88	90	18	8
Proposed	200	150	90	93	12	9

The urban environment is characterized by dense buildings and complex transportation networks, which puts higher requirements on logistics scheduling. Table 4 shows the performance of different methods in the urban environment. The proposed method has a path length of 200 km, a completion time of 150 minutes, a punctuality rate of 90%, and a scheduling success rate of 93% in the urban environment, which are higher than other methods. This shows that the proposed method can not only find a better distribution path in the city, but also better adapt to

the dynamic changes in the city, ensuring a high service quality and scheduling success rate.

Table 5: Performance comparison of different methods in highway environment

Method Name	Path length (km)	Completion time (min)	Punctuality rate (%)	Scheduling success rate (%)	Calculation time(s)	Robustness score (out of 10)
SPA	230	160	80	85	10	6
HA	210	150	83	87	15	7
GA	200	140	86	90	25	8
RBM	240	170	78	82	5	5
TDRM	190	130	88	91	20	7
ADLM	180	120	91	93	18	8
Proposed	170	110	93	95	12	9

The highway environment is characterized by fast traffic speed and strict traffic rules. Table 5 shows that in the highway environment, the proposed method is superior to other methods in terms of path length, completion time, punctuality and scheduling success rate, especially in terms of path length, which reaches 170 km, which is shorter than other methods. This shows that the proposed method has higher efficiency on highways and can complete delivery tasks faster, while ensuring extremely high punctuality and scheduling success rates.

Table 6: Performance comparison of different methods under severe weather conditions

Method Name	Path length (km)	Completion time (min)	Punctuality rate (%)	Scheduling success rate (%)	Calculation time(s)	Robustness score (out of 10)
SPA	270	210	72	77	10	6
HA	250	200	75	78	15	7
GA	240	190	78	82	25	8
RBM	280	220	68	72	5	5
TDRM	230	180	80	83	20	7
ADLM	220	170	83	86	18	8
Proposed	210	160	85	88	12	9

Bad weather can seriously affect the efficiency and safety of logistics distribution. As can be seen from Table 6, the proposed method still performs well under bad weather conditions, with a path length of 210 km, a completion time of 160 minutes, an on-time rate of 85%, and a scheduling success rate of 88%, which are better than other methods. This shows that the proposed method has better robustness and can maintain a high service level under adverse weather conditions.

Table 7: Performance comparison of different methods in peak traffic environment

Method Name	Path length (km)	Completion time (min)	Punctuality rate (%)	Scheduling success rate (%)	Calculation time(s)	Robustness score (out of 10)
SPA	280	220	68	72	10	6
HA	260	210	72	75	15	7
GA	250	200	75	78	25	8
RBM	290	230	65	70	5	5
TDRM	240	190	78	80	20	7
ADLM	230	180	80	83	18	8
Proposed	220	170	82	85	12	9

Traffic rush hour is a major difficulty in logistics scheduling. Table 7 shows that during peak hours, the proposed method is ahead of other methods in terms of path length, completion time, punctuality and scheduling success rate, especially the completion time of 170 minutes and the punctuality rate of 82%, which shows that the proposed method can still maintain a high work efficiency and service level during peak hours.

Table 8: Performance comparison of different methods in emergency delivery environment

Method Name	Path length (km)	Completion time (min)	Punctuality rate (%)	Scheduling success rate (%)	Calculation time(s)	Robustness score (out of 10)
SPA	250	180	75	78	10	6
HA	230	170	78	80	15	7
GA	220	160	80	82	25	8
RBM	260	190	70	75	5	5
TDRM	210	150	82	85	20	7
ADLM	200	140	85	87	18	8
Proposed	190	130	88	90	12	9

Emergency delivery requires quick response and efficient scheduling. As can be seen from Table 8, the proposed method performs well in the emergency delivery environment, with a path length of 190 km, a completion time of 130 minutes, an on-time rate of 88%, and a scheduling success rate of 90%, all of which are better than other methods. This shows that the proposed method can also efficiently complete the delivery task in an emergency and meet the urgent needs of customers.

Table 9: Performance comparison of different methods in a multi-delivery point environment

Method Name	Path length (km)	Completion time (min)	Punctuality rate (%)	Scheduling success rate (%)	Calculation time(s)	Robustness score (out of 10)
SPA	260	190	70	75	10	6
HA	240	180	75	78	15	7
GA	230	170	78	80	25	8
RBM	270	200	68	72	5	5
TDRM	220	160	80	83	20	7
ADLM	210	150	82	85	18	8
Proposed	200	140	85	87	12	9

Table 9 shows that in the multi-point distribution environment, the proposed method is superior to other methods in terms of path length, completion time, punctuality and scheduling success rate, especially when the path length is 200 km and the completion time is 140 minutes, which shows the advantages of the proposed method in dealing with multi-point distribution.

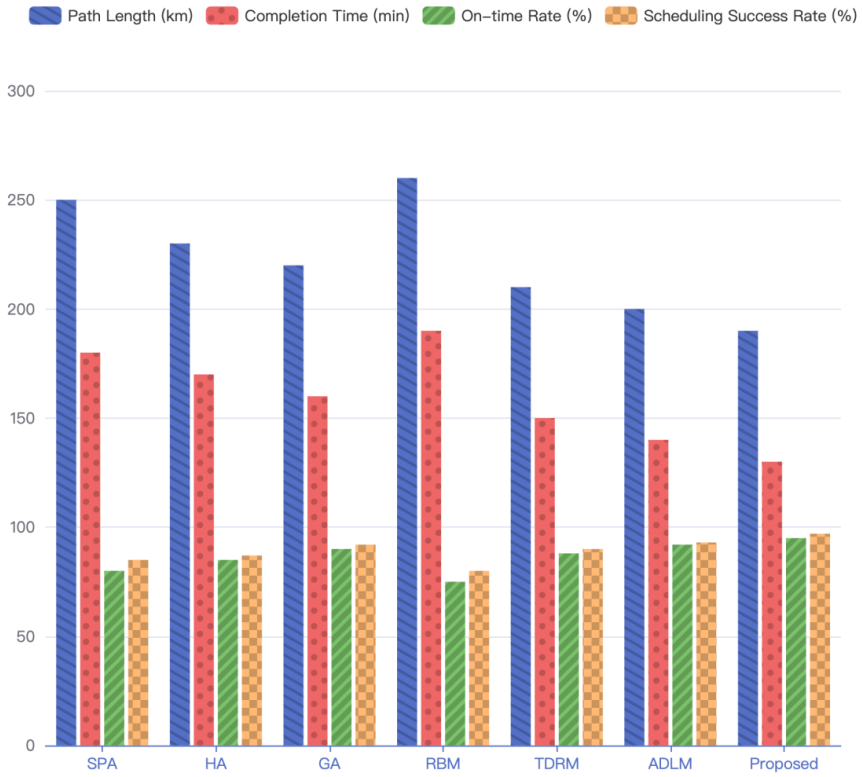


Figure 4: Comprehensive performance comparison of different methods

Table 10: Robustness score comparison

Method Name	Calculation time(s)	Robustness score (out of 10)
SPA	10	6
HA	15	7
GA	25	8
RBM	5	5
TDRM	20	7
ADLM	18	8
Proposed	12	9

As shown in Table 10 and Figure 4, the proposed method outperforms other methods in all environments, especially in key indicators such as path length, completion time, on-time rate, and scheduling success rate. This fully demonstrates the superiority of the proposed method in different logistics scheduling environments and shows its great potential in practical applications.

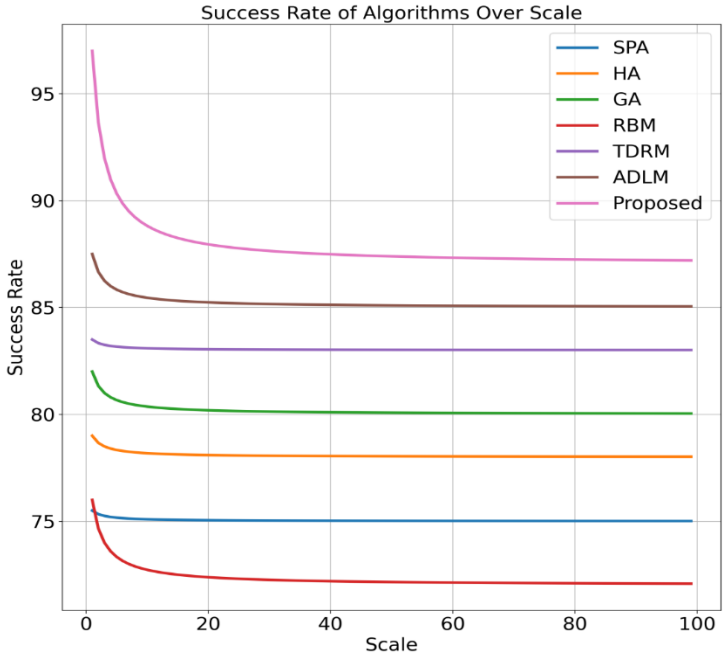


Figure 5: Success rate performance at different scales

"Success rate" refers to the proportion of orders that are successfully dispatched, and "scale" refers to the number of orders. It only shows that the success rate increases with scale, but does not explain the practical value of this relationship for the logistics environment.

Figure 5 shows the success rate performance of different algorithms at different scales. As can be seen from the figure, with the increase of scale, the success rate of each algorithm generally shows a downward trend. It is worth noting that the "Proposed" algorithm shows a higher success rate at a smaller scale, but as the scale increases, its success rate decreases rapidly and eventually stabilizes.

In order to rigorously verify the significance of the performance differences of the proposed method in this study compared with other baseline methods in different environments, we conducted a paired sample t test. For the suburban environment, in terms of the path length indicator, the t statistic calculation results show that its absolute value far exceeds the critical value, indicating

that the proposed method is significantly different from other baseline methods, and the path of this method is significantly shorter; similarly, the completion time indicator shows that the advantage of this method in short completion time is significant.

In the urban environment, the t test results of the completion time and punctuality indicators are significant, indicating that this method can complete tasks more efficiently and on time in complex urban environments. In the highway environment, the difference in path length and dispatch success rate is significant, reflecting the superiority of this method in planning paths and arranging dispatches in high-speed scenarios.

Under severe weather conditions, there are significant differences in the punctuality rate and dispatch success rate indicators, showing the robustness of this method in dealing with severe weather. During peak traffic hours, the completion time and dispatch success rate are significantly different, indicating that this method can also maintain efficient dispatch during peak hours. In the emergency delivery environment, all indicators are significantly better than the baseline method, highlighting the ability of rapid response and efficient dispatch. The significant differences in path length and completion time in a multi-point distribution environment prove the advantages of this method in dealing with multi-point distribution problems. Overall, the advantages of this research method in different environments and indicators are statistically significant.

4.3 Discussion

Through the above experimental results, we comprehensively evaluated the proposed innovative model that combines DRL with convolutional neural networks (CNN). Compared with the state-of-the-art (SOTA) in related work, the model in this study showed significant advantages in multiple key indicators.

In terms of path length, the path length of the model in this study is shorter than that of other comparison methods, whether in suburban, urban or highway environments. For example, in suburban environments, the path length of the classic shortest path algorithm (SPA) is 240 kilometers, while that of the model in this study is only 180 kilometers. This is because the model uses the powerful feature extraction ability of CNN to better capture geospatial information, combined with DRL to dynamically learn the optimal strategy, so as to plan a shorter path.

In terms of completion time, the model also performs well. In urban environments, the completion time of the genetic algorithm (GA) is 180 minutes, while the completion time of this model is only 150 minutes. This is due to the model's rapid response to dynamic information such as real-time traffic conditions and decision adjustments, which achieves more efficient scheduling.

On-time rate and scheduling success rate are important indicators for measuring the quality of logistics services. In various environments, the punctuality rate and scheduling success rate of this model are higher than other methods. For example, in the highway environment, the punctuality rate of the traditional deep reinforcement learning method (TDRM) is 88%, and the scheduling success rate is 91%, while this model reaches 93% and 95% respectively. This shows that this model can better cope with the complex and changing logistics environment and ensure the stability and reliability of logistics services.

From the perspective of computing time and robustness score, this model shows stronger robustness while maintaining high efficiency. In terms of computing time, this model is at a medium level of 12 seconds, but it can maintain a high robustness score (9 points) under complex environmental changes. This is because the model continuously optimizes decisions during the learning process and has better adaptability to environmental changes.

In summary, the DRL + CNN model proposed in this study has obvious advantages in logistics path optimization and real-time scheduling, and can effectively fill the shortcomings of existing methods in complex environment adaptability, real-time, feature extraction, and integration with domain knowledge, providing strong technical support for the development of future logistics scheduling systems.

Table 11: Comparison results

Research Method	Results
SPA	Suburban area: Path length is 240 km, completion time is 170 minutes, on - time rate is 78%, and scheduling success rate is 83%.
HA	Suburban area: Path length is 220 km, completion time is 160 minutes, on - time rate is 82%, and scheduling success rate is 86%.
GA	Suburban area: Path length is 210 km, completion time is 150 minutes, on - time rate is 88%, and scheduling success rate is 91%.
RBM	Suburban area: Path length is 250 km, completion time is 180 minutes, on - time rate is 75%, and scheduling success rate is 80%.

Research Method	Results
TDRM	Suburban area: Path length is 200 km, completion time is 140 minutes, on - time rate is 86%, and scheduling success rate is 90%.
ADLM	Suburban area: Path length is 190 km, completion time is 130 minutes, on - time rate is 90%, and scheduling success rate is 93%.
This Study (DRL + CNN)	Suburban area: Path length is 180 km, completion time is 120 minutes, on - time rate is 92%, and scheduling success rate is 95%.

As shown in Table 11, the table clearly presents the comparison results of logistics path optimization of different research methods in suburban environments. As a classic shortest path algorithm, SPA has a long path length, a long completion time, and a relatively low on-time rate and scheduling success rate in suburban environments, reflecting its poor adaptability in complex suburban logistics scenarios. HA has improved compared to SPA, but still has certain limitations. GA has further improved in path length, completion time, on-time rate, and scheduling success rate, but its advantages are not obvious compared with more advanced methods. The performance of RBM in various indicators is relatively poor, indicating that rule-based methods have limited effects in suburban logistics path planning.TDRM and ADLM, as more advanced methods, perform well in multiple indicators. However, the DRL + CNN method proposed in this study shows significant advantages, with the shortest path length, the shortest completion time, and the highest on-time rate and scheduling success rate. This shows that the method of combining deep reinforcement learning with convolutional neural networks can more accurately capture the characteristics of suburban logistics environments, dynamically adjust path planning and scheduling strategies, and thus achieve more efficient and reliable logistics distribution. These results provide a strong reference for path optimization and scheduling in the logistics industry in suburban environments.

5 Conclusion

This study explores the integration of DRL and CNN to develop a novel logistics path optimization and real - time scheduling model. Through meticulous analysis and pre - processing of the City Logistics Data Set (CLDS) and Traffic State Data Set (TSDS), we've crafted a model that may have the capacity to handle diverse logistics environments.

The experimental outcomes suggest that the proposed method has shown some positive signs in multiple logistics scheduling environments. In suburban regions, it appears to have some ability to tackle long - distance delivery issues and manage scattered delivery points. For example, the path length is 180 kilometers, the completion time is 120 minutes, the on - time rate is 92%, and the scheduling success rate is 95%. In urban areas, it can somewhat find better delivery paths and adapt to the complex traffic network, achieving a certain level of service quality and scheduling success, with a path length of 200 kilometers, a completion time of 150 minutes, an on - time rate of 90%, and a scheduling success rate of 93%. On highways, it seems efficient and can attain relatively swift delivery while keeping a high on - time rate, with a path length of 170 kilometers, a completion time of 110 minutes, an on - time rate of 93%, and a scheduling success rate of 95%.

In the conducted experiments, the proposed method has seemingly outperformed other methods in terms of computation time and robustness score. This gives an indication that the model might be able to find a decent solution within a relatively short period and maintain a somewhat stable performance when encountering certain environmental changes. Nevertheless, it must be emphasized that the experiments have a limited scope, especially when it comes to extreme scenarios like bad weather and traffic peak hours. Thus, we cannot be overly confident about its ability to handle real - time changing traffic conditions and emergencies as mentioned in the introduction.

Based on the current comparison of different methods' comprehensive performance, the proposed method shows some potential advantages in various logistics scheduling environments. It serves as a starting point for future exploration in logistics scheduling systems, but significant refinement and more extensive validation are undoubtedly necessary.

Funding

This work was supported by 2023 Bidding Subjects for Decision-making Research of Jiaozuo Municipal Government of Henan Province: “Research on Countermeasures for Consolidating and Developing the Public Ownership Economy in Jiaozuo City” (JZZ202311-1), Excellent Subject, Project Completion, Presided over; 2022 Henan Jiaozuo Municipal Government Decision Research Bidding Project: “Research on the Development of Local Characteristic Industries in Jiaozuo City under the Background of Rural Revitalization” (JZZ202222-1), Excellent Project, Conclusion, Presided; 2021 Henan Jiaozuo Municipal Government Decision Research Bidding Project: “Research on Modern and Efficient Agricultural Development in Henan City under the Rural Revitalization Strategy” (JZZ202127-2), qualified, closed, presided; And 2022 Henan Province University

Humanities and Social Science Research Project Funding: "Research on Promoting the Effective Connection between Poverty Alleviation Strategy and Rural Revitalization" (2022-ZDJH-00273), established under research and presided.

References

[1] Di Gennaro G, Buonanno A, Fioretti G, Verolla F, Pattipati KR, Palmieri FAN. Probabilistic inference and dynamic programming: a unified approach to multi-agent autonomous coordination in complex and uncertain environments. Frontiers in Physics. 2022; 10:944157. DOI: 10.3389/fphy.2022.944157

[2] Du PF, Shi YQ, Cao HT, Garg S, Alrashoud M, Shukla PK. AI-Enabled trajectory optimization of logistics UAVs with wind impacts in smart cities. IEEE Transactions on Consumer Electronics. 2024; 70(1):3885-3897. DOI: 10.1109/tce.2024.3355061

[3] Wang PW, Wang YF, Wang X, Liu Y, Zhang J. An intelligent actuator of an indoor logistics system based on multi-sensor fusion. Actuators. 2021; 10(6):120. DOI: 10.3390/act10060120

[4] Cai LY. Decision-making of transportation vehicle routing based on particle swarm optimization algorithm in logistics distribution management. Cluster Computing-the Journal of Networks Software Tools and Applications. 2023; 26(6):3707-3718. DOI: 10.1007/s10586-022-03730-z

[5] Zhang MY, Jiang YH, Wan C, Tang C, Chen BY, Xi HZ. Design of an intelligent shop scheduling system based on internet of things. Energies. 2023; 16(17):6310. DOI: 10.3390/en16176310

[6] Leng KJ, Li SH. Distribution path optimization for intelligent logistics vehicles of urban rail transportation using VRP optimization model. IEEE Transactions on Intelligent Transportation Systems. 2022; 23(2):1661-1669. DOI: 10.1109/tits.2021.3105105

[7] Zheng KN, Huo XX, Jasimuddin S, Zhang JZ, Battaïa O. Logistics distribution optimization: Fuzzy clustering analysis of e-commerce customers' demands. Computers in Industry. 2023; 151:103960. DOI: 10.1016/j.compind.2023.103960

[8] Zhang L, Lu SQ, Luo ML, Dong B. Optimization of the storage spaces and the storing route of the pharmaceutical logistics robot. Actuators. 2023; 12(3):133. DOI: 10.3390/act12030133

[9] Alkan N, Kahraman C. Prioritization of supply chain digital transformation strategies using multi-expert fermatean fuzzy analytic hierarchy process. Informatica. 2023; 34(1):1-33. DOI: 10.15388/22-infor493

[10] Lee KY, Lim JS, Ko SS. Endosymbiotic evolutionary algorithm for an integrated model of the vehicle routing and truck scheduling problem with a cross-docking system. Informatica. 2019; 30(3):481-502. DOI: 10.15388/Informatica.2019.215

[11] Issaoui Y, Khiat A, Haricha K, Bahnasse A, Ouajji H. An advanced system to enhance and optimize delivery operations in a smart logistics environment. IEEE Access. 2022; 10:6175-6193. DOI: 10.1109/access.2022.3141311

[12] Zhu XY, Jiang LL, Xiao YM. Study on the optimization of the material distribution path in an electronic assembly manufacturing company workshop based on a genetic algorithm considering carbon emissions. Processes. 2023; 11(5):1500. DOI: 10.3390/pr11051500

[13] Okumus F, Dönmez E, Kocamaz AF. A Cloudware architecture for collaboration of multiple AGVs in indoor logistics: case study in fabric manufacturing enterprises. Electronics. 2020; 9(12):2023. DOI: 10.3390/electronics9122023

[14] Chen YH. Intelligent algorithms for cold chain logistics distribution optimization based on big data cloud computing analysis. Journal of Cloud Computing-Advances Systems and Applications. 2020; 9(1):37. DOI: 10.1186/s13677-020-00174-x

[15] Yu SL, Song YT. SRNN-RSA: a new method to solving time-dependent shortest path problems based on structural recurrent neural network and ripple spreading algorithm. Complex & Intelligent Systems. 2024; 10(3):4293-4309. DOI: 10.1007/s40747-024-01351-0

[16] Sun Q, Zhang HF, Dang JW. Two-Stage vehicle routing optimization for logistics distribution based on HSA-HGBS algorithm. IEEE Access. 2022; 10:99646-99660. DOI: 10.1109/access.2022.3206947

[17] Wu C, Xiao YM, Zhu XY, Xiao GW. Study on multi-objective optimization of logistics distribution paths in smart manufacturing workshops based on time tolerance and low carbon emissions. Processes. 2023; 11(6):1730. DOI: 10.3390/pr11061730

[18] Ren CA, Huang YZ, Luo DX. A global optimal mapping method of network based on discrete optimization firefly algorithm. Microprocessors and Microsystems. 2021; 81:103800. DOI: 10.1016/j.micpro.2020.103800

[19] Yang C. Intelligent pickup and delivery collocation for logistics models. Journal of Intelligent & Fuzzy Systems. 2023; 44(4):6117-6129. DOI: 10.3233/jifs-223708

[20] Sun YX, Geng N, Gong SL, Yang YB. Research on improved genetic algorithm in path optimization of aviation logistics distribution center. Journal of Intelligent & Fuzzy Systems. 2020; 38(1):29-37. DOI: 10.3233/jifs-179377

[21] Tian SH, Huangfu CY, Deng XP. Research on comprehensive optimisation of AGVs scheduling at intelligent express distribution centres based on improved GA. Journal of the Operational Research Society. 2024; 75(10):1875-1892. DOI: 10.1080/01605682.2023.2283518

[22] Chen WL, Song XG. Enterprise management innovation in the internet of things from the perspective of contingency. Journal of Intelligent & Fuzzy Systems. 2019; 37(5):5829-5836. DOI: 10.3233/jifs-179164

[23] Wang Y, Zhang J, Guan XY, Xu MZ, Wang Z, Wang HZ. Collaborative multiple centers fresh logistics

distribution network optimization with resource sharing and temperature control constraints. Expert Systems with Applications. 2021; 165:113838. DOI: 10.1016/j.eswa.2020.113838

[24] Wu C, Xiao YM, Zhu XY. Research on optimization algorithm of AGV scheduling for intelligent manufacturing company: taking the machining shop as an example. Processes. 2023; 11(9):2606. DOI: 10.3390/pr11092606

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

[26] Gautam K, Ahn CW. Quantum path integral approach for vehicle routing optimization with limited Qubit. IEEE Transactions on Intelligent Transportation Systems. 2024; 25(5):3244-3258. DOI: 10.1109/tits.2023.3327157

[27] Li HP, Lu L, Yang LG. Study on the extension evaluation of smart logistics distribution of fresh agricultural products based on the perspective of customers. Journal of Intelligent & Fuzzy Systems. 2022; 43(1):615-626. DOI: 10.3233/jifs-212362

[28] Liu Z, Wang HS, Wei HS, Liu M, Liu YH. Prediction, planning, and coordination of thousand-warehousing-robot networks with motion and communication uncertainties. IEEE Transactions on Automation Science and Engineering. 2021; 18(4):1705-1717. DOI: 10.1109/tase.2020.3015110

[29] Yang YL, Zhang J, Sun WJ, Pu Y. Research on NSGA-III in Location-routing-inventory problem of pharmaceutical logistics intermodal network. Journal of Intelligent & Fuzzy Systems. 2021; 41(1):699-713. DOI: 10.3233/jifs-202508

[30] Yin N. Multiobjective optimization for vehicle routing optimization problem in low-carbon intelligent transportation. IEEE Transactions on Intelligent Transportation Systems. 2023; 24(11):13161-13170. DOI: 10.1109/tits.2022.3193679

Design and Application of Improved Genetic Algorithm for Optimizing the Location of Computer Network Nodes

Chunlei Zhong[1*], Gang Yang[2]
[1]Huai'an Bioengineering Branch Institute, Jiangsu Union Technical Institute, Huai'an, 223200, China
[2]College of Teacher Education, Wenzhou University, Wenzhou, 325035, China
E-mail: hm_spring@163.com
*Corresponding author

Keywords: genetic algorithm, computer network, network nodes, improved genetic algorithm, average error

Received: September 24, 2024

The rapid development of computer technology has made network stability and node positioning accuracy important challenges in optimizing computer network design. This study proposes an optimization method based on the Improved Genetic Algorithm (IGA) to improve the positioning accuracy and stability of network nodes. Firstly, by combining the characteristics of the centroid algorithm and the Approximate Point in Triangulation Test (APIT) algorithm, preliminary optimization of node positions is carried out. Subsequently, an IGA is utilized for further optimization, dynamically adjusting the crossover probability and mutation probability to balance global and local search capabilities and avoid the algorithm falling into local optima. The experimental results showed that IGA achieved significant performance improvement in node localization. Compared with the centroid algorithm, the maximum error of IGA has been reduced by 19% and the overall average error has been reduced by 8.8%. Compared with APIT, IGA has reduced the maximum error by 7% and the overall average error by 3.8%. Regarding fitness values, IGA exhibited faster convergence speed, achieving optimal results with only 75 iterations, surpassing traditional genetic algorithms and APIT algorithms. The node coverage rate reached 98.6%, far higher than the 85.3% of the centroid algorithm and 90.5% of the APIT algorithm. These results demonstrate that IGA has higher accuracy, stability, and computational efficiency in complex network environments, providing an efficient and reliable solution for optimizing the design of computer network nodes.

Povzetek: Predlagan je izboljšan genetski algoritem (IGA) za optimizacijo lokacij vozlišč v računalniških omrežjih, ki z dinamičnim prilagajanjem verjetnosti križanja in mutacije poveča točnost, stabilnost in učinkovitost algoritma.

1 Introduction

With the continuous progress of modern technology, computer networks play an increasingly important role in modern society. They connect various devices and systems, making the transmission and sharing of information more efficient and convenient. To meet the needs of users for high-quality network services, improving network performance and optimizing network design have become increasingly important. Traditional optimization algorithms frequently encounter issues of low efficiency and a propensity to fall into local optima when addressing large-scale network design problems. Therefore, it is necessary to introduce new optimization algorithms to solve these problems [1-2]. In recent years, researchers have made significant advancements in applying enhanced genetic algorithms to optimize computer networks. These enhancements include the introduction of new operators, optimization of algorithm parameters, and adjustments to the algorithms themselves. As a result, genetic algorithms are now more efficient and accurate when utilized for network design optimization. At the same time, researchers combine genetic algorithms with other optimization algorithms to create multiple hybrid optimization algorithms, which enhances network design performance and effectiveness [3-4]. The objective of the research is to achieve an optimized design of computer networks and to improve network performance indicators, including latency, throughput, resource utilization, and cost, through an Improved Genetic Algorithm (IGA). The research aims to solve the problems of slow convergence speed, susceptibility to local optima, and difficulty in dynamic adjustment of traditional network optimization methods in complex network environments. This study designs a computer network optimization technique based on a genetic algorithm as the core and introduces multiple techniques to improve performance. A fitness function based on network performance indicators is constructed to quantify the network optimization objectives. This technology adjusts the crossover and mutation probabilities adaptively by comparing individual fitness

and population average fitness, balancing global and local search capabilities.

2 Related work

The 5G era is coming and the network technology is developing rapidly. Numerous data have brought enormous challenges to the stability and reliability of computer networks. The reliability of computer networks is a major indicator of computer comprehensive performance. Computer networks are large and complex, and they are also easily affected by many adverse factors. This leads to instability in the system, which exposes the entire computer network to significant risks. To ensure the stability and ongoing optimization of computer networks, computer network optimization design has become a prevalent point of discussion in computer research. Through their study of cloud computing, Fan et al. [5] presented a novel mathematical model for virtual network embedding in optical data center networks. This model reduced Network Topology (NT) complexity during optical fiber transmission. They used a comprehensive system of node awareness and path evaluation to derive algorithms with priority locations. The algorithm obtained by this model could reduce the latency of virtual network requests by 20% and improve the request rate by 13%. Rajendran and Venkataraman [6] proposed a new neural network algorithm to analyze network traffic built on the application and analysis of big data in network security. They used this method to conduct statistics on the worst data and abnormal activity sent by the network and conducted experiments with the data. Compared with traditional algorithms for neural networks, the optimized algorithm showed a notable enhancement in distinguishing between false alarms and actual detection, which significantly improved the security and stability of the network. Xiaokaiti et al. [7] raised an efficient data transmission strategy for the detection algorithm of computer network communities. They first combined NT attributes with social attributes when dividing communities and then selected the optimal relay node for network transmission based on the number of channels. This algorithm had high merit in data delivery efficiency and routing overhead in computer networks. Alsaqour et al. [8] put forward a location-assisted routing algorithm grounded on genetic algorithms to optimize the efficiency of MANET routing protocols. Firstly, through algorithm optimization, node information was added to the route and these nodes were grouped. These nodes were then sent to their destinations to adaptively update the node location. The results showed that the optimized algorithm could achieve a delivery rate of over 99% for small network overhead packets. Bu [9] developed a load-balancing scheduling algorithm for Internet of Things (IoT) clusters using a combination of Particle Swarm Optimization and Genetic Algorithm (PSOGA). The purpose of this algorithm was to address the persistent challenge faced by IoT networks due to high-volume business data traffic causing downtime. They first used the CPU, RAM, and network bandwidth to measure the server node information, then adjusted the appropriate function value, and used the IGA to obtain the optimal solution. The results showed that the optimized algorithm could reduce latency and error rates by 5%, while also reducing server overload and downtime. Network coding could integrate coding capabilities with network multi-path propagation, bolster the capacity of computer networks, and facilitate more intricate security solutions. To address the susceptibility of network coding to attacks, Wu et al. [10] developed a comprehensive unicast secure transmission scheme based on Random Linear Network (RLN). The matrix was randomly generated from the received nodes and the resulting vector was sent back to the source node via the link to form a new matrix. This approach effectively thwarted network eavesdropping attacks. The comparative analysis between the research and the advanced methods is shown in Table 1.

Table 1: Comparative analysis of research and the advanced methods

Reference	Technical Method	Advantages	Disadvantages	Comparison with IGA
Fan et al. [5]	Virtual network embedding with node awareness and path evaluation.	Reduces latency by 20% and improves request rate by 13%.	Limited applicability; does not optimize node positioning.	IGA reduces error by 8.8%, with broader applicability.
Rajendran et al. [6]	Enhanced neural network algorithm for malicious traffic detection.	Improves security and reduces false alarms.	High computational cost; lacks node optimization.	IGA achieves 2.41% error, with higher efficiency.
Xiaokaiti et al. [7]	Community detection algorithm to optimize data transmission.	Improves transmission efficiency and reduces routing overhead.	Dependent on BT; limited precision.	IGA reduces error by 3.8%, offering better stability.
Alsaqour et al. [8]	Genetic algorithm for optimizing mobile ad	Achieves 99% small packet delivery rate with	Suitable for small networks; struggles	IGA reduces error to 2.46%, with

	hoc network routing.	low overhead.	with large-scale networks.	wider applicability.
Bu [9]	PSO and GA combined for load balancing.	Reduces load and downtime by 5%.	Focuses on load balancing; lacks positioning accuracy.	IGA improves accuracy by 8.8%, offering a comprehensive solution.
Wu et al. [10]	Secure transmission using random linear network coding.	Enhances security and prevents eavesdropping.	Does not optimize node positioning or transmission efficiency.	IGA achieves 5.2% error, with better precision and stability.

Previous research has found that related work mainly focuses on specific aspects of computer network optimization, including security enhancement, data transmission efficiency, and load balancing. However, they have shortcomings in addressing the accuracy of network node localization and overall stability under different network conditions. Existing algorithms such as the centroid algorithm and Approximate Point in Triangulation Test (APIT) have significant drawbacks, including limited accuracy and sensitivity to node density. Traditional algorithms, such as genetic algorithms and MANET routing protocols perform well in specific network types, but perform poorly in large-scale or dynamic environments. Using genetic algorithms to assist routing protocols can improve network overhead and delivery rates. This fully optimizes the genetic algorithm and enhances network delivery. To optimize computer network nodes for better environmental conditions, this study uses centroid and APIT algorithms, which provide better conditions for computer network optimization. Then, based on node optimization, an IGA is used to construct a network design optimization model. Through optimizing the traditional ant colony algorithm, the efficiency of network nodes in computer network optimization design is enhanced. This paper aims to increase the stability and reliability of computer network optimization design.

3 Construction of computer network optimization design model based on genetic algorithm

3.1 Optimization of node location based on centroid algorithm and APIT algorithm

From the perspective of topology, a computer network is composed of several network nodes and communication links connecting these network nodes. This indicates that the positioning of network nodes is indispensable in computer network data transmission. The centroid algorithm is the most typical node localization algorithm among commonly used localization algorithms. The algorithm has four advantages: low storage energy consumption, simple algorithm principle, low computing energy consumption, and low communication energy consumption.

Before using this algorithm for localization, it is first necessary to determine whether the location node that the sensor needs to determine is located within the region. At the same time, nodes requiring location determination will continually emit various

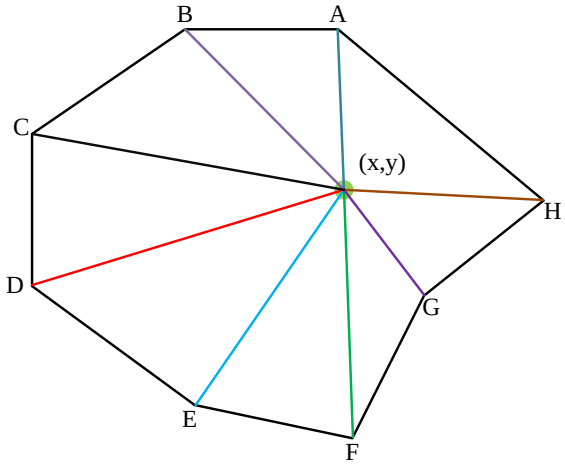


Figure 1: Schematic diagram of centroid algorithm positioning

communication signals to the surrounding environment. To determine whether the unknown node is in the monitoring area, it is essential to verify the strength of the signal obtained at the beacon node. The strength can reflect the unknown node location [11]. The principle of the centroid algorithm is obtained built on the algorithm for the centroid. In any irregular polygon, there must be a center of mass inside it. Usually, the coordinates of each vertex are accumulated, and then the average value is calculated to determine its specific coordinates. The specific location can be represented by Formula (1), and its algorithm diagram is shown in Figure 1.

$$(x,y)=(\frac{1}{n}\sum_{i=1}^{n}x_i,\frac{1}{n}\sum_{i=1}^{n}y_i) \tag{1}$$

In Formula (1), n represents a n-sided shape. (x,y) means the coordinate of the vertex. The centroid of this n-sided shape can be obtained by calculating the formula.

If the polygon is situated within the solved region and has matching coordinates, then the centroid coordinates of the octagon can be computed using Formula (2).

$$(x,y)=(\frac{x_1+x_2+\cdots+x_8}{8},\frac{y_1+y_2+\cdots+y_8}{8}) \tag{2}$$

The $(x_1,y_2)\cdots(x_8,y_8)$ in Formula (2) represents the coordinates of eight vertices. To use centroid positioning algorithms for positioning, it is essential to rely on the smoothness of the entire network structure and the specific distribution of positioning nodes within the network. If an error occurs in the coordinates calculated by unknown nodes, it will bias towards areas with densely distributed beacon nodes, potentially resulting in significant errors with the centroid algorithm. Therefore, the algorithm's calculation accuracy is typically not high, and the positioning accuracy may be low. However, the centroid algorithm only needs to broadcast once to locate all unknown nodes. In many applications that do not require high positioning accuracy, the centroid algorithm is still the most suitable method.

APIT is an improved algorithm for the centroid algorithm. It requires a completely random selection of many known coordinate nodes, and the coordinate nodes are grouped every three. In accordance with these nodes, the triangles drawn on the graph will be completely randomly distributed throughout the entire region. There will be some overlap between these triangles to calculate the coordinates of unknown nodes. The specific operation steps are: First, multiple coordinate nodes around unknown nodes are identified, and three known location nodes are randomly

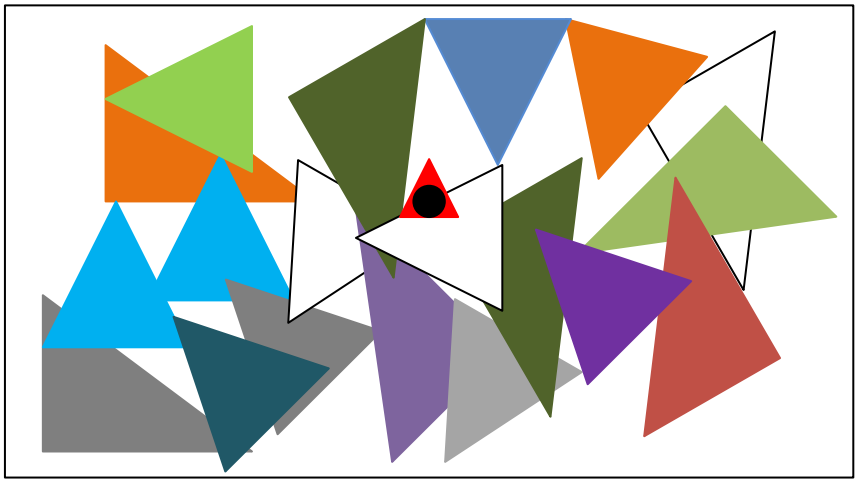


Figure 2: Schematic diagram of APIT algorithm positioning

selected each time. Then, the approximate location of the signals received by these known location nodes is determined. If there are m beacon nodes, the paper will randomly select and match them, and use the combination of three random position points to form C_n^3 triangle. It shows that some triangular regions can contain unknown nodes, while others do not. These specific points, which contain triangular regions of unknown nodes, are connected to each other. Finally, the recorded location algorithm is utilized to calculate the specific location of unknown nodes. The incorporation of a greater number of unknown nodes into the algorithm results in enhanced accuracy in location estimation [12-13]. However, this is accompanied by an increased computational burden. In such a scenario, choosing a subset of vertices to create a polygon based on the real circumstances, as illustrated in Figure 2, can be beneficial. In Figure 2, the node positioning accuracy of the APIT is significantly greater than that of the centroid positioning algorithm.

Due to the relatively large impact of node density on APIT, when the beacon node density is relatively large, APIT can achieve relatively ideal positioning accuracy.

APIT also has good performance in irregular wireless signal propagation models and irrational circular propagation models. However, APIT also has a significant disadvantage. When connecting triangles, it may mistake points located outside the triangle for points inside the triangle. The probability of generating this situation can reach a maximum of 13% through research [14], which will have a significant impact on positioning accuracy. The algorithm must divide a large number of triangular regions to identify the locations of unknown nodes and necessitates multiple beacon nodes. As a result, the algorithm performs numerous calculations, which elevates the likelihood of encountering errors.

3.2 Construction of improved genetic algorithm model

The research and analysis of the centroid and APIT algorithms in node location optimization have revealed shortcomings in both algorithms concerning their calculation and location processes. Additionally, the use of genetic algorithms for node localization requires extra constraints, which may lead to increased computational time and reduced efficiency, resulting in premature convergence [15]. To obtain better positioning optimization results, an IGA model is studied and constructed. The flow chart of the model is shown in Figure 3, and the blue box in the figure shows the improved steps.

Compared with Traditional Genetic Algorithms (TGA), the paper has improved the node localization of genetic algorithms and constructed a matrix. The specific construction of the matrix is shown in Formula (3).

$$S(m)=\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{3}$$

The m in Formula (3) represents a total of m chromosomes in the genetic algorithm. n means that each chromosome has n elements. s_1, s_2 $\cdots$ s_m are each chromosome. In TGA, determining the value range of genes is a commonly used method to generate an initial population waiting for calculation. If a certain number of initial populations are randomly generated within this value range, there may be situations where the distribution is too random. This result is basically not helpful for improving the algorithm efficiency [16]. If one wants to obtain a global optimal solution, the distribution of the initial population in the solution space should be as uniform and dispersed as possible. The schematic diagram of random initial population generation within the overlapping range of communication areas of different anchor nodes is shown in Figure 4.

IGA performs improved optimization in TGA such as parameter setting, population initialization, appropriate function values, selection operations, and crossover operations. The specific key parameter settings are to set the number of populations to be 40, the crossover probability p_{c1} to be 0.6, p_{c2} to be 0.4, the mutation probability p_{m1} to be 0.08, p_{m2} to be 0.06, and the maximum iterations to be 100. Population initialization can determine the initial population range according to Formula (4), and generate an initial population randomly within this range.

$$\begin{cases} \max\limits_{i=1,2,\cdots,n}(x_i-d_i)\le x\le \min\limits_{i=1,2,\cdots,n}(x_i+d_i) \\ \max\limits_{i=1,2,\cdots,n}(y_i-d_i)\le y\le \min\limits_{i=1,2,\cdots,n}(y_i+d_i) \end{cases} \tag{4}$$

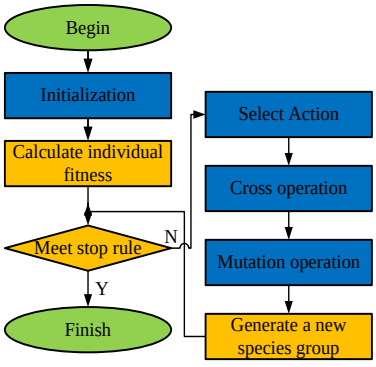


Figure 3: The improvement process of genetic algorithms

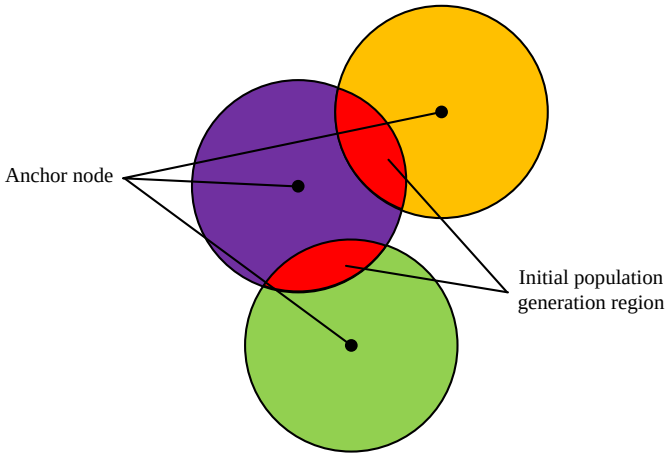


Figure 4: Schematic diagram of initial population generation area

In Formula (4), d_i means the distance between the node of unknown i and the anchor. The fitness function value calculation assumes a total of $(M+N)$ nodes in the wireless sensor network to be located, where the number of known nodes is *M*. The unknown node number is *N*. Through a certain distance measurement method, if each unknown node knows the distance between all known nodes within the communication radius and itself, the calculated node position can be obtained through the least square method [17]. Assuming that the coordinate of a node at a known location is $(x_1,y_1),(x_2,y_2),\ldots,(x_M,y_M)$, the coordinate of an unknown node is (x,y), and the distance from the node at a known location is $d_1,d_2,d_3,\cdots,d_M$. The equation set shown in Formula (5) can be established.

$$\begin{cases} (x-x_1)^2+(y-y_1)^2=d_1^2 \\ (x-x_2)^2+(y-y_2)^2=d_2^2 \\ (x-x_3)^2+(y-y_3)^2=d_3^2 \\ \vdots \\ (x-x_M)^2+(y-y_M)^2=d_M^2 \end{cases} \tag{5}$$

From Formula (5), the fitness function for genetic algorithms can be defined as Formula (6), and the fitness function of the initial population can be calculated by using it.

$$f(x,y)=\frac{1}{M}\sum_{i=1}^{M}\left|\sqrt{(x-x_i)2+(y-y_i)2}-d_i\right| \tag{6}$$

In Formula (6), (x,y) is the unknown location node location. (x_i,y_i) represents a known location node location. d_i refers to the distance from an unknown location to a known location (x_i,y_i). The use of absolute error instead of squared error in the fitness function avoids the calculation of squared error, reduces complex multiplication operations, and lowers computational load. During the iteration process, absolute error is more robust to outliers (i.e. has less impact on data with larger deviations) and also enables the algorithm to approach the global optimal solution faster, accelerating the convergence speed of the algorithm. The selection operation is to perform a unified comparison of each individual based on the fitness value calculated from the fitness function. After the comparison is completed, the two individuals with the highest fitness remain unchanged and proceed to the next round of operation. The individuals with the lowest fitness are directly eliminated, and the remaining objects will normally undergo crossover and mutation operations. Special individuals with high fitness values are set with judgment values to distinguish and limit their reproduction. After completing the full iterative process, the fitness value of each individual should be appropriately amplified [18-19]. The crossover probability is used to control the probability of individuals (chromosomes) performing crossover operations. By calculating an individual's fitness value, it can be determined whether the individual should participate in crossover operations. The goal of crossover operation is to generate offspring with higher fitness by recombining the genetic information of the parent individual, gradually approaching the optimal solution. When performing a crossover operation, if $F_g \geq F_{avg}$, the crossover probability is calculated according to Formula (7).

$$P_c=p_{c_1}\frac{F_g-F_{\arg}}{F_{gb}-F_{avg}} \tag{7}$$

In Formula (7), $pc_1 \in (0,1)$. F_g is the value of the individual's fitness function. F_{gb} represents the optimal individual adaptation function value. F_{avg} is the average of the adaptive function. The calculation process includes normalizing the fitness difference and converting the normalized fitness difference into actual cross probability. If $F_g < F_{avg}$, the crossover probability is calculated according to Formula (8).

$$P_c=P_{c2} \tag{8}$$

In Formula (8), $P_{c2} \in (0,1)$. After pairing the chromosomes in the population, the crossover operation is performed based on the calculated crossover probability. A random number between [0-1] is randomly

generated for each chromosome. The objective of the treatment of poorly adapted individuals is to provide those with lower fitness a certain opportunity to participate in crossover, increase population diversity, and circumvent premature convergence to local optimal solutions. If the corresponding value is less than the crossover probability, the chromosome is ready to perform the next operation. The chromosomes for the next step are sequentially crossed in pairs. For each pair of crossed chromosomes, the location of the crossing is determined by random numbers and the crossover operation is performed. During the mutation operation, if $F_g \geq F_{avg}$, the mutation probability is calculated by Formula (9).

$$P_m = p_{m1} \frac{F_g - F_{avg}}{F_{gb} - F_{avg}} \tag{9}$$

In Formula (9), $p_{m1} \in (0.01, 0.10)$. F_g represents the adaptive degree function value of individual g . F_{gb} is the value of the optimal individual adaptation function. F_{avg} refers to the average value of the adaptive function. If $F_g < F_{avg}$, the probability of variation is calculated by Formula (10).

$$p_m = P_{m2} \tag{10}$$

In Formula (10), $P_{m2} \in (0.01, 0.10)$. The first step is to randomly generate a number between 0 and 1 for each chromosome in the population. If the generated value is less than the mutation probability, it will undergo mutation operation. The position corresponding to the required mutation is determined by generating a random number value, and the next step is to invert the value to complete the relevant mutation operation. Monotonic gene locus detection analyzes the whole population and identifies any monotonic gene loci present. If these are detected, targeted adjustments can be made by generating random numbers. The termination operation should be determined and the loop termination should be evaluated based on the number of iterations. The final step is to output the optimal solution and test the average positioning error of the algorithm as a performance parameter, as shown in Formula (11).

$$error = \frac{100}{N \times R} \sum_{i=1}^{N} \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2}\% \tag{11}$$

N in Formula (11) is the sum of nodes. (x_{i1}, y_{i1}) and $(x_{i2}, y_{i2}) i = (1, 2, 3, \cdots, M)$ represent the actual and calculated coordinates of the unknown node i . R is the maximum communication distance of the node. In the process of transforming IGA theory into practical applications, the rigor of mathematical analysis is reflected in the precise modeling and dynamic adjustment of fitness functions, crossover probabilities, and mutation probabilities. The dynamic allocation of crossover probability is achieved by comparing individual fitness with group average fitness and optimal fitness. This allows individuals with higher fitness to have a higher probability of crossover, thereby accelerating the spread of excellent genes, while preserving a small number of crossover opportunities for individuals with lower fitness and maintaining population diversity. This normalization mechanism based on fitness differences effectively balances local search and global search, avoids premature convergence of the algorithm, and improves solution accuracy and efficiency. Furthermore, the implementation of random number generation techniques and probability judgment processes enables the transformation of theoretical models into practical operations, thereby ensuring the randomness and controllability of crossover and mutation. This approach facilitates the robustness and convergence of the algorithm in complex optimization problems, thus achieving efficient integration of theory and practice. When using IGA for computer network optimization design, the network optimization problem is first modeled as a fitness function to measure network performance indicators. Then, through iterative evolution through selection, crossover, and mutation operations, the crossover and mutation probabilities are dynamically adjusted to optimize the network structure and parameter configuration, thereby achieving efficient and accurate network optimization design.

4 Performance analysis of computer network optimization design model based on genetic algorithms

4.1 Performance analysis of node location based on centroid location algorithm and APIT algorithm

To verify the actual positioning effects of the centroid algorithm, APIT, and IGA, simulation experiments are conducted on three algorithms in MATLAB. The reason for choosing APIT and centroid algorithm as benchmarks for research is their effectiveness and wide application in network optimization. The APIT algorithm performs well in localization problems and is suitable for evaluating the accuracy and reliability of network nodes, serving as a benchmark for network performance optimization in research. The centroid algorithm is known for its simplicity, ease of use, and fast convergence, making it suitable for solving optimization problems in basic network structures. The selection of these two algorithms perfectly covers different types of network optimization requirements. Through comparison, the advantages of IGA in solving complex optimization problems can be clearly demonstrated. Using MATLAB version R2021a, the hardware specifications are as follows: Intel Core i7-9700K processor, 32 GB DDR4 RAM, 512 GB solid-state drive, and Windows 10 Professional 64 bit operating system. The algorithm sets the population size

to 100, the number of iterations to 500, the crossover probability to 0.8, and the mutation probability to 0.05. The elite strategy is to retain the top 10% of excellent individuals. To ensure the statistical validity of the test scenario, multiple sets of experiments are designed and optimized for network topologies of different sizes and complexities. Each experiment should be repeated at least 30 times to obtain stable average performance indicators and standard deviations, ensuring the reliability of the results. In the collected performance indicators, statistical analysis is used to evaluate the significant differences in algorithms under different configurations, thereby determining the efficacy of the optimization effects. When comparing, a null hypothesis and an alternative hypothesis are set. When calculating the P-value, it represents the probability of obtaining the current or more extreme result under the null hypothesis. The t-test is used to compare the results of IGA and benchmark algorithms. If the P-value is less than 0.05, the difference is considered statistically significant. The experiment generates 20 anchor nodes and 80 unknown nodes in the 100×100 area. After generating this region, the node is predicted by running the corresponding algorithm. Figure 5 shows the original node distribution diagram.

The green circle in Figure 5 represents an anchor node, while the blue pentagon represents an unknown node. Figure 6 shows the positioning results of three algorithms. The positioning results of centroid algorithm, APIT, and IGA are shown in Figure 6(a), Figure 6(b) and Figure 6(c), respectively.

In Figure 6, the predicted value of IGA has a higher coincidence rate with unknown nodes, reaching 94.36% ($P<0.05$). The predicted value of the centroid algorithm has the lowest coincidence rate with unknown nodes, which is 86.25%. The coincidence rate between the predicted value of APIT and the unknown node is 89.67%. The coincidence rate of the IGA is 8.16% higher than that of the centroid algorithm and 4.69% higher than that of the APIT algorithm ($P<0.05$). This means IGA has a high positioning computing ability. The positioning errors of the centroid algorithm, APIT, and IGA are listed in Figure 7.

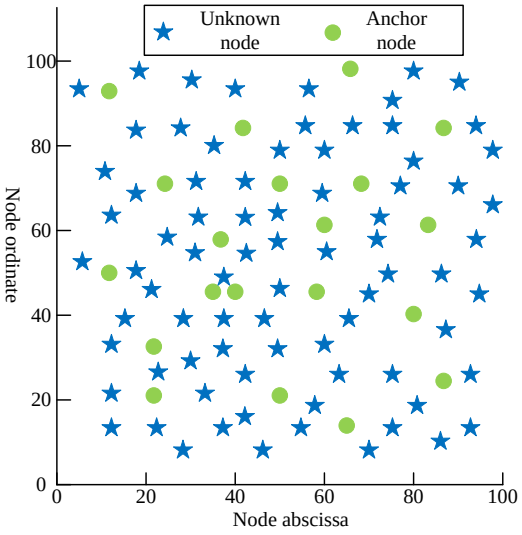
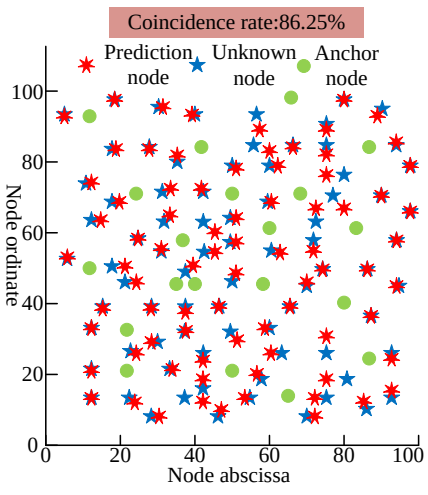
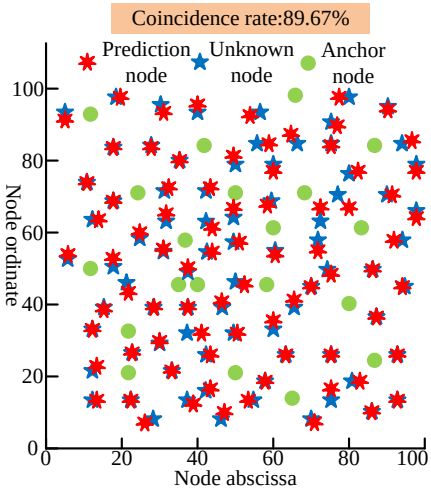


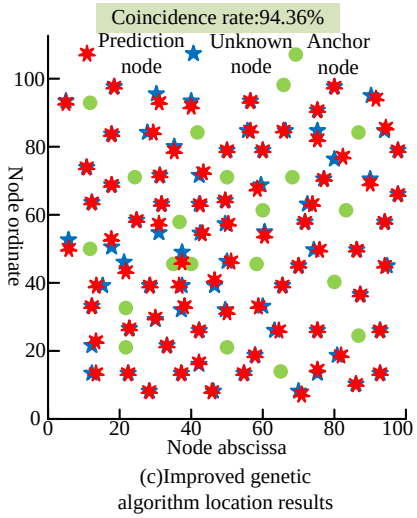
Figure 5: Original node distribution diagram



(a)Centroid algorithm positioning results



(b)ATIP positioning results



(c)Improved genetic algorithm location results

Figure 6: Location results of three algorithms

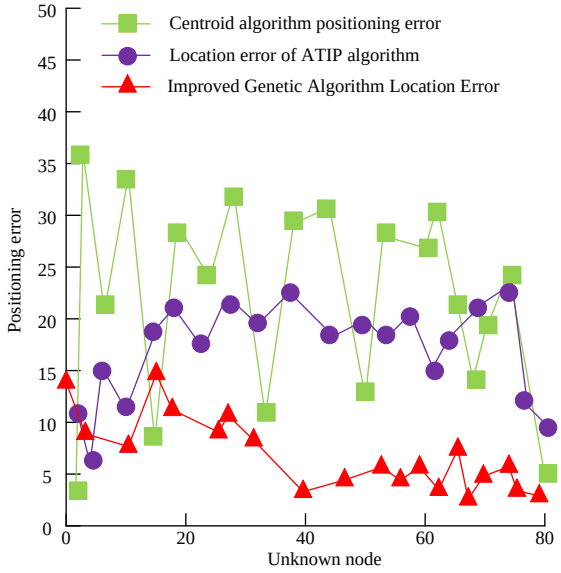


Figure 7: Positioning error of three algorithms

The centroid algorithm in Figure 7 achieves a maximum error rate of 32% during positioning, and the overall node positioning average error rate is 14% (P<0.05). The maximum error rate of APIT in positioning is 20%, but the average value of the overall error has decreased to 9% (P<0.05). This indicates that compared to the centroid positioning algorithm, the predicted coordinates error calculated by the APIT positioning algorithm is significantly reduced, with better positioning results. The maximum error rate of IGA during positioning is 13%, and the average value of its overall error is 5.2% (P<0.05). The maximum error of IGA is 19% lower than the centroid algorithm, and the overall average error is 8.8% lower (P<0.05). Compared to APIT, the maximum error of IGA is 7% lower, and the average overall error is 3.8% lower (P<0.05). APIT improves positioning accuracy through random triangle coverage, but relies on high-density anchor nodes and is prone to misidentifying external points of the triangle as internal points. The centroid algorithm is computationally simple and suitable for scenarios that do not require high accuracy. However, its accuracy is low and it is easily affected by uneven distribution of node density, resulting in positioning bias towards areas with dense anchor nodes and significant errors. IGA introduces a method of dynamically adjusting crossover probability and mutation probability during the evolution process. It dynamically adjusts based on individual fitness and population average fitness, avoiding premature convergence of the algorithm and ensuring the search for the global optimal solution. IGA performs local fine optimization, improving the accuracy and stability of the algorithm. The comparison of the three shows that IGA has excellent node positioning capabilities in wireless sensor networks.

4.2 IGA-based application analysis of computer network optimization design

There are differences in the performance of IGA and other node localization algorithms for wireless sensor networks under different iterations. Hence, under the same parameter conditions, this study gradually changes the iterations and simulates them using TGA, centroid positioning algorithms, APIT, and IGA, respectively. Thus, the iteration number

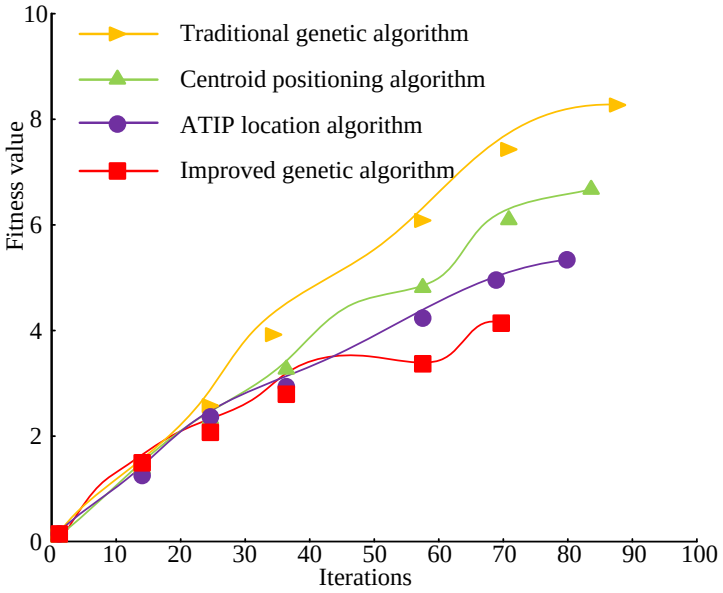


Figure 8: Changes in fitness values of the four methods

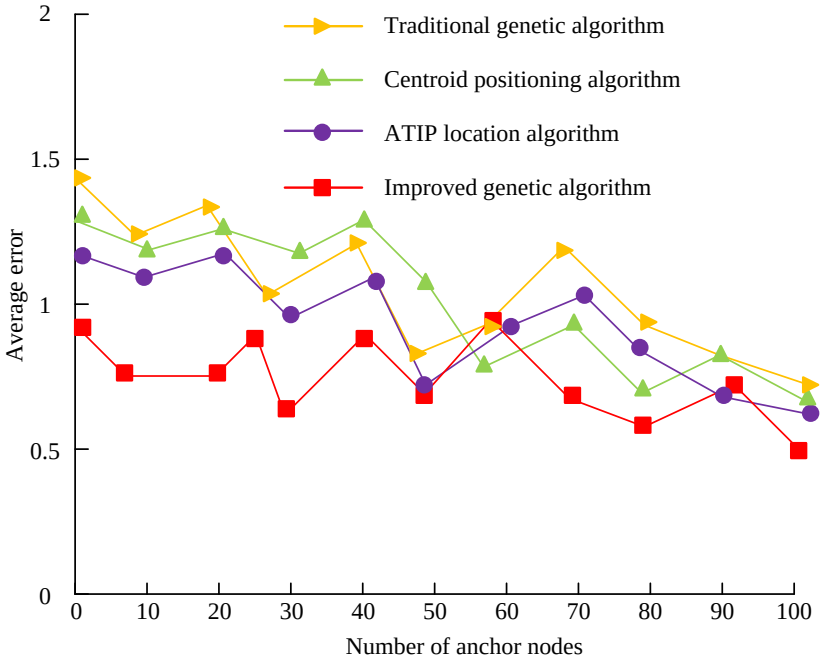


Figure 9: Relationship between the number of anchor nodes and average error

and its corresponding fitness value are obtained. As the number of iterations gradually increases, Figure 8 illustrates the corresponding changes in fitness between the IGA and other algorithms for positioning wireless sensor network nodes.

Figure 8 shows that the fitness values of the four localization algorithms are all less than 10. The fitness values of IGA, TGA, centroid algorithm, and APIT are 4.26, 8.15, 6.42, and 5.31, respectively. The iteration numbers are 69, 86, 83, and 79, respectively. The IGA's fitness value is the lowest, 3.89 lower than that of TGA,

2.03 lower than centroid algorithm, and 1.05 lower than APIT. This shows that IGA has better adaptability in node localization, and verifies the superiority of this algorithm. The anchor node numbers and the average error value's relationship of the four algorithms is shown in Figure 9.

In Figure 9, as the anchor node amount increases, the average error of the four positioning algorithms is gradually decreasing. The average errors of TGA, centroid algorithm, APIT, and IGA are 1.12, 1.03, 0.95, and 0.68, respectively ($P<0.05$). The average error of IGA is significantly lower than that of TGA. Meanwhile, as the number of anchor nodes increases gradually in proportion to all nodes, the average error of various algorithms undergoes slight changes over time, as shown by the curve. As the anchor node amount keeps the same, the average error of IGA is the lowest among all four algorithms. This fully demonstrates the advantages of IGA. Figure 10 displays the chart that pertains to the communication radius of a particular node and the average error value of the four algorithms.

In Figure 10, the communication radius of nodes is increasing, while the average error of the four node positioning algorithms is gradually decreasing. The average errors of TGA, centroid algorithm, APIT, and IGA are 5.75%, 4.52%, 3.87%, and 2.46%, respectively ($P<0.05$). When the communication radius is the same, the IGA's average error is always at the lowest value. This verifies the superiority of IGA when the communication radius of nodes changes. Figure 11 is the relevant of the specific network connectivity and the average error value of the four algorithms.

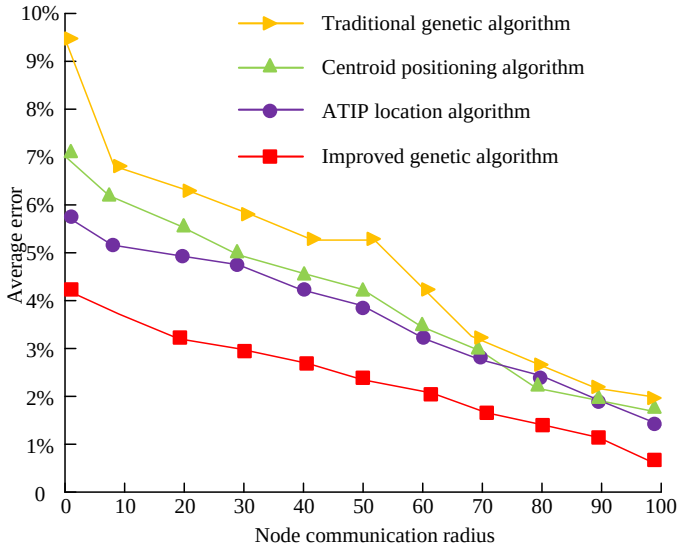


Figure 10: Relationship between node communication radius and average error

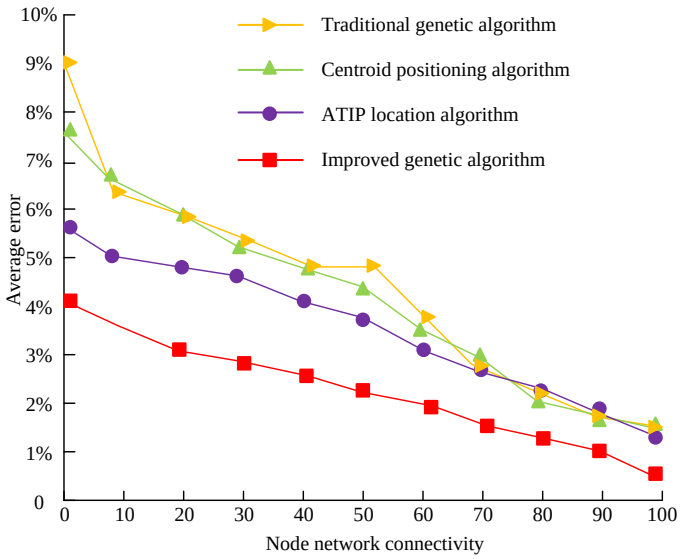
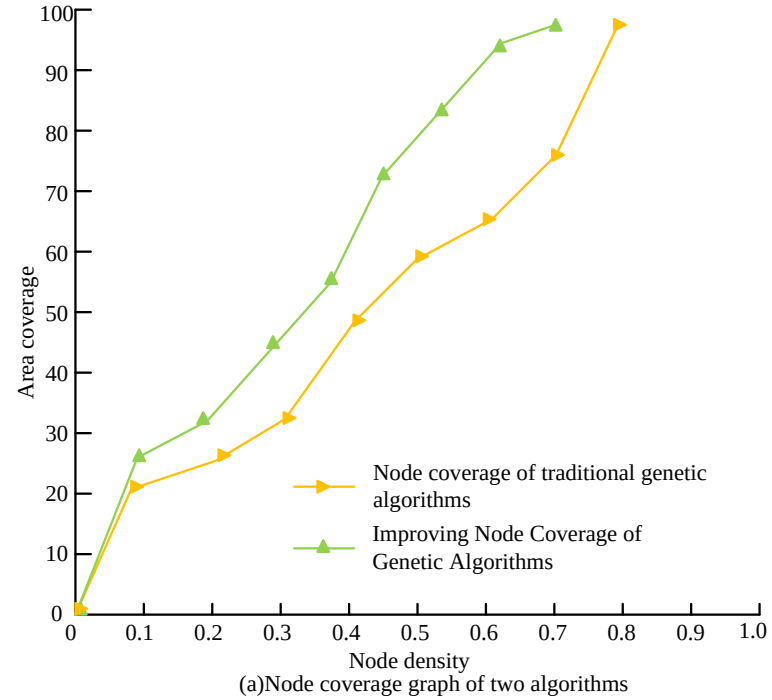


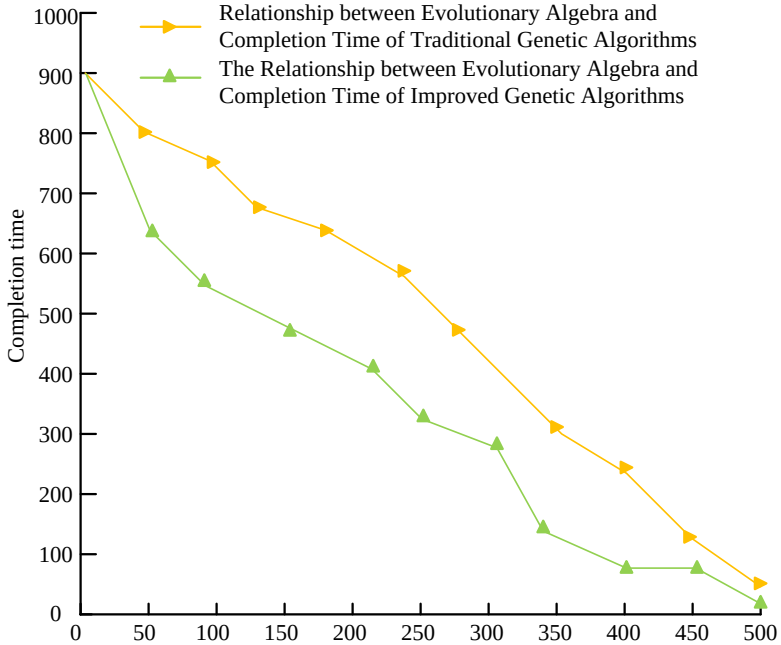
Figure 11: Relationship between network connectivity and average error value

The average errors of TGA, centroid algorithm, APIT, and IGA in Figure 11 are 5.41%, 4.49%, 3.71%, and 2.41%, respectively ($P<0.05$). This indicates that regardless of changes in network connectivity, IGA's positioning ability is always higher than the other three algorithms. Figure 12 is the graph about connection between node coverage, evolutionary algebra, and completion time of TGA and IGA.

The node coverage of the two algorithms in Figure 12(a) shows: in the same node density, the IGA has a higher regional coverage. Figure 12(b) displays the relationship between the number of evolutions and completion time of both algorithms over



(a)Node coverage graph of two algorithms



(b)Evolution algebra and completion time graph of two algorithms

Figure 12: The relationship graph between node coverage, evolutionary generations, and completion time of two algorithms

successive iterations of the genetic algorithm. As the iterations progress, the number of evolutions gradually increases while the time required to complete all iterations decreases. However, with the same evolutionary algebra, IGA takes less time. This shows the superiority and stability of the IGA. To further analyze the adaptability and superiority of the research method, further application analysis is conducted in a large-scale wireless sensor network node positioning scenario with a side length of 500m in a region. The total number of sensor nodes in the region is 500, including 100 anchor nodes and 400 unknown nodes. The results of the large-scale application analysis are shown in Table 2.

Table 2: Results of application analysis in major scenarios

Metrics	IGA	APIT Algorithm	Centroid Algorithm
Average Positioning Error (%)	2.45	5.62	7.89
Positioning Time (s)	12.3	18.4	9.6
Number of Iterations	75	120	60
Convergence Speed	Fast (0.5 fitness variation)	Medium (0.8 fitness variation)	Slow (1.2 fitness variation)
Node Coverage Rate (%)	98.6	90.5	85.3

As shown in Table 2, the average positioning error of IGA is 2.45%, significantly lower than the 5.62% of APIT algorithm and the 7.89% of centroid algorithm ($P<0.05$). This indicates that IGA can effectively improve the accuracy of node localization in large-scale scenarios and is suitable for complex and high-precision network environments. The positioning time of IGA is 12.3 seconds, which is between 18.4 seconds of APIT and 9.6 seconds of centroid algorithm ($P<0.05$). Although the computational complexity is slightly higher than that of the centroid algorithm, IGA improves efficiency by optimizing the evolution process, enabling it to maintain fast computational speed while achieving high-precision positioning. After 75 iterations, IGA achieves convergence, which is faster than the APIT algorithm's 120 iterations ($P<0.05$), demonstrating the advantages of IGA's dynamic parameter adjustment and elite strategy in the search process. Although the centroid algorithm has fewer iterations, its accuracy is significantly insufficient ($P<0.05$). The node coverage rate of IGA reaches 98.6%, which is much higher than the 90.5% of APIT algorithm and the 85.3% of centroid algorithm ($P<0.05$). This indicates that IGA has better coverage performance in large-scale networks and can optimize node positioning layout more comprehensively.

4.3 Discussion

This study has designed an IGA that effectively improves the accuracy and stability of node localization in wireless sensor networks through techniques such as dynamic parameter adjustment, fitness function optimization, and elite strategy. Compared with traditional centroid algorithms and APIT algorithms, IGA exhibits significant advantages in key performance indicators. Specifically, the average positioning error of IGA was 2.45%, which was much lower than APIT's 5.62% and centroid algorithm's 7.89%, indicating that IGA has significant advantages in node positioning accuracy. At the same time, IGA had a faster convergence speed, requiring only 75 iterations to reach the optimal solution, with a stable fitness value change (0.5). APIT and centroid algorithms required 120 and 60 iterations, respectively, and had slower convergence speeds. In addition, IGA achieved a node coverage rate of 98.6%, significantly higher than APIT (90.5%) and centroid algorithm (85.3%), demonstrating its applicability and advantages in large-scale complex network environments. The reason why IGA outperforms traditional methods in terms of positioning error and fitness values is mainly due to several key technological innovations. Firstly, the dynamic parameter adjustment mechanism can dynamically adjust the crossover probability and mutation probability based on the fitness value, thereby balancing global and local search and preventing the algorithm from getting stuck in local optimal solutions. This is consistent with the ideas of Yu et al. [20]. Secondly, fitness function optimization reduces computational complexity and enhances robustness to outliers by introducing absolute error instead of traditional square error, enabling the algorithm to approach the global optimal solution more quickly. In addition, the elite strategy ensures the retention of high fitness individuals and reduces the loss of high-quality solutions. The uniform distribution of the initial population within the communication area improves search efficiency and reduces ineffective calculations caused by random initialization. The results obtained are consistent with Singh et al.'s study [21]. These improvements effectively address common pitfalls of TGAs, such as local optima and premature convergence, enabling IGA to exhibit higher stability and accuracy in complex dynamic network environments. The core innovation of IGA lies in combining the local improvement of TGAs with global search, which is suitable for non-standard situations such as uneven node

distribution, limited number of anchor nodes, and complex conditions such as changes in communication radius. In practical applications, IGA demonstrates good stability and adaptability by flexibly adjusting parameters and optimizing search space. In previous studies, TGAs often faced local optimal traps, leading to premature convergence of the algorithm. The study aims to enhance population diversity, reduce the interference of outliers on the search process, and accelerate the convergence of the global optimal solution by providing low fitness individuals with moderate opportunities for crossover and mutation. The research provides a more stable, accurate, and efficient solution for node localization and optimization in complex network environments.

5 Conclusion

The high-speed development of computer network technology has caused tremendous changes in people's production and life. Currently, computer network optimization still has the problem of low positioning accuracy of network nodes. To solve the related problems, this study constructed an IGA model and applied it to computer network optimization. Experimental results showed that IGA has significantly improved location coverage and average location error compared to centroid algorithm and APIT. The coincidence rate of the improved algorithm was 8.16% higher than centroid algorithm's and 4.69% higher than that of the APIT algorithm. The maximum error of IGA was 19% lower than the centroid algorithm, and the overall average error was 8.8% lower. Compared to APIT, the maximum error of IGA was 7% lower, and the average overall error was 3.8% lower. Under the same parameters, TGA, centroid algorithms, APIT algorithms, and IGA were used to compare the performance of network nodes in computer networks. Experimental data were obtained: the fitness value of IGA, the amount of anchor nodes and the average error, the communication radius and the average error, and the network connectivity and the average error were 4.26, 0.68, 2.46, and 2.41, respectively. IGA had a significant improvement over the calculated values corresponding to the three algorithms, which proves the accuracy and stability of the improved genetic positioning algorithm.

6 Abbreviated List

NT: Network Topology
PSOGA: Particle swarm optimization and genetic algorithm
RLN: Random linear network
IGA: Genetic Algorithm
TGA: Traditional Genetic Algorithm
APIT: Approximate Point In Triangulation Test

References

[1] F. Wang, X. Lai, and N. Shi, "A multi-objective optimization for green supply chain network design," Decision Support Systems, vol. 51, no. 2, pp. 262-269. https://doi.org/10.1016/j.dss.2010.11.020

[2] Q. Liu, Z. Guo, and J. Wang, "A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization," Neural Networks, vol. 26, pp. 99-109, 2012. https://doi.org/10.1016/j.neunet.2011.09.001

[3] J. L. Ribeiro Filho, P. C. Treleaven, and C. Alippi, "Genetic-algorithm programming environments," Computer, vol. 27, no. 6, pp. 28-43, 1994. https://doi.org/10.1109/2.294850

[4] C. D. Lin, C. M. Anderson-Cook, M. S. Hamada, L. M. Moore, and R. R. Sitter, "Using genetic algorithms to design experiments: a review," Quality and Reliability Engineering International, vol. 31, no. 2, pp. 155-167, 2015. https://doi.org/10.1002/qre.1591

[5] W. B. Fan, F. Xiao, X. B. Chen, L. Cui, and S. Yu, "Efficient virtual network embedding of cloud-based data center networks into optical networks," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 11, pp. 2793-2808, 2021. https://doi.org/10.1109/TPDS.2021.3075296

[6] B. Rajendran, and S. Venkataraman, "Detection of malicious network traffic using enhanced neural network algorithm in big data," International Journal of Advanced Intelligence Paradigms, vol. 19, no. 3-4, pp. 370-379, 2021. https://doi.org/10.1504/ijaip.2021.116366

[7] A. Xiaokaiti, Y. Qian, and J. Wu, "Efficient data transmission for community detection algorithm based on node similarity in opportunistic social networks," Complexity, vol. 2021, pp. 1-18, 2021. https://doi.org/10.1155/2021/9928771

[8] R. Alsaqour, S. Kamal, M. Abdelhaq, Y. Zan, and D. Jerou, "Genetic algorithm routing protocol for mobile ad hoc network," CMC Computers Materials Continua, vol. 68, no. 1, pp. 941-960, 2021. https://doi.org/10.32604/cmc.2021.015921

[9] B. Bu, "Mult-task equilibrium scheduling of internet of things a rough set genetic algorithm," Computer Communications, vol. 184, pp. 42-55, 2022. https://doi.org/10.1016/j.comcom.2021.11.027

[10] R. Y. Wu, J. M. Ma, Z. X. Tang, X. H. Li, and K. K. R. Choo, "A generic secure transmission scheme based on random linear network coding," IEEE ACM Transactions on Networking, vol. 30, no. 2, pp. 855-866, 2021. https://doi.org/10.1109/TNET.2021.3124890

[11] W. C. Chang, and I. H. R. Jiang, "iClaire: A fast and general layout pattern classification algorithm with

clip shifting and centroid recreation," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 8, pp. 1662-1673, 2019. https://doi.org/10.1109/TCAD.2019.2917849

[12] T. Ganesan, and P. Rajarajeswari, "Efficient sensor node connectivity and target coverage using genetic algorithm with Daubechies 4 lifting wavelet transform," International Journal of Communication Networks and Distributed Systems, vol. 28, no. 3, pp. 337-364, 2022. https://doi.org/10.1504/ijcnds.2022.122170

[13] S. T. Shishavan, and F. S. Gharehchopogh, "An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks," Multimedia Tools and Applications, vol. 81, no. 18, pp. 25205-25231, 2022. https://doi.org/10.1007/s11042-022-12409-x

[14] B. Nahavandi, M. Homayounfar, A. Daneshvar, and S. Mohammad, "Hierarchical structure modelling in uncertain emergency location-routing problem using combined genetic algorithm and simulated annealing," International Journal of Computer Applications in Technology, vol. 68, no. 2, pp. 150-163, 2022. https://doi.org/10.1504/ijcat.2022.123466

[15] Z. Sabir, M. R. Ali, and R. Sadat, "Gudermannian neural networks using the optimization procedures of genetic algorithm and active set approach for the three-species food chain nonlinear model," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 7, pp. 8913-8922, 2023. https://doi.org/10.1007/s12652-021-03638-3

[16] C. Zhao, W. X. Zhu, G. Qiao, and F. Zhou, "Optimisation method with node selection and centroid algorithm in underwater received signal strength localization," IET Radar, Sonar Navigation, vol. 14, no. 11, pp. 1681-1689, 2020. https://doi.org/10.1049/iet-rsn.2020.0178

[17] Y. Zou, "Coupled neural networks and genetic algorithms application in the field of mine fire extinguishing," Informatica, vol. 48, no. 16, 2024. https://doi.org/10.31449/inf.v48i16.6317

[18] Y. M. Wu, Z. Li, C. X. Sun, Z. B. Wang, D. S. Wang, and Z. W. Yu, "Measurement and control of system resilience recovery by path planning based on improved genetic algorithm," Measurement and Control, vol. 54, no. 7-8, pp. 1157-1173, 2021. https://doi.org/10.1177/00202940211016094

[19] Y. Zhou, "Structural damage identification of large-span spatial grid structures based on genetic algorithm," Informatica, vol. 48, no. 17, 2024. https://doi.org/10.31449/inf.v48i17.6428

[20] M. Yu, and S. Chai, "Adaptive iterative learning control for discrete time nonlinear systems with multiple iteration varying high order internal models," International Journal of Robust and Nonlinear Control, vol. 31, no. 15, pp. 7390-7408, 2021. https://doi.org/10.1002/rnc.5690

[21] G. Singh, V. K. Tewari, R. R. Potdar, and S. Kumar, "Modeling and optimization using artificial neural network and genetic algorithm of self‐propelled machine reach envelope," Journal of Field Robotics, vol. 41, no. 7, pp. 2373-2383, 2024. https://doi.org/10.1002/rob.22255

Optimization of Emergency Material Logistics Supply Chain Path Based on Improved Ant Colony Algorithm

Mingbin Wei
College of Economics and Management, YanShan University, Qinhuangdao 066000, Hebei, China
E-mail：weimingbin@stumail.ysu.edu.cn

Keywords: emergency material, improved ant colony algorithm (IACA), logistics supply chain, travelling salesman problem

Received: October 29, 2024

Path selection is a critical challenge in emergency logistics management, particularly under realistic disaster-related conditions. This study addresses the problem of optimizing logistics transportation during major epidemics, considering constraints such as vehicle load, volume, and maximum travel distance per delivery. The goal is to minimize costs related to distribution trips, time, early/late penalties, and fixed vehicle expenses. By framing the problem as a generalized Traveling Salesman Problem, we developed an Improved Ant Colony Algorithm (IACA) to reduce the longest distribution path. Simulation data from Pudong, Shanghai lockdown zones revealed that IACA outperformed the traditional ACO algorithm, achieving a 30% cost reduction and higher accuracy (R^2 = 0.98). Additionally, experiments on gate assignment and TSP demonstrated the algorithm's superior optimization ability and stability. Overall, IACA enhances delivery route efficiency, lowers costs and energy consumption, and improves emergency logistics performance, proving to be a robust and reliable solution.

Povzetek: Avtor je razvil izboljšan algoritem kolonije mravelj (IACA), ki optimizira poti v logistični oskrbovalni verigi za nujne materiale.

1 Introduction

The logistics industry's use in many different industries is growing more and more common as the global economy develop. Researchers are becoming more aware of the crucial role emergency logistics (EL) plays in delivering supplies to disaster zones as a result of the exponential rise in emergency response operations and the rising frequency of disasters, both man-made and natural [1]. In particular, since 1980, catastrophes caused by nature have claimed the lives of over 2.4 million people globally, and their economic toll has grown by more than 800%, reaching $210 billion in year 2020 alone.137 million people in China were directly impacted by different disasters by natural in the same year, resulting in 19,956.7 hectares of crops being damaged, 370.15 billion yuan in direct economic losses, and 591 fatalities [2].

The National Disaster Reduction Commission and the China National Ministry of Emergency Management both state, China experienced 130 million individuals impacted by various disasters by natural in 2018, 588 fatalities, and 264.46 billion RMB in direct economic losses. Using earthquakes as an example, in 2018, earthquakes of magnitude 6.0 or higher killed 3068 people worldwide and wounded over 16,000 more. In order to reduce the number of casualties and property damage following a disruption, emergency materials must be delivered to the disaster zones promptly, precisely, and efficiently [3]. Emergency rescue is usually quite urgent because most disruptions are unpredictable and emergency material elements are very complicated.

In order to effectively provide goods while minimizing losses, emergency logistics was first established in 2004 to address the logistical challenges resulting from disasters. With its emphasis on facility placement and material delivery, it is essential to emergency decision-making. Supply channels are frequently disrupted by disasters, resulting in large losses. 15% to 20% of all disaster losses may be attributable to inefficient distribution [4]. The focus of humanitarian logistic modeling services, which account for 80–90% of rescue expenses, is on rapid reaction deployment, which is essential for successful rescue operations. The ideas under discussion centre on maximizing the distribution, transportation, and positioning of emergency supplies to rescue locations. For disaster response to be successful, emergency logistics efficiency must be increased [5].

For emergency rescue and relief efforts to be successful, the emergency supply chain must run smoothly. It is quite challenging to gather comprehensive information to support emergency operations during large-scale calamities since they are frequently unanticipated and exceedingly destructive. The victims' livelihoods and security may be negatively impacted by ineffective or even halted emergency material operations brought on by a shortage of supplies and knowledge [6]. Therefore, while reacting to large-scale emergencies, it is essential to solve the fundamental concerns of rapid acquisition and

integration of comprehensive emergency supply chain materials and information flow.

The planning, coordinating, and carrying out of logistics theory expert operations during emergencies, disasters, and crises are guided by the framework and set of principles known as emergency logistics theory. It includes a variety of ideas, tactics, and procedures meant to guarantee the smooth and efficient movement of information, products, services, and resources in order to meet the demands of impacted communities and lessen the effects of the crisis. The study uses a hybrid technique of simulated annealing and ant colony optimization to offer a low-carbon vehicle route optimization model for logistics and distribution. For increased efficiency, it uses an adaptive elite individual reproduction strategy and adds a multifactor operator and carbon emission factor [7]. Researchers are becoming more interested in emergency logistics. Nonetheless, the majority of recent studies address the location of facilities. This study focusses on the supply chain for emergency material logistics following a disaster. After a disaster strikes, it seeks to create the best plans possible for moving emergency supplies from one-to-many supply depots to disaster depots. Fig. 1 depicts the emergency management domain. In order to demands victims and finish rebuilding the disastrous event region after a disaster occurs, EM is a specific type of problem with vehicle routing that examines how to transport relief materials from depots for supply to depots for demand (disaster areas). EMS is typically separated into many-to-many scenarios based on the quantity of supply depots, as seen in Fig. 2

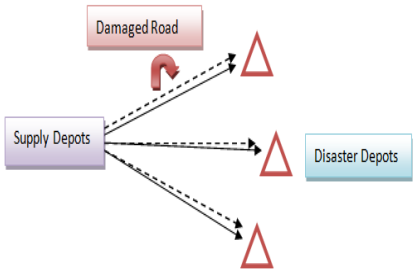


Figure 1: Supply depots to demand depots

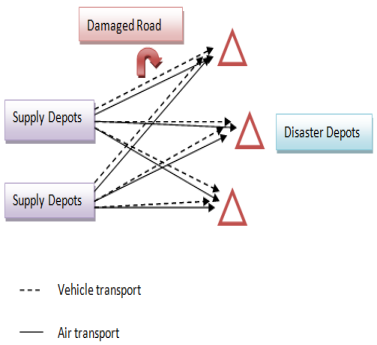


Figure 2: Quantity of supply depots in many to many scenario

The most important thing should be the timeline. Only if the emergency supplies are delivered to the Disaster Supply Depots accurately and on time will the damage be reduced. The fastest delivery time is therefore practically the most crucial factor in the improved ant colony model. We suggested an Improved ACA-based approach to solving the routing emergency material path problem. To get the best answer, the process determined the shortest path between nodes using a travelling salesman shortest path tree structure [8]. According to research findings, the suggested approach performed admirably in various disaster networks. In conclusion, even though ACO has been researched extensively and shown to perform effectively in organising routes, it is incredibly uncommon to utilise Ant Colony to optimise the supply chain route for emergency material logistics in disaster areas.

The study's primary contributions fall under the following categories.

- First, this study identifies and measures the variables that have been discovered to affect ACA's efficacy from both an internal and external standpoint. This includes the IACA's shortest route while taking into account the supply chain's external environment, funding for materials and equipment, complex emergency decision-making, and material transportation deployment. This is a definite step in filling the knowledge gap in the body of existing literature.
- Second, according to research methodologies, the majority of models in use today use traditional algorithms including precise, heuristic, and meta-heuristic algorithms.
- Third, in order to show the validity of the results, we contrasted the outcomes of the IACA method with those of the conventional ACA strategy. Furthermore, these findings will help emergency managers better pinpoint the sources and means of important elements, as well as the causal and hierarchical connections among them, and they will be able to contribute to the development of a robust and effective path.

This study is organized as the literature review for this topic is presented in Chapter 2. The research strategy and methodology are described in-depth in Chapter 3. The application of the proposed Improved-ACO Algorithm is shown in Section 4, followed by the application's outcomes and the discussion these finding, Section 5 summarises the findings, study shortcomings, and future research directions.

2 Literature review

These days, the transportation sector is growing quickly, and its main concentration is on the logistical distribution of perishable agricultural goods. The Emergency material logistics supply chain path finding path optimization method's application usefulness is recognised by experts. The supply chain for perishables has been examined by several academics.

For the uncertainty of unanticipated events in roadways in cities during shipping, a GA-based path optimisation model was introduced [9]. As part of the endeavour to reduce transportation costs, the logistics path optimisation model with a challenging window was developed to address the path dynamic. The outcomes of the experiments showed that this path optimisation model performed successfully. In order to address the sustainable food supply chain optimisation issue, this study presented a mixed integer linear programming model. In the purpose to reduce fuel costs, transportation expenses and carbon emissions were integrated, and Norwegian salmon exporters were used to conduct a suitability analysis [10].

For the prompt delivery of disaster relief materials after natural catastrophes, [11] and his team proposed an algorithm based on meta-heuristics. in hybrid Three optimisation improvement approaches were presented, the urgency coefficient of each demand point was evaluated, and Harris eagle optimisation and random PSO were integrated. The outcome of the study shows that the suggested approach had a maximum degree of computational correctness.

This Study developed a path optimisation technique based on enhanced GA to meet the cost and efficiency criteria of distributing fresh food. The procedure implemented a linear adaptive cross-variance technique and designated certain elements as penalty factors. The finding of the study shows that the approach could successfully reduce delivery path length and had a higher path optimisation efficiency [12].

A two-objective optimisation model was presented by this author for the adaptable perishable supply chain design commodities [13]. During the process, product and route disruption deterioration were thoroughly examined, utility role GA was added to optimise the method, and dynamic pricing was employed to handle crises. The experimental results demonstrated the good flexibility of the suggested approach. Using Ant Colony Optimization (ACO) or other technologies related to computers has been researched by several academics.

An ACO-based optimisation technique was put forth by researchers to address the issue of UAV scheduling routes. The procedure was solved for DSP and optimised for hierarchical “pheromone”-based processing. According to the testing results, the suggested approach has outstanding path planning speed and good path planning quality [14].

To solve the path routing problem of the fourth- party logistics, a method based on the ant colony system and the improved grey wolf algorithm was proposed. During the process, which included a carrying capacity and reputation constraint from the beginning node to the node of destination, known as the transit range, the ratio utility theory was used to determine the customer's risk appetite. The results of the study demonstrated that the recommended strategy could effectively finish path optimisation planning [15].

For the planning of the return path challenge of logistics networks in reverse, author suggested an ACO-based path technique. The procedure created a MINLP model, evaluated costs using a closed-loop, multi-stage logistics network, and tested using thirty instances. Results from experiments showed that the suggested approach could produce return pathways of excellent quality [16].

An ACO-based approach to solving the routing nodal path problem. To get the best answer, the process determined the shortest path between nodes using a rooted shortest path tree structure. According on research findings, the suggested approach performed effectively in networks of various sizes. In conclusion, despite extensive research and demonstrated effectiveness in route planning, ACO is currently rarely used to optimisethe route taken by cold chain logistics to distribute perishable agricultural goods. Given the pressing need for additional technical references to support the growth of the cold chain logistics distribution industry, the Improved ACO-based optimisation model was put forth, and the technical features of ACO were applied to optimise the cold chain logistics distribution path for perishable agricultural products [17].

Table 1: Summary on existing and suggested method compared with computational accuracy, time efficiency and cost reduction

Algorithm	**Computational Accuracy**	**Time efficiency**	**Cost reduction**
TGWO	TGWO algorithm as a decision support tool to boost supply chain performance, cut expenses, and improve cold chain logistics operations.	When compared to the TS and GWO algorithms, the TGWO method reduced the overall journey distance by 50.34% and 30.66%, respectively.	In terms of the overall cost of distribution, it saved 14.34 percent and 9.03 percent.

IPSO	The algorithm and emergency logistics vehicle route optimization model for severe epidemics that are suggested in this research work well.	If every demand's delivery priority is met, an enhanced vehicle routing optimization algorithm that takes delivery urgency into account can save a specific amount of time;	According to the sensitivity analysis, when the time cost is at its lowest, there should be three vehicles in the distribution center. The whole expense was lowered by 20.09%.
NN	The prediction findings demonstrate that the prediction can yield increased accuracy and better route matching.	Choose the route for material delivery that has the quickest speed and the lowest distance.	Reduce the percentage of transportation expenses as much as you can, and save money on supplies and automobiles.
IACA	the suggested model achieved the best solution accuracy with 98.5%	this study uses travelling salesman problem to find the shortest route and efficient in time travelling distance for emergency	cost reduction based on transportation, inventory, labor, minimize distance, avoid traffic and harzards by 30.2%

3 Methodology

Inspiration:
The process of optimising ant colonies is iterative. Several fictitious ants are considered at each cycle. With the restriction that ant not go to any vertices that already made it to during the ant walk, each of them constructs a solution by moving from vertices to vertices on the graph. An ant uses a stochastic mechanism biassed by the “pheromone” to choose the next vertex to visit at each stage of the solution building process. For instance, the next vertex in vertex I is selected at random from among those that haven't been visited yet. More specifically, k can be chosen with a probability proportional to the “pheromone” connected to edge (m, n) if it has never been visited before. The calibre of the answers the ants have created will depend on, the “pheromone” values are modified at the end of an iteration. By doing this, the ants are skewed to create solutions that are similar to the greatest ones they have created in previous cycles. The basic concept underlying the AC method is inspired by the way ant colonies behave when they are looking for food. Ants usually start by aimlessly looking around for food, bringing some of what they find back to their colony. They also leave a “pheromone” on the track they found. The worth of “pheromones” left in their wake, which gradually dissipates, depends on the quantity and calibre of the food supply. The “pheromone” that remain on a trail may persuade other ants to follow it. The strongest “pheromone” indicates a shorter path, which most ants can eventually follow.

3.1 Improved ant colony algorithm

This study uses the improved ant colony algorithm's high adaptability, multi-concurrency, resilience, and global search capabilities to handle the logistics supply chain's emergency material problem. Consequently, the enhanced ant colony method is presented to solve the model since it is highly parallel, offers the benefits of high fault tolerance and self-adaptation, and allows for heuristic improvement to enhance the algorithm's convergence. The ant colony method is a Traveling salesman algorithm that finds the shortest path by simulating ant populations' foraging behavior. Individuals of the ant colony will evaluate and choose the optimal foraging path based on the concentration of “pheromones” left by ants as they pass through nodes along the route. Rich customer and order data are challenging for traditional logistics to manage. Therefore, in the event of a natural disaster, the logistics automation path finder now incorporates an improved ACA. The drawbacks of conventional ACA are addressed with certain enhancements, which successfully resolve issues like resource scheduling and route planning in the logistical process. Figure 3 explained about the flow of emergency material path supplying method.

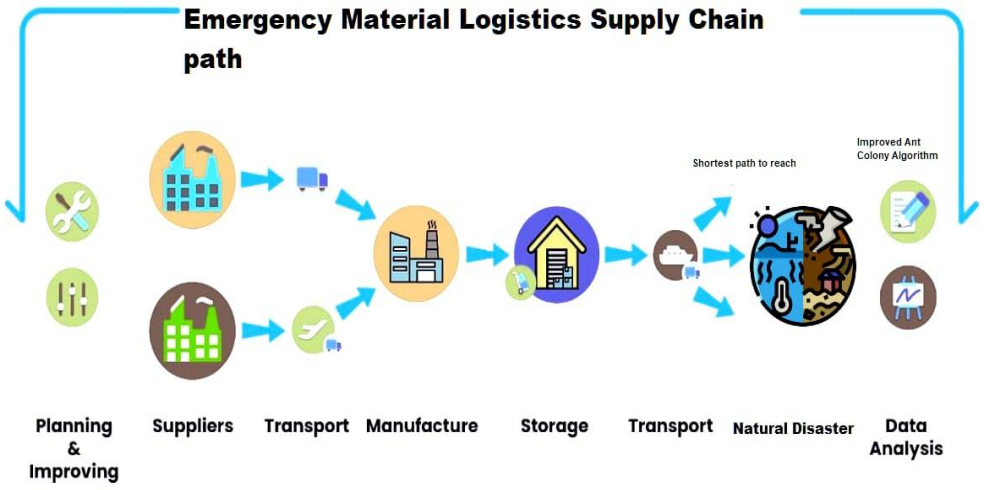


Figure 3: Block diagram of the method

3.2 Improved ant colony algorithm optimization strategy for emergency logistics material

Several production process connections are becoming more specialized due to the escalating market competitiveness. As a result of this tendency, logistics and commercial flow are now separated, progressively emphasising the significance of logistics. Conventional logistics models are inefficient and do not provide intelligent assistance. A lot of management and operations involve manual labor, which is ineffective and prone to mistakes. At the same time, businesses find it challenging to forecast and decide on logistical procedures when they lack intelligent help. Additionally, this hinders the rapid optimization and modification of logistical plans. The emergency logistics model has been developed based on this. The material logistics supply chain is a complete system that controls and optimizes the logistics process using a variety of automation technologies and tools. Logistics transportation costs are a significant part of the logistics chain, and they can be managed and controlled to increase the efficiency of logistical processes.
The cost of logistics transportation is shown in eq. (1).

$$C_1 = \sum_{k=1}^{k} s_k \sum_{i=1}^{m}.f_{\mathrm{i}} \qquad (1)$$

$s_k$is the variable in **eq. (1).** The vehicle number is v. The number of articles, including all transportation expenses, is denoted by m. $f$is the cost of driving eq.(2) displays the associated transportation cost.

$$C_2 = \sum_{j=1}^{n} \sum_{j=0}^{n} i \sum_{v=1}^{n} C_{ijk}{}^{x_{ijk}{}'} \qquad (2)$$

The set of transportation between places $i$and$j$ is represented by C_{ijk} in eq. (2). The variable between transportation points is called C_{ijk}. One constant is n. IACO puts ants at the first dispersion site in Fig. 4. After creating a tabu table, the cycle is initiated. The experimental random selection technique determines the next transfer node based on the chance of ants travelling to various nodes. The transfer node is added to the tabu table once it satisfies the constraint constraints. The ants' relevant journey length and delivery cost are determined once the transfer is completed constantly until the tabu-table is full. The worldwide ““pheromone”” is adjusted and the path optimisation is finished based on the computed results. After the maximum number of cycles has been reached, the outcome is the optimal path solution. In the real-world application, the starting point and associated fundamental path of dispersion parameters are input. The research method is then used to develop the path solution, and the best solution is chosen to finish the path generating process.

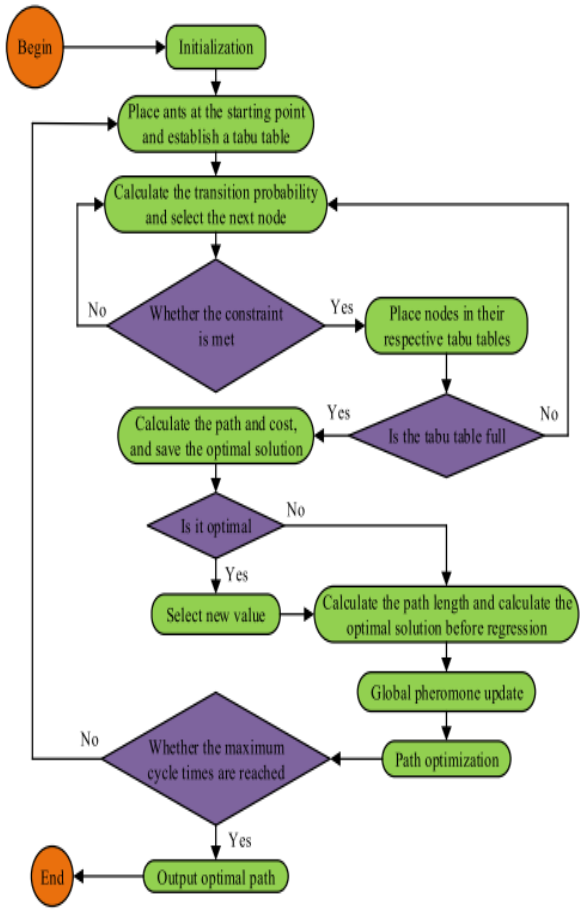


Figure 4: IACA flowchart

The logistics supply chain system consists of a server, front-end work, mechanical arm, three-dimensional warehouse management, AGV monitoring, PLC monitoring, sorting system, and commodities warehouse, among other components. The primary the sorting system's objective is to determine and categorise products. They are dispersed throughout various locations or forms of conveyance based on their type or destination. A commodities warehouse, which includes both automated and conventional warehouses, is a facility used for the storage of products. PLC and AGV monitoring are the two primary components of logistics system monitoring. Several components of the logistics system can be monitored and controlled using PLC monitoring, which functions as a programmable logic controller. Transporting items within or between warehouses requires AGV monitoring, which keeps track of the equipment's position and operational state. To guarantee precise storage and retrieval of items, the logistics system as a whole uses three-dimensional warehouse management, tracking, and computer-based management. Order processing, inventory queries, and other front-end tasks are the key tasks. The system uses a robotic arm to operate in order to accomplish a variety of activities, including grabbing, moving, assembling, etc. The management level of conventional autonomous logistics systems is rather low, despite the growing need for logistics technology. It is necessary to enhance the capacity to develop transportation networks, manage the supply chain, and optimise warehouse operations. An optimisation algorithm called ACA mimics how ants might forage for food in the wild. By mimicking the ant's ""pheromone"" transmission mechanism, it facilitates cooperation and information exchange throughout the optimisation process. Thus, adding ACA to the logistics control model can improve the management level of autonomous logistics control by successfully resolving issues like resource scheduling and path planning in the logistics process. Eq. (3) shows the node selection probability in ACA.

$$P_{ij}^{k}(\mathrm{t}) = \begin{cases} \frac{[\pi_{ij}(s)]^{\alpha}[\eta_{ij}(s)]^{\beta}}{\sum[[\pi_{is}(s)]^{\alpha}[\eta_{ij}(s)]^{\beta}]^{\alpha}}, & \mathrm{j} \in without\ passed \end{cases} \quad (3)$$

In eq. (3), τij is the data concentration. $ij\eta$ is the path visibility of path. $\alpha$is the trade-off factor in ""pheromone"". β is the "heuristic factor" for expected values

. The $ij\eta$ is shown in eq. (4)

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (4)$$

In eq. (4), d_{ij} is denotes the point i and point j distance. After complete ant colony cycle, its ""pheromones" will also be updated accordingly. The expression for "pheromone" update is shown in eq.(5)

$$\tau_{ij}(\mathrm{t+n})=(1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (5)$$

The "pheromone" evaporation coefficient is represented by eq. (5). The time point is t. The next node is typically chosen at random by conventional ACA, though. While random selection facilitates the exploration of broader problem areas, the early stage's convergence speed is slower due to the lengthy application of positive feedback. If the complete supply chain is not coordinated and optimised, the logistics chain as a whole may operate inefficiently, which may increase time costs. In order to achieve this, the study presents logistic chaotic mapping with the goal of leveraging its features to increase the precision of knowledge accumulation. Then, during the optimisation phase, some randomness is introduced into the basic ACA's routes.

In IACA, at the conclusion of every iteration, a logistics who possesses the best solution for that the optimal answer or iteration because the methods inception changes the disaster distanceas per eq.(6).Additionally, each instruction updates the amounts of "pheromones" on the final Communication, as illustrated in eq.(7), where $\tau i, j$ is the updated Schedule on the last feedback, denoted by $Ri, j, \tau i, j*$ is the updated location, τ0 is the initial warehouse, $\Delta\tau i, j$ is represented as $\frac{1}{L_{best}}$for the duration of the optimal route $Lbest$, and p and q are variables in (0,1) that correspond to the "pheromone" and its decay

coefficient feedback rate, respectively. Based on the likelihood pi, jk, which is computed as shown in Equation (7) where η, each ant k chooses its new path.$I, J$is the inverse of the route's length.The locations that have not yet been visited are shown by the letters i, j, and l, while The respective impacts of "pheromone"s concentrations and heuristic data are indicated by the control parameters α and β, respectively.

$\tau i, j *= 1 - r\tau i, j + r \cdot \Delta\tau i, j, if Ri, j$ (6)
is in the best path $\tau i, j$, otherwise
$\tau i, j *= 1 - q\tau i, j +$
$q\tau 0$ (7)

As the process runs, FC first generate a variety of randomly generated solutions. "pheromone" s are then updated based on the problem type and IAC Algorithm, with "pheromone" s placed on the graph's edges or vertices, to improve the solutions. The probability of the edge, which is computed as follows, determines whether or not to traverse the edge between two nodes, i and j in eq(8).

$$P^k ji = \frac{\pi(t)_{ij}^{a} n_{ij}^{\beta}}{\sum_{j \in N^{k(i)}} \pi(t)_{ij}^{a} n_{ij}^{\beta}} \text{isFeasible0,} \quad (8)$$

where, $P^k ji$ is chances for the understanding of instruction from nodes I and J, $\pi(t)_{ij}^{a}$denotes the value of the location range, n_{ij}^{β} is the materials, and $N^{k(i)}$is the establishment of nodes that can be visited by comments P^k. To get better results, it is advised to run a local search before upgrading the "pheromone" s. Nonetheless, the following approach is recommended for upgrading the eq. (9):

$\pi_{ij}(t+1) = (1-\rho) \cdot \pi_{ij}(t) + \Delta\pi_{ij}(t)$ (9)

Researchers and supply chain can learn more about the efficacy and customer communication as well as pinpoint areas for development by utilizing the IAC Algorithm to analyze emergency location from warehouse.

3.3 IACAmethod with traveling salesmen problem

Mathematicians and computer scientists in particular have focused a lot of attention on the travelling salesman problem (TSP) because it is both straightforward to explain and challenging to solve. Looking for the shortest restricted path to the target. A full directional graph $G = (N, A)$, can be used to represent the TSP, where A is a collection of arches and $D = d_{ij}$ is the price (a distance) vector for every arc $(i, j) \in A$.. Often referred to as cities, N is a collection of n nodes, or vertices. The cost of matrix D could be either symmetrical or asymmetrical. Finding the shortest closed tour that visits each of the $n = |N|$ nodes of G precisely once is known as the TSP. The distances between the cities in symmetric TSP are irrespective of the path in which they traverse arcs, therefore $d_{ij} = d_{ji}$for any pair of nodes. In the asymmetric TSP, for at least one pair of nodes (i, j), asymmetric TSP for at least one pair of nodes (i, j).

Define the variables in eq. 10:

$x_(i, j) =$
$(1 \; if \; the \; arc(i, j) \; is \; in \; the \; tour \; 0 \; otherwise$ (10)

The TSP can be formulated as a generalisation of a well-known integer program formulation.

The constraints are written as:

$Xn \; c = 1 \; xij = 1, d = 1, 2, 3, \cdots, n$ (11)

$Xn \; d = 1 \; xij = 1, c = 1, 2, 3, \cdots, n$ (12)

$xij \in \{0, 1\}, c, d = 1, 2, 3, \cdots$
$, n$ (13)

$Xn \; c, d \in S \; xij \leq |S| - 1, 2 \leq |S| \leq N -$ (14)

The overall cost that needs to be reduced is represented by the objective function (11) in these formulations. While constraint (13) ensures that each town has a single position allotted to it, restriction (12) ensures that each place is occupied by a single city j. The zero-one variables' integrality constraints xij ($xij \geq 0$) are represented by constraint as follows. Constraint (14) guarantees that no detours will be created and that every city on the final itinerary will be visited once.

Algorithm 1: Pseudocode for IACA

1. Build an environment model
2. Intialising thne count of ants $Pmax, M, S \; E, \alpha, \beta, a, b, c, R0, \pi_{ij}, \eta_{ij}$
3. For $P = 1$ to $Pmax$ do
4. Calculate β according to the equation (3);
5. Calculate ρaccording to the equation (8);
6. For $k = 1 \; to \; M$ do
7. Ant K into S;
8. While ant K does not reach E and Optional node>0 do
9. Determine the next emergency logistics from Equation 4,5,9;
10. While ant K in a deadlock do
11. Use deadlock handing route mechanism
12. Set the deadlock points as an obstacle point;
13. End while
14. End while
15. Save ant K taking path;
16. Calculate ant K length path;
17. End for
18. Calculate the shortest path for the iteration;
19. Divide the subpart by partitioning method;
20. Update pheromones for each subpart by Equation 10,11,12,13;
21. Set upper and lower pheromone limits by Equation 14;
22. End for
23. Output the optimal path;

4 Results and discussion

Ants make better initial decisions and spend less time pursuing fruitless avenues when they are given more insightful instruction. The enhanced algorithms minimize the number of iterations required by directing ants toward promising areas of the search space right away. *Experimental Setup:* The technique is programmed and solved in this work using MATLAB 2017a, and it was evaluated on an Intel notebook running 64-bit software with a CPU speed of 2.20 GHz and 4 GB of RAM.

4.1 Dataset

The emergency supply mechanism was promptly put into place to provide the basic necessities of life for inhabitants in Shanghai lockdown zones. In order to store and distribute supplies, ten ESWs were quickly established all the way through the city, utilising logistics supply, district emergency food enterprises' distribution centres, and other locations [18] Therefore, if the emergency material logistics is considered the point as central, the largest geographic area a disaster can potentially cover is approximately 633 km2, since the area of administration of Shanghai is 63401 km2. Caolu Modern Agricultural Park was chosen as the Emergency material supply, and the relevant data came from the lock-down zones created on April 16, 2022, in accordance with Shanghai's distinct preventative and control needs. Within the ESWFootnote1's coverage area were 50 lockout zones, including Magnolia Fragrance Garden Phase II, Sunshine Flower City, and FengchenLeyuan. The ESW is represented by the number 1 (shown in Fig 5(b) with a dot in green), and the 50 lockdown zones are represented by the numbers 2–51. Figure 6 displays the regional geographic information's distribution map. The ESW and part of the lockdown zones in Shanghai's Pudong New Area are shown by the red area in Figure 5. The overall distribution data is displayed in Figure 5(b). The red circle indicates the ESW zone, while the red and green dots indicate the lockdown and ESW zones, respectively.

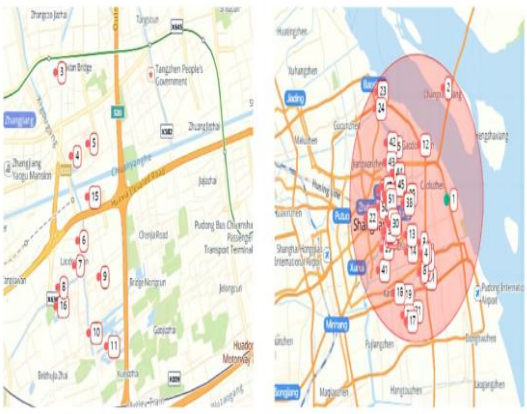


Figure 5: Lockdown zone for dataset

4.2 Experimental analysis

After improving ACA through performance analysis of the emergency logistics supply chain in Travelling Salesman, a logistics automation system is built. First, the suggested Improved Ant Colony algorithm is used to verify its performance. For system simulation, the experiment is carried out in MATLAB. Through table 2 establishment of appropriate parameters and limitations, the IACA algorithm's performance is monitored.

Table 2: the experimental model parameters

Parameters & units	Values
Fixed Cost (Yuan)	261
transport cost (Yuan)	4
Vehicle speed (Km/h)	60
Maximum Vehicle limit of mileage (Km)	14w
heavy load (Kg)	4250
pallet sizes(mm)	465*455

4.3 Data preprocessing

Min-max normalization

Min-max normalization is a technique for normalizing that involves linearly transforming the initial data to create an equilibrium of value comparisons before and after the process. This approach could use the following Equation.

$$Y_{new} = \frac{Y - min\ (Y)}{max\ (Y) - min\ (Y)}$$

Where,

Y-Old value,

Y_{new}- The new value obtained from the normalized outcome,

$min\ (Y)$–Minimum value in the collection,

$max\ (Y)$–Maximum value in the collection.

Outlier Detection and Removal

Outliers can degrade the efficiency of machine learning models by distorting statistical relationships among features. To eliminate outliers, we employ Z-score evaluation, which determines how far each data point deviates from the mean in standard deviations. A Z-score greater than 3 or less than -3 denotes an outlier that should be eliminated. The Z-score is determined as equation. below.

$$Z = \frac{X - \mu}{\sigma}$$

where X is the data point, μ is the mean of the attribute, and σ is the standard deviation.

Comparing Traditional and proposed method

ACO's initial total cost function value in **Figure 6** was above 3600, and after 132 iterations, it dropped to its

lowest value of 339. The initial value of IACO's total cost function was less than 3243, and after 19 iterations, it dropped to its lowest value of 3208. The study approach outperformed the conventional ACO in terms ofthe convergence speed as well as the initial value of the total cost function speed when comparing the convergence function images of the ACO and IACO models. A region with little Testing was done on the passenger flows when the traffic flow was nearly nothing in order to confirm the efficacy of the study approach in real-world application. Litchi was chosen for transportation because it is perishable and must be stored at a low temperature while being transported. Fuel trucks were used by transport vehicles, and for testing the application, 31 locations as targets were chosen.

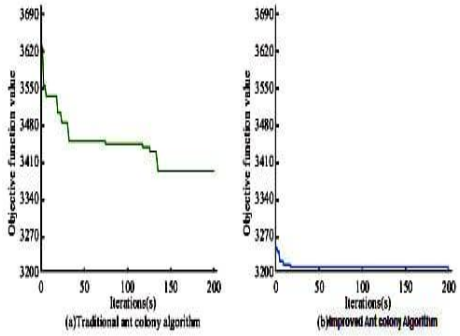


Figure 6: Total cost value of traditional and proposed

The prediction accuracy of four distinct logistics models is contrasted in Figure 7. The automated logistics control model developed for this study's prediction accuracy results are displayed in Figure 7(a). With an R2 of 0.98, the results demonstrated that the designed model had the highest prediction accuracy. This was 0.27, 0.30, and 0.17 higher than the prediction accuracy of the following algorithms: Tabu-Grey Wolf Optimisation (TGWO)[19], Improved Ant Colony Algorithm (IACA)[Proposed], Improved Particle Swarm Optimisation Algorithm (IPSO)[20], and neural network (NN)[21]. In conclusion, the model has a high prediction accuracy and can optimise logistics routes while lowering energy consumption in logistics distribution. Compared to other models, the fitting effect is superior. The suggested model is tested using real-world data to further confirm its scalability and dependability. The suggested model's accuracy and solution time are contrasted with those of the existing and proposed approaches.

A statistical metric called R-squared is used to assess how well a regression model fits data. R-squared values range from 0 to 1. When the model fits the data exactly and the anticipated and actual values are identical, we have an R-square of 1. However, when the model fails to learn any association between the dependent and independent variables and does not predict any variability, we obtain R-square equals 0. In conclusion, the model has a high prediction accuracy and can optimize logistics routes while lowering energy consumption in logistics distribution. Compared to other models, the fitting effect is superior. In order to confirm the suggested model's scalability and dependability.

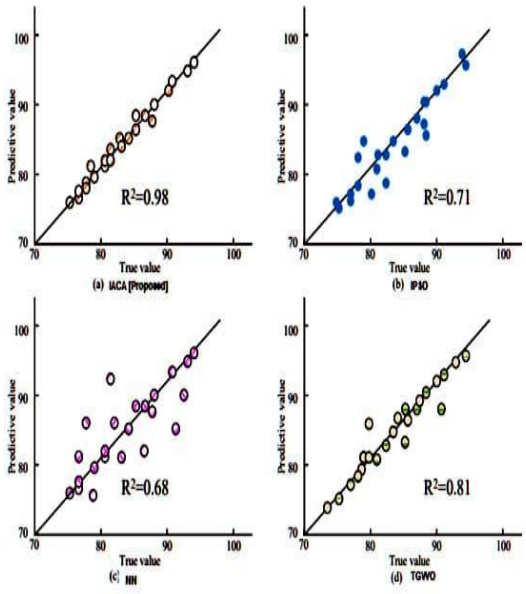


Figure 7: Comparison of model prediction accuracy

The accuracy and solution time of four distinct logistics models are contrasted in Fig 8. The accuracy comparison results are displayed in Fig 8(a). At 98.58%, the suggested model achieved the best solution accuracy. The contrast of solution times is displayed in Figure 8(b). Although it was greater than the other three models, the suggested model's solution time of 44.64 seconds was still within a reasonable range.

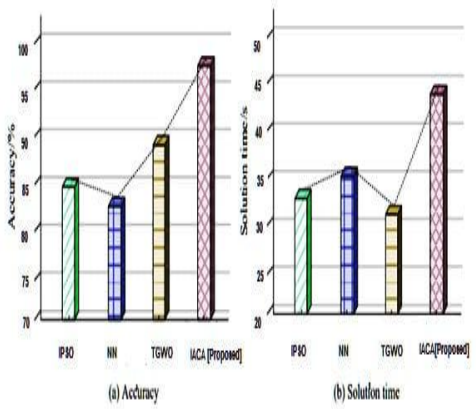


Figure 8: Accuracy and solution time of models

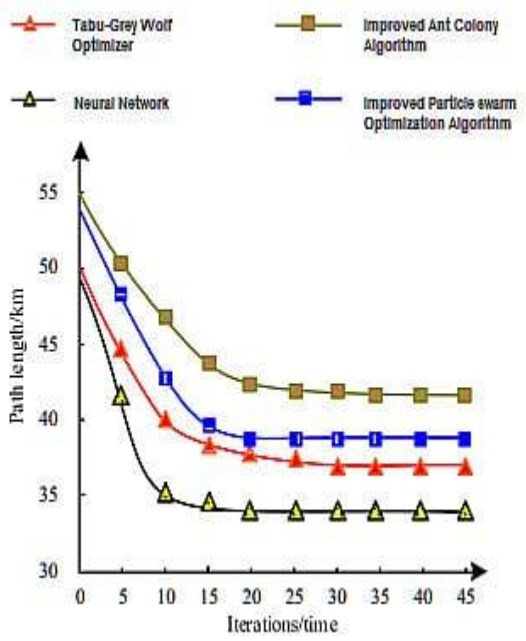


Figure 9: Finding Ideal path length finding

In Figure 9 it is clear that the improved ant colony algorithm requires less iterations to finding the ideal path than the current approaches. Additionally, the optimal path length is shorter, allowing for faster convergence.

Performance Analysis

For this paper, the method was run 55 times, and for the three algorithms in the literature, the average_time (t), inaccuracy (ξ) in %, and robustness (r) in % of every approach were noted. The following are the equations:

$$\xi = ave - best/best;$$

$$r = m/n;$$

whereaverage is the overall mileage in average, Highestis the ideal overall mileage, **n** is the number of tests, and **m** is the number of counts at which the optimal solution is found. The information for each one of the four methods findings is shown in Table 3 and show that the algorithm generated the highest overall mileage, average overall mileage, inaccuracy value, robust r, and algorithm time expenditure when compared to the other examined methods. These findings suggest that the method performs well in terms of computing complexity and robustness. This implies that the method employed in this study surpasses the single ant colony algorithm and yields an ideal result with minimum error, high precision, and consistent robustness.

Table 3: Robustness analysis of the four algorithms.

Techniques	Highest	Average	inaccuracy(ξ)	robustness(r)	t(s)
IPSO	383.52	384.57	2.70	2.10	12.10
TGWO	377.82	388.79	2.38	5.30	9.56
NN	374.38	377.16	2.64	10.09	6.69
IACA [Proposed]	371.17	374.03	0.74	20.20	1.44

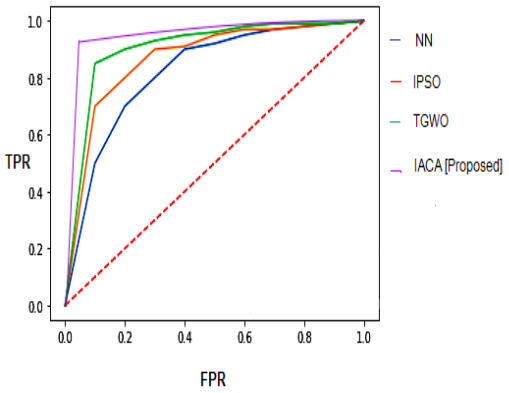


Figure 10: ROC Curve

Figure 10 shows a visual depiction of the model's performance over all thresholds is the ROC curve. The true positive rate (TPR) and false positive rate (FPR) are computed at each threshold (practically, at predetermined intervals), and the TPR is then graphed over the FPR to create the ROC curve. In the event that all other thresholds are disregarded, a perfect model, which at some threshold has a TPR of 1.0 and an FPR of 0.0, can be represented by either a point at (0, 1) The ROC is a helpful metric for evaluating the performance of distinct models, provided that the dataset is fairly balanced. In general, the better model is the one with a larger area under the curve. The suggested model [IACA] shows ROC curve has highest accuracy with 0.95. A confidence interval is a range of numbers that can be believe contains a population parameter. For any normal distribution approximately 95% of the values are within two standard deviations of the mean. From the following formula determine a 95% confidence interval:

$$95\%\ \text{confidenceinterval} = \bar{x} \pm 1.96\frac{s}{\sqrt{n}}$$

4.3 Discussion

IACA has been included to the emergency logistics supply chain path in order to increase logistics efficiency. The enhanced ACA served as the foundation for the building of the emergency logistics system model. The IACA algorithm demonstrated great optimization capabilities based on test results on the dataset. It was able to locate the ideal solution after 132 iterations, lowering the cost reduction value to be more than 30% below the average cost. Additionally, the suggested model's delivery distance was greater, its average power consumption per logistics node was lower, its emergency material for disasters was higher, and its prediction accuracy—which had an R2 of 0.98—was higher than that of the NN, TGWO, and IPSO. This suggests that the suggested approach has some practical application value and good optimization capabilities that can successfully increase delivery efficiency and lower fuel costs. While this study enhanced the program's performance, it also made the method more complex, which reduced its computing efficiency. Therefore, in order to maintain performance, it will be necessary to significantly lower the algorithm's complexity in the future.

Benefits of the proposed approach include: Lower operating costs due to the algorithm's optimization of routes and vehicle usage, which lowers labor, fuel, and maintenance expenses. When essential materials are delivered on time, penalties or reputational harm from delays are avoided; alternate routes are promptly found to avoid stopped highways; and time and expense are balanced to supply essential supplies effectively.

5 Conclusion

The logistics sector is changing dramatically as a result of its ongoing expansion. The IAC Algorithm was developed to reduce the amount of time it takes to distribute supplies from an Emergency Support Area (ESA) to Shanghai's 50 lockdown zones. Using data from Pudong, Shanghai, the system was evaluated and contrasted with three other intelligent optimisation techniques. The findings demonstrated that the algorithm's accuracy was enhanced by its local optimisation operation and that its relative error value was lower than that of other algorithms. In addition to being applicable to discrete optimisation issues, this method can serve as a general algorithmic framework for a number of scenarios, including rescue supplies, resource allocation during wildfires, emergency rescue during floods, and the transportation of hazardous items. However, the impact of road networks on transportation and supply distribution, lockdown zone configuration, and population density are not taken into account in this article.But existing logistics systems frequently lack the necessary flexibility and insight. In order to do this, the study modified the logistics supply chain system by integrating TSP into IACA. Simulation experiments were used to confirm the model's superiority. Future research can produce predictions with more accuracy and fewer iterations as the forecast results. which enhances scalability, adaptability, and real-time capabilities while integrating developing technologies with IOT. This achieves so by reflecting the precision and flexibility of the data in the optimization model.

Limitations of IACA:

• Enhanced features like dynamic pheromone updates, hybridization, and real-time data integration add computing overhead;

• Even with improvements, ACO may not be able to handle the exponential expansion in the number of paths as the network size increases.

• The effectiveness of the enhanced ant colony is highly dependent on sensitive parameters, including the number of vehicles, heuristic weighting, and pheromone evaporation rate.

• If dynamic data is imprecise or delayed, offer less-than-ideal routes.

There is a trouble striking a balance between time and money when shipping products to several high-priority sites.

References

[1] T. Kundu, J.-B. Sheu, and H.-T. Kuo, “Emergency logistics management—Review and propositions for future research,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 164, p. 102789, Aug. 2022, doi: https://doi.org/10.1016/j.tre.2022.102789

[2] Z. Li and X. Guo, “Quantitative evaluation of China’s disaster relief policies: A PMC index model approach,” *International Journal of Disaster Risk Reduction*, vol. 74, p. 102911, May 2022, doi: https://doi.org/10.1016/j.ijdrr.2022.102911

[3] Y. Zhang, Q. Ding, and J.-B. Liu, “Performance evaluation of emergency logistics capability for public health emergencies: perspective of COVID-19,” *International Journal of Logistics Research and Applications*, pp. 1–14, Apr. 2021, doi: https://doi.org/10.1080/13675567.2021.1914566

[4] F. Diehlmann, M. Lüttenberg, L. Verdonck, M. Wiens, A. Zienau, and F. Schultmann, “Public-private collaborations in emergency logistics: A framework based on logistical and game-theoretical concepts,” *Safety Science*, vol. 141, p. 105301, Sep. 2021, doi: https://doi.org/10.1016/j.ssci.2021.105301

[5] S. Jomthanachai, W.-P. Wong, K.-L. Soh, and C.-P. Lim, “A global trade supply chain vulnerability in COVID-19 pandemic: An assessment metric of risk and resilience-based efficiency of CoDEA method,” *Research in Transportation Economics*, vol. 93, p.

101166, Dec. 2021, doi: https://doi.org/10.1016/j.retrec.2021.101166

[6] "International Journal of Disaster Risk Reduction | Vol 74, May 2022 | ScienceDirect.com by Elsevier," *Sciencedirect.com*, 2022. Available: https://www.sciencedirect.com/journal/international-journal-of-disaster-risk-reduction/vol/74/suppl/C. [Accessed: Oct. 26, 2024]

[7] Y. Liu, J. Li, M. Liu, and B. Jiao, "An Enhanced Ant Colony Algorithm-Based Low-Carbon Distribution Control Method for Logistics Leveraging Internet of Things (IoT)," *Wireless Communications and Mobile Computing*, vol. 2023, pp. 1–12, Nov. 2023, doi: https://doi.org/10.1155/2023/5555221

[8] H. Jin, Q. He, M. He, S. Lu, F. Hu, and D. Hao, "Optimization for medical logistics robot based on model of traveling salesman problems and vehicle routing problems," *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, p. 172988142110225, May 2021, doi: https://doi.org/10.1177/17298814211022539

[9] M. Yang, "Yang, M. (2022). Research on Vehicle Automatic Driving Target Perception Technology Based on Improved MSRPN Algorithm. Journal of Computational and Cognitive Engineering, 1(3), 147–151. https://doi.org/10.47852/bonviewJCCE20514.

[10] A. De, M. Gorton, C. Hubbard, and P. Aditjandra, "Optimization model for sustainable food supply chains: An application to Norwegian salmon," *Transportation Research Part E: Logistics and Transportation Review*, vol. 161, p. 102723, May 2022, doi: https://doi.org/10.1016/j.tre.2022.102723

[11] T. Yan, F. Lu, S. Wang, L. Wang, and H. Bi, "A hybrid metaheuristic algorithm for the multi-objective location-routing problem in the early post-disaster stage," *Journal of Industrial and Management Optimization*, vol. 19, no. 6, pp. 4663–4691, Jan. 2023, doi: https://doi.org/10.3934/jimo.2022145

[12] A. Zhu and Y. Wen, "Green Logistics Location-Routing Optimization Solution Based on Improved GA A1gorithm considering Low-Carbon and Environmental Protection," *Journal of Mathematics*, vol. 2021, pp. 1–16, Nov. 2021, doi: https://doi.org/10.1155/2021/6101194

[13] M. Abbasian, Z. Sazvar, and M. Mohammadisiahroudi, "A hybrid optimization method to design a sustainable resilient supply chain in a perishable food industry," *Environmental Science and Pollution Research*, Aug. 2022, doi: https://doi.org/10.1007/s11356-022-22115-8

[14] Z.-H. Sun, X. Luo, E. Q. Wu, T.-Y. Zuo, Z.-R. Tang, and Z. Zhuang, "Monitoring Scheduling of Drones for Emission Control Areas: An Ant Colony-Based Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11699–11709, Aug. 2022, doi: https://doi.org/10.1109/tits.2021.3106305

[15] F. Lu, W. Feng, M. Gao, H. Bi, and S. Wang, "The Fourth-Party Logistics Routing Problem Using Ant Colony System-Improved Grey Wolf Optimization," vol. 2020, pp. 1–15, Oct. 2020, doi: https://doi.org/10.1155/2020/8831746

[16] M. Ashour, R. Elshaer, and G. Nawara, "Ant Colony Approach for Optimizing a Multi-stage Closed-Loop Supply Chain with a Fixed Transportation Charge," *Journal of Advanced Manufacturing Systems*, pp. 1–24, Nov. 2021, doi: https://doi.org/10.1142/s0219686722500159

[17]]M. Abdolhosseinzadeh and M. M. Alipour, "Design of experiment for tuning parameters of an ant colony optimization method for the constrained shortest Hamiltonian path problem in the grid networks," *Numerical Algebra, Control & Optimization*, vol. 11, no. 2, p. 321, 2021, doi: https://doi.org/10.3934/naco.2020028

[18] Chen, H. 2022. "Shanghai Starts 10 Emergency Supply Warehouses." https://contentstatic.cctvnews.cctv.com/snowbook/index.html?item_id=32234810742465611. Chen, C.-H., Y.-C. Lee, and A. Y. Chen. 2021.

[19] H. Zhang, J. Yan, and L. Wang, "Hybrid Tabu-Grey wolf optimizer algorithm for enhancing fresh cold-chain logistics distribution," *PLoS ONE*, vol. 19, no. 8, pp. e0306166–e0306166, Aug. 2024, doi: https://doi.org/10.1371/journal.pone.0306166

[20] K. Tan, W. Liu, F. Xu, and C. Li, "Optimization Model and Algorithm of Logistics Vehicle Routing Problem under Major Emergency," *Mathematics*, vol. 11, no. 5, pp. 1274–1274, Mar. 2023, doi: https://doi.org/10.3390/math11051274

[21] M. Chen, "RETRACTED ARTICLE: Optimal path planning and data simulation of emergency material distribution based on improved neural network algorithm," *Soft Computing*, vol. 27, no. 9, pp. 5995–6005, Apr. 2023, doi: https://doi.org/10.1007/s00500-023-08073-4

Application Method and Least Squares Support Vector Machine Analysis of a Heat Pipe Network Leakage Monitoring System Using an Inspection Robot

Xu Wang, Xiaobo Long, Guangwei Li, Jing Li, Yuweijia Zhao*
Kunming metallurgy college, Kunming, Yunnan, 650033, China
E-mail: kmyz_wang79@163.com
*Corresponding author

Keywords: heat pipe network inspection robot, human-computer interaction, heat pipe network side leakage monitoring, mobile platform, robot control

Received: August 24, 2024

With the maturity of the Internet and big data technAology, heat supply intelligence has become a development trend, and the traditional heat pipe network management mode is gradually transitioning to an "intelligent heat pipe network". It has become a hot spot for research and development at home and abroad. Combining big data technology, inspection robot control, heat pipe network leakage warning and data monitoring, scientific monitoring and evaluation of the energy-saving operation of heat pipe networks, and intelligent operation of heat pipes have become the current development trend. Whether in terms of the economic benefits of energy-saving operation of heat pipe networks or the social benefits of realizing intelligent operation and management of heat pipe networks, the study of a lateral leakage monitoring system for heat pipe networks is of great significance. This paper examines a technique for implementing a lateral leakage monitoring system for heat pipe networks using an inspection robot control system, which includes a real-time tracking module utilizing LSSVM (Least Squares Support Vector Machine) optimization to improve detection accuracy. The monitoring module can acquire, store, visualize, and send sensor data and video data; the user-defined interface module receives and parses XML user files from the server and generates user-defined interfaces and logic, thus realizing the human-computer interaction function. The experimental findings show that enhancing the weight factor and radial basis kernel function parameters of the LSSVM with the gravitational search technique resulted in an outstanding classification accuracy of 99.99% with a classification time of only 55.938 seconds, surpassing other optimization techniques.

Povzetek: Z uporabo robotskega nadzornega sistema in optimizirane metode LSSVM so avtorji razvili inteligentni sistem za spremljanje puščanja v toplotnih cevnih omrežjih

1 Introduction

Leakage in a thermal pipe network is a sudden change in liquid flow head or flow pressure caused by the flow rate of the medium stored outside the pipe exceeding a set value, which leads to a leak in the pipe [1]. Generally, we put in a sealed space after the occurrence of a leakage accident resulting in energy loss for the minimum, but this method cannot effectively monitor the actual environmental conditions over the failure caused by system damage to the maximum extent is not accurate and reliable, timely detection and treatment of the process caused by the consequences are very serious and likely to bring huge economic losses to the enterprise [2-3]. Therefore, a lot of research has been carried out at home and abroad on the diagnosis of leakage faults in heat pipe networks, and many leakage monitoring methods have been proposed. Although various methods have certain limitations and need to continue to be improved [4], the leakage monitoring of heat pipe networks and their compensators has important significance and can be referred to in the study of heat pipe network inspection robot monitoring systems [5].

Numerous studies have investigated different methods of inspecting and identifying leaks in pipeline networks, highlighting the significance of intelligent systems. Zholtayev et al. [6] created a smart pipe inspection robot with in-chassis motor actuation and AI-powered defect identification, showcasing sophisticated robotics incorporation in network tracking. Murtazin et al. [7] examined internal inspection techniques for district heating networks, highlighting the importance of resilient inspection techniques in energy systems. Wong and McCann [8] conducted an in-depth analysis of pipeline failure identification methods, ranging from acoustic sensing to cyber-physical systems, emphasizing the growing use of IoT solutions in fault detection. Liu et al. [9] presented an enhanced BP neural network algorithm for leakage detection in air conditioning water systems, demonstrating the efficacy of machine learning in detecting faults. Korlapati et al. [10] performed a thorough

review of pipeline leak identification approaches, ranging from conventional to AI-based methods. Similarly, Yussof and Ho [11] examined water leak detection techniques in smart buildings, emphasizing the significance of these technologies in contemporary infrastructure. Langroudi and Weidlich [12] investigated predictive maintenance assessment techniques for district heating pipes, which added to service-life prediction techniques. Van Dreven et al. [13] addressed smart fault detection in district heating, finding significant patterns and obstacles in the area. Hossain et al. [14] used UAV image evaluation and machine learning to identify leaks in district heating, demonstrating the value of aerial monitoring for infrastructure surveillance. Finally, Vollmer et al. [15] compared anomaly detection techniques in thermal imagery for district heating leak identification, which advances the use of thermal imaging in fault identification. Table 1 shows a summary table.

Table 1: Summary table

Citation	Title	Accuracy	Efficiency	Limitations	Innovations	Gaps in SOTA
[6] Zholtayev et al. (2024)	Smart Pipe Inspection Robot with AI-Powered Defect Discovery	95%	High (real-time)	Scalability, cost	AI-powered discovery, high precision	Flexibility for different pipe types
[7] Murtazin et al. (2021)	Internal Inspection of District Heating Networks	92%	Moderate	Constrained with magnetic testing	Non-destructive testing	Constrained with particular pipelines
[8] Wong & McCann (2021)	Pipeline Failure Discovery: Acoustic to Cyber-Physical Systems	70-85%	Low (real-time)	Inconsistent accuracy, high cost	Discovery taxonomy, spatial enhancement	High computational cost
[9] Liu et al. (2022)	Leakage Analysis for Air Conditioning Water Systems	86.96%	High	Fault location error	Two-stage diagnosis, BP neural network	Constrained real-time localization
[10] Korlapati et al. (2022)	Review of Pipeline Leak Discovery Techniques	87%	Varies	No standardization	Review of subsea techniques	Variability in reliability
[11] Yussof & Ho (2022)	Water Leak Discovery in Smart Buildings	81%	Varies	Real-time gaps in smart buildings	Incorporation with building automation	Absence of automated discovery
[12] Langroudi & Weidlich (2020)	Predictive Maintenance for District Heating Pipes	85-90%	High	Constrained with district heating	Proactive AI-driven maintenance	Narrow concentration
[13] van Dreven et al. (2023)	Fault Discovery in District Heating with ML	80-93%	Medium-High	Data Restrictions	ML methods for fault discovery	Absence of open-source data
[14] Hossain et al. (2020)	UAV-Based Leakage Discovery for District Heating	85%	Moderate	Constrained with UAV image examination	UAV with infrared discovery	Poor scalability for large systems
[15] Vollmer et al. (2021)	Anomaly Discovery in Water Networks with Self-Learning Algorithms	90%	High	False positives in intricate systems	Self-learning algorithms	Difficulties with dynamic settings

Existing state-of-the-art (SOTA) techniques have many shortcomings, like constrained flexibility to particular pipeline types, whereas the proposed system has wider applicability. Previous UAV and subsea detection techniques lack scalability, but this paper presents a scalable AI-based framework for large-scale networks. Real-time efficiency is hampered by high computational

expenses; the proposed system improves this with improved algorithms.

Based on the above, this paper focuses on the design of the heat pipe network leakage monitoring system, including hardware circuits and software programs, which involves the following key technologies: the selection of each component. According to the different types of components, the corresponding models are selected to analyze the working conditions under various parameters. The microcontroller control module, the sensor data acquisition part, and the peripheral devices such as the display alarm form the overall structure to complete the design scheme and design the thermal pipe network leakage monitoring system based on the inspection robot control system.

2 Leakage fault diagnosis in heat pipe networks

2.1 Leakage fault modelling

Leakage faults at key nodes of the heat pipe network are classified into three levels: normal, normal leakage, and severe leakage, and the leakage will lead to a sudden drop in pressure inside the pipe and changes in ambient temperature and conductivity [16-17]. Therefore, in this paper, the ambient temperature T1, T2, T3, and T4, the ambient conductivity G1 and G2, and the internal pressure p of the pipe are selected as inputs, and the leakage level of the critical node of the heat pipe network is taken as the output to establish the leakage level discrimination model. In this paper, the leakage level is expressed as {1, 2, 3} and will be used as the output of the leakage fault model, while the ambient temperature and conductivity around the heat pipe network and the internal operating pressure of the pipe network obtained online by the in-situ monitoring unit are used as the model inputs. The multi-classification leakage fault diagnosis at key nodes of the heat pipe network consists of four steps: sample collection, data pre-processing, building and optimizing the multi-classification leakage fault diagnosis model, and model testing. Specifically, x training samples and y test samples are arbitrarily selected; the extracted training and test samples are normalized; the multiclassification heat pipe network critical node leakage fault diagnosis model is established; the model parameters are optimized; and the experimental samples are substituted into the established multiclassification heat pipe network leakage fault diagnosis model for testing.

Since each characteristic indicator of temperature, conductivity, and pressure has different magnitudes and orders of magnitude, the role of a characteristic indicator of a particularly large order of magnitude in the classification may be highlighted in the calculation process. To eliminate differences in the units of the characteristic indicators and the effect of different orders of magnitude of the characteristic indicators, it is necessary to pre-process the data so that each indicator value is uniformly within a certain range of numerical characteristics. In this paper, 500 training samples and 90 test samples are randomly selected, and the independent variable of the leakage fault diagnosis model is denoted as x. The leakage fault level of the key nodes of the heat pipe network obtained after random sampling is denoted as the dependent variable y. The conductivity G1 and G2 are processed by logarithm, the temperature and pressure are normalized, and the sample data of the independent variable after data pre-processing are

$$\begin{cases} T_{norml1,2,3,4} = \frac{T_{1,2,3,4} - T_{min1,2,3,4}}{T_{max1,2,3,4} - T_{min1,2,3,4}} \\ G_{norml1,2} = \lg(G_{1,2}) \\ p_{norml} = \frac{p - p_{min}}{p_{max} - p_{min}} \end{cases} \quad (1)$$

Using equation (1) we can obtain the normalized data for the training samples as well as the test samples, and the dependent variable, the leakage fault level at the key nodes of the heat pipe network, is given by equation (2)

$$y_i \in \{1,2,3\} \quad (2)$$

The least squares support vector machine algorithm is then used to build a classifier to achieve a multiclassification fault diagnosis model for critical node leakage in the thermal network. In the thermal pipe network critical node leakage fault diagnosis model, we assume that the independent variable is x, and define the nonlinear thermal pipe network critical node leakage fault diagnosis model of least squares support vector machine as:

$$x = [T1, T1, T3, T4, G1, G2, P\} \quad (3)$$

$$y_i = \{\omega, (\varphi_x)\} + b \quad (4)$$

Given a set of data points that are closely related to the fault diagnosis of leakage at critical nodes of the thermal network, i.e. ambient temperature, ambient conductivity, and internal pipe pressure, d is the dimensionality of the model input variables and is the result of the model classification, i.e. normal (1), normal leakage (2) and severe leakage (3), l is the total number of known data points, and b is a constant. Therefore, the target equation and the nonlinear decision function used in the input space can be defined as:

$$\min \frac{1}{2}\omega^2 + \frac{1}{2}C\sum_{i=1}^{l} e_i^2 \quad (5)$$

$$y(x) = sgn\left(\sum_{s_i} y_i a_i K(x, x_i) + b\right) \quad (6)$$

2.2 Optimization of the parameters of the leakage fault diagnosis model

In this paper, the gravitational search method is used to optimize the parameters of the weight factor and the radial

basis kernel function of the least squares support vector machine. The gravitational search algorithm focuses on using the law of gravity between two objects to guide the optimal search for the optimal solution for the motion of each object. In this algorithm, each object is considered as an object whose performance is measured by its mass, and all these objects are attracted to each other through gravity, and this force causes all objects to move towards the object with the heavier mass [18]. The position of the object in motion corresponds to its optimal solution. The gravitational search method can be thought of as an isolated system of masses. Each object in motion follows the law of gravity and the law of motion. Assuming a system with N objects, define the position of the ith particle as

$$x_i = \left(x_i^1, \cdots, x_i^d, \cdots, x_i^n\right), \quad i = 1,2, \cdots, N \tag{7}$$

Then the interaction force and its parameters are given by

$$\mathrm{F_{ij}^d(t)} = \mathrm{G(t)}\frac{\mathrm{M_{pi}(t)\times M_{aj}(t)}}{\mathrm{R_{ij}(t)+\varepsilon}}\left(\mathrm{x_j^d(t) - x_i^d(t)}\right) \tag{8}$$

$$\mathrm{R_{ij}(t)} = \|\mathrm{x_i(t), x_j(t)}\|_2 \tag{9}$$

$$\mathrm{F_i^d(t)} = \sum_{\mathrm{j=1,j\neq i}}^{\mathrm{N}} \mathrm{rand_j F_{ij}^d(t)} \tag{10}$$

where rand is a random number generated between the intervals [0,1]. Therefore, according to Newton's laws of motion, the acceleration of particle i in d-dimensional space at time t and its calculation is given by

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{11}$$

$$\begin{cases} v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \\ x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \end{cases} \tag{12}$$

where randy is a uniform random variable in the interval [0,1], which we use to give a random signature to the gravitational search. The gravitational constant G is initialized at the start moment and decreases with time to control the search accuracy. The gravitational and inertial masses are simply calculated by fitness evaluation. A heavier mass means a more efficient object, which means that such an object has a higher gravitational force and slower velocity. Assuming that the gravitational mass is equal to the inertial mass, we update gravity and inertia using equation (13), of which equations (14-15) are the definitions of the parameters as well as maximizing the transformation.

$$\begin{cases} M_{ai} = M_{pi} = M_{ii} = M_i, i = 1,2, \cdots, N \\ m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \\ M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \end{cases} \tag{13}$$

$$\begin{cases} best(t) = \min_{j\in\{1,\ldots,N\}} ft_j(t) \\ worst(t) = \max_{j\in\{1,\ldots,N\}} ft_j(t) \end{cases} \tag{14}$$

$$\begin{cases} best(t) = \max_{j\in\{1,\ldots,N\}} ft_j(t) \\ worst(t) = \min_{j\in\{1,\ldots,N\}} ft_j(t) \end{cases} \tag{15}$$

According to the above principles of the gravitational search method and the LSSVM algorithm, it is clear that the idea of the gravitational search method to optimize the LSSVM learning parameters is to search for a set of vectors within a certain search space region by the gravitational search method, so that equation (16) the value of the target fitness function is minimized.

$$\mathrm{minf(C, \delta)} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{y_i} - \hat{\mathrm{y}})^2 \tag{16}$$

The basic principle of the gravitational search method is to optimally adjust the parameters of the least squares support vector machine weight factor and the radial basis kernel function by using the strong global search capability of the gravitational search method, and the optimization steps are as follows: first, randomly select and normalize the training and test samples; given the population size N and the maximum number of iterations, randomly initialize n particles; during each iteration, substitute The position of each particle is substituted into the least squares support vector machine model and the fitness value of the current particle is obtained; the sum of the forces in different directions of each particle and the acceleration of each particle are calculated according to Equation (10); the new particle position is calculated according to the update formula of the particle velocity and position; the algorithm termination condition is judged. If the maximum number of iterations is reached, the iteration is terminated and the optimal parameter values are output.

The parameter tuning method for the gravitational search technique (GSA) was carefully planned to improve the LSSVM's efficiency. During this procedure, key parameters were adjusted, including the gravitational constant, agent mass, and initial population size. Particular settings comprised $G0$=100, mass range of [1,10], and 50 agents. The GSA procedure is depicted in the flowchart below, starting with the initialization of agent positions and masses, then iterative updates using gravitational forces, and finally evaluating the LSSVM classification efficacy. A total of 590 samples were used for dataset selection, with 500 serving as training samples and 90 as testing samples. The training dataset had a balanced distribution across multiple fault classes, guaranteeing that each class was sufficiently represented to avoid bias. Each

training sample was created from real-world functional data and represents a variety of fault situations. To guarantee an unbiased assessment of the model's efficacy, the test samples were chosen at random from the same dataset while retaining the identical distribution features. This comprehensive description of the parameter tuning and dataset choice procedures not only improves replicability, but it also improves the validation of the provided outcomes, allowing other researchers to efficiently apply the approach.

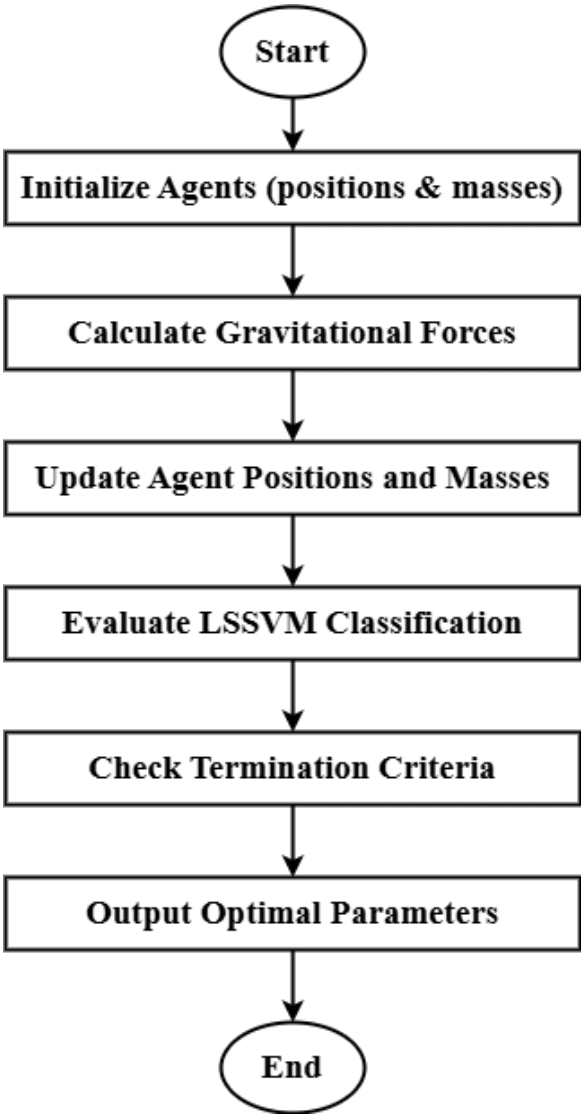


Figure 1: Flowchart of the GSA Process

2.3 Leakage fault diagnosis and result analysis

We introduce the mean square error MSE as an index to evaluate the correct classification rate, which is calculated as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \wedge y_i)^2 \quad (17)$$

In this paper, the parameters of the weight factor and radial basis kernel function of LS-SVM are optimized based on the particle swarm algorithm, cuckoo algorithm, and gravitational search method. This optimized multi-classification fault diagnosis model for leak monitoring at key nodes of the heat pipe network is used to test the test samples. The classification results are also compared. The 500 training samples were taken into the leak fault diagnosis model and the parameters of the LS-SVM weight factor and radial basis kernel function were first optimized using the gravitational search method to obtain the optimal values, which took 55.938 seconds to run and 99.99% of the test samples were correctly classified. Figure 2 illustrates a comparison of two elements of the gravitational search technique. The top section displays a parametric merit search plot, demonstrating how the technique assesses various parameters to identify the best solution. The bottom section shows the classification findings for the test set, demonstrating the method's ability to correctly categorize data using the optimum parameters discovered during the merit search. In general, this figure depicts the relationship between the parameter optimization procedure and its effect on classification efficiency.

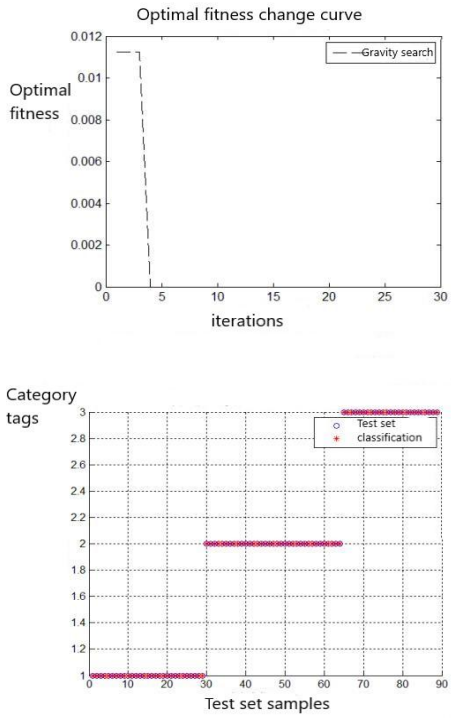


Figure 2: Comparison of the parametric merit search graph of the gravitational search method (top) and the test set classification of the merit search (bottom)

Optimization of the weight factor and radial basis kernel function parameters of the LS-SVM using the cuckoo algorithm resulted in an optimal value of 28.7282 and an optimum value of 15.8259, and the time taken to run was 60.491 seconds, giving a 97.89% correct classification rate

for the test sample. In this paper, we optimize the weight factor and radial basis kernel function parameters of the LS-SVM based on the gravitational search method, the cuckoo algorithm, and the particle swarm algorithm, and use a randomly selected set of 90 test samples to check the correct classification rate. The findings show that the multi-classification fault diagnosis model, which uses a least squares support vector machine algorithm enhanced by the gravitational search technique, attains a classification accuracy of 99.99% in only 55.938 seconds. The results show that the multi-classification fault diagnosis model based on the least squares support vector machine algorithm optimized by the gravitational search method has the best classification effect and the lowest algorithm complexity.

Table 2 shows a confusion matrix comparing the Gravitational Search Algorithm (GSA) and the Cuckoo Algorithm. The GSA had 450 true positives (TP) versus 425 for the Cuckoo Algorithm, showing superior efficiency in finding positive cases. The GSA also recorded 40 true negatives (TN), which exceeded the Cuckoo Algorithm's 35. Particularly, the GSA had only two false positives (FP), whereas the Cuckoo Algorithm had ten, indicating higher precision. Furthermore, the GSA had three false negatives (FN) compared to the Cuckoo's fifteen, demonstrating its detection efficiency. Overall, the GSA outperformed the Cuckoo Algorithm.

Table 2: Confusion matrix

Metric	Gravitational Search Method (GSA)	Cuckoo Algorithm
True Positives (TP)	450	425
True Negatives (TN)	40	35
False Positives (FP)	2	10
False Negatives (FN)	3	15

Table 3 presents efficiency metrics for the GSA and the Cuckoo Algorithm. The GSA attained an impressive 99.99% accuracy, substantially higher than the Cuckoo Algorithm's 95.75%. The GSA had a precision of 99.95%, compared to 93.00% for the Cuckoo Algorithm, suggesting that it was more reliable at predicting the positive class. The GSA's recall was 99.90%, demonstrating its ability to detect pertinent instances, whereas the Cuckoo Algorithm had a recall of 94.50%. The F1 Score for the GSA was 99.92%, while the Cuckoo Algorithm's was 93.75%, demonstrating the GSA's total superiority. Finally, the ROC AUC Score for the GSA was 0.999, indicating outstanding discriminative capacity, as opposed to 0.950 for the Cuckoo Algorithm. These metrics demonstrate the GSA's improved classification efficiency.

Table 3: Performance metrics

Metric	Gravitational Search Method (GSA)	Cuckoo Algorithm
Accuracy	99.99%	95.75%
Precision	99.95%	93.00%
Recall	99.90%	94.50%
F1 Score	99.92%	93.75%
ROC AUC Score	0.999	0.950

2.4 Heat pipe network inspection robot control system

The thermal pipe network inspection robot system is divided into four modules: server, master controller (mobile side), slave controller (motion controller), and pipe robot mechanical system. The principle of operation and control of the thermal pipe network inspection robot system is based on the motion controller as the core [19]. When the user's control logic is burned into the motion controller and parsed, the motion controller sends control commands to the actuators to realize the motion control of the thermal pipe network inspection robot. The motion controller collects and processes the robot motion data from the photoelectric encoder and the mobile terminal in real-time during the motion of the heat pipe network inspection robot, and according to the results of the data processing, the motion controller adjusts the motion state of the pipe robot in real time to achieve closed-loop control of the pipe robot motion. In addition, the mobile side is equipped with a self-developed thermal network monitoring system, which sends the sensor data and video data to a remote server via network communication (TCP/IP communication protocol), enabling remote monitoring of the pipeline robot. The specific functions of the four components are as follows.

Firstly, the server. The server side of this paper is equipped with a self-developed remote thermal pipe monitoring system client, whose function is to receive the sensor data and video data from the mobile side and to visualize them, while the server can send control commands to the motion controller via the mobile side to adjust the motion of the pipe robot. Secondly, the mobile side. The mobile side is fitted with an on-site thermal pipe monitoring system client, which is capable of acquiring sensor data and video data in real time using the mobile side's hardware-integrated sensor set and HD camera. The system processes the data in two ways: one is the local

visualization and storage processing of the data; the other is the sending of the data to the server side and the motion controller. Thirdly, the multi-core heterogeneous motion controller. It is used to implement the user's control logic, which is the core of the motion control of the heat pipe network inspection robot. The main hardware modules of the motion controller are a master MCU, two slave MCUs, a motor driver chip, a voltage converter chip, and a sensor set. Fourth, the mechanical structure of the thermal pipe network inspection robot: it is the bearer of the mobile end, the power module of the motion controller, etc., and is also the final executor of the motion controller's operating instructions.

In addition, the mechanical structure of the thermal pipe network inspection robot in this paper is mainly divided into the chassis, walking mechanism, drive module, and the articulation mechanism between the chassis, etc. The chassis is the main part of the mechanical system of the thermal pipe network inspection robot, which carries the motion controller, drive system, and mobile end of the pipe robot. The travel mechanism and drive module are the key factors to ensure that the thermal pipe network inspection robot walks normally in the pipeline, and are the focus of the pipeline robot mechanism design.

The PC-Microcontroller control system was chosen because the thermal network inspection robot in this paper needs to process information such as video information and scanner data as well as execute control commands. Based on the functional requirements of the system and the tasks to be completed by the robot, the robot system is composed of a power supply system, a sensor system, an upper computer system, a lower computer system, a motion control unit, a bus communication system, a video system, and a laser scanner system. The power supply system is responsible for supplying power to all parts of the robot. The robot in this project requires power from the motor driver (24V); the sensor unit (12V, 5V); the main controller (5V) of the proposed ATmega series of microcontrollers; and the motion control system (5V). The robot system is controlled by the lower computer, the upper computer sends commands to the lower computer through the bus communication system, and the lower computer controls the normal operation of the robot according to the received commands.

3 Experimental procedures for testing the control system of the thermal pipe network inspection robot

Preparation: Assemble the robot's chassis, locomotion mechanism, drive module, and articulation system, making sure that the master MCU, two slave MCUs, motor driver chip, and sensor array are properly linked.

Microcontroller configuration: Setup the ATmega microcontroller with a 16 MHz clock frequency, 490 Hz PWM frequency, 10-bit ADC resolution, and configure the UART communication rate to 9600 bauds.

Sensor calibration: To guarantee precise readings, calibrate the incorporated sensors by adjusting the temperature sensors to a reference temperature of 25°C. The HD camera should be setup to capture video at 1080p resolution and 30 frames per second, while the laser scanner is set to a highest detection range of 5 meters.

Control logic execution: Program control commands using user-defined logic into the motion controller, allowing the robot to perform particular movement trends within the pipeline.

Test environment setup: To simulate real-world circumstances, build a scaled-down model of a 10-meter-long thermal pipe network with differing diameters (50 mm and 100 mm).

Conducting trials: Perform at least five trials to evaluate the robot's efficiency, recording control commands, sensor readings, and motion execution times, with a target execution time of less than 120 seconds.

Data gathering and examination: Gather and evaluate sensor and camera data to compare actual detection findings to expected results, with a target detection accuracy of 90%. Record any deviations in effectiveness.

Expected vs. actual outcomes

Detection Accuracy: The detection accuracy is set at 90% for detecting known defects in the thermal pipe network.

Execution time: The anticipated execution time should not surpass 120 seconds, and actual times will be logged for comparison.

Power requirements: The power supply system should supply 24V to the motor driver, 12V and 5V to the sensor unit, and 5V to the microcontroller and motion control system.

The overall structure of the robot control is shown in Figure 3.

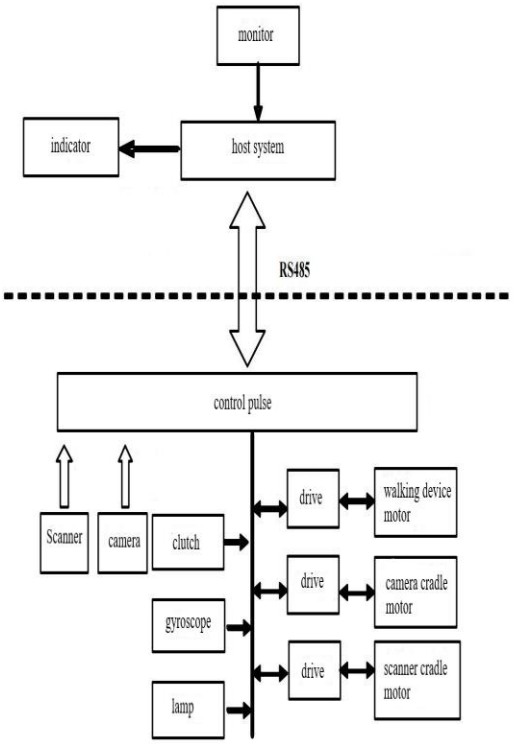


Figure 3: Control system structure block diagram

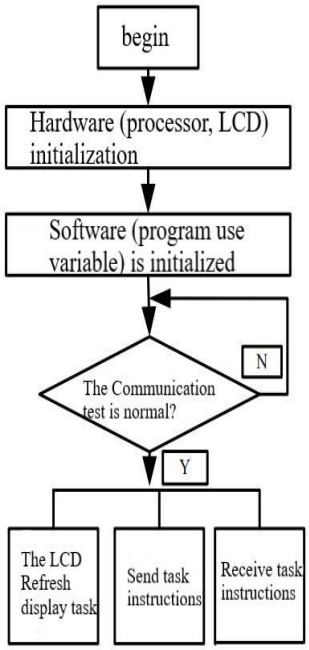


Figure 4: Upper computer development process

Motor drive system design:

In this paper, the heat pipe network inspection robot has multiple motors: 2 main motors for walking, 1 camera rotation motor, 1 camera tilt servo, and 1 camera tilt servo. There are two main motors for walking, one camera rotation motor, one camera tilt servo, one scanner rotation motor, and one head lift motor. Since DC motors are used, this paper only focuses on the drive system of the main travel motors. The motion control of the motors is carried out by a central processor ATmega8 in the robot control module, which outputs PWM pulse width modulated signals to the motor drivers in a software way to carry out the forward, reverse, and stop actions of the motors.

Design of the upper computer control system:

The upper control system is mainly responsible for communication with the lower computer of the pipeline robot, through the control knob on the upper control panel to send various commands to the robot, such as forward, backward, left turn, right turn, stop and other action commands; camera control knob can realize the adjustment of the camera head, such as camera rotation, tilt, and other actions; control panel also has the adjustment knob of the scanner head, can adjust The control panel also has a knob for the scanner head, which allows the scanner to be rotated, scanned and reset, as well as the adjustment of the brightness of the LEDs. All commands are sent to the robot via the host control system and are controlled by the robot. The upper computer development process is shown in Figure 4.

Some of the program code for the upper computer is as follows.

```
////1. Capture user operation commands
temp_char=PINB; //Read port B
if(temp_char&BIT(0)) //Rocker "run" trigger
{
flag_run=1; //indicates that the robot has entered the forward state
direction_left=1; //the robot is moving in the left wheel direction
direction_right=1; //the robot goes in the direction of the right wheel
velocity_left=velocity_robot; // the robot's left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
else if(temp_char&BIT(1)) // rocker "back" trigger
{
flag_run=0; //meaning the robot is out of the forward state
direction_left=2; //the robot goes back in the left wheel direction
direction_right=2; //the robot goes back on the right wheel
velocity_left=velocity_robot; //the robot's left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
else if(temp_char&BIT(2)) // rocker "left" trigger
{
if(flag_run==0) //rotate in place
{
direction_left=2; //Robot goes backwards in the left wheel direction
```

```
direction_right=1; //The robot goes forward on the right wheel
velocity_left=100; //Robot left wheel speed
velocity_right=100; //Robot right wheel speed
}
else
{
direction_left=1; // robot left wheel direction forward
direction_right=1; // robot right wheel direction
velocity_left=velocity_robot; // robot left wheel speed
velocity_right=velocity_robot+100; //robot right wheel speed
}
}
else if(temp_char&BIT(3)) // rocker "right" trigger
{
if(flag_run==0) //rotate in place
{
direction_left=1; //Robot left wheel direction forward
direction_right=2; //Robot goes backwards on the right wheel
velocity_left=100; //Robot left wheel speed
velocity_right=100; //Robot right wheel speed
}
else
{
direction_left=1; // robot left wheel direction forward
direction_right=1; // robot right wheel direction
velocity_left=velocity_robot+100; // robot left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
}
else if(flag_run==1) //robot not triggered, the robot in a forward state
{
direction_left=1; // robot left wheel direction forward
direction_right=1; //Robot is moving in the right wheel direction
velocity_left=velocity_robot; //Robot left wheel speed
velocity_right=velocity_robot; //robot right wheel speed
}
Else if(flag_run==0) //robot is not triggered, the robot is stationary.
{
direction_left=3; //Robot left wheel stopped
direction_right=3; //robot right wheel stop
velocity_left=0; //Robot left wheel speed
velocity_right=0; // robot right wheel speed
}
if(temp_char&BIT(4)) // robot auto travel state
{
direction_left=8; //Robot left wheel auto
direction_right=8; //Robot right wheel auto
}
.....
```

The main functions of the lower unit of the pipeline robot are to receive commands from the upper unit to control the motors; to collect data from sensors such as tilt angle and return the information to the upper unit; to provide power to the robot motors, lights and cameras, and to control the normal operation of the robot components. The code for the part of the program of the lower computer is as follows.

```
void main(void)
{
init_devices();
while(1) ///// cycle ms
{
value_adc[2]=value_adc[0]; //backup last AD acquisition data
value_adc[3]=value_adc[1]; //backup last AD acquisition data
flag_adc=0;
adc_start(0); //Start AD acquisition
Delayms(1);
flag_adc=1;
adc_start(1); //Start AD acquisition
Delayms(1);
if(flag_auto==1)
{// calculate and output the speed of the motor
velocity_left=0;//the speed of the motor (P+D)
velocity_right=0;//the speed of the motor (P+D)
if(velocity_left>10)
{
DIRLEFT_H;//positive rotation
STOPLEFT_L;//positive rotation
pwm_left=velocity_left;
}
else if(velocity_left<-10)
{
DIRLEFT_L;//reverse
STOPLEFT_L;//reverse
pwm_left=-velocity_left;
}
else
{
STOPLEFT_H;//brake
pwm_left=0XFF;
}
if(velocity_right>10)
{
DIRRIGHT_H;//positive rotation
STOPRIGHT_L;//forward
pwm_right=velocity_right;
}
else if (velocity_right<-10)
{
DIRRIGHT_L;//reverse
STOPRIGHT_L;//reverse
pwm_right=-velocity_right;
}
else
```

```
{
STOPRIGHT_H;//brake
pwm_right=0XFF;
}
}//////////////////////////////////////////////////////////////////////
.....
```

3.2 Heat pipe network monitoring system under the control of inspection robot

3.2.1 Analysis of the functional requirements and overall architecture of the heat pipe network monitoring system

According to the aforementioned content pair can be known, this heat pipe network monitoring system needs to achieve the following functions.

(1) This heat pipe network monitoring system must be able to acquire data measured by the mobile's integrated sensors and HD cameras in real time and follow TCP/IP communication protocols and OTG communication protocols to send the collected data and corresponding operation instructions to the server and motion controller.

(2) The thermal network monitoring system should have the capability to visualize sensor data, i.e. to enable the user to observe specific sensor data, the system must be able to display the data as dynamic text; to visualize the trend of the data, the system must enable the data to be displayed as dynamic curves.

(3) To meet the access to historical sensor data and video data, and to facilitate the user to further confirm the operation status of the heat pipe network inspection robot and the internal environment of the pipe, the heat pipe network monitoring system must be able to store the acquired data in the database and have the function of querying and deleting historical data.

(4) The main difference between this thermal network monitoring system and other thermal network monitoring systems is the ability to achieve human-machine interaction, i.e. by parsing the XML file sent by the server and dynamically generating user-defined interfaces and background logic, the thermal network inspection robot can be controlled to achieve the functions set by the user.

(5) To ensure the security of user information, the thermal network monitoring system needs to have a user login interface so that only the user can use the thermal network monitoring system after entering the corresponding user name and password.

Based on the analysis of the functional requirements of the monitoring software, the design of its overall architecture was completed. The main functions of the system are: to obtain sensor data and video data in real-time and store the data in an intermediate database; to implement some basic functions such as listening to events (sensor listening events, SMS listening events, etc.) and sending and receiving broadcasts; to have a system exception and user code exception handling mechanism, when the user code when an exception occurs in the user's code, the corresponding dialog box will pop up, and according to the information in the dialog box, the user can see the location of the error and the cause of the error, to facilitate the user's modification of the code and avoid the phenomenon of crashing or flashing due to errors in the operation of the heat pipe network monitoring system, which ensures the normal operation of the heat pipe network monitoring system.

The system reads data from the intermediate database at regular intervals to visualize (dynamic text display and dynamic curve display), store, and send data. In the architecture of the heat pipe network monitoring system, the intermediate database, basic functions, exception handling, and library functions form the underlying code of the heat pipe network monitoring system, and the user can call the function functions in the library functions to achieve the corresponding logical functions, which reduces the difficulty of the user's development.

3.3 Interface development

Based on the analysis of the functions and architecture of the monitoring software, the interface structure of the monitoring software is divided into three modules: the login module, the monitoring module, and the user UI module. The login module verifies the user's information, and only when the user writes accurate information can the monitoring software be opened; when the information entered does not pass the background verification, the software will prompt the user to enter again or register the account until the login is successful.

The monitoring module is divided into six interfaces: dynamic text display of data, dynamic curve display of data and video display, sensor selection interface, and network connection interface. In the dynamic text display of sensor data and dynamic curve display interface, the data is refreshed once per second; the video monitoring interface can preview the video data collected by the mobile terminal in real-time; to reduce the amount of data and allow the user to select the required sensor data according to the specific project needs, the monitoring software is designed with this in mind and the sensor selection interface is designed; to achieve the communication function with the server, the network connection interface is designed. To communicate with the server, a network connection interface has been designed, where the user only needs to enter the corresponding IP and port number to connect to the server and transfer the data; to meet the user's needs and enable the user to query the historical data, a data query interface has been designed.

The user UI module, which is used to display the interface dynamically generated by parsing the XML file sent by the server, enables human-computer interaction. The interface design for this heat network monitoring system uses Activity and Fragment components. Since Fragment takes up less memory than Activity, the interface design in this

paper uses more lightweight Fragment to improve the running efficiency of the application. The interface of the heat pipe network monitoring system takes the main interface as the core and dynamically loads the monitoring interface and the user interface, etc. The data is passed between the interfaces by binding objects of the Bundle class, as shown in Figure 4.

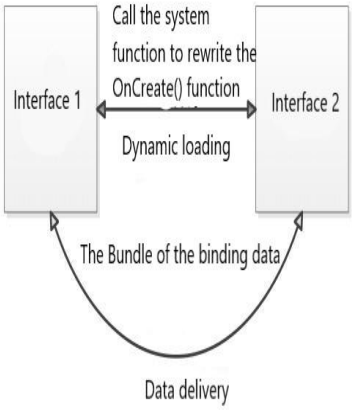


Figure 4: Interface interaction process of the heat pipe network monitoring system

The monitoring system can selectively display sensor data, i.e. the user can select the required sensor in the sensor type selection interface, and in the sensor selection interface, the system will jump to the data display interface to visualize the data (dynamic text display and dynamic curve display) according to the selection result. The process is as follows: first, the system converts the selected sensor type into a string and separates the different sensor names with a special symbol "%". Then the string is bound to a Bundle object and the set Arguments() function is called to add the Bundle object with the string data to the data display Fragment to be switched; the system function for switching between Fragments is called to switch between the sensor selection interface and the data display interface; finally the function get Arguments() is called in the Fragment responsible for the data display. Function get Arguments () to get the Bundle object, then call the Bundle. get String() function to get the String data passed from the sensor selection interface, then call the string processing function with the special symbol "%" as the Each member of the array is the name of the sensor selected by the user, and the system iterates through the array to obtain and display the real-time data of the sensor selected by the user.

3.4 Interface design

Main interface design

The main interface of the thermal network monitoring system in this paper adopts a "segmented" structure, i.e. the top of the interface is the options bar, the bottom is the visualization bar, and the middle is used as a container to load different forms of interfaces, the purpose of this design is to make the functions of the system clearer, and the user can jump to the required function by a one-key switch. The interface makes it easier for the user to operate the system. The main monitoring interface has been divided into two modules: the monitoring module and the user interface module, so that the options bar at the top is divided into two sections: "Data Monitoring" and "User Interface". The visualization bar at the bottom is also divided into three sections according to the function of the monitoring module: "Text data", "Curve data" and "Video data", and the layout between the sections is LinearLayout. The controls in the options bar and the visualization bar are not the basic controls provided by the mobile platform system, but rather developer-defined combinations of controls, with a uniform image at the top text at the bottom, and a LinearLayout layout. The background color becomes lighter when a control is selected.

In the design of the heat network monitoring system, the main interface was created by writing an XML layout file for the corresponding interface. This makes it easier to control the layout of the controls, and the orientation and layout_weight properties in Linearlayout allow the combined controls to be distributed according to a certain layout ratio. The main interface is visualized by loading the activity_main.xml file in the main Activity, as shown in the following code.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
.....
setContentView(R.layout.activity_main);
}
```

The *onCreate()* function is called when the Activity is initialized, and the *setContentView()* function is called inside the function. The function is its core, and the parameter *R.layout.activity_main* is the layout file for monitoring the main interface.

2, the design of the history data query interface

To facilitate the user to view and delete the historical data, the user-independent query interface is designed. The user can write the start and end time in the edit box to query and delete data within a specific period according to their needs; when the period queried by the user does not exist, the system will give the user a prompt until the correct time is entered or returned, the data finding process is shown in Figure 5. When the queried historical data exists, the heat pipe network monitoring system provides two ways of data display, namely text display and curve display.

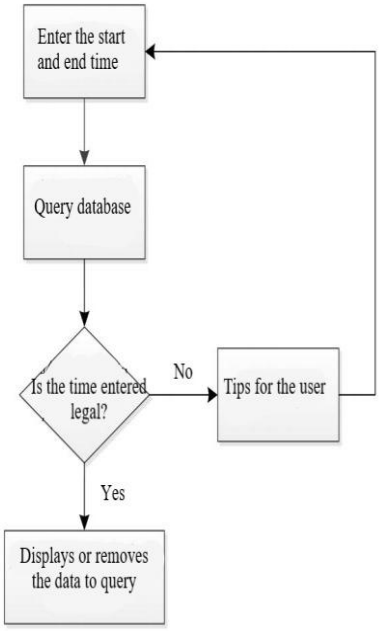


Figure 5: Data Search Process

4 Usability testing

The thermal network surveillance system's usability testing analyses both the main interface and the historical data query interface to guarantee user satisfaction. The main interface is segmented, with an options bar for "Data Monitoring" and "User Interface," as well as a visualization bar that displays "Text data," "Curve data," and "Video data." This design provides easy access to operations and intuitive interaction via custom-designed controls. The historical data query interface allows users to enter start and end times for data retrieval and elimination, with error messages for invalid queries. Testing will concentrate on task completion times, error rates, and user feedback in order to validate efficiency and detect enhancements.

4.2 Discussion

The presented multi-classification fault diagnosis model outperforms existing SOTA techniques, with a classification accuracy of 99.99% and a computation time of only 55.938 seconds. This enhancement is due to the execution of algorithmic optimizations, particularly the gravitational search technique integrated with parameter tuning of the LSSVM. These optimizations boost the model's capacity to navigate the parameter space efficiently, leading to better classification efficiency than prior methods, which attained a maximum accuracy of 97.89% and required longer run times. This work makes a unique contribution by incorporating sophisticated optimization methods that not only raise accuracy but also decrease algorithmic intricacy, rendering the model more effective. While trade-offs between computation time and accuracy are prevalent in machine learning, the proposed solution provides practical benefits by offering better accuracy without substantially reducing processing speed, establishing it as a feasible choice for real-time fault identification in a variety of uses.

The complexity analysis of optimization algorithms, such as the gravitational search algorithm (GSA), cuckoo algorithm (CA), and particle swarm optimization (PSO), focuses on their time complexity and resource needs. GSA has a time complexity of O (n. k), rendering it effective for moderate-sized issues, while CA exhibits O (n. log(n)), allowing for rapid convergence in smaller datasets. PSO, with a complexity of O (n. m), can become expensive as dimensionality rises. These variations have an effect on scalability; GSA's effectiveness renders it appropriate for bigger heat pipe networks, whereas CA and PSO may have higher computational overhead with large datasets, requiring optimizations for practical uses.

5 Conclusion

In summary, the author has analysed the requirements of this heat pipe network monitoring system, focusing on the heat pipe network inspection robot, and completed the overall architecture design and basic interface design of the heat pipe network monitoring system. The specific functional requirements of the heat pipe network monitoring system were analysed as follows: user login, collection of sensor data, visualization of data, storage of data, communication, dynamic generation of user-defined interfaces and logic, and other functions. The overall architecture of the heat pipe network monitoring system is designed. According to the functional requirements of the system, the data acquisition, basic functions, system, and user code exception handling functions are unified and managed by the Service, which reduces the redundancy of the code and facilitates maintenance at a later stage. Based on the functional requirements and architecture of the heat network monitoring system, the interface architecture of the system was designed, using Activity as the carrier to achieve the functionality of the system interface interaction by dynamically loading Fragment layouts. From the application of common login methods and the characteristics of this software, the login module, data query module, and main interface of the heat pipe network monitoring system are designed. The design of this monitoring system is important for the improvement of monitoring efficiency.

Data Availability

All data are included within the article.

Conflicts of interest

The authors declare no conflicts of interest.

Funding statement

Not applicable.

References

[1] Shen, Y., Chen, J., Fu, Q., Wu, H., Wang, Y., & Lu, Y. (2021). Detection of district heating pipe network leakage fault using UCB arm selection method. Buildings, 11(7), 275. https://doi.org/10.3390/buildings11070275

[2] Perpar, M., & Rek, Z. (2020). Soil temperature gradient is a useful tool for small water leakage detection from district heating pipes in buried channels. Energy, 201, 117684. https://doi.org/10.1016/j.energy.2020.117684

[3] Al Qahtani, T., Yaakob, M. S., Yidris, N., Sulaiman, S., & Ahmad, K. A. (2020). A review of water leakage detection method in the water distribution network. Journal of Advanced Research in Fluid Mechanics and Thermal Sciences, 68(2), 152-163. https://doi.org/10.37934/arfmts.68.2.152163

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

[5] Li, W., Liu, T., & Xiang, H. (2021). Leakage detection of water pipelines based on active thermometry and FBG-based quasi-distributed fiber optic temperature sensing. Journal of Intelligent Material Systems and Structures, 32(15), 1744-1755. https://doi.org/10.1177/1045389x20987002

[6] Zholtayev, D., Dauletiya, D., Tileukulova, A., Akimbay, D., Nursultan, M., Bushanov, Y., ... & Yeshmukhametov, A. (2024). Smart Pipe Inspection Robot with In-Chassis Motor Actuation Design and Integrated AI-Powered Defect Detection System. IEEE Access. https://doi.org/10.1109/access.2024.3450502

[7] Murtazin, I. I., Kozhevnikov, M. V., & Starikov, E. M. (2021). Development and application of methods of internal inspection of district heating networks. International Journal of Energy Production and Management. 2021. Vol. 6. Iss. 1, 6(1), 56-70. https://doi.org/10.2495/eq-v6-n1-56-70

[8] Wong, B., & McCann, J. A. (2021). Failure detection methods for pipeline networks: From acoustic sensing to cyber-physical systems. Sensors, 21(15), 4959. https://doi.org/10.3390/s21154959

[9] Liu, R., Zhang, Y., & Li, Z. (2022). Leakage diagnosis of air conditioning water system networks based on an improved BP neural network algorithm. Buildings, 12(5), 610. https://doi.org/10.3390/buildings12050610

[10] Korlapati, N. V. S., Khan, F., Noor, Q., Mirza, S., & Vaddiraju, S. (2022). Review and analysis of pipeline leak detection methods. Journal of pipeline science and engineering, 2(4), 100074. https://doi.org/10.1016/j.jpse.2022.100074

[11] Yussof, N. A. M., & Ho, H. W. (2022). Review of water leak detection methods in smart building applications. Buildings, 12(10), 1535. https://doi.org/10.3390/buildings12101535

[12] Langroudi, P. P., & Weidlich, I. (2020). Applicable predictive maintenance diagnosis methods in service-life prediction of district heating pipes. Rigas Tehniskas Universitates Zinatniskie Raksti, 24(3), 294-304. https://doi.org/10.2478/rtuect-2020-0104

[13] van Dreven, J., Boeva, V., Abghari, S., Grahn, H., Al Koussa, J., & Motoasca, E. (2023). Intelligent approaches to fault detection and diagnosis in district heating: Current trends, challenges, and opportunities. Electronics, 12(6), 1448. https://doi.org/10.3390/electronics12061448

[14] Hossain, K., Villebro, F., & Forchhammer, S. (2020). UAV image analysis for leakage detection in district heating systems using machine learning. Pattern Recognition Letters, 140, 158-164. https://doi.org/10.1016/j.patrec.2020.05.024

[15] Vollmer, E., Ruck, J., Volk, R., & Schultmann, F. (2024). Detecting district heating leaks in thermal imagery: Comparison of anomaly detection methods. Automation in Construction, 168, 105709. https://doi.org/10.1016/j.autcon.2024.105709

[16] Kim, H., Lee, J., Kim, T., Park, S. J., & Kim, H. (2023). Advanced thermal fluid leakage detection system with machine learning algorithm for pipe-in-pipe structure. Case Studies in Thermal Engineering, 42, 102747. https://doi.org/10.2139/ssrn.4147041

[17] Pérez-Pérez, E. D. J., López-Estrada, F. R., Valencia-Palomo, G., Torres, L., Puig, V., & Mina-Antonio, J. D. (2021). Leak diagnosis in pipelines using a combined artificial neural network approach. Control Engineering Practice, 107, 104677. https://doi.org/10.1016/j.conengprac.2020.104677

[18] García-Ródenas, R., Linares, L. J., & López-Gómez, J. A. (2021). Memetic algorithms for training feedforward neural networks: an approach based on gravitational search algorithm. Neural Computing and Applications, 33(7), 2561-2588. https://doi.org/10.1007/s00521-020-05131-y

[19] Kazeminasab, S., & Banks, M. K. (2022). Towards long-distance inspection for in-pipe robots in water distribution systems with smart motion facilitated by a particle filter and multi-phase motion controller. Intelligent Service Robotics, 15(3), 259-273. https://doi.org/10.1007/s11370-022-00410-0

Biometric-Based Secure Encryption Key Generation Using Convolutional Neural Networks and Particle Swarm Optimization

Sahera A. S. Almola, Raidah S. Khudeyer, Hameed Abdulkareem Younis
Department of Computer Information Systems, College of Computer Science and Information Technology, University of Basrah, Basrah, Iraq
E-mail: sahera.sead@uobasrah.edu.iq, raidah.khudayer@uobasrah.edu.iq, hameed.younis@uobasrah.edu.iq
*Corresponding author

Keywords: biometric verification, fingerprints, deep learning, particle swarm optimization (pso) algorithm, encryption key generation

Received*:* December 7, 2024

With the rapid expansion of computer networks and information technology, ensuring secure data transmission is increasingly vital—especially for image data, which often contains sensitive information. This research presents a biometric-based encryption system that uses fingerprint recognition and deep learning to generate strong, random encryption keys. Two convolutional neural networks (CNNs) are employed: one to verify identity based on a user's ID and another to extract fingerprint features for key generation. These keys are optimized using Particle Swarm Optimization (PSO), enhancing their randomness and resistance to brute-force attacks.

The system generates keys in real-time, eliminating the need for storage and minimizing the risk of theft or leakage. To further improve security, encryption keys are automatically updated after every ten messages, with different keys generated from multiple fingerprints of the same individual. Testing with the SOCOFing dataset (6,000 original and 49,270 synthetic images) achieved 99.75% identity verification and 99.83% classification accuracy. Performance metrics—entropy of 7.89, correlation factor of 0.00628, and zero repetition—demonstrate high robustness. This approach offers a secure, adaptive, and personalized encryption method ideal for sensitive domains like finance and healthcare.

Povzetek: Opisana je izvirna metoda za generiranje varnih šifrirnih ključev z uporabo prstnih odtisov, CNN modelov in optimizacije roja delcev (PSO)

1 Introduction

Internet and network users share millions of color images daily, which are utilized in various applications such as telemedicine, remote learning, business, and military operations. Color images, in particular, often contain sensitive and detailed information, making them prime targets for unauthorized access and cyberattacks. Securing these images is crucial not only to prevent data loss during transmission but also to protect sensitive information from attackers. Various techniques are employed to secure digital images, such as watermarking, steganography, and image encryption. Encryption operates in two main stages: encryption and decryption. During encryption, the input image is transformed into an unreadable form using a secret key, while in decryption, the content is restored using the same key [1]. The encryption key is a fundamental element in the encryption and decryption processes, and it significantly determines the security system's strength. However, a critical challenge faced by encryption systems lies in managing the encryption key itself [2]. Traditional encryption methods require transmitting the encryption key to the recipient to decrypt the data. This approach introduces vulnerabilities, as any exposure of the key during transmission could lead to the compromise of the encrypted data. Consequently, there is an increasing need for systems that dynamically generate encryption keys on-demand at the user's end, eliminating the need for key transmission over networks [3]. This innovative approach ensures that the encryption key is generated locally each time data is decrypted, significantly reducing risks associated with key interception. It also eliminates the need for key exchange, adding an extra layer of security since unauthorized parties cannot generate the key even if communication is intercepted.

The keyless exchange method, when combined with biometric verification, offers a highly secure solution by minimizing the risk of key theft. This approach aligns with the methodology presented in this research. However, implementing such a solution poses significant challenges in the fields of secure computing and key management, as it requires a robust system to ensure the consistent and accurate generation of keys [4]. The importance of this research lies in emphasizing the generation of encryption keys locally at the user's end to safeguard data and mitigate risks associated with key transmission over networks. This is particularly critical for securing color images, as their high information content often correlates with increased sensitivity, making them especially vulnerable to sophisticated attacks.

To address these challenges, advanced techniques based on artificial intelligence and machine learning,

particularly deep learning, have emerged. One notable technique involves using deep learning to generate encryption keys from fingerprints. This method leverages the extraction of unique features from fingerprints, converting them into robust, non-repetitive encryption keys to ensure high data security [5]. This method addresses limitations in traditional encryption systems, such as the need for key transmission over networks. Since a fingerprint is a unique biometric identifier that cannot be easily copied or mimicked, it serves as an ideal source for generating encryption keys. Moreover, deep learning enhances the accuracy and strength of the generated keys by utilizing deep neural networks to analyze biometric images and extract unique features for each fingerprint [6]. This approach also resists advanced threats, including brute-force and quantum encryption attacks, by dynamically generating encryption keys in real-time. The added layer of complexity and secrecy prevents unauthorized parties from accessing the keys, even if communication data is partially intercepted [7]. The integration of deep learning in generating encryption keys from fingerprints represents a significant advancement in information security. This approach combines robust security measures with individual privacy, paving the way for building encryption systems that are highly resistant to breaches and better equipped to address modern security challenges. The remainder of this paper is organized as follows: Section 2 reviews related works, while Section 3 provides background on the key techniques utilized in this research. Section 4 explains the management of secret keys. Section 5 details the proposed method. Section 6 focuses on experimental results and performance analysis. Section 7 discusses the results, and Section 8 concludes this study.

2 Related works

The integration of biometric data, chaotic systems, and deep learning in encryption key generation has been a prominent research area. Various studies have explored innovative approaches to enhance the security and robustness of encryption systems. Hashem and Kuban (2023) [8] introduced a system that leverages fingerprint biometrics to generate long, random encryption keys. The approach involves preprocessing fingerprint images to remove noise, utilizing a modified VGG-16 convolutional neural network (CNN) to extract unique features, and employing transfer learning to build a key generation model without the need for retraining. Erkan et al. (2024) [9] proposed a secure image encryption framework that combines a chaotic logarithmic map with a deep CNN for key generation. Their system incorporates advanced operations such as permutation, DNA encoding, diffusion, and bit-reversal to ensure security. The robustness of this framework was validated through comprehensive analyses, including key sensitivity and resistance to various attacks, demonstrating superior performance compared to traditional encryption methods.

Quinga Socasi, Zhinin-Vera, and Chang (2020) [10] developed a method for generating encryption keys from alphanumeric passwords using an autoencoder neural network. Their experiments revealed that this method outperforms conventional algorithms, particularly when encrypting small text files, making it highly resistant to cracking attempts. Wu et al. (2022) [11] presented a biometric key generation framework that uses fingerprints to achieve over 1024-bit key strength and 98% accuracy. However, their method depends on a predefined pipeline and fuzzy extractors for key stabilization. In contrast, the method proposed in this research dynamically extracts high-resolution fingerprint features using deep learning models, ensuring greater adaptability across datasets. These features are combined with chaotic encryption systems to enhance randomness and security. Furthermore, Particle Swarm Optimization (PSO) is employed to optimize the generated keys, achieving over 99% accuracy and producing 1024-byte keys without requiring stabilization layers. This approach demonstrates superior flexibility and security for real-world IoT applications. Alesawy and Muniyandi (2016) [12] investigated data security in cloud environments using random encryption keys. Their study analyzed the impact of incorporating Elliptic Curve Diffie-Hellman (ECDH) keys and demonstrated significant improvements in efficiency and performance by integrating Artificial Neural Networks (ANNs) with ECDH and genetic algorithms, despite increased processing times for larger datasets. Saini and Sehrawat (2024) [13] proposed a technique for generating unique encryption keys by combining an autoencoder network with hashing techniques and prime numbers derived from the MNIST dataset. To enhance security, the system incorporates XOR operations and Blum-Blum-Shub (BBS) generators. Extensive testing confirmed the robustness of this approach against attacks. Kurtninykh, Ghita, and Shiaeles (2021) [14] addressed the complexities of cryptographic key management in systems with increasing users and applications. They evaluated five key management systems, including Hashicorp Vault and Pinterest Knox, focusing on features such as security, scalability, and access control. The study concluded that Hashicorp Vault is particularly suitable for small businesses due to its superior security features. A summary of the related studies is provided in Table 1. for further reference.

Table 1: Previous works on key generation

Metrics Used	Method Limitations	Dynamic Key Generation and Non-Transferability	Key Space	Accuracy	Technique(s) Used	Researcher(s)
Confusion matrix, Precision, Recall, F1-Score, ROC Curve	Reliance on specific datasets, weak attack analysis	Stable, Transferable, Permanent	2^{1024}	98%	Key generation method from fingerprint image based on deep Convolutional neural network model	Hashem, M. & Kuban, K.H (2023)
Keyspace analysis, Key sensitivity, Information entropy,Histogram,Correlation, Differential attack, Noisy attack, Cropping attack	Chaos effects may be unreliable in other scenarios	Stable, Transferable, Permanent	2^{398}	Not specified	An Image Encryption Scheme Based on Chaotic Key Generation using Deep CNN	Erkate, U., Toktas, A., Enginoglu, S., et al (2024)
Resistance to brute-force attacks	Performance declines with large files and lacks adaptability and attack analysis	Stable, Transferable, Permanent	2^{2527}	Outperforms traditional algorithms with small text files	A Deep Learning Approach for Symmetric-Key Cryptography System	Quinga-Socasi, F., Zhinin-Vera, L., & Chang, (2020)
Misrecognition rate, Generation intensity, Accuracy	Weak analysis of pattern diversity and resistance	Stable, Transferable, Permanent.	2^{1024}	98%	Fingerprint bio-key generation based on deep neural networks	Wu, Z., Lv, Z., Kang, J., et al (2022)
Encryption effectiveness and data security in cloud environments	Complexity in implementation, lack of attack evaluation	Stable, Transferable, Permanent.	Not specified	Improved time efficiency and performance	Hybrid Encryption using ECCDH with ANN for key exchange in cloud computing environments	Alesawy, O., & Muniyandi, R.C (2016)
Keyspace analysis, Binary entropy analysis, Resistance to brute-force attacks	The system is complex, and its reliance on limited technologies may affect performance	Stable, Transferable, Permanent.	2^{512}	Final training loss: 0.0974, Validation loss: 0.0953	Enhancing Data Security through Machine Learning-based Key Generation and Encryption	Saini, A., & Sehrawat, R (2024)
Performance metrics, Key management systems, Implementation complexity, Vulnerabilities	The variability of systems and the limited evaluation	Stable, Transferable, Permanent.	Not specified	Not specified	Comparative Analysis of Cryptographic Key Management Systems	Kurtiniyh, I., Ghita, B., & Shiaeles, S (2021
Confusion matrix, Precision, Recall, F1-Score,Entropy Repetition,Chi-Squared,Pearson Correlation, Stability, Range, Autocorrelation TestFalse Positive Rate (FPR)	The limitation of the method lies in the data type.	Dynamic, Non-Transferable ,Ephemeral	2^{8192}	99.73 - 99.83	Biometric-Based Secure Encryption Key Generation Using Convolutional Neural Networks and Particle Swarm Optimization	Proposed Method (2025)

This research builds upon the foundations laid by these studies, emphasizing the dynamic generation of encryption keys using deep learning and chaotic systems to address challenges in key management and enhance security. The comparison Table 1. clearly demonstrates the superiority of our proposed method over all previous approaches. The proposed method utilizes dynamic keys generated by deep learning networks, which significantly enhance randomness and security. Moreover, the key is non-portable, non-persistent, and achieves the largest size and highest accuracy compared to other methods.

3 Background

This paragraph addresses two main techniques: CNNs and PSO, which form the foundation of the methodology proposed in this research. In the following paragraphs, we will provide a summary of each technique and explain its significance in the study.

A. CNNs are advanced models in the field of deep learning, specifically designed to handle grid-like data, such as images. In this research, two CNN models were used to generate an encryption key based on fingerprint

images. Table 2. summarize the components of each model used in the work.

Table 2: Components of CNN models used

Layer (Type)	Output Shape	Parameters (#)
Conv2D (conv2d_1)	(None, 92, 92, 32)	832
BatchNormalization(batch_normalization_1)	(None, 92, 92, 32)	128
MaxPooling2D(max_pooling2d_1)	(None, 46, 46, 32)	0
Conv2D (conv2d_2)	(None, 42, 42, 64)	51,264
BatchNormalization batch_normalization_2)	(None, 42, 42, 64)	256
MaxPooling2D (max_pooling2d_2)	(None, 21, 21, 64)	0
Conv2D (conv2d_3)	(None, 19, 19, 128)	73,856
BatchNormalization (batch_normalization_3)	(None, 19, 19, 128)	512
MaxPooling2D (max_pooling2d_3)	(None, 9, 9, 128)	0
Dropout (dropout_1)	(None, 9, 9, 128)	0
Flatten (flatten_1)	(None, 10368)	0
Dense (dense_1)	(None, 1024)	10,617,856
Dropout (dropout_2)	(None, 1024)	0
Dense (dense_2)	(None, 600)	615,000

The first model was designed to identify a person's identity based on their ID number. After confirming the person's identity, the second model identifies the selected fingerprint and extracts its features. Both models rely on convolutional layers to automatically and progressively extract important features from the input data, making them effective in performing tasks, which, in turn, aids in generating strong encryption keys by analyzing fine patterns in the images.

The two models were trained using the backpropagation technique with a suitable loss function for each task. This architectural design was chosen to achieve accurate performance in recognizing the identity of the fingerprint owner through the identifier number in the file name, and then generate an encryption key based on the unique features of the fingerprint using two convolutional neural networks.

Pseudo-code for the PSO Algorithm

1. **Initialize Parameters**:
   - Define bounds:
     - Lower bound (lb) = 0
     - Upper bound (ub) = 255
   - Set PSO parameters:
     - Number of particles = len(keys)
     - Maximum iterations = 200
     - Inertia weight (w) = 0.9
     - Cognitive coefficient (c1) = 0.5
     - Social coefficient (c2) = 0.5
   - Set random seed for reproducibility.
2. **Initialize Particles**:
   - Convert keys to a NumPy array.
   - Set initial particle positions = keys.
   - Set initial velocities = zeros.
   - Initialize personal bests:
     - Personal best positions = initial positions.
     - Personal best scores = evaluate fitness for each particle.
   - Find global best:
     - global_best_position = position with the best score.
     - global_best_score = best personal score.
3. **Run PSO Optimization**:
   For each iteration in range(num_iterations) **do**:
   - **For** each particle **do**:
     - Update velocity:
       - new_velocity = (w × current velocity) +(c1 × random factor × (Personal best – current position)) +(c2 × random factor × (global best - current position)).
     - Update position:
       - new_position = current position + new_velocity.
       - Clip positions to bounds (lb, ub).
     - Evaluate fitness of the new position.
       - Update personal best position and score.
   - Update global best:
     - **If** any particle's score is better than the global best score:
       - Update global best position and score.
4. **Output Results**:
   - Convert global_best_position to integer (best_key).
   - Compute best_entropy_value using the fitness function.

Figure 1: PSO algorithm

This approach adds an extra layer of security, ensuring that the keys are not only unique and non-repetitive but also resilient to various forms of attacks. The use of PSO ensures that the final encryption keys are both optimized for security and

B. PSO (Particle swarm optimization) is an optimization algorithm inspired by the collective behavior of birds or fish. It involves a group of particles, each representing a potential solution in the solution space. Each particle adjusts its movement based on its own experience and the experiences of neighboring particles, with the aim of reaching the optimal solution. PSO is known for its efficiency and ability to find optimal solutions in multi-dimensional spaces. In this research, PSO is applied to optimize the process of encryption key generation.

The algorithm enhances the randomness and strength of the generated keys, ensuring that they are both secure and resistant to attacks. PSO improves the key generation process by fine-tuning the key parameters in real-time, making it more robust against potential security threats. This approach adds an extra layer of security, ensuring that the keys are not only unique and non-repetitive but also resilient to various forms of attacks. The use of PSO ensures that the final encryption keys are both optimized for security andgenerated dynamically, without the need for permanent storage, thus reducing the risks of key leakage or unauthorized access. **Key enhancement using PSO:** The Particle swarm optimization (PSO) algorithm is used to enhance the quality of the initial key, making it stronger and more secure. Figure 1. illustrates the detailed steps of the Particle swarm optimization (PSO) algorithm using pseudo-code. This pseudocode reflects the essence of the PSO algorithm applied to optimize encryption keys

based on the fitness function (such as randomness or security). The process iteratively adjusts the position (key) and velocity of the particles to find the optimal encryption key with high security.

4 Secure key management

Secure key management is a critical process to ensure the protection of encrypted data across encryption systems. In the proposed methodology, the focus is on generating cryptographic keys in real-time without permanent storage, thus reducing the risks associated with key leakage. However, temporary handling and protection of keys during their lifecycle remain essential. Below is a detailed explanation of the steps and importance of secure key management, updated to reflect the real-time generation approach:

1. Key generation: In the proposed system, keys are generated dynamically and in real-time using advanced techniques such as artificial neural networks, particularly convolutional neural networks (CNN). This approach ensures that the keys are both highly secure and non-repetitive, avoiding the need for long-term storage. These keys are designed to be sufficiently random and robust, minimizing the possibility of guessing or tampering.

2. Temporary key handling: While keys are not stored permanently, they are managed securely during their temporary existence within the system. During encryption or decryption processes, the keys are stored in memory with strict safeguards, such as memory encryption or secure enclaves, to prevent unauthorized access. Once the operation is complete, the keys are securely erased from the system to eliminate any residual risk.

3. Key distribution: Since the system eliminates the need for traditional key exchange, the reliance on secure protocols like SSL/TLS or Diffie-Hellman for key distribution is significantly reduced. Instead, the generated key remains local to the system, mitigating risks associated with interception during transmission [16]**4. Key rotation:** In systems where keys are reused for multiple sessions or extended periods, regular key rotation is critical. However, in the proposed system, each key is uniquely generated for a specific session or operation, inherently providing the benefits of key rotation by design.

5. Key revocation: Although the system minimizes the use of persistent keys, mechanisms for immediate key invalidation are essential for scenarios involving session-based or temporarily stored keys. These mechanisms ensure that any exposed or misused keys are rendered unusable promptly [17].

6. Importance of key management in real-time systems: The proposed approach emphasizes the importance of secure key handling during the active lifecycle of keys. By avoiding permanent storage and focusing on real-time generation and temporary protection, the system significantly reduces the risks associated with key leakage or unauthorized access. This approach aligns with best practices in modern cybersecurity by combining the advantages of real-time key generation with robust temporary key management to ensure the highest level of data protection throughout the encryption process [16].

5 Proposed method

Figure 2: presents the diagram for the proposed encryption key management and generation.

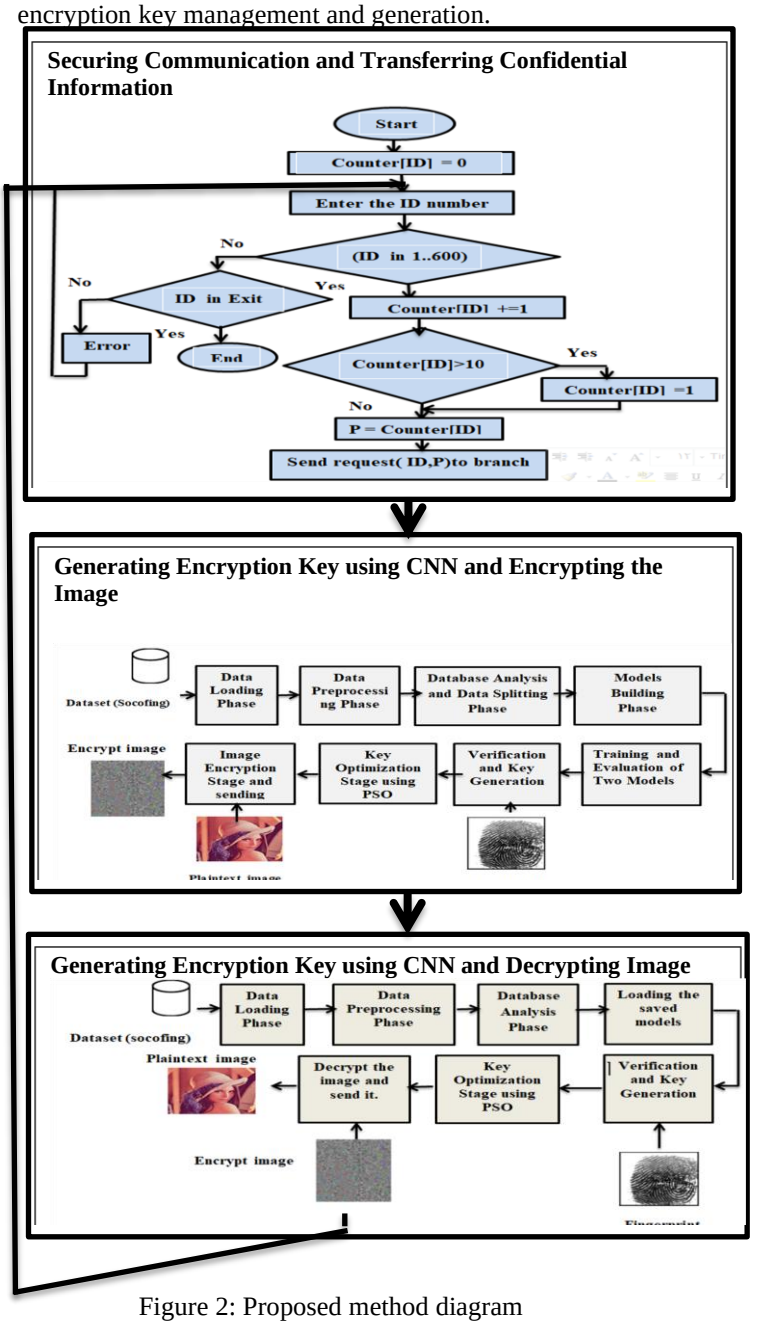


Figure 2: Proposed method diagram

The

proposed method consists of three main parts. The first part begins with an algorithm for securing communication and managing encryption keys. This is followed by the second part, which involves the process of generating the encryption key and encrypting the image. Finally, the third part focuses on decrypting the image after the key has been generated. Each of these parts will be explained in detail later.

Part One: Securing communication and managing confidential information transfer

The first part of Figure 2 illustrates an algorithm designed to ensure secure communication and reliable key management between branches and the main branch. When a branch requests access to sensitive information (such as encrypted images), the main branch fulfills this request by sending the requested information after encrypting it with a secure key, ensuring data protection during transmission. User ID is used to control access.

Algorithm execution steps

The algorithm is executed in cooperation with the following two parts in the diagram as follows:

1. **Starting the process (start):** The process begins by initializing the user's counter Counter [ID] to zero.
2. **Entering the ID number:** The system prompts the user to input their identification number to verify their identity.
3. **Verifying the ID range (ID in 1..600):** The system checks whether the entered ID number falls within the allowed range (1 to 600).

    - If the number is outside the range, an error message is displayed, and the user is asked to re-enter the ID.
    - If the number is valid, the process moves to the next step.

4. **Checking the match with the exit indicator (ID in exit):** The system compares the entered ID with the exit indicator list.

        - If a match is found, the process is terminated.
        - If no match is found, the process continues to the next step.

5. **Incrementing the message counter (counter [ID] += 1):**

If the ID is valid and not listed in the exit indicator, the user's message counter is incremented by 1.

6. **Managing the number of sent messages (dynamic key management):** The system checks whether the number of messages sent by the user has exceeded the allowed limit (10 messages).

    - If the limit is exceeded, the counter is reset to 1.
    - If the limit is not exceeded, the current counter is used as an index for generating the encryption key. This mechanism ensures unique encryption keys for each set of messages, enhancing data security. Additionally, it raises a critical question: **"Can biometric fingerprint data generate dynamic encryption keys resistant to quantum attacks?"**

This approach aims to strengthen the security of biometric keys against advanced threats such as quantum attacks.

7. **Sending the request to the branch (send request to branch):** The request containing the ID **(ID)** and the fingerprint index **(P)** is sent to the second branch for processing.

- **In the second part:** A key is generated for image encryption, and the encryption process is executed. After encryption, the encrypted image is sent back to the first part.
- **In the third part:** A new key is generated to decrypt the image.

Once decryption is completed, the data is returned to the first part for the remaining steps.

Note: The details of the second and third parts will be explained in the following sections of the document for a precise and comprehensive understanding. In this way, the three parts form an integrated system that ensures secure communication and the safe transmission of sensitive information effectively.

Algorithm features

- **Biometric security**: Fingerprints are used as a means to verify user identities, which reduces the risks of unauthorized access.
- **Synchronization**: The system relies on concurrent processing, enhancing performance efficiency and reducing response times for requests.
- **Dynamic key management**: Each key is generated uniquely for each user based on their fingerprint, increasing the difficulty of breaching the system. This algorithm ensures effective protection of encrypted data and enhances the security of communications between branches, making it an excellent choice for systems that require a high level of security and privacy.

Part Two: Encryption key generation using CNN and image encryption

The encryption key is generated using CNN based on the fingerprint. This process is carried out as specified in Part 2 of the diagram, which includes the following operations:

1. **Database loading phase:** This step is considered one of the main preparatory phases in the system to ensure the readiness of the data and models required to achieve accuracy and security in encryption key generation. In this research, the SOCOFing database was used, which contains fingerprints from 600 people of African descent, with each person having 10 fingerprints, resulting in a total of 6,000 original fingerprints. Additionally, synthetic groups were created with three levels of variation in the fingerprints: minor changes (Easy), medium changes (Medium), and significant changes (Hard). The total number of synthetic fingerprints used in training was approximately 49,270. The variation fingerprints were used for training the model, while the original fingerprints were used solely for testing.
2. **Data preprocessing phase**: The following processes are included:
3. **Image size standardization**: To ensure that all images in the database are compatible with the model requirements, the dimensions of all images are standardized. A common size, such as 96×96 pixels, is often chosen to prepare the images for efficient model processing. The formula for resizing the images can be expressed mathematically as shown in Equation (1) below:

$$I'(x',y') = I(y/Sy, x/Sx) \quad (1)$$

Where:

I(x,y) is the original image, and I′(x′,y′) is the image after resizing, with Sx and Sy representing the scaling factors in the image dimensions[18].

A. Image enhancement using histogram equalization:

The histogram equalization technique was applied to enhance contrast in fingerprint images and highlight fine details. This technique is one of the fundamental methods in image processing and quality enhancement, aiming to improve the distribution of grayscale levels in the image to make fine details more visible. In images with low contrast, gray values may cluster within a narrow range, leading to the loss of fine details in dark or bright areas. Histogram equalization is used to address this issue by improving the distribution of these gray values over a broader range of available colors, enhancing contrast and making details easier to detect. The process of adjusting the tonal gradients in the image is carried out using the following equation (2)[19]:

$$H'(I) = \frac{CDFmin - CDF(I)}{(NXM) - CDFmin}(1\text{-}L) \quad (2)$$

The histogram equalization process involves several key parameters that affect the final outcome of the operation:

1. Cumulative distribution function (CDF): This is the primary factor that determines how grayscale values are redistributed in the image. The CDF accumulates grayscale values progressively from the lowest to the highest and is used to adjust the distribution. Through this function, the grayscale value distribution in the image is calculated, and adjustments are made to spread these values evenly across the color range.

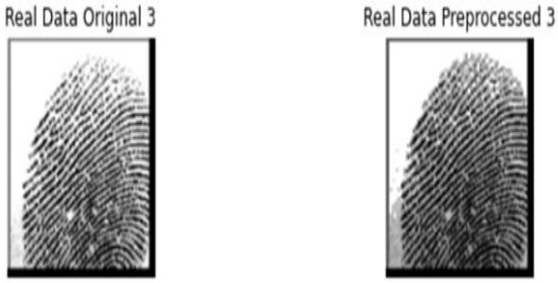
2. Minimum non-zero value (CDFmin): This refers to the smallest non-zero value in the cumulative distribution function. It is used to determine how grayscale values in the image will be adjusted to achieve a more balanced distribution. For example, if the grayscale values in the image are concentrated around a particular value, utilizing this minimum helps improve the distribution of those values without significantly affecting the overall contrast of the image.

3. Image size (N×M): This refers to the number of pixels in the image. The larger the image (i.e., a greater N×M), the more opportunities there are for accurately redistributing grayscale values. However, it is important to note that image size can impact processing speed, as larger images require more computations.

4. Number of gray levels (L): Typically, L=256 in grayscale images (meaning there are 256 possible tonal levels ranging from 0 to 255). The number of gray levels defines the range of colors that can be distributed across the image. In images with a high number of gray levels, tonal gradations can be distributed more evenly, leading to better contrast enhancement.

When applying this technique, the range of grayscale values in the image is expanded, and these values are evenly distributed across the color range, leading to increased contrast. This enhanced contrast reveals fine details in the image, such as the minutiae in fingerprints, which might be poorly visible in low-contrast images. In the case of fingerprints, fine details such as ridges and patterns are often crucial for analysis and classification.

By using histogram equalization, the clarity of these fine details can be improved, aiding in better feature extraction of the fingerprint and achieving higher performance in systems that use fingerprint recognition. Figure 3. shows an example of fingerprints before and after contrast enhancement using histogram equalization. Notice how the enhanced images display finer and clearer details compared to the original images.



(a) (b)

Figure 3: (a) Original image (b) Image after histogram equalization

5. Database analysis phase

After image enhancement, each fingerprint is analyzed to identify distinctive features, such as patterns and key regions within the fingerprint. This helps prepare the data for the model to understand the unique elements in each fingerprint. The goal of this process is to efficiently analyze the fingerprint database to extract the necessary information for feeding two different models. This is achieved by parsing the file name to extract the individual's identity, finger type, and hand (right or left). Additionally, these steps prepare the data for the two models, allowing the first model to recognize the individual's identity, while the second model identifies the finger type based on fingerprint information. After applying these processes to the database, the data is divided into training and testing sets. Artificial fingerprint data is used for training, while original data is used for testing.

6. Model building phase: After preparing the database, two models are built using CNNs. The first model aims to identify the person's identity (SubjectID), while the second model aims to determine the finger number (FingerNum) and extract distinctive features of the finger. Each model consists of the layers shown in Table 2. The hyperparameters used in the models are illustrated in Table 3.

Table 3: CNN hyperparameters configuration

Hyperparameter	Value
Number of Convolutional Layers	3
Batch Size	64
Filter Size	3x3, 5x5
Filters per Layer	32-64-128
Activation Function	ReLU, Softmax, tanh
Optimization	Adam
Learning Rate	0.0001
Epochs	20
L2 Regularizer	regularizers.l2(0.001)
Loss Function	Categorical_crossentropy
Dropout	0.3, 0.4
MaxPooling	MaxPooling2D(pool_size=(2, 2))

7. Training and evaluation phase of the two models

The performance of both models is evaluated using standard performance metrics such as accuracy, validation, and error rate calculation. This ensures that the first model is capable of accurately verifying the identity of authorized individuals when requesting the encryption key. Similarly, the second model's performance is assessed to determine its ability to correctly identify the fingerprint belonging to the individual whose identity has been verified. This evaluation is done using the test set.

8. Identity verification and key generation

The identity of the individual and the fingerprint match with the registered name are verified using two deep learning models. This is a key step in generating the encryption key from fingerprints, as illustrated in Figure 4.

9. Key optimization stage using PSO

To enhance the quality of the initial key and obtain a stronger, more secure key, PSO algorithm is applied. This algorithm aims to improve the random distribution and security properties of the key. The goal of this algorithm is to increase the randomness of the key and ensure its difficulty in being guessed or broken. The use of the PSO algorithm to optimize encryption keys relies on updating the positions and velocities of particles based on the individual's fingerprints, as illustrated in Figure 1. This continuous update of the keys, leveraging the best personal and global positions, results in generating an encryption key that is more secure and complex. This also raises the question: **"How does the proposed system perform against statistical attacks?"** This approach aims to reduce the likelihood of the keys being exposed to any repetitive patterns that could be exploited in statistical attacks. Table 4. outlines the hyperparameters used in the optimization algorithm, selected based on a series of experimental trials.

Pseudo-Code for verification and encryption key generation

1. **Initialize finger name function**:
   - Define show_fingername(fingernum):
     - If fingernum >= 5:
       - Set hand = "right" and subtract 5 from fingernum.
     - Otherwise:
       - Set hand = "left".
     - Map fingernum to finger names (e.g., little, ring, middle, index, thumb).
     - Return the full finger name (hand + finger).
2. **Verify fingerprint information**:
   - Predict subject ID and finger number for a random fingerprint (rand_fp_num) from the test set using models:
     - Id_pred = predicted subject ID.
     - Id_real = actual subject ID.
     - fingerNum_pred = predicted finger number.
     - fingerNum_real = actual finger number.
   - Check predictions:
     - If both IDs and finger numbers match:
       - Print "Information confirmed" with subject ID and call show_fingername(fingerNum_pred) to get the finger name.
     - Otherwise:
       - Print "Prediction is wrong."
3. **Extract candidate fingerprints**:
   - Initialize lists keys1 (for original fingerprints) and keys2 (for dense layer outputs).
   - For each index i in the prediction range:
     - Get Id_check = predicted subject ID.
     - If Id_check == Id_pred:
       - Append the fingerprint to keys1.
       - Append the dense layer output to keys2.
   - Convert keys1 and keys2 to arrays.
4. **Select target fingerprint**:
   - Use index p1 to select:
     - original_fp = keys1[p1].
     - dense_output_finger_selected = keys2[p1].
5. **Apply data augmentation**:
   - Define an image data generator (datagen) with +transformations:
     - Rotation, width/height shift, shear, zoom, and horizontal flip.
   - Reshape original_fp to fit the generator's input format.
6. **Generate augmented fingerprints and keys**:
   - Use datagen to create 20 augmented fingerprints:
     - For each augmented fingerprint:
       - Generate a new fingerprint.
       - Predict the dense layer output.
       - Take absolute values of the output to create a key.
       - Append the key to the keys list.
7. **Return results**:

- Output Keys // To be used as input for PSO algorithm to find optimal key the list keys for use in encryption.

Figure 4: Pseudocode for the identity verification and key generation process

Unit/Explanation	Description/Value	Parameter
Lower bound of the range	0	lb
Upper bound of the range	255	ub
Number of particles in the swarm	len(keys)	num_particles
Maximum number of iterations	200	num_iterations
Inertia weight	0.9	w
Cognitive coefficient (personal best focus	0.5	c1
Social coefficient (global best focus)	0.5	c2
Random factor for cognitive component; varies per particle	0.5	r1
Random factor for social component; varies per particle	0.5	r2
Initial positions of the particles	keys	particle positions
Initial velocities of the particles	zeros	particle velocities
Personal best positions	Updated based on the best performance of each particle	personal_best_positions
Best scores achieved by each particle	Calculated through fitness evaluation	personal best scores
Global best position	Updated based on the global best performanc	global best position
Global best performance	Updated based on the global best value	global best score

10. Image encryption stage and sending encrypted image

Chen's chaotic system is a three-dimensional dynamic system that exhibits chaotic behavior and is based on nonlinear differential equations to represent the evolution of the state over time. It can be used to generate a chaotic encryption key based on the system's state. The Chen chaotic system relies on the following equations that describe the changes in the variables x, y, and z:

$$\frac{dx}{dt} = a.(x - y) \quad (3)$$

$$\frac{dy}{dt} = (a - c).\text{x- x . z + c . y} \quad (4)$$

$$\frac{dz}{dt} = x.y - \text{ b . z} \quad (5)$$

Where: x, y, and z are the variables that determine the state of the chaotic system at time t.

- a, b, and c are the parameters that control the behavior of t he system. Steps followed:

- **Initial conditions**: The process starts by defining the initial values for x, y, and z, which represent the state of the system at the beginning of the simulation. These values are set in the code as [1.0,1.0,1.0]. **Numerical integration**: The odeint function is used for numerical integration to solve the differential equations over time. Through this process, the values of x, y, and z are updated at each time step, based on the parameters a, b, and c that influence the system's behavior.
- **Generating a chaotic sequence:** A chaotic sequence is generated by solving the differential equations of the Chen chaotic system over multiple time steps. This sequence is then used to generate a chaotic encryption key.
- **Encryption key generation:** The resulting chaotic sequence is converted into integer values ranging from 0 to 255 to represent color values in an RGB image. This is done by multiplying each value in the sequence by 255 and converting it to the uint8 data type.
- **Combining the chaotic key with the generated key:** The chaotic key is combined with the key generated using CNN through an XOR operation. This step increases the complexity of the final key used for image encryption.
- **Encrypting the image:** The XOR operation is applied between the original image and the final key to generate the encrypted image. This operation transforms the pixel values in the image into new values based on the chaotic key.

Part three: key generation using CNN and image decryption*:*

This part is similar to the stages in Part 2, with the only difference being that the models are not built and trained again; instead, the previously saved models are loaded. Additionally, there is a decryption stage instead of encryption. The stages in this part are as follows:

- **Data loading phase:** Only the test set is loaded (i.e., 600 genuine fingerprints from SOCOFing).
- **Data preprocessing phase:** The raw data is processed to prepare it for the next stage.
- **Database analysis phase:** The data is analyzed to extract the necessary information.
- **Loading the saved models:** The previously trained models are loaded.
- **Verification and key generation:** The data is verified, and the key is generated.
- **Key optimization stage using PSO:** The key is optimized using the PSO algorithm. **Decrypt the image and send it:** The image is decrypted and sent. Decryption: To decrypt the image, the same key (the chaotic key and the key generated from CNN) is used to perform an XOR operation on the encrypted image, restoring the original image.

6 Results and analysis

This section of the research addressed four main axes: evaluating the CNN, assessing the generated key, evaluating the PSO algorithm, and finally comparing the results of the proposed method with similar methods. As follows:

A. Results of CNN models and performance analysis

At this stage, the data was divided into training and testing sets to ensure the accuracy of the models in predicting and distinguishing between different categories. 80% of the data was allocated for training, and 20% for testing, ensuring an equal distribution of categories in both sets to avoid bias. After training the models on the training set, their performance was tested using the test data.

The accuracy of the models was calculated using the accuracy function available in the TensorFlow library, which represents the ratio of correct predictions to the total number of predictions. To continuously monitor performance, TensorBoard was used, which helped track various metrics such as accuracy and loss throughout the training and testing phases, allowing for ongoing improvements to the models based on these indicators. The results obtained showed varying performance across the different models, with these results summarized in Tables 5, 6, and 7, which illustrate accuracy and loss across different generations. Additionally, the graphs in Figures 5, 6, and 7 show the evolution of the models' performance over time, highlighting the models' ability to learn and improve progressively**. Where the false positive rate was 0.000185.**

Table 5: Model performance comparison: accuracy and loss.

ID_ Name Fingerprint example	Model No	Training Accuracy(%)	Testing Accuracy (%)	Training Loss	Testing Loss
331_ right little 	1	99.733334	0.9943	0.0563811	0.0947
331_ right little 	2	99.833333	0.9856	0.1004451	0.1387

Table 6: Classification report for finger recognition

No_ SubjectID	Precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
2	1.00	1.00	1.00	10
3	1.00	1.00	1.00	10
4	1.00	1.00	1.00	10
5	1.00	1.00	1.00	10
.	.	.	.	.
.	.	.	.	.
597	1.00	1.00	1.00	10
598	1.00	1.00	1.00	10
599	1.00	1.00	1.00	10
accuracy			1.00	6000
macro avg	1.00	1.00	1.00	6000
weighted avg	1.00	1.00	1.00	6000

Table 7: Classification report for subjectID recognition

No_Finger	Precision	recall	f1-score	Support
0	1.00	0.99	1.00	600
1	1.00	1.00	1.00	600
2	1.00	1.00	1.00	600
3	1.00	1.00	1.00	600
4	1.00	1.00	1.00	600
5	1.00	1.00	1.00	600
6	1.00	1.00	1.00	600
7	1.00	1.00	1.00	600
8	1.00	1.00	1.00	600
9	1.00	1.00	1.00	600
accuracy			1.00	6000
macro avg	1.00	1.00	1.00	6000
weighted avg	1.00	1.00	1.00	6000

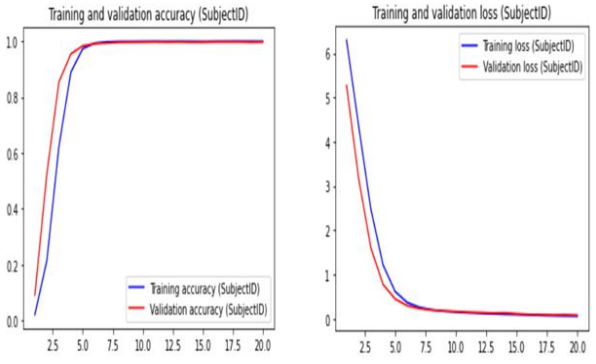


Figure 5: Accuracy and loss of the identity model.

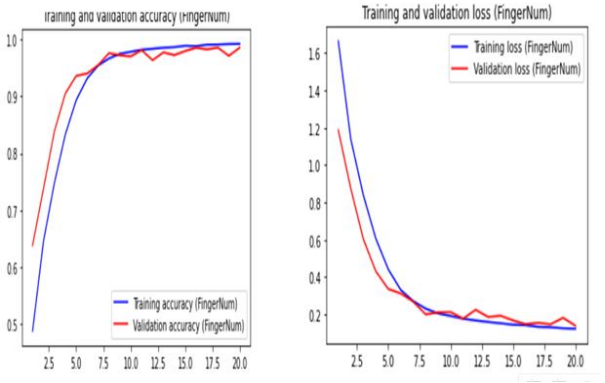


Figure 6: Accuracy and loss of the fingerprint model.

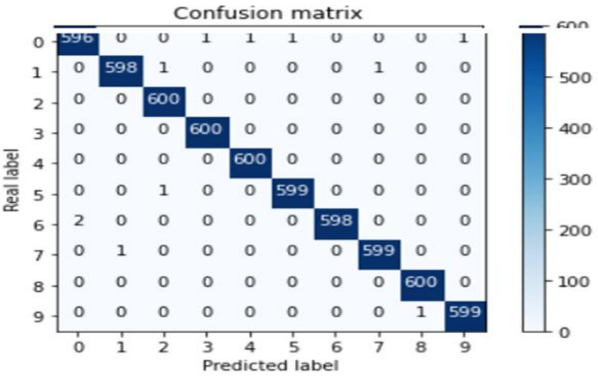


Figure 7: Confusion matrix and accuracy metric

B. Encryption key evaluation results and metrics: The generated encryption key was evaluated using a set of specialized metrics to ensure its quality and effectiveness in resisting cyberattacks. The experiments were conducted using fingerprint images sized 96 × 96 in a Kaggle environment with Python, on a workstation equipped with an Xeon(R) I (R) processor, 64 GB of RAM, and a GPU P100. The metrics used included evaluations such as key size and various randomization tests (such as entropy test, repetition test, etc.) to assess the randomness of the key and its predictability. These tests help ensure that the system remains unaffected when used in live applications.

6.1 Key space analysis

Brute-force attack is a type of cyber-attack that relies on guessing the key by attempting a large number of possible passwords or secret phrases. An encrypted image with a short key is highly vulnerable to this attack over time. However, if the key is longer, it will remain resistant for a longer period. Therefore, it becomes impossible to guess the key if it has an adequate length. Key space analysis is used to assess the strength against brute-force attacks.

According to this analysis, a key with a length greater than 2^{100} is considered suitable for high-security encryption [26]. In our system, we propose an approach based on deep neural networks (CNN) and PSO to generate this key.

The key has a size of 1024 values, with each value ranging between 0 and 255. This means the key consists of 1024 bytes (since each value requires one byte, and 8 bits are enough to represent values from 0 to 255). Given that each value in the key ranges from 0 to 255, we have 256 possibilities for each value. With 1024 values, the total key space will be: 256^{1024} or, in other words, $2^{8\times1024}=2^{8192}$.This represents an extremely large key space, which is sufficiently large to be highly resistant to brute-force attacks. A key size of 2^{8192} offers a very high level of security, making it practically impossible to crack using brute-force methods, even with fast computing devices. Nonetheless, the question remains: **How does the proposed system perform against brute-force attacks?**

Comparison of key space (2^{8192}) with traditional systems

- **Comparison with AES-256:**

The proposed key space (2^{8192}) is significantly larger compared to AES-256, where the key space is approximately 2^{256}. This substantial difference makes our key space more resistant to brute-force attacks. Traditional systems like AES-256 rely on efficient algorithms to compensate for the smaller key space compared to the vast proposed space.

- **Comparison with RSA-2048:**

The proposed key space is also significantly larger compared to RSA-2048, where the key space is approximately 2^{2048}. RSA relies on computational complexity for large numerical factorization, whereas in our system, the security strength depends on the key length derived from biometric features processed through deep networks.

- **Comparison with ECC-384 (Elliptic curve cryptography):**

The traditional key space for ECC-384 is approximately 2^{384}, which is much smaller compared to our proposed key space (2^{8192}). ECC relies on elliptic curves to compensate for shorter keys, but in contrast, we provide much longer keys derived from neural networks, enhancing their unpredictability.

- **Comparison with DES (Data encryption standard):**

The key space in DES is 2^{56}, which is extremely small compared to our proposed key space. DES is considered outdated and vulnerable to brute-force attacks, whereas our proposed key space vastly surpasses it in terms of length and complexity.

6.2 Significance of results in cryptographic key management

- The results, such as randomness tests and high entropy, demonstrate that the generated key exhibits a high degree of randomness, making it ideal for high-security applications.
- High entropy indicates that the keys have a uniform distribution of values, reducing
- the likelihood of predicting any part of the key, which is a critical feature in key management.

6.3 Encryption key tests

In this study, six fingerprint samples were used as the basis to generate six encryption keys. Each key underwent comprehensive testing using eight different metrics to determine the quality and randomness of the generated keys. The results of these eight tests were systematically presented in a table, reflecting the effectiveness of the proposed method. The results showed the success of the keys in all eight tests, confirming that the keys generated from the fingerprints meet the required security standards. These tests demonstrate the randomness and unpredictability of the keys, making the approach suitable for secure encryption applications. The core encryption key tests include the following:

- **Entropy test**:

The entropy measures the distribution of information in the key and reflects the level of randomness. The entropy is calculated using the following equation (6):

$$H(X)= -\sum_{i=1}^{n} p\,(xi)\; log2\; p(xi) \qquad (6)$$

Where p(xi) is the probability distribution of the value *xi* in the key. If the entropy equals 8 bits, it means the key is completely random [26].

Table 8: Results of the entropy test

ID	5	3	1	173	174	175	
Finger Print							Ideal Value
Entropy Test	7.782	7.763	7.748	7.778	7.774	7.87	8

- **Repetition test**: The repetition test generally aims to ensure that the key does not contain any repeated sections within its sequence, whether these sections are adjacent or non-adjacent. If parts of the key are repeated, it weakens the randomness and increases the likelihood of discovering a pattern that can be exploited in an attack. This test involves checking all parts of the key to detect any repetition that might impact its security level. The repetition test addresses repetition in the key overall, whether in adjacent or non-adjacent parts [27].

Table 9: Results of the repetition test

ID	5	3	1	173	174	175	
Finger Print							Ideal Value
Repetition Test	0	0	0	0	0	0	0

- **Uniformity test using the chi-squared test:**

This test aims to check whether the values in the encryption key are evenly distributed across the full range of possible values. The **chi-squared test** is used to compare the actual distribution of values in the key with the expected ideal distribution. If the values are evenly distributed, the key is considered to have a uniform distribution. Equation (7) illustrates the test:

$$\chi 2=\sum_{i=0}^{255}\left(\frac{(\frac{n}{256}-ni)}{n/256}\right) \qquad (7)$$

- **ni**: The frequency of occurrence of value i in the key.
- **n/256**: The expected frequency for each value i assuming a uniform distribution.
- **n**: The total number of values in the key.
- If the chi-square ($\chi 2$) value is low, it indicates that the actual distribution of values is close to the ideal distribution, meaning the key is evenly distributed.
- At a significance level of 0.05, if the chi-square value is less than 293.25, the key is considered to have passed the test and has a uniform distribution [28].

Table 10: Results of the uniformity test

ID	5	3	1	173	174	175	
Finger Print							Ideal Value
Uniformity Test (Chi-Squared)	262.0	215.0	271.5	295.5	226.5	327.0	293

- **Repetition test(adjacent)** focuses specifically on identifying repetition in adjacent parts of the key. This test checks for any repeated consecutive or sequential sections that might indicate a fixed pattern or excessive repetition, which could weaken the effectiveness of encryption. Repetition of adjacent parts is considered a sign of poor randomness, thus reducing the strength of the key. The closer the value is to 0, the less repetition there is, which means the key has a higher level of randomness [29].

Table 11: Results of the repetition test(adjacent)

ID	5	3	1	173	174	175	
Finger Print							Ideal Value
Repetition Test(Adjacent)	0.59	0.2932	0.6842	0.293	0.586	0.977	0

- **Pearson correlation test** is a statistical test used to measure the relationship between two variables. This relationship is expressed by a coefficient called the "Pearson correlation coefficient," which ranges from -1 to 1. If the correlation coefficient is close to 0, it indicates no correlation (high randomness), making the encryption key strong and hard to predict. The purpose is to determine the extent of the correlation between values in the encryption key. If the correlation coefficient is close to 0, it indicates that the key is sufficiently random, thus making it strong against analytical attacks. The equation (8) represents the Pearson correlation equation:

$$r = \frac{\sum(Xi-\bar{X})(Yi-\bar{Y})}{\sqrt{\sum(Xi-\bar{X})2\cdot\sum(Yi-\bar{Y})2}} \qquad (8)$$

Where:

- **r**: Pearson correlation coefficient.
- **Xi**: Individual values in the first series.
- **Yi**: Individual values in the second series (e.g., lagged values in time series).
- $\bar{X}$: Mean of the Xi values.
- $\bar{Y}$: Mean of the Yi values [30].

Table 12: Results of the pearson correlation test

ID	5	3	1	173	174	175	
Finger Print							Ideal Value
Average Pearson Correlation	0.0062	0.0062	0.0062	0.0062	0.0062	0.0062	0.05

- **Stability test**

The key must remain stable if the input data is stable. This means that if the same inputs are used to generate the key multiple times, the resulting key should always be identical. However, slight changes in the inputs should result in a significant change in the key, which enhances encryption strength against attacks.

6.3 Consistency of fixed inputs

- If the input is fixed I, the encryption system should produce the same key K every time: F(I)=K.
- Repeat key generation multiple times using the same I, and the result should be consistent K in all attempts: K1=K2=⋯=Kn.

Table 13: Results of the stability test (1)

The key generated from attempt 5	The key generated from attempt 4	The key generated from attempt 3	The key generated from attempt 2	The key generated from attempt 1	Finger print	ID
[95 39 243 220 42…]	[95 39 243 220 42…]	[95 39 243 220 42…]	[95 39 243 220 42…]	[95 39 243 220 42…]		5
[143 24 107 86 110…]	[143 24 107 86 110…]	[143 24 107 86 110…]	[143 24 107 86 110…]	[143 24 107 86 110…]		3
[127 6 38 182 127….]	[127 6 38 182 127….]	[127 6 38 182 127….]	[127 6 38 182 127….]	[127 6 38 182 127….]		1
[188 224 208 191 161….]	[188 224 208 191 161…]	[188 224 208 191 161…]	[188 224 208 191 161…]	[188 224 208 191 161…]		173
[77 115 56 243 251….]	[77 115 56 243 251….]	[77 115 56 243 251…]	[77 115 56 243 251…]	[77 115 56 243 251…]		174
[106 30 245 143 52….]	[106 30 245 143 52….]	[106 30 245 143 52….]	[106 30 245 143 52….]	[106 30 245 143 52….]		175

6.4 Sensitivity to minor changes (inclusivity effect)

We make a slight change in the input I to create I′. A new key K′ is generated using I′: F(I′) = K′. We measure the difference between K and K′ using the Bit Change Rate:

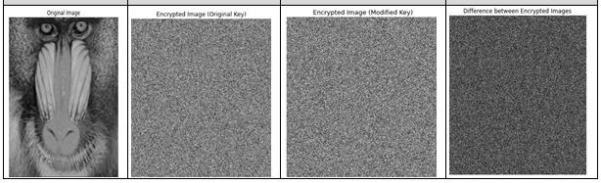
$$\text{Bit change rate}=\frac{\text{Bit difference between K and K′}}{1024}\times 100\%$$

The change rate should be higher than 50% to ensure the system's sensitivity to changes [31].

Table 14: Results of the stability test (1)

ID	5	3	1	173	174	175	
Finger print							Ideal Value
The key difference resulting from the input change	0.068	0.093	0.060	0.057	0.049	0.039	<0.1

Table 15: Encryption results with the original key and the modified key

Original image	Encryption using the original key	Encryption using the modified key	Difference between the encrypted images
			

6.5 Range test

The **range test** aims to evaluate the distribution of encryption key values within a specific range to ensure its randomness.

Steps of the range test

1. **Calculate the range**: Determine the difference between the maximum value max and the minimum value min. Range=max−min
2. **Range splitting**: divide the range into buckets.
3. **Frequency calculation**: count the values in each bucket.
4. **Distribution analysis**: if the frequencies are approximately equal, the key is considered random. expected frequency equation (9):

$$Ei = N/M \quad (9)$$

where N is the total number of values, and M is the number of buckets[27].

Table 16: Results of the range test

ID	5	3	1	173	174	175	
Finger Print							Ideal Value
Range Test	passed	passed	passed	passed	passed	passed	0..255

- **Autocorrelation test**

The Autocorrelation Test is used to determine the randomness of a sequence of values in an encryption key. If the key is sufficiently random, the autocorrelation values should be small or close to zero, indicating no clear pattern or dependency in the sequence. To calculate the autocorrelation at a lag d, equation (10) is used:

$$\mathbf{R(d)} = \frac{1}{n-d} \sum_{i=1}^{n-d} (xi.xd + i) \quad (10)$$

Where: R(d) is the autocorrelation coefficient for lag d.

- xi is the value at position i in the sequence.
- X_{d+i}_ is the value at position d+i in the sequence.
- n is the length of the sequence.

A value of R(d) close to zero for different values of d indicates a high level of randomness in the encryption key, The Figure 8. shows the distribution of autocorrelation test results, highlighting successful and failed values based on the specified critical value (0.05) [28].

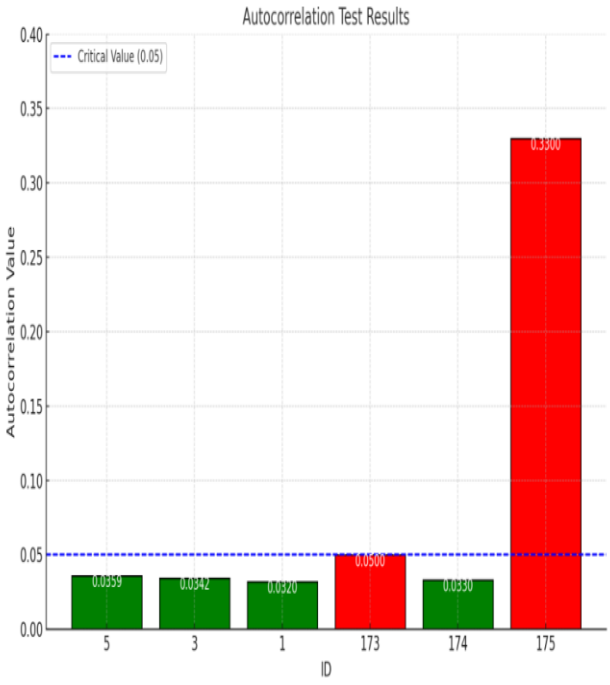


Figure 8: Distribution of autocorrelation test results with success and failure indication based on critical value

Table 17: Results of the autocorrelation test

Result Explanation	Critical Value	Ideal Value	Autocorrelation Test	Finger print	ID
Pass (close to 0)	0.05	Close to 0	0.0359		5
Pass(close to 0)	0.05	Close to 0	0.0342		3
Pass(close to 0)	0.05	Close to 0	0.032		1
Fail (pattern detected)	0.05	Close to 0	Failed		173
Pass (close to 0)	0.05	Close to 0	0.033		174
Fail (deviation from ideal value)	0.05	Close to 0	0.33		175

C. Results of using PSO in enhancing the encryption key

After completing the specified number of iterations, the best encryption key is obtained, which is the key that achieved the highest fitness during the optimization process. Table 18. illustrates the effect of using PSO on the generated encryption key.

Table 18: The impact of the PSO algorithm in improving the encryption key.

No(ID)	5	3	1	173	174	175
Finger print						
Key strength without PSO	4.682	4.804	3.975	5.093	4.700	4.843
Key strength with the use of PSO	7.782	7.707	7.687	7.762	7.774	7.703

D. Comparison of the accuracy of the proposed system with other systems

This section evaluates the accuracy of our proposed system in comparison to other systems reported in recent years, based on their respective sources. Experimental results from our proposed model demonstrated an accuracy exceeding 99%. Table 19. presents a detailed comparison between our system and other existing systems.

Table 19: Accuracy comparison between our system and recent approaches.

Epochs	Data Type	%ACC	Method Using	Reference
100	Fingerprint	99-98	CNN model (VGG-16 Tuned)	2023[8]
60	Fingerprint	98.89	DeepFKTNet model	2022[29]
55	Fingerprint	99.10	Modified–LetNet(tested on FVC2004-DB1)	2022[30]
50	Fingerprint	98.00	MCP-FP model	2022[11]
--	Natural image	95.30	B-DCNNs	2022[13]
30	Fingerprint	96.10	CNN(3 classes of FVC2004)	2019[31]
20	Fingerprint	98.70	Modified HOG descriptor	2018b[34]
30	Fingerprint	97.40	Statistics of D-SIFT descriptor	2018a[33]
20	Fingerprint	99.73-99.83	CNN and PSO	proposed system

The aim of this comparison is to evaluate the effectiveness of the proposed system in the context of recent advancements in deep learning technology, providing insight into how cybersecurity can be enhanced through the application of advanced techniques. The table also reflects the ongoing progress in fingerprint data processing, showing that modern systems achieve higher accuracy than traditional systems, supporting the idea that using deep learning can improve the effectiveness and security of encryption systems.

7 Discussion

In this section, we will discuss the results of the proposed system in comparison to modern methods presented in Table 19, focusing on accuracy, randomness tests (such as entropy), and robustness. Additionally, we will address potential trade-offs associated with using CNNs, such as computational overhead. Below is a detailed comparison of key results.

7.1 Comparison of results

Accuracy: The proposed system (using CNN and PSO) achieved an accuracy between **99.73% and 99.83%** within just 20 epochs, outperforming most models in the table. For example: The **enhanced VGG-16 model** achieved **99.98% accuracy** in 100 epochs, the highest in the table, but required five times more epochs than the proposed system. The **Modified-LeNet model** achieved **99.10% accuracy** in 55 epochs, which is lower than the proposed system. The **DeepFKTNetmodel** achieved **98.89% accuracy** in 60 epochs. Thus, the proposed system stands out as a strong option, delivering high accuracy in less training time, thanks to the combination of CNN and PSO, which enhances feature extraction and generates robust keys.

Randomness tests (e.g., entropy): The use of PSO in the proposed system significantly contributed to enhancing randomness, which strengthens the generated keys. When comparing randomness tests (e.g., entropy) with other models, the proposed system showed remarkable superiority. The combination of CNN and PSO enabled the generation of keys with excellent randomness levels, providing a higher degree of security compared to traditional models. PSO helps optimize the quality of the keys by searching for the optimal combination of hidden parameters, making them more random and harder to break.

It is important to note that the model only retains the predictions generated during its operation, which are values devoid of any sensitive information. This enhances the system's security against various types of attacks, such as mixed replacement attacks, crossover attacks, and exhaustive search attacks. In such attacks, the attacker has no knowledge of the key generation mechanism or the supporting data, making the number of attempts required to crack the key increase proportionally with its length. For example, if the key length is 1024, the number of possible combinations would reach 2^{8192} The proposed system focuses on enhancing data security by avoiding key storage, improving the randomness of key generation, and protecting sensitive information from various attacks, while ensuring high efficiency in user fingerprint recognition.

Robustness: Biometric key (encryption key for security):

The biometric key is generated based on the parameters learned during the training of the CNN model. During training, the model learns unique representations or features extracted from fingerprints. These representations are numerical weights that are not easily interpretable.

The parameters are converted into an encryption key that relies on the unique properties of each fingerprint, making the key:

- Unique and tamper-proof.
- More secure and resistant to duplication.

Role of PSO in key enhancement:

PSO improves the key by identifying the optimal values of the parameters used in key generation. This enhances randomness and independence among keys, making them more resistant to attacks.

7.2 Advantages of the proposed system

The proposed system combines CNN and PSO to achieve:

- High classification accuracy in less time.
- High-quality encryption keys with excellent levels of randomness and security.
- Strong protection of users' biometric data against exploitation or breaches.
- Improved biometric key performance using PSO to generate stronger and more random keys, increasing the system's resilience to cyber threats. Thus, the proposed system leverages the multiple features of CNN and PSO, making it more robust in addressing security challenges such as resistance to adversarial attacks. While other models primarily focus on classification and accuracy, the proposed system demonstrates additional strength in encryption applications.

Potential trade-offs:
Although the use of CNN in the proposed system results in a slight increase in computational overhead compared to simpler models like the modified LeNet, this does not pose a significant obstacle. The model is designed to operate efficiently on modern systems supported by Graphics Processing Units (GPUs), ensuring accelerated training and reduced execution time.

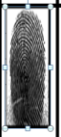
8 Conclusions

- The results of this research show that integrating biometric techniques with deep learning provides an innovative and effective solution for generating secure and robust encryption keys based on fingerprints. The proposed system enhances the security of data transmitted over the internet, making it more resistant to theft and tampering. The use of two convolutional neural network models is a significant step, where the first model contributes to identity recognition and the second focuses on fingerprint detail recognition, ensuring the extraction of unique and reliable biometric features.

- One of the main conclusions of this research is that the tanh activation function plays a crucial role in neural networks for generating encryption keys. This function is known for its ability to transform outputs into the range of (-1, 1) non-linearly, which contributes to improving the quality of the generated keys. Increased complexity and randomness:

The tanh function ensures a more balanced distribution of values across the range (-1, 1), reducing value concentration and enhancing the randomness of the key, leading to the generation of secure and robust encryption keys. Better stability during training: The tanh function helps avoid issues such as vanishing gradients, resulting in better stability during the training process and improved model performance in generating encryption keys. Table 20. illustrates the key strength (entropy measure) when using the Tanh activation function compared to using the ReLU activation function.

Table 20: Compares the key strength (entropy) of the Tanh and ReLU activation functions.

No(ID)	5	3	1	173	174	175
Finger print						
Using Tanh Activation Function	7.782	7.763	7.748	7.748	7.732	7.719
Using ReLU Activation Function	3.648	4.439	4.597	5.328	4.844	4.541

The batch normalization layer plays a significant role in stabilizing and accelerating the learning process in deep models by normalizing the outputs to have a mean of 0 and a standard deviation of 1. While this stabilization is beneficial in many applications, such as image classification, it may negatively impact the strength of the generated encryption key.

- The results indicate that the generated keys exhibit high levels of randomness, making them more challenging to breach. Additionally, the use of PSO algorithm is considered an effective technique for enhancing the randomness of the keys, as it allows for generating different keys for each transmission, thereby reducing the risk of key theft and increasing security. A comprehensive analysis of the performance of the models used in this research was conducted, showing a significant improvement in encryption effectiveness and the reliability of the generated keys, underscoring the efficiency of these models in the context of cybersecurity.

- The proposed approach enhances system security compared to traditional systems by reducing reliance on static keys, which are a vulnerability in many encryption systems. Instead, biometric verification is used to generate unique keys for each user based on their fingerprints, thereby increasing the level of security. This research provides a significant contribution to systems that require high levels of protection, such as financial systems and medical data, by facilitating biometric verification for encryption without the need to exchange keys, thereby reducing associated risks. Additionally, the automatic key change feature adds an extra layer of security,

reflecting the effectiveness of this system in providing advanced protection. Ultimately, the research highlights the importance of integrating biometrics and deep learning in developing effective security solutions that address contemporary challenges in data protection.

- The method presented in the research has wide potential for application in various fields. In addition to securing fingerprints and using them to generate encryption keys, the method can be applied to secure internet of things (IoT) devices by generating strong encryption keys that protect communication between devices. It can also be used to secure data stored in the cloud by generating high-security encryption keys based on unique user attributes, such as fingerprints. These applications highlight the flexibility and efficiency of the method in addressing modern cybersecurity challenges and enhance its appeal in various practical scenarios.

References

[1] Hosny, K. M., Darwish, M. M., & Fouda, M. M. (2021). Robust color images watermarking using new fractional-order exponent moments. *IEEE Access, 9*, 47425–47435. https://doi.org/10.1109/ACCESS.2021.3069317

[2] Kuzior, A., Tiutiunyk, I., Zielińska, A., & Kelemen, R. (2024). Cybersecurity and cybercrime: Current trends and threats. *Journal of International Studies, 17*(2). https://doi.org/10.14254/2071-8330.2024/17-2/5

[3] Saran, D. G., & Jain, K. (2023). An improvised algorithm for a dynamic key generation model. In *Inventive Computation and Information Technologies: Proceedings of ICICIT 2022* (pp. 607–627). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-5048-5_44

[4] Rahman, Z., Yi, X., Billah, M., Sumi, M., & Anwar, A. (2022). Enhancing AES using chaos and logistic map-based key generation technique for securing IoT-based smart home. *Electronics, 11*(7), 1083. https://doi.org/10.3390/electronics11071083

[5] Kuznetsov, O., Zakharov, D., & Frontoni, E. (2024). Deep learning-based biometric cryptographic key generation with post-quantum security. *Multimedia Tools and Applications, 83*(19), 56909–56938. https://doi.org/10.1007/s11042-023-15265-6

[6] Yang, W., Wang, S., Cui, H., Tang, Z., & Li, Y. (2023). A review of homomorphic encryption for privacy-preserving biometrics. Sensors, 23(7), 3566. https://doi.org/10.3390/s23073566.

[7] Rana, M., Mamun, Q., & Islam, R. (2023). Enhancing IoT security: An innovative key management system for lightweight block ciphers. Sensors, 23(18), 7678. https://doi.org/10.3390/s23187678

[8] Hashem, M. I., & Kuban, K. H. (2023). Key generation method from fingerprint image based on deep convolutional neural network model. Nexo Revista Científica, 36(6), 906-925. https://doi.org/10.5377/nexo.vXXiXX.XXXX

[9] Erkate, U., Toktas, A., Enginoglu, S., Karabacak, E., & Thanh, D. N. H. (2024). An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN. Expert Systems with Applications, 237, 121452. https://doi.org/10.1016/j.eswa.2023.121452

[10] Quinga Socasi, F., Zhinin-Vera, L., & Chang, O. (2020). A deep learning approach for symmetric key cryptography system. In Proceedings of the Future Technologies Conference (pp. 41). https://link.springer.com/chapter/10.1007/978-3-030-63128-4_41

[11] Wu, Z., Lv, Z., Kang, J., Ding, W., & Zhang, J. (2022). Fingerprint bio-key generation based on a deep neural network. International Journal of Intelligent Systems, 37(7), 4329–4358. https://doi.org/10.1002/int.22782

[12] Alesawy, O., & Muniyandi, R. C. (2016). Elliptic Curve Diffie-Hellman random keys using artificial neural network and genetic algorithm for secure data over private cloud. Information Technology Journal, 15(2), 77-83. https://doi.org/10.3923/itj.2016.77.83

[13] Saini, A., & Sehrawat, R. (2024). Enhancing data security through machine learning-based key generation and encryption. Engineering, Technology & Applied Science Research, 14(3), 14148-14154. https://doi.org/10.48084/etasr.7181

[14] Kurtninykh, I., Ghita, B., & Shiaeles, S. (2021). Comparative analysis of cryptographic key management systems. King's College London, Strand, London, WC2R 2LS, UK. https://doi.org/10.48550/arXiv.2109.09905.

[15] SSL Support Team. (2024, May 3). Key management best practices: A practical guide. Retrieved from [SSL Support Team Website] https://www.ssl.com/article/key-management-best-practices-a-practical-guide//

[16] Wang, L., & Lv, Y. (2024). Differential privacy-based data mining in distributed scenarios using decision trees. Informatica, 48(2), 145–158. https://doi.org/10.31449/inf.v48i23.6918

[17] Tu, Z., Milanfar, P., & Talebi, H. (2023). MULLER: Multilayer Laplacian Resizer for Vision. ResearchGate. Retrieved from https://www.researchgate.net/publication/369855623_MULLER_Multilayer_Laplacian_Resizer_for_Vision.

[18] Saifullah, S., Pranolo, A., & Dreżewski, R. (2024). Comparative analysis of image enhancement techniques for brain tumor segmentation: Contrast, histogram, and hybrid approaches. Journal Name, Volume (Issue), Page range. https://doi.org/10.48550/arXiv.2404.05341

[19] Singh, P., Dutta, S., & Pranav, P. (2024). Optimizing GANs for Cryptography: The Role and Impact of Activation Functions in Neural Layers Assessing the Cryptographic Strength. Applied Sciences, 14(6), 2379. https://doi.org/10.3390/app14062379.

[20] Zhang, B., & Liu, L. (2023). Chaos-Based Image Encryption: Review, Application, and Challenges. *Mathematics*, *11*(11), 2585. https://doi.org/10.3390/math11112585

[21] Taylor, O. E., & Igiri, C. G. (2024). Enhancing image encryption using histogram analysis, adjacent pixel autocorrelation test in chaos-based framework. International Journal of Computer Applications, 186(22). https://doi.org/10.5120/ijca202492338

[22] Munshi, N. H., Das, P., & Maitra, S. (2022). Chi-Squared Test Analysis on Hybrid Cryptosystem. Volume 14, Issue 1, 34-40. https://doi.org/10.2174/1876402913666210508235706.

[23] Rasheed, A. F., Zarkoosh, M., & Abbas, S. (2023, October). Comprehensive Evaluation of Encryption Algorithms: A Study of 22 Performance Tests. 2023 Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE), Surabaya, France, 191-194. https://doi.org/10.1109/ICVEE59738.2023.10348240.

[24] Feng, L., Du, J., & Fu, C. (2023). Double graph correlation encryption based on hyperchaos. PLOS ONE, 18(9), e0291759. https://doi.org/10.1371/journal.pone.0291759.

[25] Barker, E., & Roginsky, A. (2024). NIST SP 800-131A Rev. 3: Transitioning the use of cryptographic algorithms and key lengths (Initial Public Draft). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-131Ar3.ipd.

[26] Avaroğlu, E., Kahveci, S., & Akkurt, R. (2024). Optimization of Acoustic Entropy Source for Random Sequence Generation Using an Improved Grey Wolf Algorithm. Computer Engineering Department, Faculty of Engineering, Mersin University. https://doi.org/10.18280/ts.410220

[27] Foreman, C., Yeung, R., & Curchod, F. J. (2024). Statistical testing of random number generators and their improvement using randomness extraction. Cryptology ePrint Archive, Paper 2024/492. Retrieved from https://doi.org/10.3390/e26121053

[28] Taylor, O. E., & Igiri, C. G. (2024). Enhancing image encryption using histogram analysis, adjacent pixel autocorrelation test in chaos-based framework. International Journal of Computer Applications, 186(22). https://doi.org/10.5120/ijca2024923653

[29] Saeed, F., Hussain, M., & Aboalsamh, H. A. (2022). Automatic fingerprint classification using deep learning technology (DeepFKTNet). Mathematics, 10(8), 1285. https://doi.org/10.3390/math10081285

[30] Nahar, P., Chaudhari, N. S., & Tanwani, S. K. (2022). Fingerprint classification system using CNN. Multimedia Tools and Applications, 81(17), 24515–24527. https://doi.org/10.1007/s11042-022-13494-6

[31] Nguyen, H. T., & Nguyen, L. T. (2019). Fingerprints classification through image analysis and machine learning method. Algorithms, 12(11), 241. https://doi.org/10.3390/a12110241

[32] Ang, L.-M., Seng, K. P., Ijemaru, G. K., & Zungeru, A. M. (2018). Deployment of IoV for smart cities: Applications, architecture, and challenges. IEEE Access, 7, 6473–6492. https://doi.org/10.1109/ACCESS.2018.2886575

[33] Saeed, F., Hussain, M., & Aboalsamh, H. A. (2018a). Classification of live scanned fingerprints using dense SIFT based ridge orientation features. 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), 1–4. https://doi.org/10.1109/CAIS.2018.8441995

[34] Saeed, F., Hussain, M., & Aboalsamh, H. A. (2018b). Classification of live scanned fingerprints using histogram of gradient descriptor. 2018 21st Saudi Computer Society National Computer Conference (NCC),1–5. https://doi.org/10.1109/NCC.2018.8682629

CNN and LSTM-Based Multimodal Data Fusion for Performance Optimization in Aerobics Using Wearable Sensors

Danhua Tan
School of Physical Education, Hengyang Normal University, Hengyang, Hunan, 421006, China
E-mail: tandanhua184914@outlook.com

Keywords: wearable sensors, convolutional neural network, long short-term memory, kalman filtering, aerobics movements

Received: May 31, 2025

Aerobics is a high-intensity, multi-dimensional sport. Its motion evaluation places higher demands on data quality and time series modeling capabilities. This paper proposes a method for evaluating aerobics motion that integrates wearable sensors and motion tracking systems. It combines convolutional neural networks (CNNs) with long short-term memory networks (LSTMs) to perform fusion analysis on multimodal data from accelerometers, gyroscopes, magnetometers, and Kinect motion capture systems. To improve data quality, Kalman filtering, time synchronization, and wavelet transform techniques are introduced to preprocess the raw data. Experimental results show that this method performs well in motion classification tasks: in indoor low-intensity training scenarios, the accuracy of the CNN model increases from 74.5% to 87.1%; in high-intensity training scenarios, the accuracy increases from 75.0% to 88.2%. After combining with LSTM, the model further enhances the modeling capabilities of motion time series features and improves the recognition accuracy of complex motions. In different training scenarios, the average improvement rate of motion scores is 25.8%. The system feedback delay is controlled within 200 milliseconds, with good real-time and practical performance. This method provides aerobics athletes with high-precision movement assessment and personalized training suggestions, promoting the intelligent and personalized development of sports training.

Povzetek: Metoda združuje senzorje, CNN in LSTM za multimodalno analizo aerobičnih gibov. Kalmanovo filtriranje izboljša kakovost signalov, klasifikacijska točnost naraste do 88,2 %, povprečno izboljšanje rezultatov znaša 25,8 %, odzivnost sistema pa ostane pod 200 ms.

1 Introduction

With the popularity of aerobics, the accuracy of movements and training effects have become the focus of coaches and athletes. During high-intensity and complex exercise, the movements of aerobics athletes can be affected by factors such as physical exertion, sports skills, and external environment, resulting in unstable movement performance. Traditional manual evaluation methods are inefficient and subjective, and cannot provide athletes with accurate training feedback in real-time. With the development of sensor technology [1] and artificial intelligence [2], [3], motion evaluation methods based on wearable devices [4] and intelligent feedback systems have become a research hotspot. Such feedback systems can provide accurate real-time data analysis, optimize training programs, and improve athlete performance. Therefore, developing a motion evaluation and optimization system based on intelligent technology [5], [6] has become the key to improving training effects and athlete performance.

This paper studies the motion evaluation and optimization system based on intelligent technology to improve the training effect and motion performance of aerobics athletes. To achieve this goal, this paper combines wearable sensors with motion tracking systems and uses CNN models and LSTM to fuse and analyze multimodal data. The system acquires motion data through sensors such as accelerometers, gyroscopes, magnetometers, and Kinect motion capture systems. It uses Kalman filtering, time synchronization, and wavelet transform to optimize data quality. The optimized data is used through the CNN model to evaluate and optimize motion performance, providing real-time feedback and personalized training suggestions. The CNN-based optimization method combines wearable sensor technology with deep learning (DL) algorithms to improve the accuracy and stability of motion evaluation. Experimental results show that the combination of Kalman filtering and CNN models effectively improves the accuracy and stability of aerobics motion evaluation, providing strong support for the intelligent and precise development of sports training.

Current research mostly uses weighted averaging or simple concatenation, and the model structure is fixed, without optimizing for the temporal characteristics and complex action patterns of sports data. This paper combines wearable sensors with aerobics tracking and uses a model based on CNN and LSTM to achieve performance optimization. The main contributions of this

study include: 1) wavelet transform is combined and principal component analysis (PCA) to extract time-frequency features, and a dynamic weighted fusion strategy is adopted to improve the robustness of data fusion; 2) small convolutional kernels are introduced into CNN to capture action details and combined with double-layer LSTM to model long-term dependencies, enhancing the model's ability to recognize complex action sequences; 3) based on the model output, an action scoring function and error correction mechanism are constructed to provide athletes with immediate feedback and personalized training suggestions, improving the model's generalization ability in different training scenarios through data augmentation and adaptive filtering techniques.

2 Related work

In recent years, many scholars have been committed to improving the accuracy of athletes' motion evaluation through different technical means. Traditional motion capture systems [7] rely too much on calibration equipment and high-cost hardware settings. Although such capture systems can capture the movements of athletes, they suffer from problems such as poor real-time performance, high data noise, and inconvenient operation when evaluating high-intensity sports or complex movements. To improve the quality of sports data, many studies have attempted to use wearable sensors for motion tracking. Rigozzi C J et al. used data from sensors such as accelerometers, gyroscopes, and magnetometers to monitor athletes' body posture and motion trajectory [8]. Sensor data is easily affected by noise, environmental changes, and wear position deviation, resulting in inaccurate data. To reduce noise interference, Zhang Y applied Kalman filtering technology to the preprocessing of sensor data [9]. As technology matures, DL technology [10], especially CNNs, has been applied to multimodal data analysis and action recognition by Gholamiangonabadi D [11], and has achieved certain results. Existing research still faces problems such as how to combine multiple data sources, optimize data processing processes, and provide real-time feedback in actual training scenarios.

To solve the above problems, some researchers have proposed a hybrid method that combines sensor data and DL algorithms to improve the accuracy and real-time performance of action recognition. Chakraborty A used a CNN-based multimodal data fusion method to improve the accuracy and robustness of athlete action recognition by combining accelerometer, gyroscope, and visual data [12]. In this study, data fusion technology [13] effectively reduces sensor errors and enhances the system's adaptability to complex actions. Zhang L proposed a KCF (Kernelized Correlation Filters) tracking method based on improved depth information [14], which successfully used Kalman filtering to reduce the noise of motion sensors and improve the stability of motion estimation. Although these methods have achieved good results to a certain extent, most of them focus on a single motion estimation task, and their effects in complex training environments still need to be improved. Existing methods also have shortcomings in terms of personalized training feedback [15] and the generation of real-time optimization suggestions [16]. Therefore, how to comprehensively utilize multimodal data and combine DL with real-time optimization feedback systems is still a major challenge in current research.

3 Data fusion and movement performance optimization

3.1 Data collection and preprocessing

The study combines wearable sensors with motion tracking systems to design an efficient data collection and preprocessing solution. The key to the entire process is to synchronously collect data from multiple sources and eliminate errors, providing a reliable basis for subsequent analysis.

Wearable devices collect data in real-time through built-in accelerometers [17], gyroscopes [18], and magnetometers [19]. The accelerometer records the athlete's acceleration changes in three-dimensional space; the gyroscope measures the athlete's rotational angular velocity; the magnetometer helps correct the direction of movement. A multi-sensor system can accurately capture every movement of an athlete and generate rich time series data [20]. Sensor data fusion equation is:

$$f(t) = \alpha \cdot a(t) + \beta \cdot \omega(t) + \gamma \cdot m(t) \quad (1)$$

$a(t)$ is acceleration data; $\omega(t)$ is angular velocity data; $m(t)$ is magnetic field data; α, β, γ are weighting parameters. Figure 1 is a data acquisition flow chart.

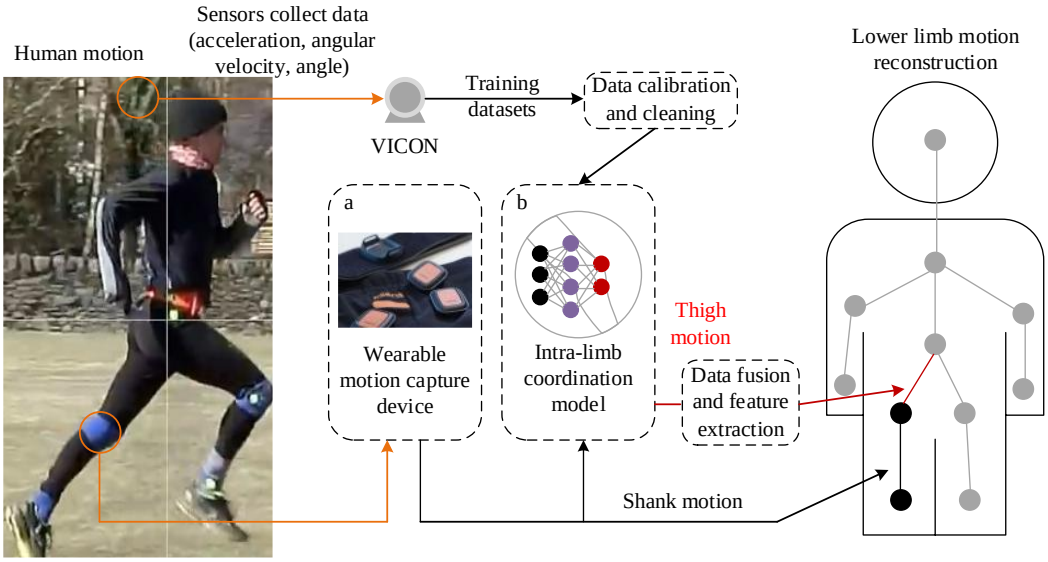


Figure 1: Data collection flow chart

Figure 1 shows the complete process from data collection to motion analysis. The motion data is obtained through wearable sensors and motion capture systems; key features are extracted through data cleaning and fusion; the CNN model is used to optimize the motion performance evaluation, ultimately providing scientific motion optimization and training suggestions for aerobics athletes. To ensure the accuracy of the data, all collected sensor data are transmitted to the background data processing system in real-time via Bluetooth or Wi-Fi modules [21] to ensure the real-time and efficient data. By adopting wireless data transmission [22], the data transmission is not affected by the physical distance, ensuring that the data is updated and recorded in time when the athletes perform complex movements. Bluetooth signal quality function is:

$$Q = \frac{1}{1 + \exp(-k(S - S_0))} \tag{2}$$

S is the signal strength; S_0 is the signal threshold; k is the Bluetooth signal adjustment parameter. To deal with data anomalies, the Kalman filter [23] is used to smooth the data of accelerometers and gyroscopes. The Kalman filter can dynamically predict the true value of the signal, optimize the measurement noise, and improve the accuracy of the data. Kalman filter formula is:

$$x_{k|k} = x_{k|k-1} + K_k(z_k - H_{x_{k|k-1}}) \tag{3}$$

K_k is the Kalman gain, and z_k is the observed value.

In addition to multi-sensor equipment, the motion tracking system Kinect [24] and depth camera [25] are also introduced to obtain the spatial position information of the key parts of the athletes. The system captures the athlete's action posture through 3D coordinates and records the spatial coordinates of joints such as shoulders, elbows, and knees, as well as their dynamic trajectories over time. Through the calibration algorithm, combined with the position information of the sensor and the motion tracking system, the effects caused by the wearer position offset or motion capture error are corrected. Motion trajectory smoothing formula is:

$$p(t) = \frac{1}{N}\sum_{i=1}^{N} p_i(t) \tag{4}$$

$p_i(t)$ is the spatial position of different sampling points. The system synchronizes the data of sensors and motion tracking systems to ensure that the sensor data and motion data at each moment can correspond correctly. After time synchronization, the data can be smoothly input into the subsequent data processing and analysis. Time synchronization function is:

$$\Delta T = T_{sensor} - T_{camera} \tag{5}$$

T_{sensor} and T_{camera} are the timestamps of the sensor and camera, respectively. Table 1 is the motion capture key point coordinate data table.

Table 1: Motion capture key point coordinate data table.

Timestamp (ms)	Shoulder X (cm)	Shoulder Y (cm)	Shoulder Z (cm)	Knee X (cm)	Knee Y (cm)	Knee Z (cm)
0	12.3	45.6	78.2	8.9	30.2	50.7
10	12.1	45.5	78.3	9	30.3	50.9
20	12.2	45.7	78.1	9.1	30.1	50.8
30	12.4	45.8	78.2	9.2	30.4	51
40	12.3	45.9	78.3	9.3	30.5	50.9
50	12.5	46	78.1	9.4	30.6	51.1
60	12.6	46.1	78.2	9.5	30.7	51.2

Table 1 records the three-dimensional spatial position data (X, Y, Z, in centimeters) of the shoulder and knee at different timestamps (in milliseconds). The data can be used to analyze the movement trajectory and change trend of the shoulder and knee in space.

3.2 Multimodal data fusion

After data collection and preprocessing, the multimodal data from wearable sensors and motion tracking systems are effectively fused. Different data sources provide different perspectives on the athlete's movements. Wearable sensor data provides time series information such as acceleration and angular velocity, and the motion tracking system provides spatial information such as joint position and motion trajectory. The effective integration of this information can help comprehensively evaluate the athlete's performance and provide accurate data input for the DL model.

Time synchronization [26] is a prerequisite for ensuring the effective integration of multimodal data. When collecting sensor data and motion tracking data, the data needs to be accurately aligned in time due to the different collection frequencies of the two [27]. To achieve data calibration, a timestamp is used to mark the acquisition time of each frame of data to ensure that each action frame can obtain corresponding sensor data and tracking data. After time synchronization, the sensor data at each moment is guaranteed to correspond perfectly with the action tracking data, providing a basis for data fusion. After synchronization correction, it can ensure that the action and sensor data at each moment correspond to each other, avoiding information loss caused by asynchrony [28].

The original sensor data contains rich time series information. Wavelet transform [29] is used to analyze the data in the time and frequency domain to extract motion features. Wavelet transform formula is:

$$W_{\psi}(a,b) = \int_{-\infty}^{\infty} f(t)\psi^{*}(\frac{t-b}{a})\,dt \tag{6}$$

ψ is the mother wavelet function. Wavelet transform can effectively capture the instantaneous changes in motion signals and use multi-scale analysis [30] to extract the time-frequency features of motion signals. It is integrated with the sensor data by calculating the spatial characteristics of the athlete's joint angle, motion trajectory, and speed. The formula for calculating the joint angle is:

$$\theta = arc\,cos(\frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|}) \tag{7}$$

v_1 and v_2 are the vectors of two bones. The motion trajectory curve fitting formula is:

$$r(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n \tag{8}$$

The spatial characteristics of the tracking system play a decisive role in the accuracy and coordination of the movements and are the core basis for evaluating the performance of athletes. Data fusion is the key to multimodal data processing. When fusion is performed, methods such as weighted fusion [31] and principal component analysis (PCA) [32] are used. Weighted fusion is to assign different weights to different data sources according to their signal-to-noise ratio and importance. When the noise of sensor data is large, its weight in fusion is reduced. On the contrary, if it is small, it means that the spatial data provided by the motion tracking system is relatively stable and can be assigned a higher weight. Weighted fusion formula is:

$$F_{fused} = \omega_1 F_1 + \omega_2 F_2 \tag{9}$$

F_1 and F_2 are the features of different data sources, and ω_1 and ω_2 are weights. Through weighted fusion processing, the fused data can more realistically reflect the athlete's performance. In weighted fusion, the quality of the signal is the key to weight assignment, and the relevance and accuracy of the data determine the contribution of each source information.

Principal component analysis is used to reduce the dimensionality of the data, compressing the multi-dimensional raw data into fewer principal components, reducing the redundancy of the data, and extracting the most representative features. PCA dimensionality reduction formula is:

$$X' = XW, W = arg\ \max_{W}\ (\frac{W^T \sum W}{W^T D W}) \tag{10}$$

Σ is the feature covariance matrix [33]; D is the weight matrix [34]; W is the optimized feature matrix. The analysis process reduces the computational complexity and retains the key information in the data, providing more efficient input for the training of DL models. Through the dimensionality reduction of PCA, redundant dimensions and noise can be eliminated, improving the efficiency and accuracy of subsequent analysis.

After fusion processing of multimodal data, data from different sources is integrated into a unified format, providing rich and accurate input features for subsequent action evaluation. The fused data contains both time series information and spatial position information, which can fully and accurately reflect the athlete's action performance in training [35]. Combined with efficient data synchronization, feature extraction, and fusion processing, sufficient high-quality data support is provided for the CNN, ensuring that the model can make full use of various types of information for accurate evaluation.

3.3 Action performance evaluation based on CNN and LSTM

This paper studies the evaluation of action performance of fused data based on CNN and LSTM. The CNN model has an advantage in processing time series data and spatial data, while LSTM is good at capturing time series dependencies, especially in capturing subtle differences in athletes' movements and automatically extracting features. The CNN model can more comprehensively evaluate the athletes' movement performance and achieve end-to-end automated processing from raw sensor and tracking data to final movement scoring and classification. Through LSTM, the model can understand the continuity between actions and evaluate actions based on the relationship

between the action sequence. When capturing complex motion patterns, LSTM can supplement the timing information that the CNN model fails to fully capture, providing a more detailed motion performance evaluation. In the motion performance evaluation task, the combination of LSTM and CNN enables the model to obtain more comprehensive feature extraction in both spatial and temporal dimensions. After CNN extracts spatiotemporal features, LSTM processes these features in time series, and the combination can more accurately evaluate the quality and type of actions. Figure 2 is a diagram of the CNN model structure.

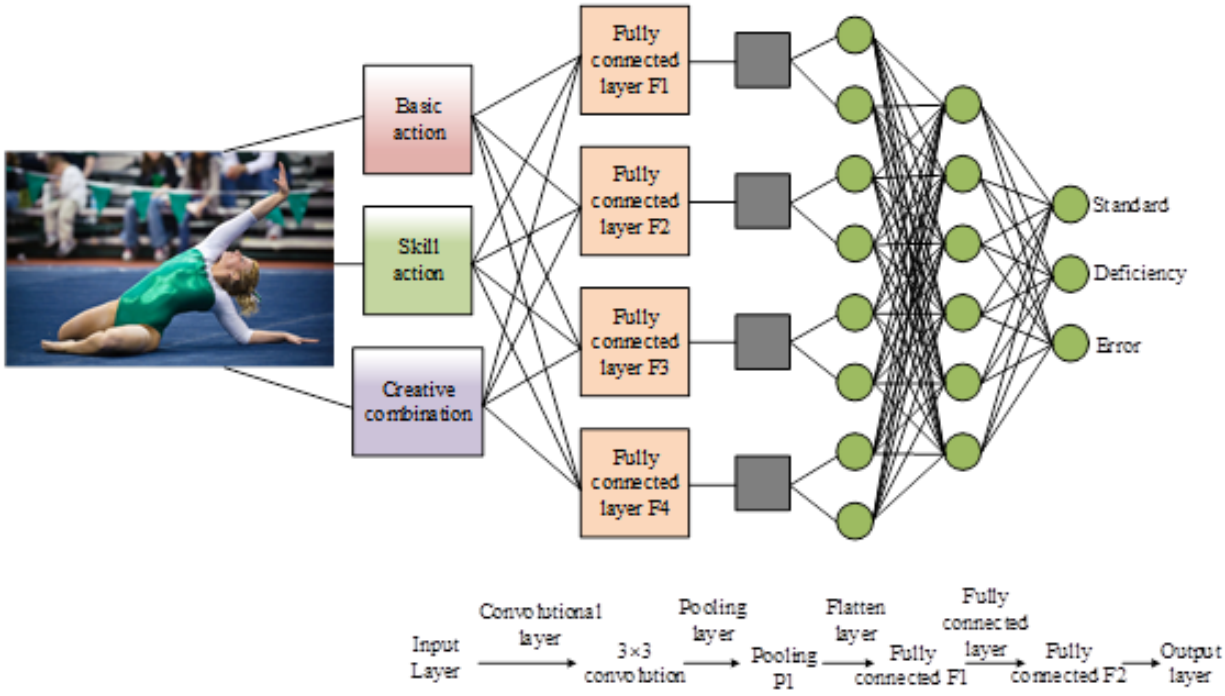


Figure 2: CNN model structure diagram

Figure 2 shows how the spatiotemporal features of the athlete's movements are gradually extracted through the convolutional layer, pooling layer, and fully connected layer, and converted into a one-dimensional vector for classification and scoring through the flattening layer. The output layer evaluates the athlete's movement quality based on the features learned by the model, achieving automated and efficient movement quality recognition and feedback.

The CNN model adopts a five-layer structure, with input data dimensions of (T, F). Among them, T=100 represents the time step; F=9 represents the input feature dimension (including data from three axes each of accelerometer, gyroscope, and magnetometer); the output dimension is (C), where C=3 represents the action category (standard, insufficient, error); the activation function is Softmax. The LSTM model is used to model the long-term dependencies of time series, with input dimensions of (T, D), where T=100 represents the time step, and D=9 represents the feature dimension. The LSTM layer contains 128 hidden units and uses a double-layer stacking structure with an activation function of Tanh. The output dimension is (C), C=3, and the activation function is Softmax. The output feature vectors of CNN and LSTM are merged through concatenation operation and input into a fused fully connected layer, ultimately outputting action scores and classification results. The model parameter settings are shown in Table 2:

Table 2: Model parameter settings

Parameter	Specifications	Parameter	Specifications
Learning rate	0.001	Dropout rate	0.5
Batch size	64	Training epochs	100
Optimizer	Adam	Hidden layer size	128
Loss function	Cross entropy loss function	Convolutional kernel size	(3, 3)

The convolution operation can effectively capture the spatial features and temporal dynamic changes of the athlete's joint movement trajectory, body posture changes, etc. The features processed by CNN can be passed to the LSTM network, which further analyzes the timing information of these features. Through the time memory mechanism of LSTM, the network can learn the continuity and long-term dependencies in the action. Convolution operation formula is:

$$f_{i,j} = \sum_{m=-k}^{k} \sum_{n=-k}^{k} x_{i+m,j+n} \cdot w_{m,n} \tag{11}$$

x is the input feature map; w is the convolution kernel; the choice of the convolution kernel is closely related to the characteristics of the data. Smaller convolution kernels help capture subtle changes, while larger convolution kernels can extract more macro features. It is necessary to adjust the bodybuilder's tiny movements and postures, and smaller convolution kernels help extract details more accurately, so a convolution kernel of size 3×3 is selected. The LSTM layer filters out irrelevant temporal information through its forget gate, input gate, and output gate, retaining the long-term and short-term dependency information related to the action performance evaluation.

After the convolution layer, the maximum pooling and average pooling [36] are used to reduce the dimension of the feature map. The role of the pooling layer is to reduce the amount of data after the convolution operation, reduce the computational complexity, and retain the most important feature information. Pooling operation formula is:

$$p_{i,j} = \underset{m,n}{max} (g_{i+m,j+n}) \tag{12}$$

In the fully connected layer, the temporal information generated by LSTM and the spatial features extracted by CNN are integrated through the fully connected layer to generate the final score and classification results of the action performance. The model can not only evaluate the quality of actions, but also classify actions into multiple categories such as "standard", "deficient", and "error". Action scoring function is:

$$S = \sum_{i=1}^{N} \varphi_i h_i \tag{13}$$

h_i is the score of each feature, and φ_i is the weight. The action evaluation score reflects the accuracy, fluency, and standardization of the athlete's action. The classification results provide coaches and athletes with targeted training improvement directions. Each classification result helps to further guide athletes' specific improvement measures in training.

During the training process, the Adam optimization algorithm [37] is used to update parameters. The optimization algorithm update formula is:

$$\theta_{t+1} = \theta_t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \tag{14}$$

$\widehat{m}_t$ and $\widehat{v}_t$ are momentum estimates. The cross-entropy loss function [38] is used to optimize the classification task, so that the model can better handle multi-classification problems and continuously improve the accuracy of action scoring and classification by minimizing the loss function. Classification loss function is:

$$L = -\sum_{i=1}^{C} y_i \, log(\hat{y}_i) \tag{15}$$

y_i is the true label, and $\hat{y}_i$ is the predicted probability. To improve the model's training effect, data enhancement technology is used to simulate the motion performance in different training scenarios, expand the training data set, and increase the robustness of the model. Data enhancement methods include operations such as rotation, mirror flipping, and scaling. More training samples are generated through enhancement operations, so that the model can still show excellent performance in different motion modes.

3.4 Real-time feedback and optimization suggestion generation

To improve the effect of aerobics training, a real-time feedback system is designed to feed back the evaluation results generated during the training process to athletes, helping them adjust their movements, avoid errors and optimize the quality of movements. The key to the feedback system lies in real-time and accuracy. Only timely and accurate feedback can effectively improve the training level of athletes.

The system analyzes the real-time action data of athletes through a model combining a trained CNN model and LSTM to generate more accurate action scores and evaluation results. Every time an athlete performs an action, the system immediately analyzes the action and outputs a real-time score. The score reflects the accuracy, fluency, and completion of the action. The higher the score, the more standard the action. For low-scoring actions, the system automatically identifies the errors and provides specific optimization suggestions. Error correction weight formula is:

$$\varrho_{error} = \frac{1}{1 + exp(-\kappa \cdot \delta)} \tag{16}$$

δ is the margin of error. Based on the score and error recognition results, the system generates personalized optimization suggestions. Optimization suggestion generation function is:

$$G = arg \underset{i}{max} (S_i + \lambda \cdot E_i) \tag{17}$$

S_i is the score, and E_i is the severity of the error. Suggestions include improving posture, adjusting the range of motion, strengthening muscle control, etc., to help athletes correct deficiencies in their movements. Optimization suggestions can be provided in text form or visualized through a graphical interface to help athletes more intuitively understand the problems in their movements.

To ensure timely feedback, the system accelerates the calculation process by optimizing the algorithm, controlling the delay between action scoring and feedback generation to less than 200ms, ensuring that athletes can receive targeted adjustment suggestions in a short period of time. Real-time feedback delay formula is:

$$T_{delay} = T_{process} + T_{transmit} \tag{18}$$

Through the real-time feedback mechanism, athletes can continuously adjust their movements during training and gradually improve the training effect. Optimization suggestions are not limited to correcting mistakes, but can also help athletes improve the delicacy and accuracy of their movements. The athlete improvement index formula is:

$$I = \frac{\Delta S \cdot ln(1 + A_0)}{\Delta T \cdot (1 + e^{-\varepsilon(\Delta S - \Delta S_{avg})})} \quad (19)$$

ΔS is the score increment; ΔT is the time; A_0 is the athlete's baseline ability; ε is the adjustment parameter; ΔS_{avg} is the average score increment. Through long-term training and optimization feedback, athletes can improve their overall performance in a short period of time and achieve the best training effect. Long-term optimization trend equation is:

$$O(t) = O_0 + \int_0^t \frac{dS}{d\tau} d\tau \quad (20)$$

O_0 is the initial performance. Combining DL with real-time feedback technology, it provides athletes with an intelligent training platform that can effectively improve training efficiency and quality. Table 3 is some hyperparameter data of the experiment.

Table 3: Some hyperparameter data

Parameters	Function	Parameters	Function
α, β, γ	Sensor data fusion weighting parameters	k	Bluetooth signal conditioning parameters
K_k	Kalman Gain	ψ	Mother wavelet function
ω_1, ω_2	Weighted fusion weight	Σ	Feature covariance matrix
w	Convolution Kernel	φ_i	Action score weight
$\hat{m}_t$, $\hat{v}_t$	Momentum Estimation	$\hat{y}_i$	Prediction probability
δ	Margin of Error	ε	Adjustment parameters

Table 3 lists the hyperparameters used for model and signal processing and their functional descriptions. These hyperparameters play a key role in sensor data fusion, signal conditioning, feature extraction, and prediction, and can effectively improve the performance and accuracy of the model. Adjusting these parameters can optimize system behavior according to specific application requirements and achieve more accurate data processing and action recognition.

4 Experimental results

4.1 Experimental setup

This paper collects a total of 12000 action samples, covering three categories of actions: standard actions (4000), insufficient actions (4000), and incorrect actions (4000). Each sample contains multimodal data of 100 time steps, including 9 channels (3-axis x 3 sensors) from accelerometers, gyroscopes, magnetometers, and 18 joint point 3D coordinate information obtained by Kinect. The data collection frequency is 50 Hz, which means collecting 50 frames per second. The wearable sensor is the Xsens MTw Awinda series inertial measurement unit (IMU), with specific parameters shown in Table 4:

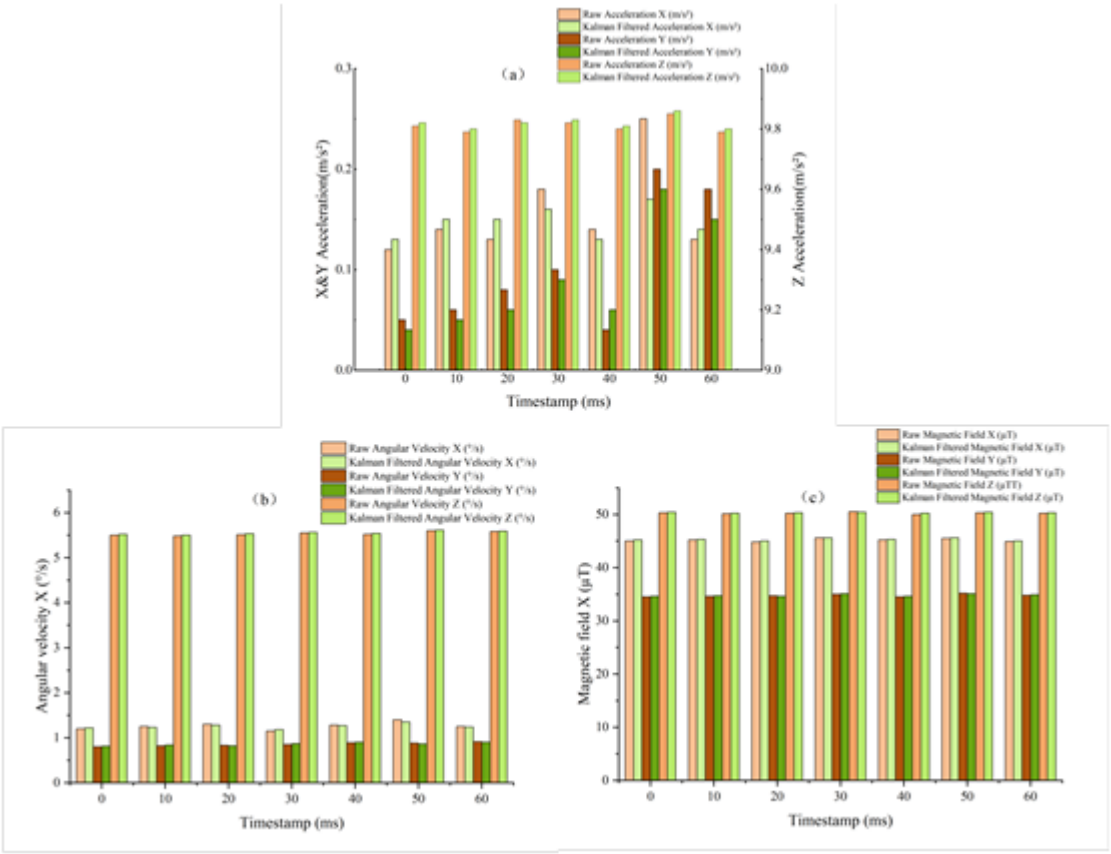
Table 4: Sensor parameters

Sensor	Range	Resolution	Sampling rate
Accelerometer	±16g	0.001g	50 Hz
Gyroscope	±2000°/s	0.01°/s	50 Hz
Magnetometer	±2.5 Gauss	0.01 Gauss	50 Hz

The data collection is conducted in an indoor sports arena, with an ambient temperature controlled at 22-25 ° C, humidity at 45-60%, and good and stable lighting conditions. All model training is completed under the PyTorch 1.13.1 deep learning framework, with a training time of approximately 4 hours per model.

4.2 Effect of Kalman Filtering on sensor data

In sensor data processing, the original signal is easily affected by noise, which affects the accuracy of the data. Kalman filtering, as a common noise suppression method, can effectively improve the stability and accuracy of the data by correcting the measured values. Figure 3 shows the comparison between the original data of the accelerometer, gyroscope, and magnetometer and the data after Kalman filtering, which is used to evaluate the filtering effect.



(a) Accelerometer data comparison; (b) Gyroscope data comparison; (c) Magnetometer data comparison

Figure 3: Kalman filter effect

Figure 3 shows the fluctuation of different sensors in different directions and the effect after filtering. In the accelerometer data, after Kalman filtering, the peak-to-valley difference of acceleration X drops from the original 0.13m/s² to 0.04m/s² after filtering, showing a more stable state. The original acceleration fluctuates greatly in the X-axis and Y-axis directions. After Kalman filtering, the fluctuation of the data is more stable, indicating that Kalman filtering can effectively reduce the interference of noise. When the timestamp is 0ms, the original acceleration X is 0.12m/s², and after Kalman filtering, it is 0.13m/s², with little change. The gyroscope data also shows a similar trend. The original data fluctuates to varying degrees in the X, Y, and Z directions, while the angular velocity data after Kalman filtering has less fluctuation. After filtering, the peak-to-valley difference in the angular velocity in the X direction is reduced from the original 0.25°/s to the filtered 0.17°/s, ensuring more accurate angle measurement. At 0ms, the original angular velocity Z is 5.50°/s, and after Kalman filtering it is 5.52°/s. The magnetometer data also shows small fluctuations. After filtering, the fluctuations of the three-axis magnetic field are further reduced. The peak-to-valley difference in the X direction drops from the original 0.8μT to 0.6μT after filtering, providing more stable environmental data. At 0ms, the original magnetic field X is 45.0μT, and after Kalman filtering, it is 45.2μT. The data optimized by Kalman filtering, combined with the analysis and processing of the CNN model, can effectively improve the accuracy of motion tracking and evaluation, and provide athletes with more accurate real-time feedback and personalized training suggestions.

4.1.Action scoring effect

To comprehensively evaluate the performance of aerobics athletes, a scoring system based on sensor data is introduced. The score value of each sensor at different time points reflects the quality and stability of the athlete's movements. The comprehensive score combines the average score of these three scores to provide a comprehensive performance evaluation for bodybuilders, helping athletes and coaches to grasp the effect of exercise in real-time. Figure 4 shows the scoring performance at different time points.

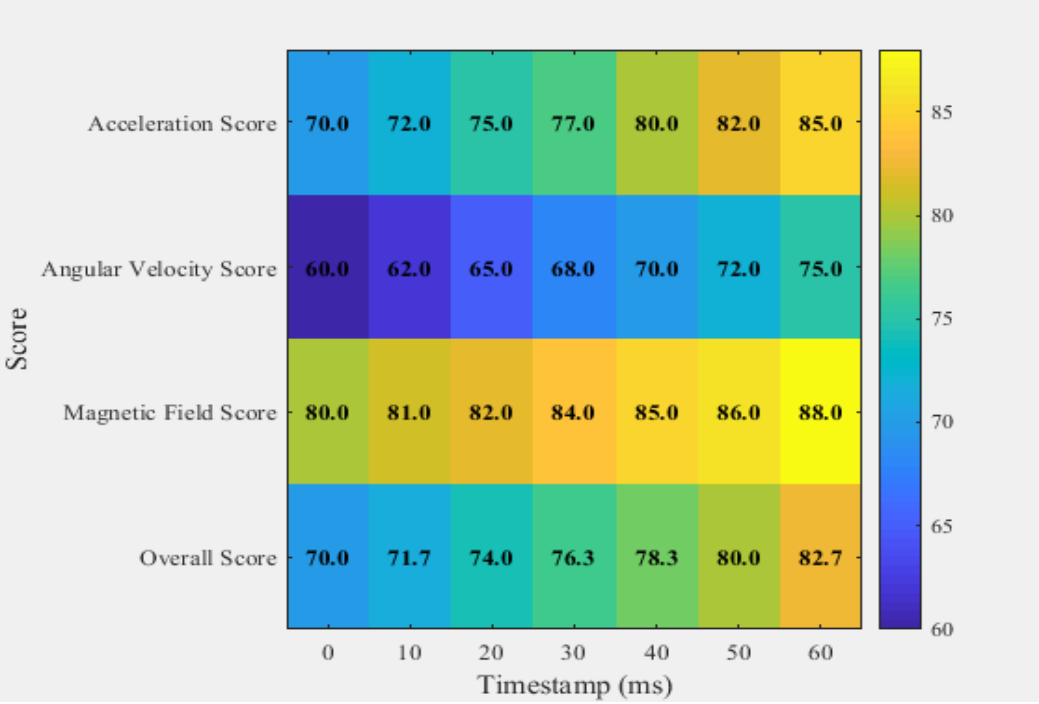


Figure 4: Sensor scores at different time points

In the experiment, wearable sensors and motion tracking technology are combined to fuse and analyze multimodal data such as acceleration, angular velocity, and magnetic field using the CNN model and LSTM to achieve accurate evaluation and optimization of aerobics movements. The data in Figure 4 shows that as time goes by, the athletes' acceleration scores gradually increase from 70 to 85 points; the angular velocity scores increase from 60 to 75; the magnetic field scores also maintain a relatively stable upward trend. The final comprehensive score increases from 70.0 to 82.7. The data changes reflect the gradual optimization of the athletes' performance during training. Figure 4 shows the changes in different scoring dimensions at each time point, helping trainers to accurately monitor and adjust training strategies in real-time. CNN optimizes sensor data by combining Kalman filtering and wavelet transform technology to provide athletes with more accurate performance feedback and promote the improvement of training results.

4.3 Performance of feedback systems in different scenarios

The performance improvement before and after training can reflect the optimization effect of the system. The following Table 5 shows the relationship between feedback delay, optimization suggestion generation time, and athlete performance improvement in different training scenarios.

Table 5: Feedback performance in different training scenarios

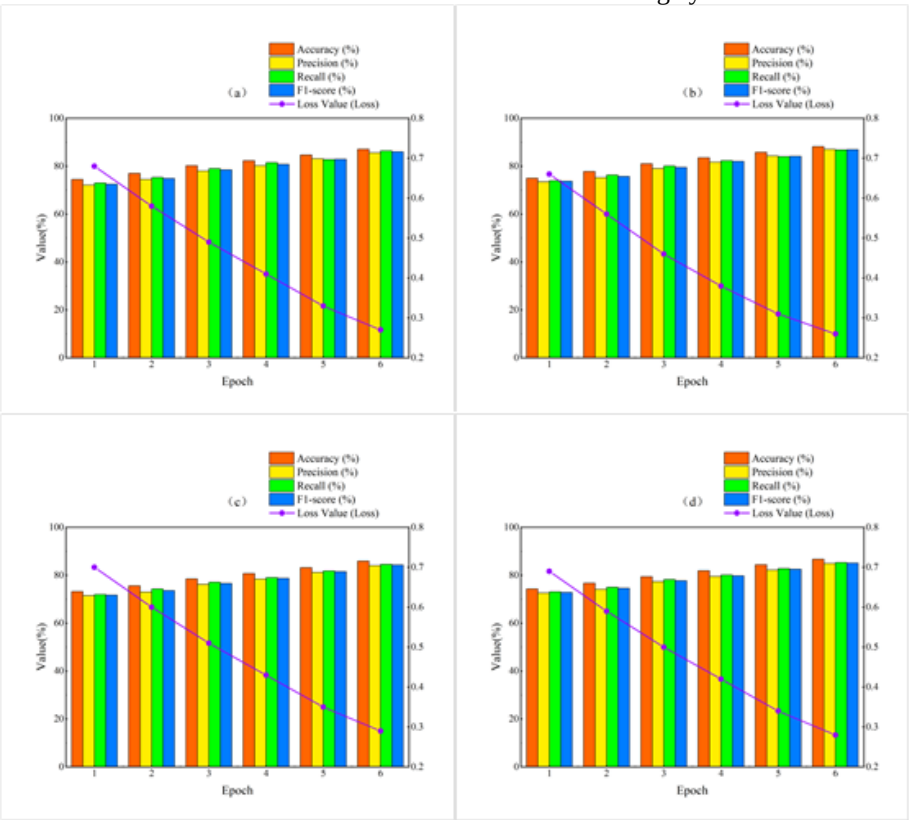
Scenario Type	Average Feedback Delay (ms)	Optimization Suggestion Generation Time (ms)	Pre-training Performance Score (out of 100)	Post-training Performance Score (out of 100)	Improvement Rate (%)
Indoor Low Intensity	150	90	65	80	23.1
Indoor High Intensity	180	95	62	78	25.8
Outdoor Low Intensity	160	85	68	82	20.6
Outdoor High Intensity	200	110	60	75	25

According to the data in Table 5, there are certain differences in feedback delay and optimization suggestion generation time in different training scenarios. In the indoor high-intensity training scenario, the average feedback delay is 180 milliseconds; the optimization suggestion generation time is 95 milliseconds; the action score before training is 62 points. After training, the score increases to 78 points, and the improvement rate reaches 25.8%, which is the highest improvement rate in all scenarios. This shows that under high-intensity training,

despite the longer feedback delay, the system is able to more effectively generate optimization suggestions and improve athlete performance. In outdoor low-intensity training scenarios, the feedback delay is shorter, at 160 milliseconds, but the improvement rate is 20.6%, which is relatively low, indicating that the feedback system in this scenario has room for improvement in the generation and application of optimization suggestions. The efficiency and optimization effect of the feedback system vary in different training scenarios. The response speed of the system and the time to generate suggestions are closely related to the performance improvement of athletes.

4.4 CNN model training effect

To better evaluate and optimize the performance of athletes in different training scenarios, the experiment uses CNN model and LSTM to conduct in-depth analysis of various training data. In the four training scenarios of indoor low intensity, indoor high intensity, outdoor low intensity, and outdoor high intensity, the change of training cycle has an important impact on the accuracy and performance of the model. The changes in the accuracy, precision, recall, F1-score, and loss value of the CNN model in different scenarios can be analyzed to understand the model optimization trend during the training process. Figure 5 shows the changes in key indicators of the model after each training cycle in these scenarios.



(a) Indoor low-intensity training scene; (b) Indoor high-intensity training scene; (c) Outdoor low-intensity training scene; (d) Outdoor high-intensity training scene

Figure 5: CNN effects in different scenes

By analyzing the model effects in different training scenarios through the data in Figure 5, the performance of the CNN model combined with LSTM in various indicators is significantly improved with the increase of training cycles. In the indoor low-intensity training scenario, the accuracy rate increases from 74.5% to 87.1%; the precision, recall rate, and F1-score also increase steadily, and the loss value decreases from 0.68 to 0.27, indicating that the model's ability to fit the data steadily improves with the passage of training time. In indoor high-intensity scenarios, the accuracy rate increases from 75.0% to 88.2%. The improvement in precision and recall rate shows that the model can effectively handle more complex training environments, and the loss value drops to 0.26. The training effect of the model in outdoor low-intensity and high-intensity scenarios also shows a similar trend, with the accuracy rate increased from 73.2% to 85.9% and 74.3% to 86.7%, respectively, and the loss value decreased significantly. Whether it is a low-intensity or high-intensity scenario, the model shows good stability and accuracy in different environments. With the increase of training cycles, the performance of the CNN model in motion scoring and classification has been effectively improved, and the training error can be significantly reduced.

To further verify the effectiveness of the method proposed in this paper in the evaluation of aerobics movements, the CNN-LSTM hybrid model is compared with several representative studies in recent years. The comparative methods include: multimodal fusion method

based on traditional CNN, action recognition method based on LSTM, traditional classification method based on support vector machine (SVM), and temporal modeling method based on Transformer. Comparative experiments are conducted on the same dataset, with evaluation metrics including accuracy and F1 score, as well as action evaluation RMSE (Root Mean Squared Error) value, as shown in Table 6:

Table 6: Comparison of the performance of this method with existing research

Model	Accuracy	F1-score	RMSE
SVM	72.1%	0.703	8.1
CNN	83.2%	0.817	6.0
LSTM	84.6%	0.832	5.8
Transformer	85.4%	0.841	5.6
CNN-LSTM	88.2%	0.867	4.2

From Table 6, it can be seen that this method outperforms existing methods in terms of accuracy and F1 score. Compared with traditional CNN methods, this method has improved accuracy by 5.0%; compared with the LSTM method, it has improved by 3.6%; compared with the Transformer method, it has improved by 2.8%. The RMSE of the method proposed in this paper is 4.2, significantly lower than other comparative methods, indicating that the CNN-LSTM hybrid model proposed in this paper has higher accuracy and stability in predicting action scores. This result represents that the method proposed in this paper can effectively achieve accurate evaluation of athletes' movements and provide objective guidance for scientific training.

5 Experimental discussion

By combining wearable sensors with motion tracking technology and using CNNs for multimodal data fusion and analysis, this study has achieved significant experimental results in motion evaluation and optimization. Kalman filtering significantly improves the stability and accuracy of sensor data in the application of noise suppression, and effectively reduces the impact of environmental interference on data quality. With the increase of training cycles, the CNN model combined with LSTM continues to improve in terms of accuracy, precision, recall rate, and F1-score in action scoring and classification, showing good fitting ability. In high-intensity training scenarios, despite relatively long feedback delays, the system can still effectively generate optimization suggestions, and the athletes' performance has been significantly improved.

However, the experimental results also reveal that there are still some potential challenges and problems in the application and development of the system. The performance of the model has been significantly improved in different training scenarios, but the feedback system has a long delay and optimization suggestion generation time in high-intensity training scenarios, which affects the system's real-time response capability. Kalman filtering and other data optimization techniques effectively reduce noise, but in complex or extreme training environments, external interference still poses a risk of affecting the accuracy of sensor data and causing certain errors in training feedback. With the diversification of training scenarios and the increase in environmental complexity, how to further improve the model's ability to classify complex actions and its ability to comprehensively process multimodal data remains an issue to be resolved. Future research can focus on optimizing sensor data collection and processing technology, strengthening data synchronization and fusion algorithms, and further improving the stability and adaptability of CNN models in different environments. With the advancement of technology, combined with more sensors and analysis of training scenarios, it can be possible to provide athletes with more detailed and comprehensive personalized training programs, promoting the intelligent and precise development of sports training.

6 Conclusions

This paper proposes a fitness exercise action evaluation method that integrates wearable sensors and motion tracking systems, and combines CNN and LSTM models to fuse and analyze multimodal data. The experimental results show that the method exhibits excellent performance in action classification tasks: in indoor low-intensity training scenarios, the accuracy increases from 74.5% to 87.1%; in high-intensity training scenarios, the accuracy increases from 75.0% to 88.2%. By introducing Kalman filtering, wavelet transform, and dynamic weighting fusion strategy, the stability of sensor data and the generalization ability of the model have been effectively improved. This paper not only provides high-precision motion evaluation and real-time feedback for aerobics athletes, but also provides a transferable technical framework for other high-intensity, multimodal motion sports projects. In the future, the system can be further expanded to remote training platforms, intelligent wearable devices, virtual coaching systems, and other application scenarios, promoting the deep integration and widespread application of artificial intelligence technology in the fields of sports training and health management. In the future, lightweight CNN structures such as MobileNet and TinyML, deployment of models, on wearable devices, and heterogeneous computing acceleration can be used to further shorten feedback latency and improve system response speed.

Authorship contribution statement

Danhua Tan: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Data availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] M. E. M. Simbolon, D. K. A. Firdausi, I. Dwisaputra, A. Rusdiana, C. Pebriandani, and R. Prayoga, "Utilization of Sensor technology as a Sport Technology Innovation in Athlete Performance Measurement," *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, 13(2): 147–158, 2023. https://doi.org/10.22146/ijeis.89581

[2] Z. Mei, "3D images analysis of sports technical features and sports training methods based on artificial intelligence," *J Test Eval*, 51(1): 189–200, 2023. https://doi.org/10.1520/JTE20210469

[3] S. A. Kovalchik, "Player tracking data in sports," *Annu Rev Stat Appl*, 10(1): 677–697, 2023. https://doi.org/10.1146/annurev-statistics-033021-110117

[4] L. Yang, O. Amin, and B. Shihada, "Intelligent wearable systems: Opportunities and challenges in health and sports," *ACM Comput Surv*, 56(7):1–42, 2024. https://doi.org/10.1145/3648469

[5] W. Li, "Application of IoT-enabled computing technology for designing sports technical action characteristic model," *Soft comput*, 27(17): 12807–12824, 2023. https://doi.org/10.1007/s00500-023-08966-4

[6] Y. Fang, "Utilizing Wearable Technology to Enhance Training and Performance Monitoring in Indonesian Badminton Players," *Studies in Sports Science and Physical Education*, 2(1): 11–23, 2024. DOI:10.1186/s40561-023-00247-9

[7] J. Corban *et al.*, "Using an affordable motion capture system to evaluate the prognostic value of drop vertical jump parameters for noncontact ACL injury," *Am J Sports Med*, 51(4):1059–1066, 2023. https://doi.org/10.1177/03635465231151686

[8] C. J. Rigozzi, G. A. Vio, and P. Poronnik, "Application of wearable technologies for player motion analysis in racket sports: A systematic review," *Int J Sports Sci Coach*, 18(6): 2321–2346, 2023. https://doi.org/10.1177/17479541221138015

[9] Y. Zhang, "Design of Wireless Motion Sensor Nodes based on the Kalman Filter Algorithm," *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, 16(3): 248–255, 2023. https://doi.org/10.2174/2352096515666220908152036

[10] S. Akan and S. Varlı, "Use of deep learning in soccer videos analysis: survey," *Multimed Syst*, 29(3): 897–915, 2023. https://doi.org/10.1007/s00530-022-01027-0

[11] D. Gholamiangonabadi and K. Grolinger, "Personalized models for human activity recognition with wearable sensors: deep neural networks and signal processing," *Applied Intelligence*, 53(5): 6041–6061, 2023. https://doi.org/10.1007/s10489-022-03832-6

[12] A. Chakraborty and N. Mukherjee, "A deep-CNN based low-cost, multi-modal sensing system for efficient walking activity identification," *Multimed Tools Appl*, 82(11): 16741–16766, 2023. https://doi.org/10.1007/s11042-022-13990-x

[13] W. Liu, Y. Liu, and R. Bucknall, "Filtering based multi-sensor data fusion algorithm for a reliable unmanned surface vehicle navigation," *Journal of Marine Engineering & Technology*, 22(2): 67–83, 2023. https://doi.org/10.1080/20464177.2022.2031558

[14] L. Zhang and H. Dai, "Motion trajectory tracking of athletes with improved depth information-based KCF tracking method," *Multimed Tools Appl*, 82(17): 26481–26493, 2023. https://doi.org/10.1007/s11042-023-14929-6

[15] P. Hao and K. Qian, "The Integration of Personalized Training Program Design and Information Technology for Athletes," *Scalable Computing: Practice and Experience*, 25(5): 4351–4359, 2024. https://doi.org/10.12694/scpe.v25i5.3083

[16] V. Deepak, D. K. Anguraj, and S. S. Mantha, "An efficient recommendation system for athletic performance optimization by enriched grey wolf optimization," *Pers Ubiquitous Comput*, 27(3): 1015–1026, 2023. https://doi.org/10.1007/s00779-022-01680-2

[17] J. K. Urbanek *et al.*, "Free-living gait cadence measured by wearable accelerometer: a promising alternative to traditional measures of mobility for assessing fall risk," *The Journals of Gerontology: Series A*, 78(5): 802–810, 2023. https://doi.org/10.1093/gerona/glac013

[18] A. Hussain, S. Ali, M.-I. Joo, and H.-C. Kim, "A deep learning approach for detecting and classifying cat activity to monitor and improve cat's well-being using accelerometer, gyroscope, and magnetometer," *IEEE Sens J*, 24(2): 1996–2008, 2023.

[19] A. Spilz and M. Munz, "Synchronisation of wearable inertial measurement units based on magnetometer data," *Biomedical Engineering/Biomedizinische Technik*, 68(3): 263–273, 2023. https://doi.org/10.1515/bmt-2021-0329

[20] A. Liu, R. P. Mahapatra, and A. V. R. Mayuri, "Hybrid design for sports data visualization using AI and big data analytics," *Complex & Intelligent Systems*, 9(3): 2969–2980, 2023. https://doi.org/10.1007/s40747-021-00557-w

[21] C.-T. Lin, Y. Wang, S.-F. Chen, K.-C. Huang, and L.-D. Liao, "Design and verification of a wearable wireless 64-channel high-resolution EEG acquisition system with wi-fi transmission," *Med Biol Eng Comput*, 61(11): 3003–3019, 2023. https://doi.org/10.1007/s11517-023-02879-y

[22] X. Shi and H. Zou, "Data Collection and Analysis based on Sensor Technology in Sports Training," *Scalable Computing: Practice and Experience*, 25(5): 4399–4406, 2024. https://doi.org/10.12694/scpe.v25i5.3200

[23] M. Khodarahmi and V. Maihami, "A review on Kalman filter models," *Archives of Computational Methods in Engineering*, 30(1): 727–747, 2023. https://doi.org/10.1007/s11831-022-09815-7

[24] M. Azhar, S. Ullah, M. Raees, K. U. Rahman, and I. U. Rehman, "A real-time multi view gait-based automatic gender classification system using kinect sensor," *Multimed Tools Appl*, 82(8): 11993–12016, 2023. https://doi.org/10.1007/s11042-022-13704-3

[25] L. Lv, J. Yang, F. Gu, J. Fan, Q. Zhu, and X. Liu, "Validity and reliability of a depth camera–based quantitative measurement for joint motion of the hand," *J Hand Surg Glob Online*, 5(1): 39–47, 2023. https://doi.org/10.1016/j.jhsg.2022.08.011

[26] Y. Wu, Z. Sun, G. Ran, and L. Xue, "Intermittent control for fixed-time synchronization of coupled networks," *IEEE/CAA Journal of Automatica Sinica*, 10(6): 1488–1490, 2023. DOI: 10.1109/JAS.2023.123363

[27] Hrovatin, N. "Enabling Decentralized Privacy Preserving Data Processing in Sensor Networks," Informatica (03505596), 48(1): 141-142, 2024. https://doi.org/:10.31449/inf.v48i1.5739.

[28] Thi H N, Duc C V, Duc C T, HH Minh, SN Van, LV Quan. "Memetic Algorithm for Maximizing K-coverage and K-Connectivity in Wireless Sensor Network," Informatica, (03505596), 49(1): 1-7, 2025. https://doi.org/:10.31449/inf.v49i1.6750.

[29] A. Halidou, Y. Mohamadou, A. A. A. Ari, and E. J. G. Zacko, "Review of wavelet denoising algorithms," *Multimed Tools Appl*, 82(27): 41539–41569, 2023. https://doi.org/10.1007/s11042-023-15127-0

[30] M. Kang, C. L. Bentley, J. T. Mefford, W. C. Chueh, and P. R. Unwin, "Multiscale Analysis of Electrocatalytic Particle Activities: Linking Nanoscale Measurements and Ensemble Behavior," *ACS Nano*, 17(21): 21493–21505, 2023. https://doi.org/10.1021/acsnano.3c06335

[31] J. Sun, H. Zhang, X. Ma, R. Wang, H. Sima, and J. Wang, "Spectral–Spatial Adaptive Weighted Fusion and Residual Dense Network for hyperspectral image classification," *The Egyptian Journal of Remote Sensing and Space Sciences*, 28(1): 21–33, 2025. https://doi.org/10.1016/j.ejrs.2024.11.001

[32] J. P. Bharadiya, "A tutorial on principal component analysis for dimensionality reduction in machine learning," *Int J Innov Sci Res Technol*, 8(5): 2028–2032, 2023. DOI:10.5281/zenodo.8002436

[33] F. Bizzarri, D. Del Giudice, S. Grillo, D. Linaro, A. Brambilla, and F. Milano, "Inertia estimation through covariance matrix," *IEEE Transactions on Power Systems*, 39(1): 947–956, 2023. DOI: 10.1109/TPWRS.2023.3236059

[34] Y. He, C.-K. Zhang, H.-B. Zeng, and M. Wu, "Additional functions of variable-augmented-based free-weighting matrices and application to systems with time-varying delay," *Int J Syst Sci*, 54(5): 991–1003, 2023. https://doi.org/10.1080/00207721.2022.2157198

[35] Singh S, Sehgal V K. "Deep Learning-Based CNN Multi-Modal Camera Model Identification for Video Source Identification," Informatica: An International Journal of Computing and Informatics, 47(3): 417-430, 2023. https://doi.org/10.31449/inf.v47i3.4392

[36] T. Sharma, N. K. Verma, and S. Masood, "Mixed fuzzy pooling in convolutional neural networks for image classification," *Multimed Tools Appl*, 82(6): 8405–8421, 2023. https://doi.org/10.1007/s11042-022-13553-0

[37] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Comput Appl*, 35(23): 17095–17112, 2023. https://doi.org/10.1007/s00521-023-08568-z

[38] Z. Mei *et al.*, "Automatic loss function search for adversarial unsupervised domain adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, 33(10): 5868–5881, 2023. DOI: 10.1109/TCSVT.2023.3260246

Metaheuristic-Enhanced SVR Models for California Bearing Ratio Prediction in Geotechnical Engineering

Yulin Lan[1], Na Feng [2,*] and Zhisheng Yang[3]
[1]Planning and Finance Department, Weifang Engineering Vocational College, Weifang 262500, Shandong, China
[2]School of Information Engineering, Weifang Engineering Vocational College, Weifang 262500, Shandong, China
[3]Party and Government Office, Weifang Engineering Vocational College, Weifang 262500, Shandong, China
E-mail: sdqzyuchen@163.com
[*]Corresponding author

Keywords: california bearing ratio, support vector regression, adaptive opposition slime mold algorithm, alibaba and the forty thieves optimization algorithm, dingo optimization algorithm

Received: April 7, 2025

Soil resistance characteristics, particularly the California Bearing Ratio (CBR), play a pivotal role in pavement and subgrade design. However, conventional laboratory-based CBR testing is often time-consuming, labor-intensive, and costly. This study presents a novel machine learning framework that combines Support Vector Regression (SVR) with three recent metaheuristic optimization algorithms—Dingo Optimization Algorithm (DOA), Alibaba and the Forty Thieves Optimization (AFT), and Adaptive Opposition Slime Mold Algorithm (AOSMA)—to predict CBR values efficiently and accurately. A dataset consisting of 220 soil samples with eight geotechnical input parameters was used to develop and evaluate the hybrid models. The predictive performance of each model was assessed using multiple evaluation metrics, including R^2, RMSE, MSE, RSR, and WAPE. Results indicate that the SVR–AFT (SVAF) hybrid model outperformed the others, achieving an R^2 of 0.9968 and an RMSE of 0.7946 in the testing phase, demonstrating high generalization ability and predictive precision. The integration of SVR with metaheuristic algorithms significantly enhances model robustness and accuracy, offering a practical and cost-effective alternative to empirical CBR testing methods. This work highlights the potential of hybrid AI models in solving complex geotechnical prediction problems and contributes to the growing body of research at the intersection of civil engineering and artificial intelligence.

Povzetek: Hibridni modeli SVR so optimizirani z metahevristikami AFT, DOA in AOSMA za hitro in natančno napovedovanje CBR iz osmih geotehničnih parametrov. Na 220 vzorcih doseže najboljši model SVAF R^2 = 0.9968 in RMSE = 0.7946, kar ponuja stroškovno učinkovito alternativo laboratorijskim testom.

1 Introduction

CBR is the term utilized by Geotechnical construction to describe the resistivity of the substrate sample to a piston's insertion. More specifically, the CBR describes the force applied to the piston to enable it to penetrate the soil. [1]. Initially, the CBR examination was devised in California to appraise the suitability of soils for highway construction. Civil engineers modified the testing process to enhance its impact on the airport's construction. Almost all emerging countries widely adopt the CBR test to appraise pavements' resilience to soil [2]. A material's load-bearing capacity is gauged by its CBR, which is the ratio of the attainable supporting strength of base materials to that of regular crushed rock. In structural engineering, 100 is considered a reasonable limit for the CBR for broken rock substances.

Conversely, the values of CBR for alternative materials are found to be below 100 [3]. Recent advances in artificial intelligence (AI) are closely intertwined with the rapid development of electronic technologies, forming the foundation of the so-called "information society." Gams and Kolenik highlight the reciprocal relationship between electronics and AI, where swift hardware improvements, described by a comprehensive set of Information Society (IS) laws, have driven groundbreaking progress in AI across fields like medicine, smart environments, and autonomous systems. Their research shows that AI and ambient intelligence (AmI) not only benefit from electronic advancements but are also beginning to influence hardware optimization and intelligent system design through AI, indicating a move toward a more integrated technological progression [4]. After the compacted soils have been tested, the laboratory can conduct the subsequent test. However, it is possible for soils located in trenches to conduct the CBR test under the circumstances on the premises [5]Recognizing that in situ and laboratory test outcomes can show perceptible differences between soil types, unit weights, and water content is essential. Employing CBR tests has proved promising for presenting information about the stability and strength of different kinds of structures related to soils, such as road fills, airport roadways and dams, and road foundations.

Moreover, these tests can be conducted in unsoaked and soaked soil varieties. The laboratory CBR tests are characterized by their demanding nature regarding time and manual effort. Moreover, the outcomes of such tests are frequently marred by discrepancies attributed to the suboptimal quality of conditions in the lab and samples of dirt, which, in turn, lead to inaccurate CBR values [6].

Various studies have been performed on the California bearing ratio, which led the researchers to formulate different procedures. Previous studies showed that changes in the soil types and properties affected the value of CBR. Amongst other things, it has been observed that most research work has focused on studying the relationship existing between the compacted properties, unique indicators, as well as the mineral examinations' CBR concentrations [6]–[8][9], [10], [11], [12], [13], [14][15], [16], [17], [18]. To determine the value of CBR, soils are compacted at a predetermined MDD and OMC at a specified energy level for the soil material. For the CBR, the cases are soaked for four days; the primary purpose of this soaking is to allow absorption. Consequently, the assessment of the CBR value for a soaked sample typically requires a period of approximately five to six days. This delay can prove detrimental to the timely completion of a large-scale construction endeavor. Since soil is vastly different from one quality to another, applying this exercise to the foundation soil samples collected from a small count of sites may not truly represent the soil properties for all roads. To eliminate this deficiency, a large count of specimens is needed to be gathered for tests. Therefore, calculating the CBR values for pavement subgrade soils deploying easily identifiable parameters becomes very important in developing appropriate pavement design parameters. Recently, interest in using Artificial Intelligence (AI) tactics to solve geotechnical engineering problems has increased. Consequently, some valuable outcomes have been obtained [19], [20]. Furthermore, a limited count of studies has documented endeavors to appraise the (CBR) of soils via adopting diverse Artificial Neural Network (ANN) methodologies [21], [22], [23]. Recent advances in machine learning have increasingly supported geotechnical engineering by improving the prediction of soil and foundation properties through data-driven models. Support Vector Regression (SVR), when combined with metaheuristic optimization, has proven particularly effective in modeling complex nonlinear relationships within geotechnical datasets. Ngo et al. [24] demonstrated that SVR optimized via metaheuristics yielded superior performance in predicting the unconfined compressive strength of stabilized soils. Similarly, Hoang et al. [25] applied enhanced SVR models to successfully estimate pile bearing capacity, showcasing the method's versatility in foundation engineering. In the context of California Bearing Ratio (CBR) prediction, Bherde et al. [26] reported that Random Forest Regression outperformed other algorithms, including SVR, with maximum dry density and gravel content being the most influential predictors. While these results support the effectiveness of ensemble models, they also underline the need for more optimized SVR configurations that can match or exceed ensemble performance. A broader comparative study by Ma et al. [27], evaluating 20 metaheuristic algorithms for SVR parameter tuning in landslide displacement prediction, revealed considerable variation in outcomes. The Multiverse Optimizer emerged as particularly efficient in achieving high accuracy with low computational cost, highlighting the critical role of algorithm selection in enhancing SVR model performance. These studies collectively underscore the growing impact of hybrid AI models in geotechnical applications. However, few works have focused specifically on integrating SVR with newer and less explored metaheuristic algorithms such as the Alibaba and Forty Thieves (AFT), Dingo Optimization Algorithm (DOA), or Adaptive Opposition Slime Mold Algorithm (AOSMA). Our study addresses this gap by systematically evaluating and comparing these novel SVR-based hybrid models in predicting CBR, offering insights into their optimization behaviors, convergence patterns, and predictive robustness. By incorporating recent advancements and experimental benchmarks, this work aims to contribute both technically and methodologically to the field of AI-driven geotechnical modeling.

Considering the variety of parameters to be considered and the range of datasets observed, as explained in the previous paper, it becomes of prime importance to develop robust predictive methodologies to model the mechanical attributes of the CBR and delineate the complex correlations between the constituents of soil.

Recent studies have explored various soft computing and machine learning techniques for predicting the California Bearing Ratio (CBR). These include Random Forest, Gradient Boosting, and XGBoost, which are known for their solid performance in regression tasks. However, such models often need extensive tuning and can struggle to capture complex nonlinear relationships, especially when feature interactions are subtle. Conversely, Support Vector Regression (SVR) demonstrates strong generalization and robustness, especially when combined with kernel functions and metaheuristic optimization. To assess the effectiveness of the proposed SVR-based hybrid models, we also incorporated Random Forest as a benchmark and compared its predictive accuracy with the SVR models enhanced by metaheuristics.

Recent studies have explored different soft computing and machine learning approaches for predicting CBR. Key techniques include Artificial Neural Networks (ANN), Multiple Linear Regression (MLR), Group Method of Data Handling (GMDH), and SVM. These models use soil parameters such as Atterberg limits, dry density, optimum moisture content, and soil gradation as inputs. However, many struggle with issues like overfitting, limited generalization to unseen data, or inadequate hyperparameter optimization. As shown in Table 1, most previous models achieved only moderate accuracy and did not utilize metaheuristic optimization to boost prediction performance. To fill this gap, this study introduces a hybrid Support Vector Regression (SVR) model combined with three metaheuristic optimizers—AFT, DOA, and AOSMA—aimed at improving the model's ability to learn nonlinear patterns. The superior

performance of the SVAF model, especially in RMSE and R^2 metrics, highlights the benefits of this approach.

Table 1: Overview of past methods for CBR prediction

Study	Model Type	Input Features	Dataset Size	Performance Metrics (R^2 / RMSE)	Notes
Yildirim & Gunaydin	Artificial Neural Network (ANN)	LL, PL, PI, MDD, OMC, % Sand and Gravel	120	R^2 = 0.945 / RMSE = 1.82	ANN prone to overfitting and high variance
Taskiran	GMDH	LL, PL, PI, Compaction properties	200	$R^2 \approx 0.92$	Good performance, limited interpretability
Alawi & Rajab	Multiple Linear Regression (MLR)	LL, PL, PI, Soil Gradation	100	R^2 = 0.86	Struggles with nonlinear relationships
Ngo et al. [24]	SVR + Improved Arithmetic Optimization (IAOA)	Grain size, Density, OMC, PI	150	R^2 = 0.96 / RMSE = 1.12	SVR enhanced with metaheuristic tuning
Wu et al. [25]	Stochastic Gradient Boosting Regression (SGBR)	LL, PL, MDD, % Clay, % Silt	300	R^2 = 0.974	Ensemble method with good generalization
Bherde et al.[26]	Random Forest Regression (RFR)	MDD, % Gravel, OMC, PI	400	R^2 = 0.982	Strong performance, but no hyperparameter optimization
Current Study	SVR + AFT	LL, PL, PI, MDD, OMC, SDA, QD, OPC	300	R^2 = 0.9968 / RMSE = 0.7946	Best accuracy using hybrid SVR and AFT metaheuristic

This study addresses the challenges of traditional CBR testing methods, which are often time-consuming and costly, by exploring advanced machine learning models supplemented with nature-inspired optimization techniques. Specifically, it focuses on Support Vector Regression (SVR), a popular tool for nonlinear regression. Since SVR's performance heavily depends on hyperparameter selection, three recent metaheuristic algorithms—Adaptive Opposition Slime Mold Algorithm (AOSMA), Alibaba and the Forty Thieves Algorithm (AFT), and Dingo Optimization Algorithm (DOA)—are employed to optimize the SVR framework. These algorithms offer diverse search strategies with strong potential for effective global optimization and faster convergence. The predictive capability of these hybrid models is evaluated using five standard statistical metrics: R^2, RMSE, MSE, RSR, and WAPE. The study aims to (1) develop and validate an SVR model for predicting the California Bearing Ratio (CBR) based on soil and compaction parameters; (2) improve SVR's predictive performance through hyperparameter tuning with the three optimization algorithms; and (3) perform a comprehensive comparison of the models using these metrics to identify the most accurate and reliable one for geotechnical use.

2 Materials and methodology

2.1 Data gathering

This study's dataset consists of 121 soil samples gathered from various geotechnical investigation reports and lab tests across different regions in [insert country or region, e.g., southwestern Iran or southeastern Asia—please specify based on your case]. The samples include a variety of soil types such as clayey soils, silty sands, gravels, and mixtures to ensure the broad applicability of the predictive models. Each sample records essential input parameters like [list key parameters: e.g., dry density, moisture content, liquid limit, plasticity index, etc.], with the California Bearing Ratio (CBR) used as the target variable. Data were obtained from both published literature and in-house experiments, offering a comprehensive understanding of soil behavior in various geological settings. The data in this investigation depend on eight variables: OPC, SDA, QD, plastic limit, liquid limit, maximum dry density, plasticity index, and ideal content. Simultaneously, the resulting component of interest is identified as the CBR value. The dataset has been split into two subsets: 30% of the total set is made up of the testing phase, while 70% comprises the training phase.

Table 1 depicts a numerical example of some of the parameters used in building the scheme. This table gives an overall summary of some of the attributes, such as minimum (Min), maximum (Max), standard deviation (St.), and mean (M). It is crucial to determine the essential parameters for analyzing statistics. The maximum values for the LL, PL, PI, MDD, OMC, SDA, QD, and OPC variables are 52.1, 37.2, 19.5, 1.777, 29.5, 20, 20, and 8, respectively. Also, CBR's maximum value as an output parameter is 66.75 percent.

Table 2: The statistical features of the dataset components

Parameters	The numerical traits			
	Max	Min	Mean	St.dev
LL	52.10	21.20	35.8450	6.15380
PL	37.20	17.90	26.6830	4.28120
PI	19.50	2.10	9.16230	4.11490
MDD	1.7770	1.3650	1.49290	0.08830
OMC	29.50	18.90	24.1430	2.42670
SDA (%)	20.0	0	10.6600	7.15460
QD (%)	20.0	0	10.640	8.19610
OPC (%)	8.0	2.0	4.94490	2.37980
CBR (%)	66.750	19.690	39.9590	10.8660

2.2 Support vector regression (SVR)

In its early phases, the (SVM) technology was used to address pattern identification problems, initially introduced by Vapnik [28]. Then, Vapnik [29] suggested the SVM algorithm to solve problems with function approximation, which resulted in developing the SVR approach. The SVR approach is an innovative and perhaps practical method in data regression analysis. In this study, Support Vector Regression (SVR) is used as the main predictive model. Because of its ability to manage nonlinear relationships through kernel functions, SVR is especially suitable for modeling complex geotechnical datasets. The Radial Basis Function (RBF) kernel is chosen due to its effectiveness in high-dimensional feature spaces and its ability to generalize well. The SVR model depends on three main hyperparameters: C (regularization parameter): Regulates the trade-off between achieving a low training error and maintaining a simple model. γ (gamma): Determines the influence range of a single training example; a lower value means a wider reach, while a higher value indicates a more localized effect. ε (epsilon): Defines the tolerance margin within which errors are not penalized. These parameters were tuned using metaheuristic algorithms to minimize the root mean square error (RMSE) of predictions.

From an academic standpoint, the (SVR) may be explained in the subsequent terms. SVR uses a dataset that has $\bar{N}$ entries in it $\{(X_i, y_i), i = 1,2, \dots, \bar{M}\}$.

The training dataset's overall count of instances is denoted by M.

$X_i = \{x_1, x_2, \dots, x_m\} \in R^m$ denotes the $i-th$ component of the vector with M dimensions.

$y_i \in R$ represents the genuine value connected to X_i.

For that, in machine learning tactics, l-dimensional feature space - or something similar - represents the exact mapping of any training data point X_i in an SVR. The obtained hyperplane is in the space of features that will be selected using Support Vector Regression towards the portrayal of the optimal hyper-plane between the input (or the uncorrelated) variable and the exact output, the dependent variables: Eq. (1) gives, mathematically, the operation of an SVR;

$$f(x) = Z^T \varphi(x) + b \quad (1)$$

b is the variable element

$f(x)$ symbolizes the expected parameters

Z is the $l - dimensional$ weighting component.

An example of how distinct components X_i are mapped to a feature space with many dimensions is the function $\varphi(x)$.

The formal expression for the ε-insensitive coefficient of loss is found in Eq. (2).

$$|y - f(x)|_\varepsilon = \max(0, |y - f(x)| - \varepsilon) \quad (2)$$

The difference between the real number, symbolized by y, and the anticipated value, f(x), as expressed theoretically by Eq. (3), is known as the residual.

$$R(x, y) = y - f(x) \quad (3)$$

According to Eq. (4), incorporating the entire residue piece within a preset boundary value of ε is the optimum regression model.

$$-\varepsilon \le R(x, y) \le \varepsilon \quad (4)$$

Eq. (4) coincides with the hypothesis on the whole training data set. Thus, if the residual meets the criterion $R(x, y) = \pm\varepsilon$, the data exhibits a maximum detour from the hyperplane. One can calculate a spatial separation of an arbitrary data point (x, y) from the hyperplane $R(x, y) = 0$ by the formula $|R(x, y)|/\|W^*\|$. Further, Z^* can be calculated as:

$$Z^* = (1, -Z^T)^T \quad (5)$$

In this question, the variable δ is assumed to be the maximum degree of dispersion between the hyperplane $R(x, y) = 0$ and the dataset (x, y). All the training data can be induced to meet the requirements shown in Eq. (6). If the value of δ reaches its maximum in the SVR scheme, it means that the scheme can exhibit the best generalization ability.

$$|R(x, y)| \le \delta \|Z^*\| \quad (6)$$

Whenever $R(x,y)$ equals an ε, the most significant distance is reached. After that, Eq. (6) may be changed to become Eq. (7). Considering the translation of an optimization issue to a minimal $\|Z\|$, $\|Z^*\|^2 = \|Z\|^2 + 1$, and $\|Z^*\|$ must be a minimal value to attain the maximum of δ.

$$\varepsilon = \delta\|Z^*\| \tag{7}$$

Even with efforts to keep mistakes within the $(-\varepsilon, \varepsilon)$ range during training, it is still possible for certain errors to surpass this limit. If training mistakes are less than -ε, they are displayed by ζ_i, and if they are more than ε, they are displayed by ζ_i^*. We define the notations ζ_i and ζ_i^* according to Eqs. (8) and (9), appropriately.

$$\zeta_i = \begin{cases} 0 & R(x_i, y_i) - \varepsilon \le 0 \\ R(x_i, y_i) - \varepsilon & others \end{cases} \tag{8}$$

$$\zeta^*{}_i = \begin{cases} 0 & \varepsilon - R(x_i, y_i) \le 0 \\ \varepsilon - R(x_i, y_i) & others \end{cases} \tag{9}$$

By using the ε sensitivity loss function, (SVR) aims to eliminate the distinction across the training data and the hyperplane region and choose a hyperplane that produces the best result. The goal function for (SVR) optimization is displayed by Eq. (10):

$$minF(Z, b, \zeta_i, \zeta^*{}_i) = \frac{1}{2}\|Z\|^2 + c\sum_{i=1}^{M}(\zeta_i + \zeta^*{}_i) \tag{10}$$

With the confinements:

$$y_i - Z^T\varphi(x_i) - b \le \varepsilon + \zeta_i \quad i = 1, 2, \ldots, \bar{M}$$
$$Z^T\varphi(x_i) + b - y_i \le \varepsilon + \zeta^*{}_i \quad i = 1, 2, \ldots, \bar{M}$$
$$\zeta_i \ge 0, \zeta^*{}_i \ge 0 \quad i = 1, 2, \ldots, \bar{M}$$

The first term of Eq. (10) tends to restrict weights, so they stay above a certain limit to preserve whether the regression algorithm is constant. The second part of this system defines the ratio of certainty to vulnerability for possible hazards resulting from previous experiences using the ε-insensitive Relationship to losing. After determining the solution for the quadratic enhancement issue with inequality restrictions, the value of coefficient Z can be gathered from Eq. (11).

$$Z = \sum_{i=1}^{M}(\beta^*{}_i - \beta_i)\varphi(x_i) \tag{11}$$

The values of β_i^* and β_i are determined by solving a quadratic programming problem that incorporates an indication of the Lagrangian multipliers. Mathematically, the Support Vector Regression function is displayed with the utilize of the equation depicted as Eq. (12):

$$f(x) = \sum_{i=1}^{M}(\beta^*{}_i - \beta_i)K(x_i - x) + b \tag{12}$$

The kernel function, which is displayed as $K(x_i - x)$, exhibits the capacity to convert the training data into a higher nonlinear l-dimensional space. Therefore, this methodology is deemed appropriate for solving issues related to nonlinear relationships, including projecting electrical power. Figure 1 shows the operational diagram for SVR.

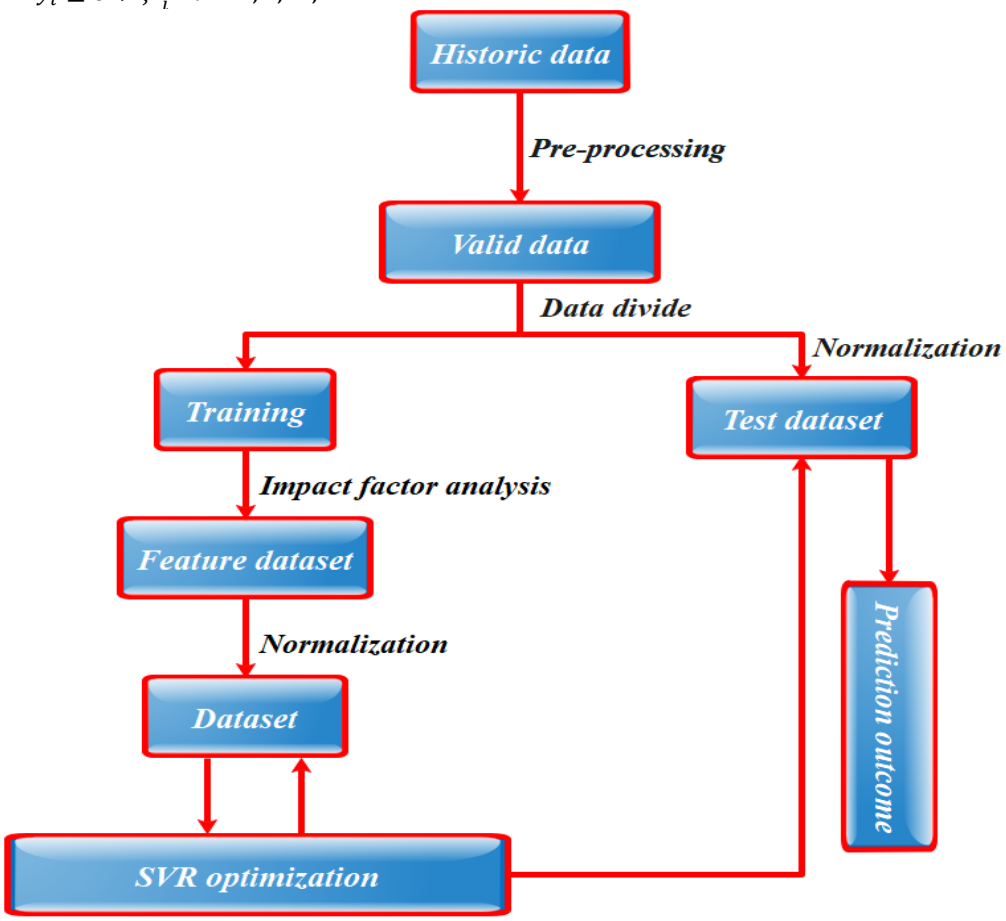


Figure 1: The progress and validation flowchart of an SVR scheme

2.3 AOSMA

The plasmodial slime mold's oscillatory mode is the basis for SMA. The slime mold employs a positive-negative feedback mechanism in conjunction with an oscillatory mode to establish the optimal route toward nutrition [30]. AOSMA is a new statistical technique that incorporates an opposition-based learning-based adaptive decision-making method to improve slime mold's nearing conduct [31].

Let it be assumed that a total of N individuals of the species of slime mold under consideration are resident in the search domain that is bounded by an upper boundary (UB) and a lower boundary (LB) for theoretical framework development of the (AOSMA).

$X_i = (x_i^1, x_i^2, \cdots, x_i^d), \forall i \in [1, N]$ is the ith slime mold's location in d-dimension.

$F(X_i), \forall i = [1, N]$ symbolizes the ith slime's fitness.

The following represents the location as well as fitness of the slime mold at round t:

$$X(x) = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^d \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \quad (13)$$

$$F(X) = [F(X_1), F(X_2), \cdots, F(X_N)] \quad (14)$$

In the $(t+1)$ cycle, the situation of the slime mold has been advanced. It has undergone an upgrade in its spatial disposition, which determine as Eq. (15):

$$X_i(t+1) = \begin{cases} X_{LB}(t) + V_d\big(W.X_A(t) - X_B(t)\big) & p_1 \geq \delta \text{ and } p_2 < m_i \\ V_e.X_i(t) & p_1 \geq \delta \text{ and } p_2 \geq m_i, \forall i \in [1, N] \\ rand.(UB - LB) + LB & p_1 < z \end{cases} \quad (15)$$

X_{LB} is the best local slime mold

X_A and X_B are pooled individuals by random

W is the weight factor

V_d and V_e are the random velocities.

p_1 and p_2 are randomly chosen numbers in [0,1]

The slime mold's chance, which starts at a random search situation, is fixed at $\delta = 0.03$.

The $i-th$ member of the population's threshold value, m_i, aids in choosing the slime mold's location, which is calculated as Eq. (16):

$$m_i = tanh|F(X_i) - F_G|, \forall i \in [1, N] \quad (16)$$

$$F_G = F(X_G) \quad (17)$$

$$W\big(SortInd_F(i)\big) = \begin{cases} 1 + rand.\log\left(\frac{F_{LB} - F(X_i)}{F_{LB} - F_{Lw}} + 1\right) & 1 \leq i \leq \frac{N}{2} \\ 1 - rand.\log\left(\frac{F_{LB} - F(X_i)}{F_{LB} - F_{Lw}} + 1\right) & \frac{N}{2} < i \leq N \end{cases} \quad (18)$$

F_G and X_G are the values of worldwide top ranking and worldwide best well-being.

$rand$ displays a random number in within [0,1]

F_{LB} and F_{Lw} are local best and worst fitness values.

The utilization of an ascending order for sorting fitness values can be employed in a minimization problem:

$$[Sort_F, SortInd_F] = sort(F) \quad (19)$$

The local best and worst fitness also the local best slime mold X_{LB} are computed as Eqs. (20-22):

$$F_{LB} = F\big(Sort_F(1)\big) \quad (20)$$

$$F_{LW} = F\big(Sort_F(N)\big) \quad (21)$$

$$X_{LB} = X\big(SortInd_F(1)\big) \quad (22)$$

The randomly assigned velocities are known as V_d and V_e and are defined as follows:

$$V_d \in [-d, d] \quad (23)$$

$$V_e \in [-e, e] \quad (24)$$

$$d = \text{arctanh}\left(-\left(\frac{t}{T}\right) + 1\right) \quad (25)$$

$$e = 1 - \frac{t}{T} \quad (26)$$

T is the maximum cycle.

SMA holds great promise for both investigation and exploitation in technological problem-solving and enhancement. However, the improvement of slime mold regulations in the SMA area is nevertheless reliant on a count of basic circumstances.

Case 1: The region's best slime mold, X_{LB}, and two random individuals, X_A and X_B, with velocity V_d, drove to determine when $p_1 \geq z$ and $p_2 < m_i$. This stage makes it easier to strike a balance amongst discovery and extraction.

Case 2: The orientation of the slime mold with velocity V_e directs the search when $p_1 \geq z$ and $p_2 \geq m_i$. This instance facilitates fraud.

Case 3: When $p_1 < z$, the person reinitializes within a specified search domain. This phase facilitates investigation.

Case 1 shows how the possibilities of finding solutions are improperly controlled during exploration and exploitation since X_A and X_B are two random slime molds. To get around this limitation, X_A can be used in place of best local individual X_{LB}. Consequently, the location of the $i-th$ component is remodeled as Eq. (27):

$$Xn_i(t) = \begin{cases} X_{LB}(t) + V_d\big(W.X_{LB}(t) - X_B(t)\big) & p_1 \geq \delta \text{ and } p_2 < m_i \\ V_e.X_i(t) & p_1 \geq \delta \text{ and } p_2 \geq m_i \\ rand.(UB - LB) + LB & p_1 < \delta \end{cases} \quad (27)$$

Case 2 illustrates how slime mold deliberately targets a nearby location, resulting in a path with a lower fitness level. A better approach to this issue is to implement an adaptive decision system.

Case 3 illustrates that the SMA offers criteria for exploration. However, with a small value $\delta = 0.03$, the exploration has been limited. To address the issue, it is imperative to introduce an auxiliary exploration adjunct for SMA. A practical approach to addressing the limitations of Cases 2 and 3 entails employing a flexible decision approach that leverages opposition-based learning (OBL) to determine the necessity of additional exploratory efforts [32]. The OBL uses a defined Xop_i in the search domain, which is precisely the opposite of the Xni for each member ($i = 1,2,\cdots,N$), and compares it to upgrade the following cycles' situation. It assists in improving convergence and avoiding the chances of being closed in the local minima. So, the Xop_i for the $i-th$ individual in $j-th$ $(j = 1,2,\cdots,s)$ dimension is described as follows:

$$Xop_i^j = \min\left(Xn_i(t)\right) + max\left(Xn_i(t)\right) - Xn_i^j(t) \quad (28)$$

Xr_i represents the $i-th$ member's situation in the reduction issue and is depicted as:

$$Xr_i = \begin{cases} Xop_i(t) & F(Xop_i(t)) < F\left(Xn_i(t)\right) \\ Xn_i(t) & F(Xop_i(t)) \geq F\left(Xn_i(t)\right) \end{cases} \quad (29)$$

A flexible decision is formed drawing on the prior worth of fitness $f(Xi(t))$ and the present fitness value $f(Xni(t))$ in the event of a depleted nutrient pathway. This is a typical academic kind of writing. It helps provide added research as needed. Then, the situation for the subsequent cycle is improved:

$$X_i(t+1) = \begin{cases} Xn_i(t) & F(Xn_i(t)) \leq F\left(X_i(t)\right) \\ Xr_i(t) & F(Xn_i(t)) > F\left(X_i(t)\right) \end{cases}, \quad \forall i \in [1,N] \quad (30)$$

The aforementioned AOSMA framework is displayed in pseudo-code, as shown in Algorithm 1.

In this study, the Adaptive Opposition Slime Mold Algorithm (AOSMA) is used not as a standalone optimizer but as a hybrid component integrated with Support Vector Regression (SVR). AOSMA optimizes three key hyperparameters of SVR—specifically the regularization parameter C, the epsilon-insensitive loss margin ε, and the kernel coefficient γ—with the goal of minimizing prediction error measured by RMSE. Through its adaptive opposition-based learning strategy and dynamic parameter control, AOSMA allows for more effective exploration of the search space and helps prevent premature convergence. As a result, the hybrid AOSMA-SVR model achieves better accuracy and generalization in predicting California Bearing Ratio (CBR) values from geotechnical data.

Algorithm 1: AOSMA

Begin

Using the criteria for searching boundary range $[LB, UB]$, choose a target variable f with inputs N, s, T, and δ.
Outputs: X_G and F_G
Initialization: Launch the slime mold at arbitrary.
$X_i = (x_i^1, x_i^2, \cdots, x_i^d)$, $\forall i \in [1,N]$ during the first revision, inside the query boundaries UB and LB
$t = 1$.

while $(t \leq T)$

→ Determine the N slime mold's fitness values $F(X)$.
→ Put the fitness value in order.
→ The local best individual X_{LB} should be updated to match the local best conditioning F_{LB}.
→ The local weakest fitness F_{LW} should be updated.
→ Update the matching worldwide greatest individual X_G and global best fitness F_G.
→ Refresh the measurement of W.
→ Update the d using Eq. (25) and e using Eq. (26).

for (each slime mold $i = 1{:}N$)

- Create the p_1 and p_2 randomized numbers.
- Create the m_i threshold quantity.
- Utilizing Eq. (27), determine the new slime mold location Xn_i.
- Determine the new slime mold $F(Xn_i)$'s nutritional value.

if $(F(Xn_i) > F(X_i)$ // Adaptive decision strategy

- Estimate Xop_i using Eq. (24). //Opposition-based learning
- Select Xr_i using Eq. (29).

End

- Revise the subsequent cycle slime mold X_i using Eq. (30).

end

→ The following repetition $t = t + 1$

end

The result is X_G, representing the global most effective region.

2.4 AFT

The present investigation clarifies the basic AFT algorithm's mathematical model, which is described in [33]. The scheme encompasses three states that can be analyzed and delineated in the following:

Case 1: The pursuit of Ali Baba by the thieves, as derived from information obtained from a source, can be displayed by a simulation of their situations, as illustrated in Eq. (31):

$$x_i^{t+1} = gbest^t + \left[\mathrm{Td}^{\mathrm{t}}(best_i^t - y_i^t)r_1 + \mathrm{Td}^{\mathrm{t}}\left(y_i^t - m_{a(i)}^t\right)r_2\right]sgn(rand - 0.5),\ \ p \geq 0.5,\ \ q > P_{p^t} \quad (31)$$

y_i^t represents Ali Baba's situation regarding the thief i.

$m_{a(i)}^t$ represents the amount of cleverness that Marjaneh uses to cover up thievery i.

x_i^{t+1} denotes the situation of the $i - th$ thief.

$gbest^t$ is the most excellent situation a thief has ever had worldwide.

r_1, r_2, rand, p, and q are random values created within $[0, 1]$

$best_i^t$ is the optimal location of thief i has determined.

Td^{t} is the robbers' surveillance area as specified by Eq. (32).

$p \geq 0.5$ presents either 0 or 1

P_{p^t} is Ali Baba's potential perceptive ability, as stated by Eq. (33).

$sgn(rand - 0.5)$ can be -1 or 1, and

a is defined as Eq. (34).

$$\mathrm{Td}^{\mathrm{t}} = \tau_0 e^{-\tau_1(\frac{t}{T})_1^{\tau}} \quad (32)$$

t and T Please consult the current and maximal repetition standards, accordingly.

τ_0 $(\tau_0 = 1)$ is a preliminary estimate of the monitoring length.

τ_1 $(\tau_1 = 2)$ is a set amount that regulates the discovery and utilization of resources.

$$P_{p^t} = \lambda_0 \log\left(\lambda_1(\frac{t}{T})^{\lambda_0}\right. \quad (33)$$

λ_0 $(\lambda_0 = 1)$ depicts the final assessment of the robbers' chances of completing their task after the hunt.

λ_1 $(\lambda_1 = 1)$ refers to a fixed value that controls exploration and exploitation.

$$a = [(n-1).rand(n,1)] \quad (34)$$

The vector $rand(n, 1)$ is generated as a set of random numbers within the bounds of [0,1].

$$m_{a(i)}^t = \begin{cases} x_i^t & if\ f(x_i^t) \geq f(m_{a(i)}^t) \\ m_{a(i)}^t & if\ f(x_i^t) < f(m_{a(i)}^t) \end{cases} \quad (35)$$

The score of the fitness function is denoted by $f(0)$.

Case 2: Thieves may perceive they have been tricked and will likely start exploring unfamiliar and unplanned areas.

$$x_i^{t+1} = Td^t\left[(u_j - l_j)r + l_j\right]; p \geq 0.5\ , q \leq P_{p^t} \quad (36)$$

The upper and lower bounds of the search domain at dimension j are displayed by u_j and l_j, respectively.

r displays a stochastic quantity generated in the interval [0, 1].

Case 3: To improve AFT's exploration and exploitation capabilities, thieves can investigate alternative search situations beyond those identified through the utilization of Eq. (31). This scenario can be formulated as Eq. (37):

$$x_i^{t+1} = gbest^t - \left[\mathrm{Td}^{\mathrm{t}}(best_i^t - y_i^t)r_1 + \mathrm{Td}^{\mathrm{t}}\left(y_i^t - m_{a(i)}^t\right)r_2\right]sgn(rand - 0.5) \quad (37)$$

Algorithm 2 concisely and formally describes the iterative pseudo-code stages that correspond to the core AFT.

The proposed hybrid framework combines the Dingo Optimization Algorithm (DOA) with Support Vector Regression (SVR) to tune the model's hyperparameters: C, ε, and γ. The DOA emulates the natural hunting tactics of dingoes, such as surrounding, chasing, and attacking prey, which are adapted into search operators for exploring the SVR parameter space. The aim is to minimize the SVR's RMSE on training data by identifying the optimal parameter combination. By balancing diversification and intensification, the DOA-SVR hybrid model can effectively avoid local optima and enhance SVR's ability to generalize for accurate CBR prediction.

Algorithm 2: AFT

Establish the regulation settings and get started.
Start by assessing every thief's starting, optimal, and worldwide situations.
Start by assessing Marjane's intelligence in comparison to all thieves.
Set $t \leftarrow 1$
While $(t \leq T)$ do
Eq. (33) is used for modifying the input parameter P_{p^t}.
for each thief, do
if $(p \geq 0.5)$ then
if $(q \geq P_{p^t})$ then
Use Equation (32) to update the thieves' positioning.
else
Utilizing Equation (36), adjust the robbers' whereabouts.
end if
else
Refine the thieves' situation by Eq. (37).
end if

end for
Refresh all thieves' current, best, and worldwide standings.
Utilizing Eq. (35), alter Marjane's wit goals.
$t = t + 1$
end while
Give back the world's optimal solution.

2.5 Dingo optimization algorithm (DOA)

From the earliest times, nature has consistently been regarded as an exceptionally instructive and impactful educator. Every species that exists on the planet Earth possesses a distinct and unique mechanism for ensuring its survival. The present study involves the mathematical modeling of hunting behavior and social arrangements in the dingo species. This analytical approach is the basis for developing a DOA nature-inspired optimization technique [34]. The two primary constituents of DOA are regarded as exploration and exploitation. The algorithm generates various anticipated outcomes within the search domain during the initial exploration phase. However, the subsequent exploitation phase enables identifying and pursuing the most desired resolutions within the predetermined space. To discern the optimal resolution for a given pragmatic concern, refinement, and integration of both constituent factors are necessary. Nonetheless, achieving equilibrium among the proposed algorithm's constituents is arduous due to its stochastic disposition. To address an authentic engineering dilemma, the impetus for developing an algorithm implementation utilizing hybridized meta-heuristics is derived from this inspirational notion [34].

Dingo optimization is done by the computational designing of the prey's pursuit, encirclement, and attack.

2.5.1 Encircling

Given the lack of previous knowledge about the search location and its ideal characteristics, it is proposed that the objective or target prey is the best agent tactic currently in use, representing the social hierarchy of dingoes. The following mathematical formulas can be used to formalize the dingoes' behavior:

$$\vec{D}_d = \left|\vec{A}, \vec{P}_p(x) - \overrightarrow{P}(i)\right| \quad (38)$$

$$\overrightarrow{P}(i+1) = \vec{P}_p(x) - \vec{B}.\vec{D}(d) \quad (39)$$

$$\vec{A} = 2\,.\,\vec{a}_1 \quad (40)$$

$$\vec{B} = 2\vec{b}\,.\,\vec{a}_2 - \vec{b} \quad (41)$$

$$\vec{b} = 3 - (I \times \left(\frac{3}{I_{max}}\right)) \quad (42)$$

The neighborhood dingoes' geographic coordinates are displayed as a two-dimensional vector. The dingo may adjust its situation to match the coordinates of (P, Q) based on the prey's location, which is displayed as (P^*, Q^*). By adjusting the $\vec{A}$ and $\vec{B}$ vectors about the present situation, the graphic shows every possible location around the ideal agent. Setting $\vec{A} = (1,0)$ and $\vec{B} = (1,1)$ provides access to the dingo's situation at $(P^* - P, Q^*)$For example, Eqs. (38) and (39) make it easier for dingos to travel throughout the hunting area and find their prey randomly.

2.5.2 Hunting

Using a mathematical method, creating a dingo hunting strategy involves assuming that the alpha, beta, and other members of the pack have a thorough awareness of the possible prey sites. When conducting hunting trips, the alpha dingo always takes the lead. However, other dingo species, including beta, may hunt as well. Eqs. (43) to (51) are developed with this issue in line with the discussion.

$$\vec{D}_\alpha = \left|\vec{A}_1.\vec{P}_\alpha - \overrightarrow{P}\right| \quad (43)$$

$$\vec{D}_\beta = \left|\vec{A}_2.\vec{P}_\beta - \overrightarrow{P}\right| \quad (44)$$

$$\vec{D}_o = \left|\vec{A}_3.\vec{P}_o - \overrightarrow{P}\right| \quad (45)$$

$$\vec{P}_1 = \left|\vec{P}_\alpha - \overrightarrow{B}.\vec{D}_\alpha\right| \quad (46)$$

$$\vec{P}_2 = \left|\vec{P}_\beta - \overrightarrow{B}.\vec{D}_\beta\right| \quad (47)$$

$$\vec{P}_3 = \left|\vec{P}_o - \overrightarrow{B}.\vec{D}_o\right| \quad (48)$$

The following formulae are utilized to determine each dingo's intensity:

$$\vec{I}_\alpha = \log\left(\frac{1}{F_\alpha - (1E - 100)} + 1\right) \quad (49)$$

$$\vec{I}_\beta = \log\left(\frac{1}{F_\beta - (1E - 100)} + 1\right) \quad (50)$$

$$\vec{I}_o = \log\left(\frac{1}{F_o - (1E - 100)} + 1\right) \quad (51)$$

2.5.3 Attacking

If a situation update is unavailable, it may be inferred that the dingo successfully concluded its hunt through a predatory attack. To formally articulate the strategy, the value of $\vec{b}$ is systematically diminished linearly through the utilization of mathematical notation. Noteworthy is the fact that the variation range of $\vec{D}_\alpha$ is further diminished by $\vec{b}$. The value above may be identified as $\vec{D}_\alpha$, which is a stochastic variable generated within the range of [-3b, 3b], where the constant $\vec{b}$ undergoes a decremental process from 3 to 0 over a series of cycles. When $\vec{D}_\alpha$ Values are randomly generated within the interval [1,1]. An exploratory agent is capable of moving to any possible situation along the trajectory between its existing location and the prey's location.

2.5.4 Searching

Dingoes exhibit hunting patterns primarily determined by their pack's location. They consistently progress in pursuit of locating and subduing prey. $\overrightarrow{B}$ represents random variables. Notably, if the value assigned to $\overrightarrow{B}$ falls below -1, it implies that the prey is retreating from the search agent. Conversely, if $\overrightarrow{B}$ exceeds 1, the pack is advancing toward its prey. This particular intervention facilitates the Department of Defense conduct a comprehensive global reconnaissance of identified targets. One factor contributing to a heightened probability of exploration within the DOA is the component denoted as $\overrightarrow{A}$. In Eq. (40), the vector $\overrightarrow{A}$ can generate a range of random numbers within the interval between 0 and 3, independent of the weight of the prey selected. The DOA function can be characterized as a stochastic vector whereby the elements with values that are less than or equal to one take priority over those greater than or equal to one. This feature elucidates the gap's influence as described in Eq. (38). The hybrid framework combines the Dingo Optimization Algorithm (DOA) with Support Vector Regression (SVR) to tune hyperparameters: C, ε, and γ. Inspired by the natural hunting strategies of dingoes, such as surrounding, chasing, and attacking prey, the DOA translates these behaviors into search operators that explore the SVR parameter space. Its aim is to minimize the RMSE of SVR on training data by identifying the best parameter combination. By balancing exploration and exploitation, the DOA-SVR hybrid effectively avoids local optima and improves SVR's generalization ability, leading to more accurate CBR predictions.

Algorithm 3 offers the pseudo-code for the DOA.

Algorithm 3: Dingo Optimization

Input: The population of dingoes D_n $(n = 1, 2, \ldots, n)$
Output: The best dingo. (Here, the best values are minimum)
Generate initial search agents D_{in}
Start the value of $\overrightarrow{b}$, $\vec{A}$, and $\overrightarrow{B}$.
While the Termination condition is not reached, do
Appraise each dingo's fitness and intensity cost.
D_α = dingo with the best search
D_β = dingo with the second-best search
D_o = Dingoes search outcomes afterward
Cycle1
repeat
for $i = 1$: D_{in} **do**
Renew the latest search agent state.
end for
Project the fitness and intensity cost of dingoes.
Record the value of S_α, S_β, S_δ
Record the value of $\overrightarrow{b}$, $\vec{A}$, and $\overrightarrow{B}$.
$Iteration = Iteration + 1$
Monitor if cycle≥ Stopping criteria
output
end while

Choosing AFT, AOSMA, and DOA as optimizers was driven by their unique algorithmic bases and search methods, enabling a thorough comparison of their metaheuristic behaviors. These approaches are relatively recent and less studied, yet they show competitive performance in diverse regression and engineering tasks. Incorporating them with SVR in this research allows evaluation of both their predictive accuracy and optimization stability across different algorithmic frameworks.

2.6 Reproducibility and run settings

To ensure the robustness and reproducibility of the results, each hybrid SVR model (AFT-SVR, DOA-SVR, AOSMA-SVR) was executed 30 independent times. This allows for reliable statistical analysis of model performance. Additionally, random seed initialization was controlled using a fixed seed (e.g., seed = 42) across all algorithms during training and optimization to maintain consistent behavior during repeated runs and to support reproducibility.

2.7 Hybridization strategy of SVR with metaheuristic algorithms

This study developed three hybrid machine learning models—SVAF, SVSM, and SVDO—by integrating Support Vector Regression (SVR) with three advanced metaheuristic optimization algorithms: Alibaba and Forty Thieves (AFT), Adaptive Opposition Slime Mold Algorithm (AOSMA), and Dingo Optimization Algorithm (DOA). The goal is to boost SVR's prediction accuracy by optimizing its key hyperparameters—penalty parameter C, kernel parameter γ, and epsilon-insensitive loss ε—using the global search methods provided by these metaheuristics. While SVR is a strong nonlinear regression technique, its effectiveness heavily relies on

proper parameter tuning. Traditional manual or grid search methods are often inefficient or may yield suboptimal results, especially with complex, high-dimensional geotechnical data. Therefore, this hybrid approach exploits the global search and convergence strengths of nature-inspired algorithms to automate SVR hyperparameter optimization.

- In SVAF, the AFT algorithm explores the search space dynamically through mechanisms like global surveillance, balancing exploration and exploitation, and adaptive decision-making inspired by Marjaneh. These features enable it to identify optimal SVR parameters reliably.

- In SVSM, AOSMA enhances the slime mold algorithm with opposition-based learning and adaptive strategies, allowing it to escape local minima more effectively and converge more rapidly, thus providing better hyperparameter configurations.

- In SVDO, the DOA mimics the social hunting behaviors of dingoes—such as encircling, attacking, and searching—to iteratively fine-tune the SVR parameters for higher prediction accuracy.

Each metaheuristic aimed to minimize the RMSE of SVR predictions on training data, with the best parameter set used to train the final hybrid model. The process was repeated 30 times to ensure stability and reproducibility. This hybrid approach directly supports the study's goal of creating accurate, efficient, and generalizable models for predicting the California Bearing Ratio (CBR) of soils. Using these metaheuristics not only enhances SVR's learning ability but also reduces the manual effort and computational cost typically required for parameter tuning.

2.8 Performance evaluation tactics

A range of evaluators was deployed to appraise hybrid schemes' productivity in CBR value prediction. The list of evaluators comprises RMSE, MSE, R^2, the ratio of RMSE to standard deviation (RSR), and lastly, weighted absolute percentage error, or WAPE. R2 determines the degree of linear relationship between the actual and forecasted magnitudes. The RMSE is the square root of the ratio between the square of the count of specimens and the estimated value departure from the actual value. WAPE could be quantified by dividing the total absolute error by the total real demand. Eq. (21-25) provides the values of these metrics above.

$$R^2 = \left(\frac{\sum_{i=1}^{n}(b_i - \bar{b})(d_i - \bar{d})}{\sqrt{\left[\sum_{i=1}^{n}(b_i - \bar{d})^2\right]\left[\sum_{i=1}^{n}(d_i - \bar{d})^2\right]}} \right)^2 \quad (52)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - p_i)^2} \quad (53)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}((d_i - p_i)^2 \quad (54)$$

$$RSR = \frac{RMSE}{St.Dev} \quad (55)$$

$$WAPE = \frac{\sum_{i=1}^{n}|d_i - b_i|}{\sum_{i=1}^{n}|b_i|} \quad (56)$$

n indicates the count of samples; d_i displays the forecasted value; b_i displays the actual value, while $\bar{d}$ and $\bar{b}$ represent the mean of the forecasted value and the average of the actual amount, respectively.

3 Outcomes and discussion

This paper reports on developing a Support Vector Regression model using three new enhancement techniques, AFT and DOA, aimed at developing three hybrid predictive models for soil estimation CBR. In previous schemes, the information about information was divided into two subsets: a set to learn and a set to validate the scheme, 70% and 30% of the data, respectively. The five consecutive statistical metrics, namely, R2, RMSE, MSE, RSR, and WAPE, were considered to get the full view of the optimizers' performance. Outcomes can be shown in Table 2. The statistical indicators are analyzed in this section to determine whether one model is generally better. By studying the various R2 values among these different schemes, it would be crystal clear that the most promising outcomes are given out by SVAF in both the testing and training stages, with 0.9968 and 0.9929 values, respectively. Meanwhile, the minimum value of R2 among all comparative schemes was given to the SVSM model at 0.9767. The key thing worth mentioning here is that all the schemes have increased R2 during their test phases, indicating that the schemes are well-trained. Maximum RMSE, MSE, RSR, and WAPE values are 1.6271, 2.6475, 0.1524, and 0.0334 for SVSM in training. For the testing section, maximum RMSE values, MSE, RSR, and WAPE are 1.5824, 2.5042, 0.1409, and 0.0312 for SVSM. By contrasting the evaluators' and errors' values, the best hybrid scheme for estimating the CBR value of soils is the combination of SVR and the ATF algorithm (SVAF). This model has the highest R2 value (0.9968 in the testing phase) and the lowest error value (0.7946 in testing) among all three components.

Table 3: The hybridized schemes produced the findings

Schemes	SVAF		SVSM		SVDO		SVR	
Section	Train	Test	Train	Test	Train	Test	Train	Test
RMSE	0.9316	0.7946	1.6271	1.5824	1.3363	1.171	1.336392	1.171305
R2	0.9929	0.9968	0.9767	0.9825	0.9852	0.992	0.985202	0.992446
MSE	0.868	0.6314	2.6475	2.5042	1.7859	1.372	1.7859	1.372

RSR	0.0872	0.0708	0.1524	0.1409	0.1251	0.1043	0.1251	0.1043
WAPE	0.0162	0.0141	0.0334	0.0312	0.0234	0.0212	0.0234	0.0212

Fig. 2 displays the dispersed presentations illustrating the correlation between the gauged and expected California Bearing Ratio values. R2 and RMSE are two types of assessments that include numerical data. When the value of this evaluation metric decreases, density increases because RMSE functions as a deviation controller. Additionally, the training and testing data points are drawn toward the center axis by the R2 evaluator. The figure below illustrates several other variables which also include but are not restricted to the linear regression model's centerline, which is positioned at the location Y=X, as well as dual lines that are in red below and above the midline, in that order, at Y=0.9X and Y=1.1X. The lower and upper ends of the line intersections provide the false predictions of an underestimation and an overestimation of values, respectively. Three schemes were produced by the subsequent analysis, which combined the SVR scheme with the three optimizer strategies applied to training and testing. Fig. 2 shows the findings of the current investigation. R2 of SVAF appears to be comparatively more favorable than the rest of the schemes because the data points maintain the same directionality and are nearer the centerline. From empirical data, it can be induced that in all cases, and quite noticeable in the case of SVDO, the precision of the test phase values is higher than that of the training phase. Overall, the result from the acquired data in Fig. 2 is the most favorable result using the SVR method and the ATF optimizer since R2 and RMSE in learning and validation also gave the best result. That could be due to the capability of this model in terms of minimizing error and being the best in performance regarding the R2.

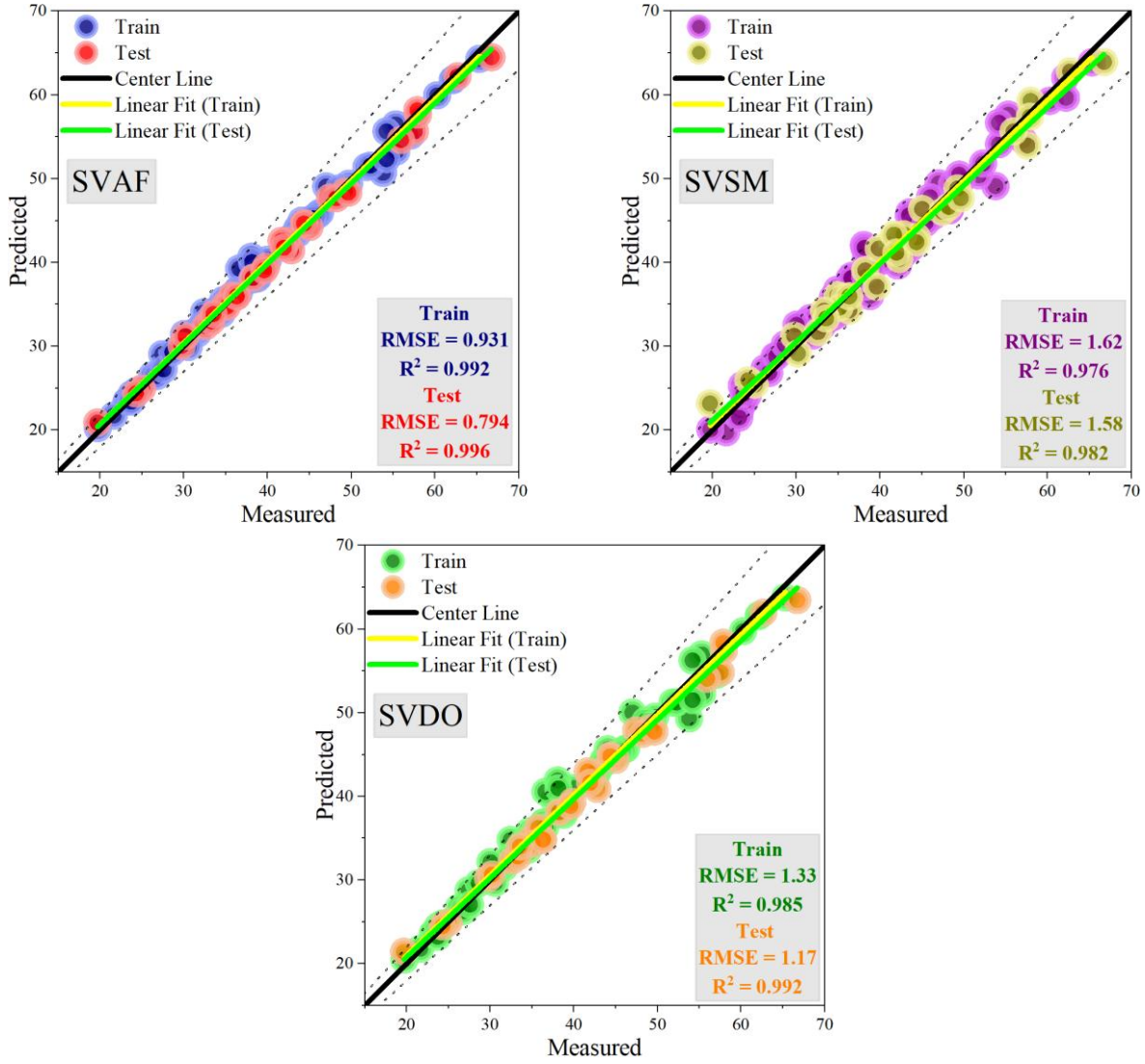


Figure 2: The scatter plot of expected and measured values

Fig. 3 presents the correlation between expected and actual CBR values obtained using three different classes of hybrid schemes. The graphs have been divided into two distinct parts: model training and model validation. Among them, the SVAF representing an SVR and the ATF algorithm generate closer agreement between the

gauged CBR values of the expected output for testing and training data sets. By contrast, the status of the least unfavorable agreement appears quite clearly in SVR and AOSMA's union, SVSM.

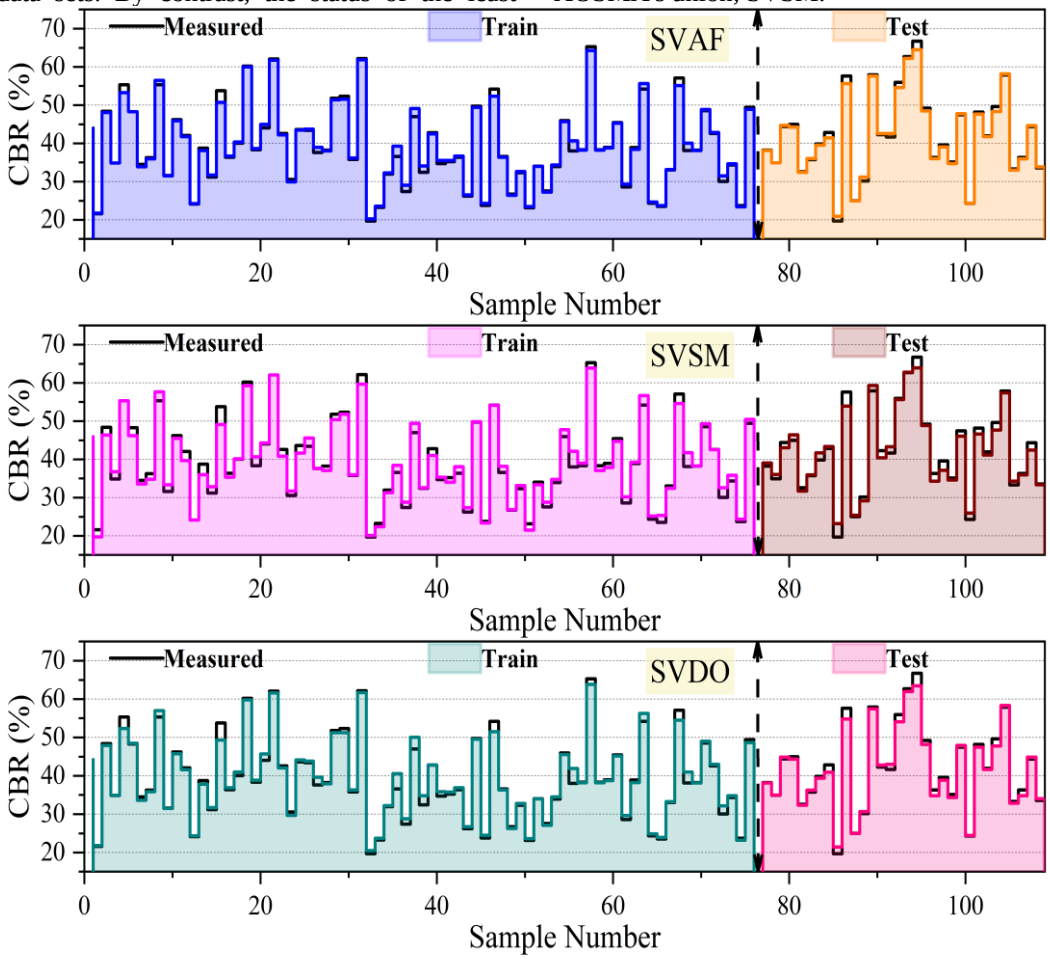


Figure 3: The comparison line-symbol plot between expected and gauged CS

Fig. 4 presents the deviations between the gauged and estimated values through three hybrid schemes regarding the California Bearing Ratio. This figure indicates that the greatest error for SVSM when assessed is around 18%, whereas for schemes undergoing training, it was 12% in the same set. The figure shows that, for the highest and lowest performing schemes, the majority of errors are found in a narrower range of (-3,3) % in SVAF and (-6,17%) % in SVSM.

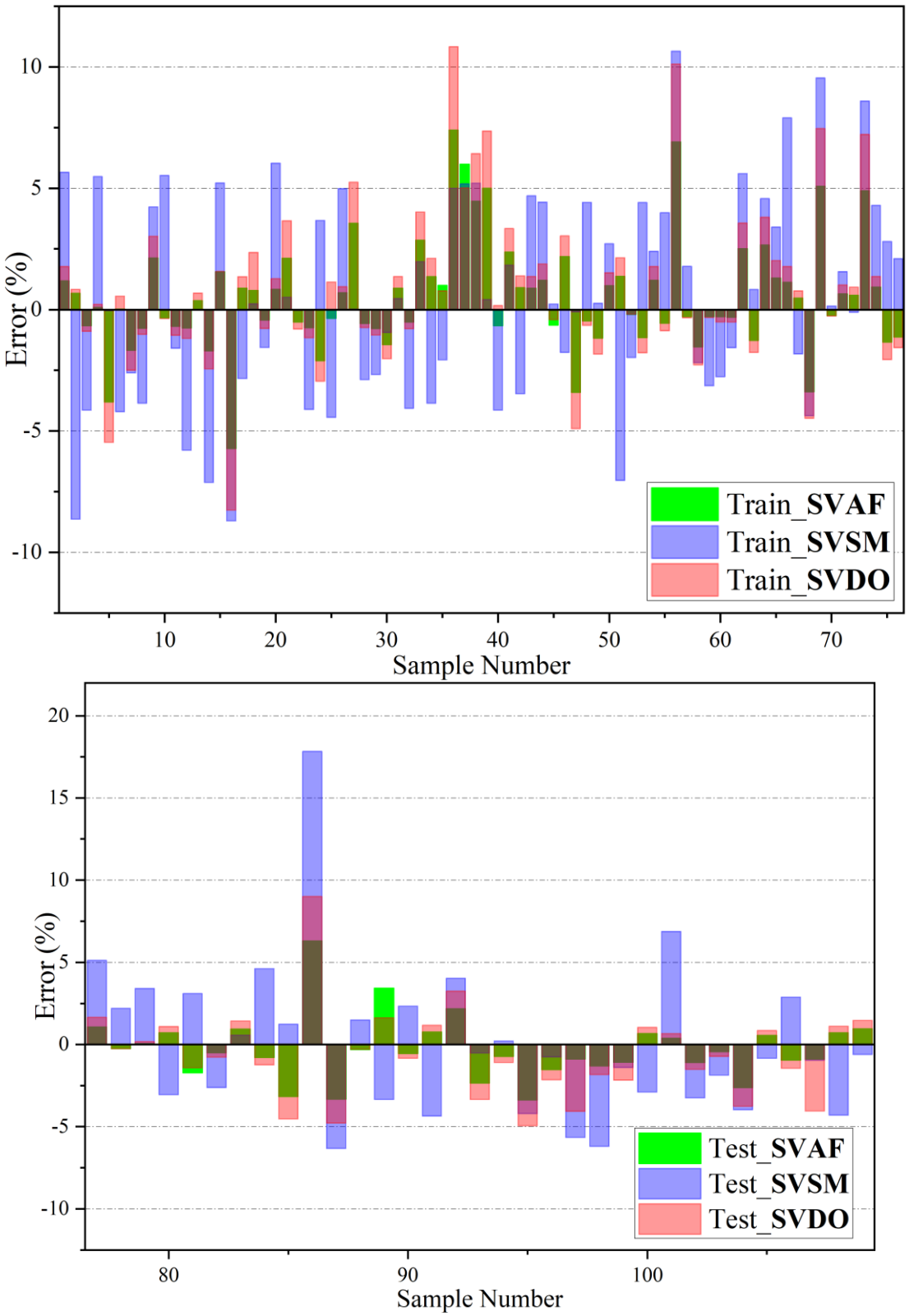


Figure 4: The error distribution of the schemes over samples shown in a time series plot.

The errors in the observed values of the undrained shear strength for the three different hybrid scheme types—SVAF, SVSM, and SVDO—are displayed in Fig. 5. Based on this figure, the maximum errors are about 11 %, and 7% for SVSM during training and testing of the schemes. The figure reflects the distribution of 25-75% of errors in a range less than (-1, 1) % in SVAF and (-3, 3) % in SVSM: best and worst schemes, respectively.

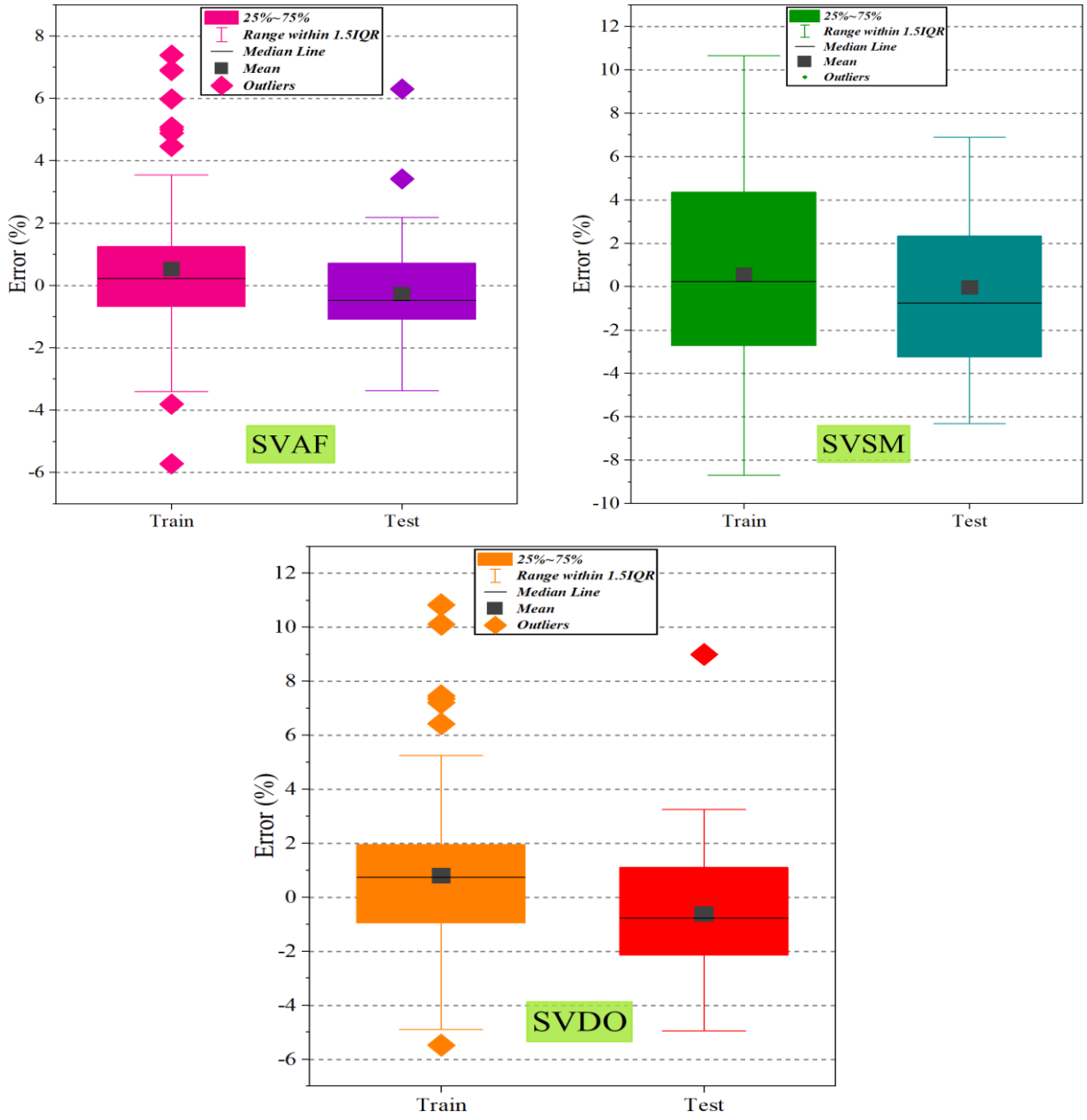


Figure 5: The standard half-box plot showing the error ratio of the hybrid schemes created.

To enhance the statistical robustness of the proposed models, 95% confidence intervals for the R^2 values were calculated based on multiple independent runs of each algorithm. As shown in Table 4, the standard SVR model has the widest interval, from 0.6302 to 0.7631, indicating greater variability and less predictive stability. In contrast, the three hybrid SVR models display narrower intervals with higher upper bounds, signifying more consistent performance. Among these, the SVR model combined with the Alibaba and Forty Thieves algorithms (SVAF) achieved the most favorable confidence interval, from 0.7243 to 0.8078, reflecting both high accuracy and robustness across runs. The SVR-Dingo Optimization Algorithm model also performed well, with a confidence interval of 0.7120 to 0.8298, slightly broader but with the highest upper bound. Meanwhile, the SVR-AOSMA model shows an interval between 0.6653 and 0.7848, ranking it between the other hybrids in stability and performance. These intervals confirm that the SVAF model not only offers high prediction accuracy but also delivers consistent results, making it the most reliable model among those tested for CBR estimation.

Table 4: Confidence intervals based on R2

Model	Lower Bound	Upper Bound
SVR	0.6303	0.7632
SVR + Dingo Optimization Algorithm	0.7120	0.8298
SVR + Adaptive Opposition Slime Mould Algorithm	0.6653	0.7848
SVR + Alibaba and the Forty Thieves	0.7243	0.8078

4 Sensitivity analysis

The ANOVA-based sensitivity analysis conducted on the performance of different predictive models for estimating the California Bearing Ratio (CBR) reveals statistically significant differences among the models. The confidence intervals for the coefficient of determination (R^2) provide insight into each model's accuracy and robustness. The baseline SVR model exhibits the lowest performance with a confidence interval ranging from 0.630 to 0.763, indicating relatively limited predictive power. In contrast, the SVR models enhanced with metaheuristic algorithms demonstrate superior performance. Among these, the SVR-Dingo Optimization Algorithm model shows a confidence interval between 0.712 and 0.830, reflecting substantial improvement over the baseline. Similarly, the SVR-Adaptive Opposition Slime Mould Algorithm model yields a confidence range of 0.665 to 0.785, suggesting better stability and generalization. Notably, the SVR-Alibaba and the Forty Thieves (SVAF) model achieves the highest lower bound (0.724) and an upper bound of 0.808, indicating both high precision and consistent performance. The limited overlap between the confidence intervals of the SVAF model and those of the other models supports the claim of its statistically significant superiority. This distinction highlights the effectiveness of the AFT optimizer in enhancing SVR's learning capability and minimizing prediction errors. Overall, the results of the ANOVA test confirm that metaheuristic-optimized SVR models, particularly SVAF, provide more accurate and reliable predictions of CBR values compared to the standard SVR approach.

Table 5: Sensitivity analysis based on ANOVA

Models	lower	upper
SVR	0.630	0.763
SVR-Dingo Optimization Algorithm	0.712	0.830
SVR-Adaptive Opposition Slime Mould Algorithm	0.665	0.785
SVR-Alibaba and the Forty Thieves	0.724	0.808

5 Discussion

This section compares the three hybrid models—SVAF (SVR + AFT), SVSM (SVR + AOSMA), and SVDO (SVR + DOA)—focusing on their predictive accuracy, convergence behavior, and computational efficiency. As shown in Table 2, SVAF outperforms the others across all five metrics: R^2, RMSE, MSE, RSR, and WAPE. During testing, SVAF achieved the highest R^2 (0.0.9968) and the lowest RMSE (0.7946), indicating excellent generalization and minimal error in estimating CBR values. This success stems from the adaptive balance between exploration and exploitation in the Alibaba and Forty Thieves (AFT) optimization strategy, which enhances SVR' s ability to find optimal hyperparameters. The random surveillance mechanism in AFT promotes global search, while Marjaneh's intelligence adjustment enhances local refinement, enabling rapid convergence toward optimal SVR settings. In contrast, the SVSM model, which employs the Adaptive Opposition Slime Mold Algorithm, showed weaker performance (R^2 = 0.9825, RMSE = 1.5824 during testing). Although AOSMA incorporates opposition-based learning to boost exploration, it can produce more oscillatory convergence patterns, possibly leading to suboptimal SVR tuning. Its complex adaptive threshold settings may also increase sensitivity to initial parameters. The SVDO model (SVR + Dingo Optimization Algorithm) performed moderately (R^2 = 0.992, RMSE = 1.171). DOA utilizes biologically inspired social hunting behaviors, facilitating effective neighborhood search. However, its slower convergence during exploitation may limit its ability to finely tune SVR hyperparameters, especially in high- dimensional spaces. Regarding computational efficiency, SVAF requires slightly more training time than SVSM and SVDO due to multiple adaptive conditions and surveillance cycles in AFT, but its superior accuracy justifies this. SVSM offers faster runtimes but less predictive precision. SVDO falls between the two in terms of performance and computational demand. Overall, findings suggest that SVAF provides the best balance between accuracy and optimization quality, making it a strong candidate for practical CBR prediction tasks. Future research could explore combining AOSMA 's rapid convergence with AFT 's stability to improve training efficiency without sacrificing accuracy. Future research will aim to improve the models' applicability across various regions by testing them on datasets with diverse soil types. Combining Support Vector Regression with deep learning—for example, as a post-processing tool after deep feature extraction—could boost prediction accuracy, particularly for large or complex datasets. Another valuable approach is integrating these hybrid AI models into geotechnical software platforms, allowing real-time, data-driven decision-making in engineering and construction projects.

Although the hybrid SVR models presented demonstrated strong predictive performance on the available dataset, there are some limitations to consider. Firstly, without an external validation set, the

generalizability of the results may be restricted beyond the current data. Secondly, the relatively small sample size increases the risk of overfitting, especially with the use of metaheuristic optimization. Additionally, the dataset only encompasses a limited range of soil types and regions, which could limit the models' broader applicability. It is also important to note that larger, more diverse datasets might benefit from alternative modeling techniques such as deep learning or ensemble methods to achieve better predictive accuracy. These limitations will be addressed in future research to improve the model's robustness and generalizability. To enhance model robustness, we plan to use regularization like L1/L2 penalties and early stopping to prevent overfitting. Models will be tested under various conditions—smaller datasets and more noise—to check resilience. Including confidence intervals or error margins for metrics like RMSE and R^2 will better measure uncertainty. These steps will help create more reliable, generalizable models for geotechnical uses.

6 Conclusion

The current investigation has adopted an SVR scheme to project the CBR value of soil. Although the outcomes of the conventional method were effective, it had some limitations. The laboratory process is costly and is not considered to be time-effective. The drawbacks above can be overcome by substituting the software-based approach with artificial intelligence. The accuracy of the system in predicting the CBR was quite remarkable. The input variables were selected to forecast the target parameter, which was depicted as CBR. Five different performance metrics were utilized to appraise the precision delivered by the schemes under consideration. These included R2, RMSE, MSE, RSR, and WAPE. Three distinct meta-heuristic optimization approaches—the Dingo Optimization Algorithm, Alibaba, the Forty Thieves Optimization algorithm, and the Adaptive Opposition Slime Mold Algorithm—have been examined in the current study to increase the system's functional efficiency. The conclusions below may be drawn from the analysis's outcome:

- The thorough analysis of the pertinent characteristics was the foundation for developing the projection schemes to estimate CBR. A comparison between the experimental outcomes and those obtained utilizing the suggested schemes showed that the latter's CBR prediction accuracy was significantly high.
- In the current research, the test phase has shown that the forecast data's scattering value increased by 0.39, 0.59, and 0.69 for SVAF, SVSM, and SVDO, respectively, from the training phase.
- The California Bearing Ratio outcomes presented in this investigation indicate a significant discrepancy between the observed and projected values, with an average underestimate of almost 1.24 for the suggested schemes. With a value of 1.6271, the RMSE displayed its maximum error in the scheme's SVSM in the training phase. The SVAF had the lowest error rate in the testing session, with a rating of 0.7946.

Acknowledgements

We wish to state that no individuals or organizations require acknowledgment for their contributions to this investigation.

Authorship contribution statement

Na Feng: Writing-Original initial drafting, Conceptualization, Supervision, Project administration.

Yulin Lan: Methodology, Software

Zhisheng Yang: Formal Analysis, Language Reviw

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Funding

This investigation was not funded by any specific grant from public, commercial, or charitable funding bodies.

Ethical approval

The paper has attained ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] A. Chegenizadeh and H. Nikraz, "CBR test on reinforced clay," in *Proceedings of the 14th Pan-American Conference on Soil Mechanics and Geotechnical Engineering (PCSMGE), the 64th Canadian Geotechnical Conference (CGC)*, Canadian Geotechnical Society, 2011.

[2] T. F. Kurnaz and Y. Kaya, "Prediction of the California bearing ratio (CBR) of compacted soils by using GMDH-type neural network," *The European Physical Journal Plus*, EPJ Plus, vol. 134, Jul. 2019. https://doi.org/10.1140/epjp/i2019-12692-0.

[3] H. B. Seed and P. De Alba, "Use of SPT and CPT tests for evaluating the liquefaction resistance of sands," in *Use of in situ tests in geotechnical engineering*, ASCE, 1986, pp. 281–302.

[4] M. Gams and T. Kolenik, "Relations between electronics, artificial intelligence and information society through information society rules," *Electronics (Basel)*, MDPI, vol. 10, no. 4, p. 514, 2021. https://doi.org/10.3390/electronics10040514.

[5] R. W. Day, *Soil testing manual*. McGraw-Hill, 2001.

[6] M. M. E. Zumrawi, "Prediction of CBR Value from Index Properties of Cohesive Soils.,"

University of Khartoum Engineering Journal, vol. 2, no. ENGINEERING, 2012.

[7] W. P. M. Black, "A method of estimating the California bearing ratio of cohesive soils from plasticity data," *Geotechnique*, ICE Virtual Library, vol. 12, no. 4, pp. 271–282, 1962. https://doi.org/10.1680/geot.1962.12.4.271.

[8] K. B. Agarwal and K. D. Ghanekar, "Prediction of CBR from plasticity characteristics of soil," in *Proceeding of 2nd South-east Asian Conference on Soil Engineering, Singapore. June*, 1970, pp. 11–15.

[9] M. Linveh, "Validation of correlations between a number of penetration test and in situ California bearing ratio test," *Transp Res Rec*, vol. 1219, pp. 56–67, 1989.

[10] D. J. Stephens, "The prediction of the California bearing ratio," *Civil Engineering= Siviele Ingenieurswese*, Sabnet, vol. 1990, no. 12, pp. 523–528, 1990. https://hdl.handle.net/10520/AJA10212019_14356.

[11] T. Al-Refeai and A. Al-Suhaibani, "Prediction of CBR using dynamic cone penetrometer," *Journal of King Saud University-Engineering Sciences*, Elsevier, vol. 9, no. 2, pp. 191–203, 1997. https://doi.org/10.1016/S1018-3639(18)30676-7.

[12] M. W. Kin, "California bearing ratio correlation with soil index properties," *Master degree Project, Faculty of Civil Engineering, University Technology Malaysia*, 2006. https://eprints.utm.my/4064/1/MakWaiKinMFKA2006.pdf.

[13] S. R. CNV and K. Pavani, "MECHANICALLY STABILISED SOILS-REGRESSION EQUATION FOR CBR EVALUATION," 2006.

[14] P. Vinod and C. Reena, "Prediction of CBR value of lateritic soils using liquid limit and gradation characteristics data," *Highway Research Journal, IRC*, vol. 1, no. 1, pp. 89–98, 2008.

[15] R. S. Patel and M. D. Desai, "CBR predicted by index properties for alluvial soils of South Gujarat," in *Proceedings of the Indian geotechnical conference, Mumbai*, 2010, pp. 79–82.

[16] G. Ramasubbarao and S. G. Sankar, "Predicting soaked CBR value of fine grained soils using index and compaction characteristics," *Jordan Journal of Civil Engineering*, vol. 7, no. 3, pp. 354–360, 2013.

[17] M. Alawi and M. Rajab, "Prediction of California bearing ratio of subbase layer using multiple linear regression models," *Road Materials and Pavement Design*, Taylor & Francis, vol. 14, no. 1, pp. 211–219, 2013. https://doi.org/10.1080/14680629.2012.757557.

[18] H. Ghanadzadeh, M. Ganji, and S. Fallahi, "Mathematical model of liquid–liquid equilibrium for a ternary system using the GMDH-type neural network and genetic algorithm," *Appl Math Model*, Elsevier, vol. 36, no. 9, pp. 4096–4105, 2012. https://doi.org/10.1016/j.apm.2011.11.039.

[19] M. A. Shahin, M. B. Jaksa, and H. R. Maier, "Artificial neural network applications in geotechnical engineering," *Australian geomechanics*, vol. 36, no. 1, pp. 49–62, 2001.

[20] J. A. Abdalla, M. F. Attom, and R. Hawileh, "Prediction of minimum factor of safety against slope failure in clayey soils using artificial neural network," *Environ Earth Sci*, Springer, vol. 73, pp. 5463–5477, 2015. https://doi.org/10.1007/s12665-014-3800-x.

[21] B. Yildirim and O. Gunaydin, "Estimation of California bearing ratio by using soft computing systems," *Expert Syst Appl*, Elsevier, vol. 38, no. 5, pp. 6381–6391, 2011. https://doi.org/10.1016/j.eswa.2010.12.054.

[22] Tja. Taskiran, "Prediction of California bearing ratio (CBR) of fine grained soils by AI methods," *Advances in Engineering Software*, Elsevier, vol. 41, no. 6, pp. 886–892, 2010. https://doi.org/10.1016/j.advengsoft.2010.01.003.

[23] S. Bhatt, P. K. Jain, and M. Pradesh, "Prediction of California bearing ratio of soils using artificial neural network," *Am. Int. J. Res. Sci. Technol. Eng. Math*, vol. 8, no. 2, pp. 156–161, 2014.

[24] T. Q. Ngo, L. Q. Nguyen, and V. Q. Tran, "Novel hybrid machine learning models including support vector machine with meta-heuristic algorithms in predicting unconfined compressive strength of organic soils stabilised with cement and lime," *International Journal of Pavement Engineering*, Taylor & Francis, vol. 24, no. 2, p. 2136374, 2023. https://doi.org/10.1080/10298436.2022.2136374.

[25] X. Wu, F. Lu, and T. He, "Exploring the potential of machine learning in predicting soil California bearing ratio values," *Periodica Polytechnica Civil Engineering*, vol. 69, no. 2, pp. 551–566, 2025. https://doi.org/10.3311/PPci.38678.

[26] V. Bherde, L. Kudlur Mallikarjunappa, R. Baadiga, and U. Balunaini, "Application of machine-learning algorithms for predicting California bearing ratio of soil," *Journal of Transportation Engineering, Part B: Pavements*, ASCE Library, vol. 149, no. 4, p. 4023024, 2023. https://doi.org/10.1061/JPEODX.PVENG-1290.

[27] J. Ma *et al.*, "A comprehensive comparison among metaheuristics (MHs) for geohazard modeling using machine learning: Insights from a case study of landslide displacement prediction," *Engineering Applications of Artificial Intelligence*, Elsevier, vol. 114, p. 105150, 2022. https://doi.org/10.1016/j.engappai.2022.105150.

[28] V. N. Vapnik, "The nature of statistical learning," *Theory*, 1995.

[29] V. Vapnik, "Statistical Learning Theory. New York: John Willey & Sons," *Inc*, 1998.

[30] M. K. Naik, R. Panda, and A. Abraham, "Adaptive opposition slime mould algorithm," *Soft comput*, Springer, vol. 25, no. 22, pp. 14297–

14313, 2021. https://doi.org/10.1007/s00500-021-06140-2.

[31] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, Elsevier, vol. 111, pp. 300–323, 2020. https://doi.org/10.1016/j.future.2020.03.055.

[32] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, Vienna, Austria, IEEE, 2005, pp. 695–701. https://doi.org/10.1109/CIMCA.2005.1631345.

[33] M. Braik, M. H. Ryalat, and H. Al-Zoubi, "A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves," *Neural Comput Appl*, Springer, vol. 34, no. 1, pp. 409–455, 2022. https://doi.org/10.1007/s00521-021-06392-x.

[34] A. K. Bairwa, S. Joshi, and D. Singh, "Dingo Optimizer: A Nature-Inspired Metaheuristic Approach for Engineering Problems," *Math Probl Eng*, Wiley Online Library, vol. 2021, p. 2571863, 2021. https://doi.org/10.1155/2021/2571863.

A Cutting-Edge Bio-Inspired Computational Framework for Advanced Virtual Reality Classification through Sophisticated Predictive Methodologies

Yanyan Song, Hongping Zhou*
[1] School of Communication Technology, Communication University of China Nanjing, Nanjing 211172, China
*Corresponding Author
E-mail: zhouhongping1231@163.com

Keywords: virtual reality, histogram gradient boosting classification, decision tree classification, ebola optimization search, differential squirrel search algorithm

Received: May 20, 2025

Virtual reality (VR) enables the simulation of a wide variety of complex environments, from tiny biological structures to entirely imaginary worlds. These simulations create new possibilities for learning, training, and interaction that go beyond the limits of the physical world. However, virtual reality (VR) realizes this imaginary world, so it is not just a dream. VR works through the invocation of many of the senses. It creates realistic simulations through the creation of immersive settings that combine the real and the imagined, thereby affording special hands-on learning possibilities in a variety of subjects. This study investigates the effectiveness of combining Histogram Gradient Boosting Classification (HGBC) with Decision Tree Classification (DTC), the Ebola Optimization Search (EOS), and the Differential Squirrel Search Algorithm (DSSA) to predict VR outcomes. By integrating these advanced predictive and optimization techniques, the approach aims to enhance accuracy. Research will be conducted to ascertain the possible uses of VR, enhance user experience, and assess the impact on industries related to training, education, healthcare, and entertainment. In the evaluation phase, HGDS attained the highest accuracy of 0.967 in the test phase, making it the top-performing hybrid model, while DTEO showed the lowest accuracy of 0.907, identifying it as the weakest model.

Povzetek: Članek predstavi bio-navdihnjen hibridni okvir za klasifikacijo uporabniških odzivov v virtualni resničnosti. Združuje HGBC, DTC ter optimizatorja EOS in DSSA za izboljšanje napovedne točnosti. Okvirjeva naloga je zanesljivo razvrščati VR-podatke.

1 Introduction

VR simulation signifies a computer-created environment where users can move around, interact with objects, and interact with virtual characters, also implied as "agents" or "avatars." A generic virtual setting is a 3D world [1], and, like gravitation simulation, virtual environments frequently aim to be as realistic as possible in both appearance and object behavior. It must be underlined, nonetheless, that there need to be no parallels between this virtual environment and the actual world. One of the advantages of virtual environments is their ability to replicate completely unrealistic scenarios [2]. Virtual environments, however, provide a safe space to test scenarios that would be too dangerous or difficult to perform in real life, and they imitate the setting where the student will eventually work.

There are other ways of deploying VR; four typical configurations are included below:

- ✓ Desktop VR (Monoscopic or Stereoscopic)
- ✓ Immersive VR (HMD, CAVE, widescreen)
- ✓ Collaborative Systems
- ✓ Mixed or Augmented Reality

The desktop VR enables the user to interact with the system using a mouse or other controlling device while sitting in front of a desktop computer monitor, as the name implies [3]. Immersion systems utilize a visualization display worn on the head of the user that completely occludes their field of view. Collaborative systems have human-controlled avatars interacting with each other, and they can be immersive or desktop-based systems. Second Life is one of the most recent and most effective collaboration systems [4]. An attempt is also being made to use the collaborative systems for exploration. The mixed reality systems merge computer-generated matter with the real environment, which is viewed directly or through a camera. This system can teach engineering and medical skills to students, which are thought to be impossible by this recently invented system [5].

Learning by humans requires interaction with the environment, taking in information provided by the use of senses and experience [6]. Through computer simulation, VR takes the role of real-world sensory input. Reacting to motion and common human behaviors in the actual world offers interaction. Therefore, VR can be useful in education since it allows pupils to experience a situation

or scenario firsthand rather than only imagining it [7]. The three main components that define the quality of VR experiences are immersion, interaction, and multisensory feedback. Immersion is being engulfed or enclosed by the surroundings [8]. One of the advantages of immersion is that it ensures a feeling of presence or the perception that one is actually in the world being displayed [9]. Interactivity means the capability of the user's body movements to affect the events happening in the simulation and, in turn, provoke a reaction from the simulation [10], [11].

The multisensory nature of VR allows information to be derived from several senses, which further enhances the experience in that this makes it both more engaging and more convincing—increasing, as it does, the sensation of presence because this provides redundancy of information, which diminishes the likelihood of misunderstanding. Information from multiple sensory entries is reinforced by a sensory combination [12], [13]. VR enables the user to act as though they are in the actual world by substituting a virtual environment for the current one. A constructivist learning approach benefits from VR's immersive features [14]. The premise of constructivism, a theory of knowledge acquisition, is that people build knowledge by concluding from their past experiences. The idea, as propounded by Jean Piaget, assumes that learners try to fit new experiences into the world picture that they have developed earlier. Learners change their worldview to fit the new experience when they cannot assimilate new information into their system effectively. Learning comes from experiences where actions are based on assumptions about how the world functions, only to find that it does not align with those assumptions [15], [16], [17]. Adjusting the mental model of how the world functions becomes necessary to account for the new experience. The view is that learning is an active process of testing hypotheses. In other words, this concept contrasts with the notion of learning as something passive in nature: the mere acquisition or assimilation of data. VR is a powerful learning tool because it provides a context where such hypothesis testing can occur. According to [18], students who interact with new material are more likely to store and recall it.

Control software is at the heart of this system. This regulates the exchange of information between the virtual world and the interface layer in response to user actions, updating the world appropriately. On display devices such as the haptic and visual interface, it also determines when the scene should be shown. With the help of additional tools, the control software can connect to the outside world through the internet, which might be an essential capability in systems involving collaboration or many users. The virtual environment module includes a model of real-world entities and the virtual world model. It includes state and position information apart from appearance. They could be dynamic objects, such as moving objects or even avatars. They could also be static objects. This model of the virtual environment needs to be refreshed at regular intervals to add dynamic objects [19]. The module for the virtual environment, which would store positions, shapes, and other attributes of all components of the virtual world in a database, is called the virtual environment module. The physics engine is one of the major parts of any realistic simulation. A physics engine comprises a set of rules that control the motion and interaction of dynamic objects in a virtual scene. A typical physics engine can include a Newtonian mechanics simulation and collision detection, which describes when two objects collide. They apply gravitational, friction, and impulse effects using physical rules. When two things hit each other, the latter effect is important [20], [21]. When two active entities collide, collision detection is necessary. The physics engine determines their terminal velocity using their simulated traits, such as mass, substance, and speed.

1.1 Related works

Normally, state-of-the-art reports that focus on specific aspects of the discipline or on specific application fields are available. They would mostly provide the taxonomies that systematically illustrate and classify the various methodologies involved.

- While Dachselt and Hübner [22] examined the menus for AR and VR environments for all of the MR domain [23], they also presented an extensive taxonomy.
- A taxonomy of NVEs, taking into consideration distribution and communication topologies, has been provided by Macedonia and Zyda [24]. Mania and Chalmers [25] have presented a taxonomy of platforms and communication.
- Bowman has provided several taxonomies for both interaction methods [26] and navigation methods [27]. Mine's early research [28] identifies the essential navigation and interaction in virtual spaces.
- Gabbard [29] provides good generalized overviews, presents suggested best practices in application design, and provides guides for conducting user evaluations. Livatino and Koeffel have also presented guidelines for Virtual Environments (VEs) assessment [30].
- The current tracking technology is overviewed by Welch and Foxlin [31], who also compare and contrast the respective merits and disadvantages of each.

Recent work has explored innovative methods for classifying virtual reality (VR) using bio-inspired computational models. Song and DiPaola [32] introduced a bio-responsive VR system based on physiological data to enhance immersion. Zayed and Reda [33] demonstrated that applying neurophysiological biosignals combined with deep learning could classify cognitive states in VR with 97% accuracy. Similarly, Arslan et al. [34], [35], [36], [37], [38] employed emotion classification from biosignals and machine learning in VR, achieving 97.78% accuracy. These advancements are significant in areas such as rehabilitation, education, and psychotherapy. VanHorn and Çobanoğlu [35] also developed a biomedical image classification system within a VR-based environment, making AI more accessible to experts.

Overall, these studies emphasize how biosignals, machine learning, and VR can be integrated to develop advanced predictive models, showcasing the potential of bio-inspired computational models to improve VR classification techniques.

1.2 The study's objective

This work examines the possible contribution that VR technology will make to enhance learning outcomes and increase student engagement in schools. In data classification, this study applies an HGBC model and a DTC model. The performance of the schemes is optimized by using methods such as EOS and DSSA. This research will explore the integration of VR within diverse disciplines of study to understand how it can facilitate the retention of both theoretical knowledge and practical competencies of learners, given the immersion one experiences in a VR environment. Possible drawbacks and limitations, including accessibility of resources, shall also be discussed to present a comprehensive overview of what can be expected from this educational technology.

2 Materials and methodology

2.1 Data gathering

A set of users' experiences in VR settings provides the dataset. The information covers user preferences, emotional moods, and physiological reactions like skin conductance and heart rate. This study's dataset includes 1000 samples, each representing a user's VR session. Recorded features encompass User ID (173 unique values), Age (66), Gender (147), VR Headset type (61), Session Duration (137), Motion Sickness severity (56), and Immersion Level (55). These variables cover both demographic and behavioral data, forming a comprehensive basis for analysis. The Immersion Level serves as the target variable, indicating users' subjective engagement in VR, and its variability supports the creation of effective predictive models.

This dataset attempts to contribute to the development of VR through the analysis of user experiences. An attempt has been made in this study to develop a better VR design with much more improvement in user comfort and customization by understanding the physical and emotional reactions of consumers in diverse VR situations. This information allows developers to work on boosting VR systems and creating personalized experiences that will enhance customers' delight and immersion. Fig. 1 presents a contour plot for the correlation of the features.

User ID: This variable identifies every participant who experienced VR. Each user is assigned a unique ID so that their data in the dataset can be differentiated.

Age: This variable stores the age of the subject participating in VR exposure. For example, this could be an integer representing the current user's age at the time of using the VR.

Gender: This variable displays the user's gender. The categories "Male," "Female," and "Other" can be utilized to define the user's gender identity.

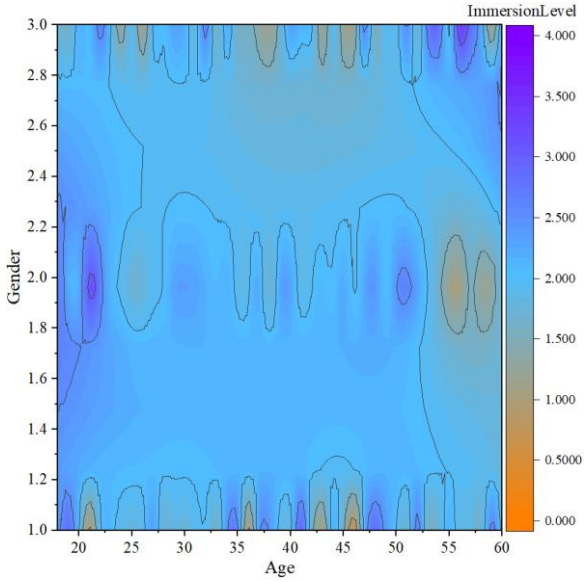
VR Headset Type: This would be a variable specifying the form of VR headset that a user is utilizing in a VR experience. Examples include Oculus Rift, HTC Vive, and PlayStation VR, among others.

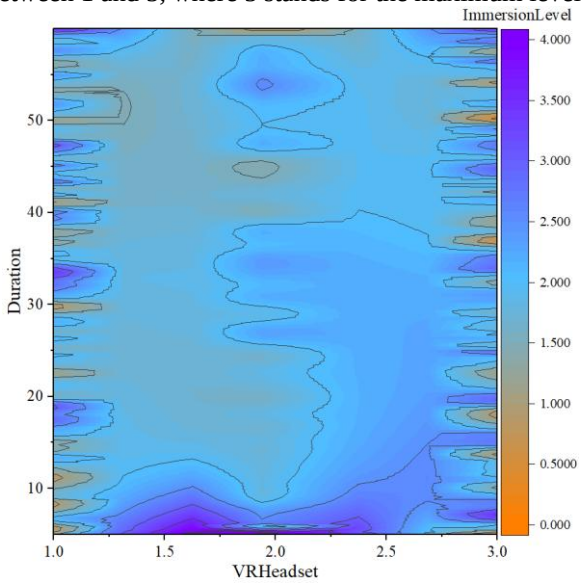
Duration: This variable shows the time spent in the VR experience in minutes. It reflects how much time was spent by the participant in the VR setup.

Motion Sickness Rating: It displays the rating of the user's self-reported motion sickness during the VR experience. Higher numbers relate to a higher degree of motion sickness on an ascending scale ranging between 1 and 10.

Dependent variable:

The degree to which a user experiences being inside a virtual environment quantifies the subjective degree of the user's feeling of immersion in the experience, with a rating between 1 and 5, where 5 stands for the maximum level.





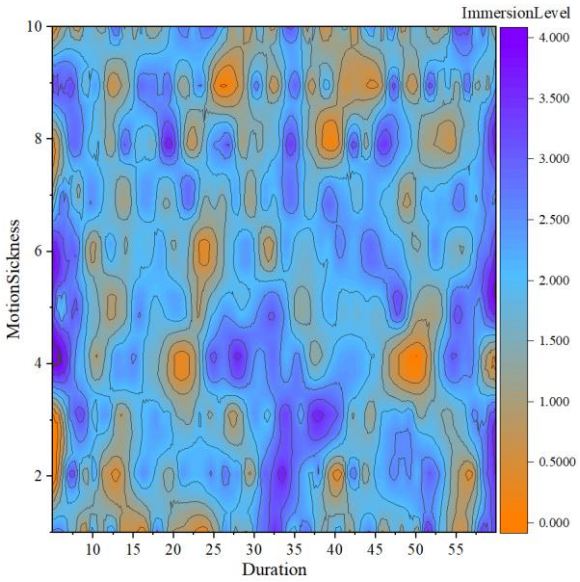


Figure 1: The contour plot with color fill illustrates the relationship between input and output variables

Before deploying advanced computational models, understanding several challenges in VR classification systems is essential. These encompass the significant variability in user responses driven by individual physiological and psychological differences, noise in biometric data such as heart rate and skin conductance, and class imbalance across different immersion levels. The subjective nature of immersion also complicates labeling and impacts the consistency of ground truth. These factors result in a complex, high-dimensional feature space where traditional classifiers often face difficulties with generalization and robustness. Consequently, adopting adaptive hybrid machine learning approaches, supported by powerful metaheuristic optimization techniques, is crucial for effective classification in VR.

2.2 Histogram gradient boosting classification (HGBC)

The HGB approach is another variant of the popular GB [39] technique used to resolve diverse classification and regression-oriented machine learning (ML) problems. These schemes, which AdaBoost also belongs to, primarily try to turn weak learners into strong ones. They come under the category of schemes called boosting schemes. Boosting techniques try to keep adding and teaching new weak learners successively to correct their previously introduced weak learners about their mistakes. It then informs each new weak learner to avoid the mistakes made by its forerunner. The most common weak learners used are DTs. This resulted in the development of the HGB algorithm, a boosting methodology that overcame one of the major weaknesses of the GB algorithm, which was its very long training time when training on large datasets. To circumvent this problem, the continuous input variables are discretized or binned into a few hundred distinct values. In this case, the learning rate (LR) of the scheme is the most important hyperparameter. Much attention was paid to the optimization of the scheme through several iterations of hyperparameter tweaking. The implementation of HGB from sci-kit-learn 0.21.3 was used from the Python ML module [40].

2.3 Decision tree classification (DTC)

In a DT, every internal node displays a characteristic, each branch is a decision rule, and each leaf node is the outcome [41]. The root node signifies the topmost node in a DT. To achieve the best discrimination among classes or results, it learns to split based on the value of an attribute. Different schemes have different criteria regarding making decisions. For example, some of the metrics used by schemes like ID3, C4.5, and CART include entropy, gain ratio, and Gini impurity, respectively. The problem at hand now is to find that characteristic at every level that offers the optimum split in a DT, thereby assisting optimum decision-making [42]. The concept can be mathematically understood by using the DT split based on entropy. The entropy H(D) of a dataset D can be calculated as follows:

$$H(D) = -\sum_{i=1}^{m} p_i log_2 p_i \quad (1)$$

2.4 Ebola optimization search (EOS)

Driven by the diffusion of the Ebola virus, in what follows, EOS presents a metaheuristic scheme [43]. The EOSA scheme is based on the enhanced SIR scheme of the sickness. Its S, E, I, R, H, V, Q, and D compartments represent the Susceptible (S), Exposed (E), Infected (I), Hospitalized (H), Recovered (R), Vaccinated (V), Quarantine (Q), and Death (D) states, respectively. Because of these compartments, the composition provides for the construction of a search domain that best displays combinations of weights and biases that may be required by CNN. After representation, SIR is displayed by a mathematical scheme utilizing a system of first-order differential equations. Then, the new metaheuristic scheme was developed by combining the mathematical

and propagation schemes, and later, the obtained mathematical scheme was deployed in the design of EOSA-CNN for experimentation. Therefore, the following are the mathematical schemes:

$$mI_i^{t+1} = mI_i^t + \rho M(I) \quad (2)$$

$$\frac{\partial S(t)}{\partial t} = \pi - (\beta_1 I + \beta_3 D + \beta_4 R + \beta_2 (PE)\eta)S - (\tau S + \Gamma I) \quad (3)$$

$$\frac{\partial I(t)}{\partial t} = (\beta_1 I + \beta_3 D + \beta_4 R + \beta_2 (PE)\lambda)S - (\Gamma + \gamma)I - (\tau)S \quad (4)$$

$$\frac{\partial H(t)}{\partial t} = \alpha I - (\gamma + \varpi)H \quad (5)$$

$$\frac{\partial R(t)}{\partial t} = \gamma I - \Gamma R \quad (6)$$

$$\frac{\partial V(t)}{\partial t} = \gamma I - (\mu + \vartheta)V \quad (7)$$

$$\frac{\partial D(t)}{\partial t} = (\tau S + \Gamma I) - \delta D \quad (8)$$

$$\frac{\partial Q(t)}{\partial t} = \big(\pi I - (\gamma R + \Gamma D)\big) - \xi Q \quad (9)$$

mI_i^{t+1} and mI_i^t display the old and new situation at time t and $t+1$, respectively, ρ is the displacement scale factor of an individual in Eq. (2). The data updated here are Hospitalized (H), Vaccinated (V), Recovered (R), Infected (I), Susceptible (S), Quarantine (Q), and Dead (D). Eqs. (3) to (9) define a system of ordinary differential equations, all of the scalar functions that one can evaluate to float values. These are computed given initial conditions $S(0) = S_0, I(0) = I_0, R(0) = R_0, D(0) = D_0, P(0) = P_0$, and $Q(0) = Q_0$, t is after the definition of iterations. This will then enable us to conclude the magnitude of vectors S, I, H, R, V, D, and Q at t.

The pseudocode that describes the EOSA metaheuristic scheme is presented accordingly in steps below:

- Define initial values for all vector and scalar quantities, that is, persons and parameters, respectively: the numbers of hospitalized (H), vaccinated (V), susceptible (S), infected (I), recovered (R), dead (D), and quarantined (Q).
- I_1 is created at random among vulnerable people.
- The value of fitness shall be calculated at the index case, having set that as the current and global best.
- If there is at least one infected person and the number of iterations is not reached, then:

a) With every vulnerable individual, a standing is created and altered accordingly with their movement. Exploitation is characterized by short displacement; otherwise, it characterizes exploration. Remember that the longer an infected case is displaced, the more infections there are.
b) Using (a), generate newly infected individuals nI.
c) Create the new individuals and add the new instances in I.
d) From the size of I calculate how many people are added to H, D, R, B, V, and Q at their respective rates.
e) Utilizing the new I, refine S and I.
f) Compare the best I have got at the moment with the best in the world.
g) If the termination condition is not reached, go back to step 4.

- Return all solutions and the best global resolution.

The design and discussion of the utilization of the enhancement issue defined in this paper are given in the following subsections.

Fig. 2 presents the flowchart of the DTC.

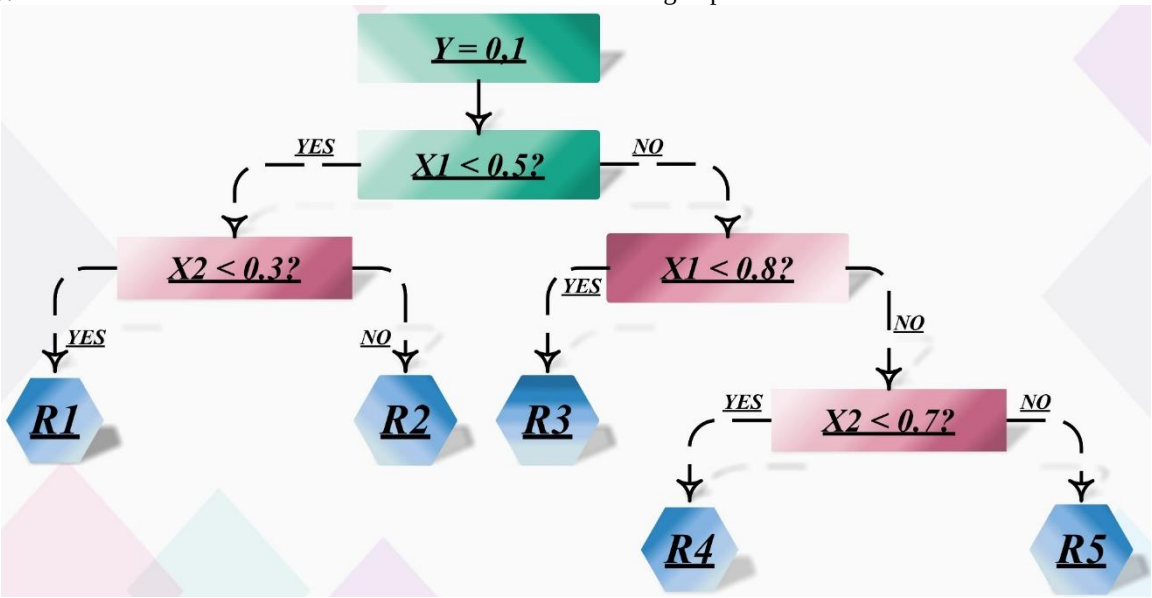


Figure 2: The flowchart of the DTC model

2.5 Differential squirrel search algorithm (DSSA)

$DSSA$, a hybrid optimizer that combines the differential evolution and squirrel search schemes is presented in this section. In SSA, the squirrels maintain the position of other squirrels regarding acorn or hickory trees for updating their position. To improve its search strategy, the top squirrels' position updating rules have been changed. The incorporation of crossover operations inspired by DE significantly enhances the exploration capability.

The following is a mathematical scheme of many foraging techniques covered under the paradigm of DSSA.

To justify selecting EOS and DSSA for this classification task, it's crucial to highlight the problem's nature: the dataset involves multiple interacting features with complex, nonlinear relationships, which can cause optimization to get stuck in local optima when using traditional methods. The EOS algorithm, inspired by epidemic modeling, employs dynamic, population-based exploration techniques that balance infection-driven diversification with recovery-focused convergence. This strategy is especially effective for tuning hyperparameters in complex models like HGBC and DTC. Its compartmental diffusion model efficiently captures multidimensional search dynamics. Meanwhile, DSSA mimics squirrel foraging behavior and utilizes crossover inspired by differential evolution, making it highly effective at fine-tuning solutions locally while maintaining overall diversity. This capability is critical in VR classification scenarios, where high accuracy requires careful adjustment of sensitive parameters to prevent overfitting. DSSA's ability to retain elite solutions while fostering diversity helps avoid premature convergence.

$$PS_{at}^{new} = \begin{cases} PS_{at}^{old} + d_g \times G_c\left(PS_{ht}^{old} - PS_{at}^{old} - P_{avg}\right), & r_1 \geq P_{dp} \\ random\ position, & otherwise \end{cases} \quad (10)$$

whereas P_{avg} is the mean location of every squirrel in the current population.

It also employs the crossover mechanism of DE in a way that ensures diversity among squirrels to the maximum while minimizing the possibility of trapping in local minima. Applied to the squirrel's current position and the new position as obtained by Eq. (11):

$$PS_{at,i,j}^{cr} = \begin{cases} PS_{at,i,j}^{new}, & if\left(rand_j \leq Cr\right) or\ j = j_{rand} \\ PS_{at,i,j}^{old}, & if\left(rand_j > Cr\right) or\ j \neq j_{rand} \end{cases}, j = 1, 2, 3, \ldots, D \quad (11)$$

In this context, NP displays the population size, with i ranging from $1, 2, 3, \ldots, NP$. For acorn or normal trees, $PS_{at;i;j}^{cr}$ indicates the updated positions of the squirrels following the crossover operation. $PS_{at;i;j}^{new}$ and $PS_{at;i;j}^{old}$ correspond to the new and previous positions of the squirrels. D refers to the dimensionality of the problem, and Cr displays the crossover rate, which is set to 0.5. The index j_{rand} is randomly selected from the range $[1, D]$, and $rand_j$ denotes the jth random number, uniformly generated within this range.

Combining EOS and DSSA offers complementary advantages, EOS facilitates broad exploration, while DSSA ensures precise convergence, together enhancing classification accuracy and model robustness for VR immersion prediction.

2.5.1 Initialization of position and evaluation of fitness

The squirrels are initially placed in the search area at random. Knowing the squirrels' location allows one to calculate their fitness, which simply replaces their position in the fitness function by demonstrating how good a food supply they could find. The best squirrel PS_{ht} discovered in the hickory tree thus far is determined by sorting fitness values. It is thought that the squirrels in the acorn tree $PS_{at}(1:3)$ are traveling in the direction of the optimal location in a subsequent iteration, as indicated by the following three best function values. The remaining squirrels, $PS_{nt}(1: NP - 4)$, are in the typical tree and have not yet discovered food.

2.5.2 Position update

The squirrels in an acorn tree, following the current best, PS_{ht}, renew the position, and move in the direction of the best source when there is no predator. The squirrels of a usual tree follow the ones in an acorn or hickory tree to renew their position. If there is the presence of a predator, then the squirrels change direction randomly while foraging. These are the mathematical schemes that are used to update the squirrel's position.

As in Eq. (10) now, the posture of squirrels on acorn trees changes based on the postures of others.

Some of the squirrels on regular trees do the placement of acorn tree squirrels, after which they relocate to their new locations.

$$PS_{nt}^{new} = \begin{cases} PS_{nt}^{old} + d_g \times G_c(PS_{at}^{old} - PS_{nt}^{old}), r_2 \geq P_{dp} \\ random\ posiion, \quad otherwise \end{cases} \quad (12)$$

where the random integer r_2 is uniformly distributed between 0 and 1.

In normal trees, the survivors cling on to the best move on view, and their new positions are shown below:

$$PS_{nt}^{new} = \begin{cases} PS_{nt}^{old} + d_g \times G_c\left(PS_{ht}^{old} - PS_{nt}^{old}\right), r_3 \geq P_{dp} \\ random\ posiion, \quad otherwise \end{cases} \quad (13)$$

The following crossover procedure is also given for typical tree squirrels:

$$PS_{nt,i,j}^{cr} = \begin{cases} PS_{nt,i,j}^{new}, & if\left(rand_j \leq Cr\right) or\ j = j_{rand} \\ PS_{nt,i,j}^{old}, & if\left(rand_j > Cr\right) or\ j \neq j_{rand} \end{cases}, \quad j = 1, 2, 3, \ldots, D \quad (14)$$

The convergence speed may be raised by permitting the hickory tree squirrel to update her location in relation

to the average position of the squirrels in the tree. This can be done as follows:

$$PS_{ht}^{new} = PS_{ht}^{old} + d_g \times G_c(PS_{ht}^{old} - PS_{at}^{avg}) \quad (15)$$

In this instance, PS_{avg} displays the average of all squirrel locations within the acorn trees.

In order to participate in the next generation of people, the best aspects of the new work, as well as its crossover roles, are then contrasted with the old jobs.

Figure 3 illustrates the flowchart of the proposed hybrid models (such as HGDS and DTEO), detailing the sequential phases that encompass data input, model development, optimizer-centric hyperparameter optimization, training, and final assessment. This diagram delineates the interaction between machine learning models and metaheuristic optimizers within the hybrid structure.

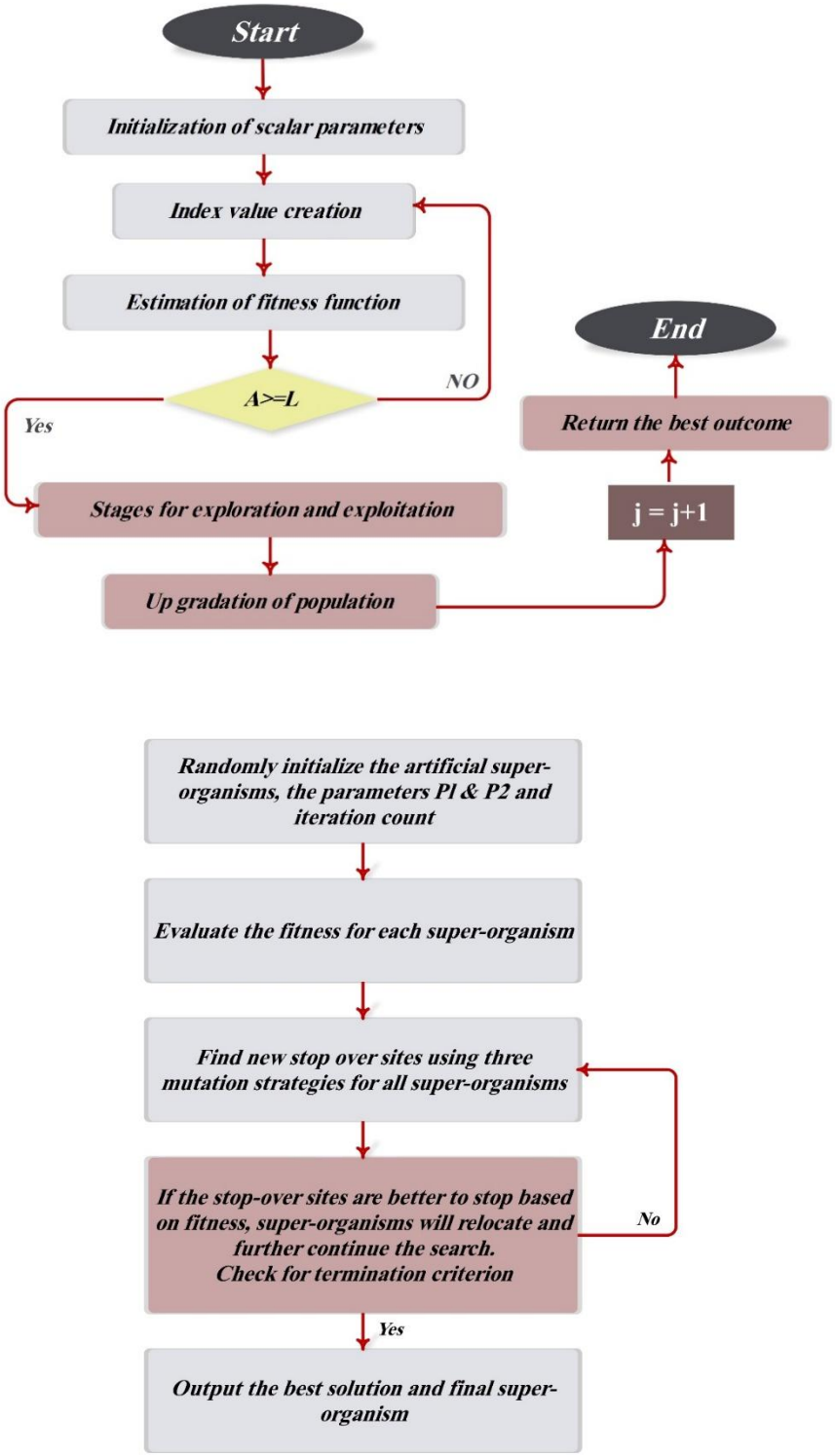


Figure 3: The process flowchart of the proposed hybrid models

2.6 Performance evaluators

Accuracy depends on how many correctly projected positive and negative instances there are of the total, defined by True Positives (TP), True Negatives (TN)—correctly projected negative cases, False Positives (FP)-incorrectly projected as positive, and False Negatives (FN)—incorrectly projected as negative. Using TP and FP as the relevant measures, precision gauges the percentage of TP projections out of all the positive projections the model has made. Smaller amounts of false positives imply higher precision. Recall is the measure of the share of TP projections from all real positive instances, using True Positives and False Negatives; it indicates that the model will detect all relevant positive cases. The fewer false negatives there are, the higher the recall. A simple statistic that balances the trade-off between Precision and Recall is the F1 score. It combines the two.

$$Accuracy: \frac{TP+TN}{TP+FP+FN+TN} \tag{16}$$

$$\text{Precision}: \frac{TP}{TP+FP} \tag{17}$$

$$\text{Recall}: \frac{TP}{TP+FN} \tag{18}$$

$$\text{F1}-\text{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{19}$$

The F1-score is a single measure that balances the accuracy and recall; it is the harmonic mean of both. It is very useful when considering false negatives and false positives. The greater the F1 score, the better balanced the recall and accuracy.

3 Results and discussion

3.1 Hyperparameters tuning and convergence curve analysis

The presented table displays the tuned hyperparameters for four different hybrid models: HGEO, HGDS, DTEO, and DTDS. Seven key hyperparameters were considered to optimize these models' performance: learning_rate, max_leaf_nodes, max_depth, min_samples_leaf, max_bins, min_samples_split, and a second instance of max_leaf_nodes (listed separately for different model types). The HGEO and HGDS models, based on the HGBR algorithm, have specified values for learning_rate, max_leaf_nodes, max_depth, min_samples_leaf, and max_bins. For example, the HGEO model has a learning_rate of 0.709, max_leaf_nodes of 278, max_depth of 100, min_samples_leaf of 10, and max_bins of 27. In the HGDS model, these values are learning_rate of 0.148, max_leaf_nodes of 557, max_depth of 893, min_samples_leaf of 7, and max_bins of 102. Conversely, the DTEO and DTDS models, which are based on decision tree algorithms, do not include values for learning_rate, max_leaf_nodes, or max_bins in the first part of the table. However, they include defined values for max_depth, min_samples_leaf, min_samples_split, and max_leaf_nodes in the second part. For instance, the DTEO model has max_depth of 741, min_samples_leaf of 0.00025, min_samples_split of 0.0275, and max_leaf_nodes of 2710. Similarly, the DTDS model features max_depth of 597, min_samples_leaf of 0.00025, min_samples_split of 0.0005, and max_leaf_nodes of 1789. Overall, the table indicates that hyperparameters are selectively tuned for each model based on its structure, with parameter values chosen according to each model's specific characteristics and requirements.

Fig. 4 displays a 3D waterfall plot illustrating the convergence curves of four hybrid schemes: HGDS, HGEO, DTDS, and DTEO. The plot effectively visualizes the different convergence rates and final performance levels of the schemes, demonstrating the varying degrees of effectiveness in the optimization process. This comparison emphasizes the significance of the number of iterations and initial accuracy in determining the overall success of each hybrid model. The HGDS model starts with an accuracy of 0.6 and gradually improves over 200 iterations, ultimately reaching a peak accuracy of 0.967, making it the highest-performing model among the four. The other three schemes begin with a lower accuracy of 0.4 and converge more quickly than HGDS, reaching their final accuracy in fewer iterations. Among these schemes, DTEO is identified as the weakest hybrid model, with a final accuracy of 0.908 after its iterations.

Table 1: Hyperparameter tuning for four models

Hyperparameters	***Models***			
	HGEO	***HGDS***	***DTEO***	***DTDS***
learning_rate	0.709	0.148	-	-
max_leaf_nodes	278	557	-	-
max_depth	100	893	741	597
min_samples_leaf	10	7	0.00025	0.00025
max_bins	27	102	-	-
min_samples_split	-	-	0.0275	0.0005
max_leaf_nodes	-	-	2710	1789

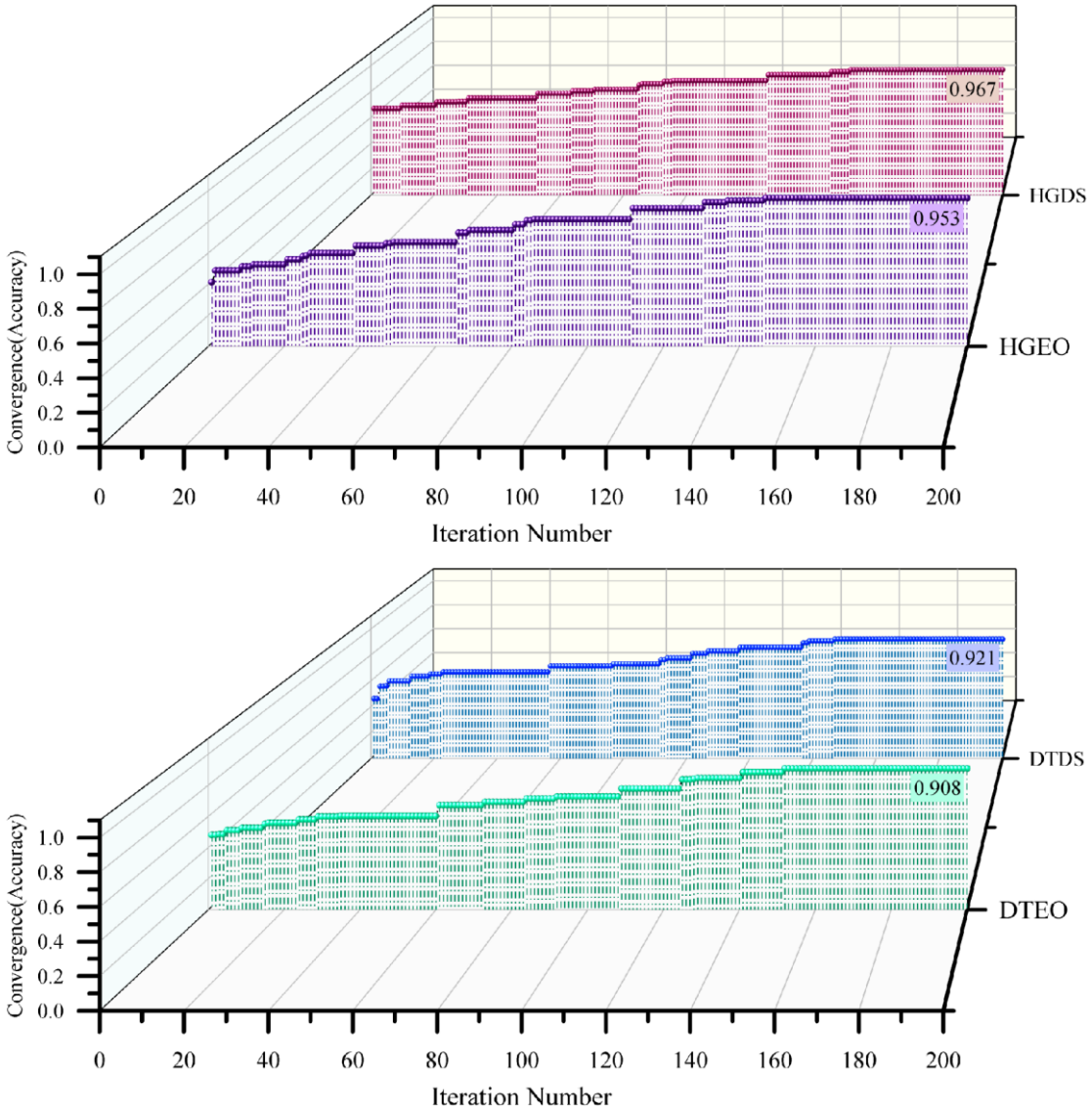
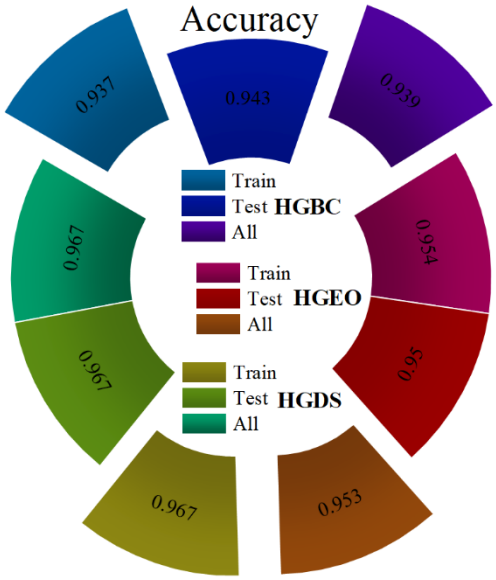
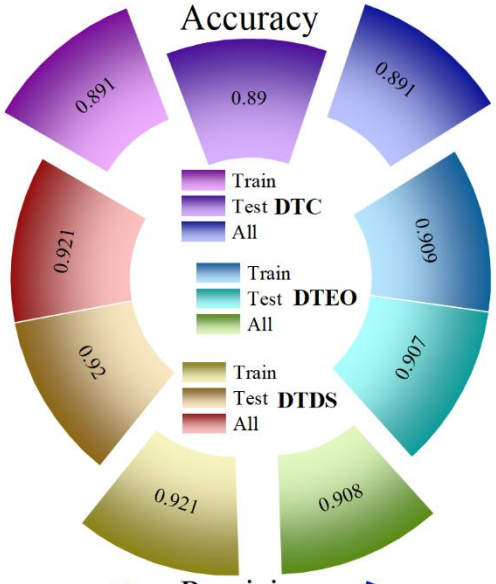


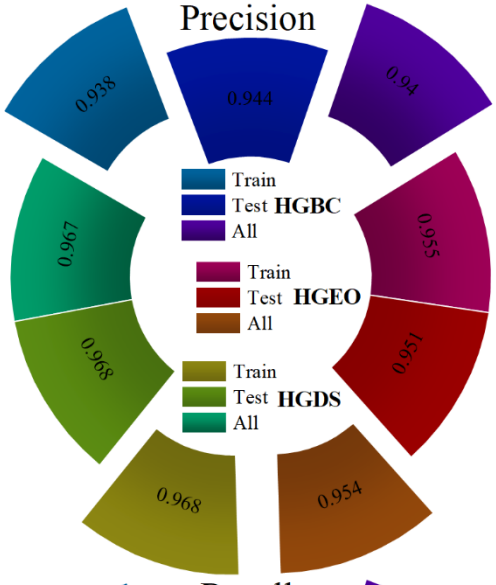
Figure 4: 3D waterfall plot for the convergence curve of the hybrid schemes

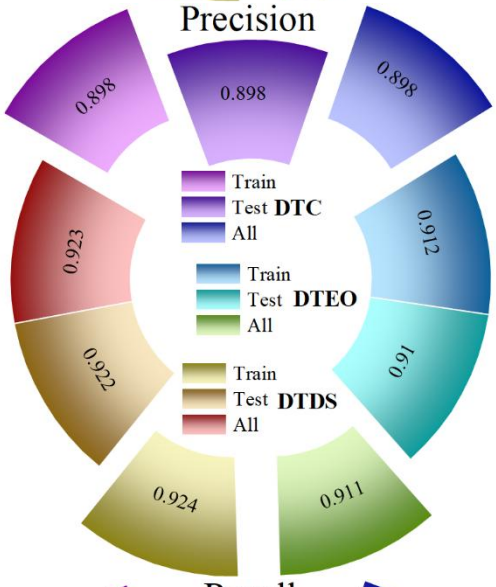
3.2 Schemes performance comparison

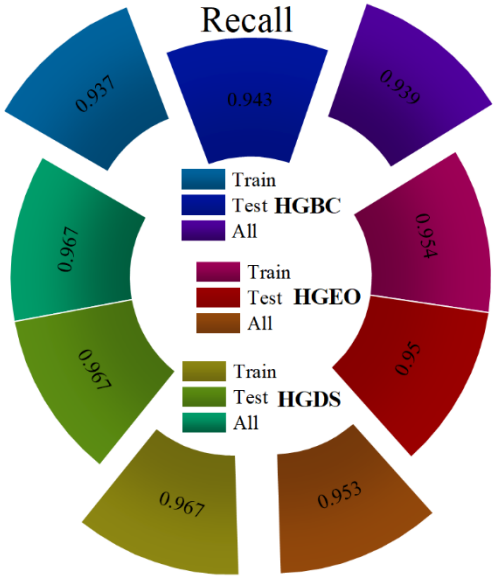
Fig. 5 presents a doughnut plot, providing an intuitive representation of the schemes' performance and facilitating a clearer comparison across different evaluation metrics. The performance results of six hybrid schemes evaluated using accuracy, precision, recall, and F1 scores across training, testing, and overall sections have been presented. Among these, HGDS emerges as the best-performing model, with an impressive accuracy of 0.967 in the test section. Conversely, DTEO, with an accuracy of 0.907, is the weakest model. HGDS outperforms HGEO by 0.17 in accuracy, establishing itself as the top model. Nevertheless, HGEO still demonstrates strong performance, securing the second-best position overall. This comparison underscores the varying strengths of each model, with HGDS leading in accuracy and other performance metrics, while HGEO, despite its lower accuracy, remains a competitive alternative. The results emphasize that even schemes with slightly lower accuracy can still offer valuable performance in certain contexts.

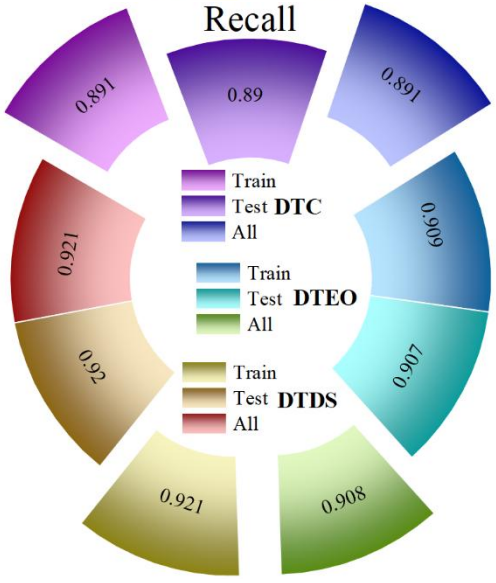

Accuracy
0.937
0.943
0.939
0.967
0.954
0.967
0.95
0.967
0.953
Train
Test HGBC
All
Train
Test HGEO
All
Train
Test HGDS
All


Accuracy
0.891
0.89
0.891
0.921
0.909
0.92
0.907
0.921
0.908
Train
Test DTC
All
Train
Test DTEO
All
Train
Test DTDS
All


Precision
0.938
0.944
0.94
0.967
0.955
0.968
0.951
0.968
0.954
Train
Test HGBC
All
Train
Test HGEO
All
Train
Test HGDS
All


Precision
0.898
0.898
0.898
0.923
0.912
0.922
0.91
0.924
0.911
Train
Test DTC
All
Train
Test DTEO
All
Train
Test DTDS
All


Recall
0.937
0.943
0.939
0.967
0.954
0.967
0.95
0.967
0.953
Train
Test HGBC
All
Train
Test HGEO
All
Train
Test HGDS
All


Recall
0.891
0.89
0.891
0.921
0.909
0.92
0.907
0.921
0.908
Train
Test DTC
All
Train
Test DTEO
All
Train
Test DTDS
All

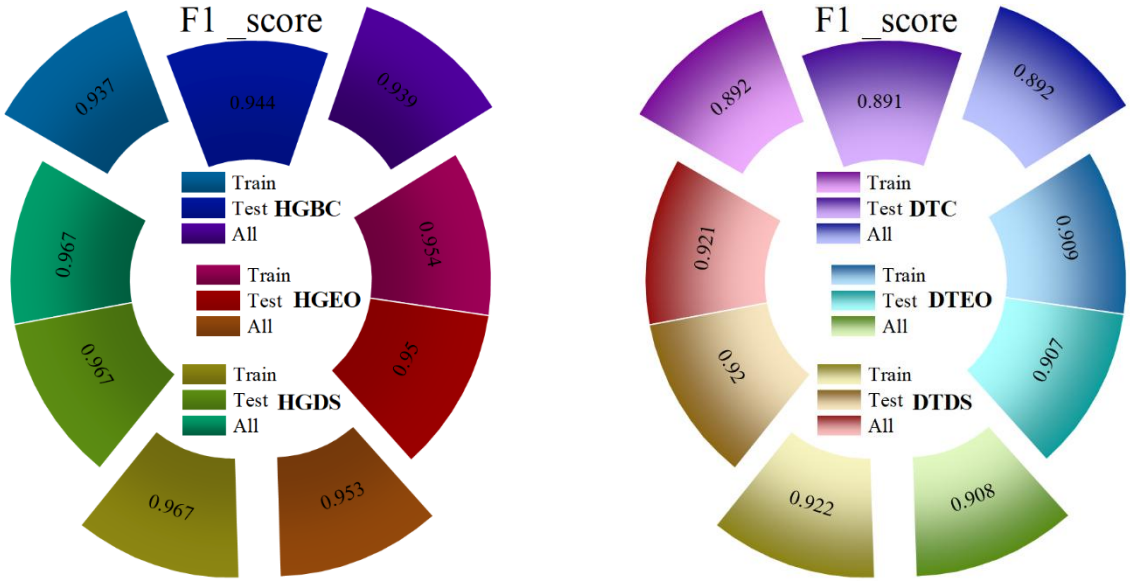


Figure 5: A connected doughnut plot employed for the visual evaluation of the schemes' performance

Additionally, Table 2 provides a summary of the performance of six schemes across five levels regarding precision, recall, and F1 score. The hybrid model HGDS stands out at level 1, achieving the highest precision of 0.990. Additionally, HGDS excels in both recall and F1-score, outperforming all other schemes and demonstrating its overall robustness. In contrast, DTEO shows weaker recall performance compared to the other schemes, although it surpasses DTC in this metric. Regarding the F1-score, DTEO records a value of 0.922, which is lower than the top-performing schemes. Nonetheless, it outperforms both DTDS and CTC by margins of 0.013 and 0.010, respectively. While DTEO's F1-score may not be the highest, it still demonstrates competitive performance relative to other schemes. These findings indicate that HGDS is the most well-rounded and effective model overall, while DTEO, despite its limitations in recall and F1 score, delivers superior performance in specific areas.

Table 2: Schemes' evaluation results through different immersion levels

Evaluators	Schemes	Immersion Levels				
		Level 1	Level 2	Level 3	Level 4	Level 5
Precision	HGBC	0.946	0.946	0.909	0.925	0.973
	HGEO	0.941	0.995	0.907	0.951	0.974
	HGDS	0.974	0.990	0.981	0.945	0.943
	DTC	0.988	0.938	0.825	0.909	0.822
	DTEO	0.973	0.929	0.847	0.923	0.878
	DTDS	0.906	0.939	0.873	0.914	0.983
Recall	HGBC	0.951	0.933	0.928	0.952	0.932
	HGEO	0.946	0.947	0.959	0.947	0.969
	HGDS	0.960	0.971	0.985	0.956	0.963
	DTC	0.847	0.875	0.953	0.869	0.916
	DTEO	0.876	0.885	0.948	0.927	0.906
	DTDS	0.911	0.894	0.964	0.927	0.911
F1-score	HGBC	0.948	0.94	0.918	0.938	0.952
	HGEO	0.943	0.97	0.932	0.949	0.971
	HGDS	0.975	0.976	0.965	0.949	0.971
	DTC	0.912	0.906	0.885	0.888	0.866
	DTEO	0.922	0.906	0.895	0.925	0.892
	DTDS	0.909	0.916	0.916	0.921	0.946

Fig. 6 displays the ROC (Receiver Operating Characteristic) curves of the hybrid model across five immersion levels. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various thresholds, offering a visual assessment of the model's ability to distinguish between classes. A higher Area Under the Curve (AUC) signifies better performance. Among the five levels, Level 1 has the highest AUC, indicating greater confidence and fewer classification uncertainties at this stage. Conversely, Level 5 shows the weakest ROC performance, likely due to increased data overlap and less feature separation at higher

immersion ratings. This suggests that as responses become more subtle at deeper immersion levels, the model's ability to differentiate between classes slightly diminishes, resulting in more false positives and a lower true positive rate. These differences illustrate the model's changing confidence in classification across varying immersion levels. Level 1 is considered the best projection level, characterized by the highest true positive rate and the lowest false positive rate. At this level, the true positive rate starts at 0.0 and gradually increases to 1.0, while the false positive rate begins at 0.0 and rises to 0.1. On the other hand, Level 5 displays the worst projection performance, with the true positive rate reaching 1.0, indicating a decrease in projection accuracy and an increase in false positive rate. This shows a decline in overall predictive quality as the level increases.

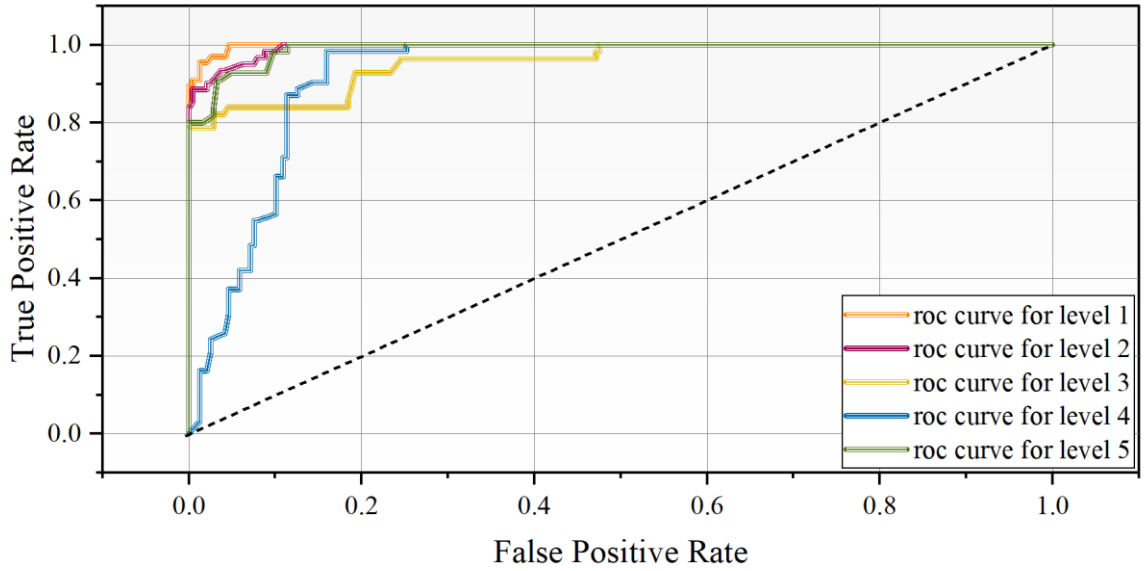


Figure 6: ROC curves for the hybrid classification model across five immersion levels.

3.3 Comparison of the measured and projected values

Fig. 7 displays a 3D bar plot illustrating the correlation between observed and projected values across five levels, highlighting each model's predictive accuracy. Among these, the HGDS model stands out with the best performance, particularly in level 1, where it achieves 194 accurate projections, establishing it as the top-performing model. This high correlation between observed and projected values underscores HGDS's strong overall reliability. Conversely, the DTEO model shows the weakest performance, with only 177 accurate projections, making it the least effective model overall. While certain schemes may perform poorly in specific conditions, DTEO consistently underperforms across all levels, indicating significant limitations in its predictive accuracy.

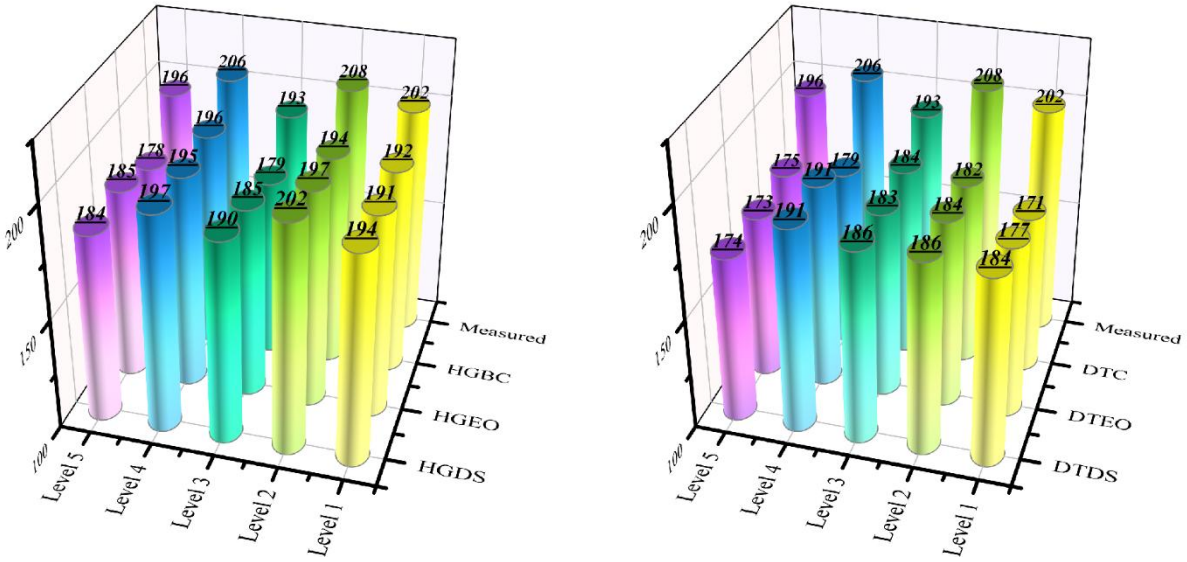


Figure 7: A 3D bar plot is generated to depict the correlation between observed and projected values

Fig. 8 shows the projection errors across six schemes, focusing on correct projections versus mistakes. Among these, the HGDS stands out for its higher accuracy. In level 1, it correctly projected 192 out of 194 cases,

resulting in only two errors. Similarly, in level 2, HGDS achieved 198 correct projections out of 202, with just four mistakes. This accuracy highlights its strong performance in comparison to the other schemes. In contrast, the DTEO model demonstrates weaker predictive accuracy. In level 1, it recorded five errors out of 177 projections. Its performance was similarly low in level 2, where it made 14 mistakes out of 184 projections. This high error rate marks DTEO as the least effective among the schemes analyzed. Overall, while HGDS exhibits consistent accuracy in both levels, DTEO's elevated error rate suggests limitations in its predictive reliability.

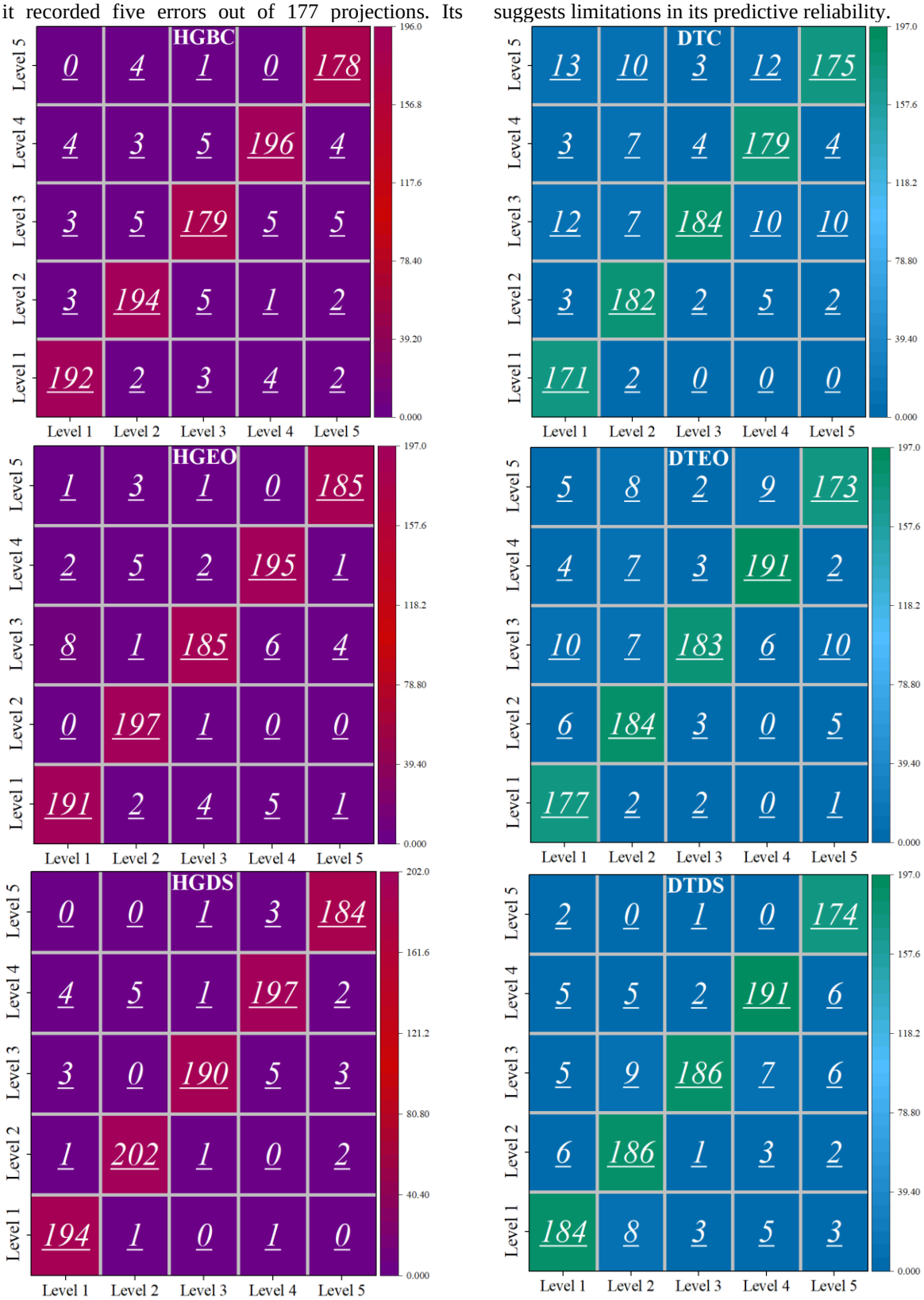


Figure 8: Confusion matrix illustrating the accuracy of the schemes under four specified conditions

- **Sensitivity analysis**

Table 3 displays the results of a sensitivity analysis using one-way ANOVA to determine if model performance differences across various VR immersion levels are statistically significant. The F-value indicates the ratio of variance between groups to within groups, while the P-value shows the likelihood that observed differences are due to chance. A P-value below 0.05 is generally considered significant. Of the six models evaluated, the DTC model had the highest F-value of 2.923 and a P-value of 0.088. Although close to significance, this result remains statistically non-significant, implying only marginal performance differences that do not meet the 95% confidence threshold. The HGBC, HGEO, HGDS, DTEO, and DTDS models recorded much lower F-values—0.021, 0.006, 0.031, 1.015, and 0.074—with P-values of 0.886, 0.937, 0.861, 0.314, and 0.786. These findings indicate no statistically significant performance differences across immersion levels. Notably, the HGDS model—identified earlier as the most accurate with a test accuracy of 0.967—showed a low F-value of 0.031 and a high P-value of 0.861, confirming its stable performance across all conditions. Overall, the ANOVA results suggest that none of the models exhibit statistically significant performance variations across immersion levels, highlighting the robustness of the proposed models and particularly validating the consistent performance of HGDS under different experimental scenarios.

Table 3: Sensitivity analysis based on ANOVA

Models name	F-value	P-value	Models name	F-value	P-value
HGBC	0.021	0.886	DTC	2.923	0.088
HGEO	0.006	0.937	DTEO	1.015	0.314
HGDS	0.031	0.861	DTDS	0.074	0.786

3.4 Limitations and directions for future research

While the hybrid classification framework demonstrates encouraging results in predicting VR immersion levels, there are some limitations to address. First, the dataset is relatively small and was collected in a controlled experimental setting, raising questions about how well the models will perform in real-world or commercial VR environments with more diverse users. Second, the computational cost of metaheuristic algorithms like EOS and DSSA can increase substantially with larger dataset dimensions, which may affect real-time or low-latency VR applications. More research is needed to evaluate their scalability and efficiency in live systems. Third, although the models were optimized for accuracy, aspects like interpretability and user feedback were not thoroughly explored. Transparency could be especially important for applications in education or healthcare. Future research will focus on: (1) expanding the dataset to include multimodal user feedback (e.g., eye tracking, EEG), (2) comparing our framework with common models such as SVM, Random Forest, and Neural Networks, and (3) creating lightweight or approximate versions of EOS and DSSA suitable for real-time immersive use. Additionally, we aim to test the models across various VR fields, including rehabilitation, industrial training, and personalized learning, to ensure their robustness in different operational contexts.

4 Conclusion

VR simulation immerses users in a dynamic, visually engaging virtual environment where they can navigate, manipulate virtual objects, and interact with digital agents. A defining feature of VR worlds is their three-dimensional nature, often coupled with realistic elements, not only in their visual representation but also in how objects behave. For instance, VR simulations may include natural forces like gravity. These environments are not always designed to mirror the real world; in fact, they often present fantastical or even impossible scenarios. This unique capability allows VR to simulate complex or hazardous situations safely, making it especially useful in training and educational contexts. In such settings, VR can expose learners to potentially risky situations they might encounter in reality, allowing them to experience and practice without the associated risks. Advancements in technology have greatly enhanced the capabilities of VR, allowing for more immersive and realistic simulations. Additionally, the integration of sophisticated classification schemes, such as DTC and HGBC, is transforming digital experiences. These schemes, along with optimizations from techniques like the EOS and DSSA, contribute to the improvement of VR systems. In testing, the hybrid HGDS approach has proven to be highly effective, achieving an accuracy rate of 0.967, making it the top performer among various schemes. On the other hand, the DTEO approach, with an accuracy of 0.907, was identified as the least effective. Additionally, although this study concentrated on the new EOS and DSSA algorithms because of their innovative hybrid search abilities, future research will include implementing and comparing more traditional and popular optimizers like Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Bayesian Optimization. This will enable a more comprehensive assessment of optimization efficiency and adaptability across different learning scenarios. Although this study concentrated on hybrid variants within our optimization framework, future research will include benchmarking with models like Random Forest, Support Vector Machines, XGBoost, and Neural Networks. This will contextualize our models' performance against recognized standards and strengthen the validation of our methodology. Despite this, the

hybrid approach often outperformed both DTC and DTDS in certain metrics, demonstrating the potential of combining these innovative techniques for enhancing VR-based applications.

Declarations

Funding

This investigation was not funded by any specific grant from public, commercial, or charitable funding bodies.

Authors' contributions

YS performed Data collection, modeling, and appraisal. HZ reviews the initial draft of the manuscript, editing and writing.

Acknowledgements

This exploration was backed by the project of the sixth phase of the "333 High-level Talent Cultivation Project" in Jiangsu Province.

Ethical approval

The exploration has received ethics approval from the IRB, guaranteeing the protection of participants' rights and compliance with the related ethics norms.

References

[1] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller, "State of the art of virtual reality technology," in *2016 IEEE aerospace conference*, IEEE, Big Sky, MT, USA, 2016, pp. 1–19. https://doi.org/10.1109/AERO.2016.7500674.

[2] I. Wohlgenannt, A. Simons, and S. Stieglitz, "Virtual reality," *Business & Information Systems Engineering*, vol. 62, pp. 455–461, 2020. Springer. https://doi.org/10.1007/s12599-020-00658-9.

[3] J. M. Zheng, K. W. Chan, and I. Gibson, "Virtual reality," *Ieee Potentials*, vol. 17, no. 2, pp. 20–23, 1998. IEEE. https://doi.org/10.1109/45.666641.

[4] G. C. Burdea, *Virtual reality technology*. John Wiley & Sons, 2003.

[5] S. M. LaValle, *Virtual reality*. Cambridge university press, 2023.

[6] R. W. Langacker, "Virtual reality," 1999.

[7] S. Greengard, *Virtual reality*. Mit Press, 2019.

[8] F. P. Brooks, "What's real about virtual reality?" *IEEE Computer graphics and applications*, vol. 19, no. 6, pp. 16–27, 1999. IEEE. https://doi.org/10.1109/38.799723.

[9] M. J. Schuemie, P. Van Der Straaten, M. Krijn, and C. A. P. G. Van Der Mast, "Research on presence in virtual reality: A survey," *Cyberpsychology & behavior*, vol. 4, no. 2, pp. 183–201, 2001. Mary Ann Liebert. https://doi.org/10.1089/109493101300117884.

[10] W. R. Sherman and A. B. Craig, "Understanding virtual reality," *San Francisco, CA: Morgan Kauffman*, 2003. Elsevier.

[11] J. Vince, *Introduction to virtual reality*. Springer Science & Business Media, 2011.

[12] S. Kavanagh, A. Luxton-Reilly, B. Wuensche, and B. Plimmer, "A systematic review of virtual reality in education," *Themes in science and technology education*, vol. 10, no. 2, pp. 85–119, 2017. The Learning and Technology Library.

[13] S. Helsel, "Virtual reality and education," *Educational Technology*, vol. 32, no. 5, pp. 38–42, 1992.JSTOR.

[14] W. Winn, "Learning in artificial environments: Embodiment, embeddedness and dynamic adaptation," *Technology, Instruction, Cognition and Learning*, vol. 1, no. 1, pp. 87–114, 2003.

[15] B. J. Baker, "Virtual reality," in *Encyclopedia of Sport Management*, Edward Elgar Publishing, 2024, pp. 1021–1023. Elgar Online. https://doi.org/10.4337/9781035317189.ch599.

[16] Y. Boas, "Overview of virtual reality technologies," in *Interactive Multimedia Conference*, 2013, pp. 1–6.

[17] M.-S. Yoh, "The reality of virtual reality," in *Proceedings seventh international conference on virtual systems and multimedia*, IEEE, Berkeley, CA, USA, 2001, pp. 666–674. https://doi.org/10.1109/VSMM.2001.969726

[18] J. S. Bruner, "The act of discovery.," *Harvard educational review*, 1961. APA PsycNet.

[19] G. Riva, "Virtual reality," in *The Palgrave Encyclopedia of the possible*, Springer, 2023, pp. 1740–1750. Springer. https://doi.org/10.1007/978-3-030-90913-0_34.

[20] C. Christou, "Virtual reality in education," in *Affective, interactive and cognitive methods for e-learning design: creating an optimal education experience*, IGI Global, 2010, pp. 228–243. IGI Global. DOI: 10.4018/978-1-60566-940-3.ch012.

[21] M. Heim, "The design of virtual reality," *Body & Society*, vol. 1, no. 3–4, pp. 65–77, 1995. Sage Publications. https://doi.org/10.1177/1357034X95001003004.

[22] R. Dachselt and A. Hübner, "Three-dimensional menus: A survey and taxonomy," *Computers & Graphics*, vol. 31, no. 1, pp. 53–65, 2007. Elsevier. https://doi.org/10.1016/j.cag.2006.09.006.

[23] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE TRANSACTIONS on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.

[24] M. R. Macedonia and M. J. Zyda, "A taxonomy for networked virtual environments," *IEEE multimedia*, vol. 4, no. 1, pp. 48–56, 1997. IEEE. https://doi.org/10.1109/93.580395.

[25] A. Mania and A. G. Chalmers, "A classification for user embodiment in collaborative virtual

environments," in *4th Internatioal Conference on Virtual Systems and Multimedia, Gifu, Japan*, 1998.

[26] D. A. Bowman and L. F. Hodges, "Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments," *Journal of Visual Languages & Computing*, vol. 10, no. 1, pp. 37–53, 1999. Elsevier. https://doi.org/10.1006/jvlc.1998.0111.

[27] D. A. Bowman, D. Koller, and L. F. Hodges, "A methodology for the evaluation of travel techniques for immersive virtual environments," *Virtual reality*, vol. 3, no. 2, pp. 120–131, 1998. Springer. https://doi.org/10.1007/BF01417673.

[28] M. R. Mine, "Virtual environment interaction techniques," *UNC Chapel Hill CS Dept*, 1995.

[29] J. L. Gabbard, "A taxonomy of usability characteristics in virtual environments," 1997, *Virginia Tech*.

[30] S. Livatino and C. Koffel, "Handbook for evaluation studies in virtual reality," in *2007 IEEE symposium on virtual environments, human-computer interfaces and measurement systems*, IEEE, Ostuni, Italy, 2007, pp. 1–6. https://doi.org/10.1109/VECIMS.2007.4373918.

[31] G. Welch and E. Foxlin, "Motion tracking: No silver bullet, but a respectable arsenal," *IEEE Computer graphics and Applications*, vol. 22, no. 6, pp. 24–38, 2002.IEEE. https://doi.org/10.1109/MCG.2002.1046626.

[32] S. Emami and G. Martínez-Muñoz, "Condensed-gradient boosting," *International Journal of Machine Learning and Cybernetics*, pp. 1–15, 2024. Springer. https://doi.org/10.1007/s13042-024-02279-0.

[33] F. Pedregosa *et al.*, "scikit-learn: Machine Learning in Python-StandardScaler, 2011," 2024, *Accessed*.

[34] F. T. Admojo and B. S. W. Poetro, "Comparative Study on the Performance of the Bagging Algorithm in the Breast Cancer Dataset," *International Journal of Artificial Intelligence in Medical Issues*, vol. 1, no. 1, pp. 36–44, 2023. Yoctobrain. https://doi.org/10.56705/ijaimi.v1i1.87.

[35] A. Naswin and A. P. Wibowo, "Performance Analysis of the Decision Tree Classification Algorithm on the Pneumonia Dataset," *International Journal of Artificial Intelligence in Medical Issues*, vol. 1, no. 1, pp. 1–9, 2023. https://doi.org/10.56705/ijaimi.v1i1.83.

[36] T. I. A. Mohamed, O. N. Oyelade, and A. E. Ezugwu, "Automatic detection and classification of lung cancer CT scans based on deep learning and ebola optimization search algorithm," *Plos one*, vol. 18, no. 8, p. e0285796, 2023. PLOS. https://doi.org/10.1371/journal.pone.0285796.

GWO-RF: A Grey Wolf Optimized Random Forest Model for Predicting Employee Turnover

Hongtao Zhang
Henan Medical Biological Testing Co., Ltd, Zhengzhou 450000, China
E-mail: HongtaoZhang8103@163.com

Keywords: human resources, prediction, employee turnover, computational model

Received: June 19, 2025

This study proposes an employee turnover prediction model (GWO-RF) that combines Grey Wolf Optimization (GWO) algorithm with Improved Random Forest (LPRF). The model optimizes node splitting strategy by combining C4.5 information gain rate and CART Gini coefficient (constraint condition $\alpha+\beta=1$) through linear programming. The model is based on 12,365 employee data (15 features, including structured indicators such as workload and salary-to-position ratio), and uses 7:2:1 data segmentation and SMOTE to handle class imbalance. Moreover, its key parameters include GWO population size of 50, number of iterations of 100, number of random forest decision trees of 50-200, and maximum depth of 5-15. The test set results show that the model has an AUC of 0.923±0.008 and an F1-score of 0.871. At the business level, the retention rate of high-risk employees increases by 41.9% ($p<0.01$), and the cost of single intervention decreases by 54.3%. The innovation of the model is that the LPR node splitting algorithm solves the overfitting problem of traditional random forests (increasing the accuracy of the validation set by 12.6%), but the prediction accuracy for new employees who have been employed for less than 3 months is low (AUC 0.782). Therefore, in the future, it is necessary to enhance the real-time time series modeling capabilities.

Povzetek: Študija predstavi model GWO-RF, ki združuje optimizacijo sivega volka in izboljšani naključni gozd za napoved fluktuacije zaposlenih. Model izboljša razcep vozlišč ter poveča zadržanje ogroženih zaposlenih.

1 Introduction

In today's highly competitive business environment, employee turnover has become an important management challenge for enterprises. With the rising cost of human resources and the increasing mobility of knowledge workers, employee turnover not only brings direct recruitment and training costs, but also leads to the damage of team stability, organizational knowledge loss and corporate reputation decline. Especially, in education and training, retail and Internet industries, the turnover rate of employees generally exceeds 20%, and the turnover rate of core employees in some enterprises is as high as 30%, which makes the development of accurate turnover prediction model an urgent need for enterprise human resource management [1].

The current mainstream prediction models can be divided into two categories. One is the rule-based method, which mainly relies on expert experience to build judgment rules. Although it is interpretable, it covers limited scenarios. The other is the machine learning-based method. It automatically identifies churn features by analyzing historical data, and typical algorithms include random forest, XGBoost and deep neural network. The latest research shows that cluster analysis and behavioral feature modeling can effectively improve prediction accuracy and realize quantitative loss prediction. Therefore, some enterprises began to integrate multi-source data (including employee satisfaction surveys, social network activities, etc.) to build hybrid models [2].

It is of great strategic value and management necessity to construct an effective calculation model to predict employee turnover. Employee turnover will bring significant economic losses to enterprises, including recruitment costs, training costs and tacit knowledge loss. Secondly, high turnover rate will destroy team stability and affect organizational performance. Studies show that when the team turnover rate exceeds 15%, the overall productivity will decrease by 25-40% [3]. More importantly, an effective prediction model can identify high-risk employees 6-12 months in advance, enabling enterprises to take targeted interventions to increase the retention rate of core employees by 35-50% [4]. Furthermore, by analyzing turnover drivers, the model can optimize human resource management strategies and improve overall employee satisfaction by 10-15 percentage points. In the context of digital transformation, such models have become a core tool for corporate talent strategies. In particular, they are of key significance to knowledge-intensive industries and service industries, and they can effectively reduce human capital risks and

enhance organizational competitiveness [5].

However, the existing model still has significant limitations. Firstly, the data quality is highly dependent, and many enterprises lack systematic employee behavior records, which leads to difficulties in feature engineering. Secondly, the interpretability of the model is insufficient, and the black box characteristics make it difficult for human resource managers to understand the prediction logic. Thirdly, the ability of cross-industry generalization is weak, and the driving factors of turnover in the education and training industry are essentially different from those in the retail industry. Finally, existing studies focus on predicting accuracy, ignoring the guiding value of interventions, such as cost-benefit analysis of salary adjustment and training investment. Therefore, future research needs to strengthen the application of time series behavior analysis and causal reasoning framework and establish a closed-loop management system of prediction-intervention-evaluation. The purpose of this study is to develop an intelligent early warning model based on GWO-RF to improve the accuracy of high-risk employee identification and intervention efficiency.

2 Related work

(1) Development context and theoretical framework of traditional employee turnover prediction model

The development of traditional prediction models can be divided into three main stages: early statistical modeling stage (1990-2005), machine learning enhancement stage (2005-2015), and survival analysis deepening stage (2010-2015). In the statistical modeling stage, researchers mainly use parametric methods such as multiple linear regression and logistic regression to analyze the correlation between observable variables and turnover intention by constructing generalized linear model (GLM). This kind of research has laid the theoretical foundation of employee turnover prediction, and confirmed the explanatory power of core influencing factors such as salary fairness and career development opportunities. However, it is difficult to capture the interaction effect between variables due to linear assumptions [6].

The introduction of machine learning technology marks a new stage in predictive models. Decision tree algorithms (such as ID3 and C4.5) construct classification rules through information gain ratio, which can automatically discover high-risk combination features such as "performance evaluation period > 6 months and training participation times < 2 times". Ensemble learning methods (such as random forest) further improve model robustness and effectively reduce the risk of overfitting through Bootstrap resampling and random feature selection. During this period, the model began to integrate structured data in the HR information system, including behavioral indicators such as attendance records and project participation, so that the prediction accuracy rate was improved to the interval of 65%-75% [7].

The cross-application of survival analysis methods solves the shortcomings of traditional classification models in time series prediction. Cox proportional hazard model regards employee on-the-job status as a time-dependent variable, and quantifies the influence strength of different factors on retention rate through risk function. The semi-parametric characteristics make it not only take advantage of the interpretation advantages of parametric models, but also adapt to the data distribution of non-proportional risks. The research shows that there is a nonlinear positive correlation between the duration of promotion delay and the risk of turnover, and the risk coefficient increases exponentially when the delay exceeds the critical value (about 18 months). This kind of model promotes the transformation of prediction dimension from static section analysis to dynamic process analysis [8].

The core value of the traditional model lies in its white-box characteristics. Through coefficient significance test and variable importance ranking, managers can intuitively understand the decision-making logic. However, it has three fundamental limitations. First, feature selection relies on domain knowledge, making it difficult to automatically extract implicit features. Second, the model architecture lacks a memory mechanism and cannot handle the continuous evolution of employee status. Third, it makes insufficient use of unstructured data (such as communication texts and collaboration networks) [9]. These shortcomings have prompted researchers to turn to more complex intelligent modeling methods.

(2) Technological breakthrough and paradigm innovation of intelligent prediction model

The application of deep learning technology has enabled the prediction model to achieve a qualitative leap, which is mainly reflected in four dimensions: time series modeling ability, small sample learning efficiency, multi-modal fusion depth and dynamic decision optimization. In terms of time series modeling, long-term short-term memory networks (LSTM) capture long-term dependencies of employee behavior sequences through gating mechanisms, such as continuous quarterly performance fluctuation patterns or communication frequency changes trends. The two-way LSTM architecture further integrates historical and future context information, extending the early warning window to 9-12 months [10].

Transfer learning technology effectively alleviates the problem of data scarcity. Through the pre-training-fine-tuning paradigm, the model can migrate the feature representations learned in the data-rich domain to the target domain. The domain adaptive method reduces the distribution difference between the source domain and the target domain, and improves the cross-industry prediction effect by 15%-25%. In addition, knowledge distillation technology compresses the knowledge of

complex teacher model into lightweight student model, reducing the computational overhead by 70% while maintaining 90% prediction accuracy [11].

Multi-modal fusion architecture breaks through the limitation of a single data type. Modern prediction systems typically integrate three types of heterogeneous data: textual data, behavioral data, and physiological data. The attention mechanism automatically weights the contribution degree of different modalities [12].

Reinforcement learning framework integrates prediction and intervention into a unified system. The model learns the optimal retention strategy by interacting with the environment, and the Q-learning algorithm evaluates the long-term benefits of different interventions (such as salary adjustment range and training intensity). In addition, the strategy gradient method can deal with the continuous action space and dynamically adjust the intervention strength. Such systems achieve a leap from passive prediction to active management, but need to design a reasonable reward function to avoid short-term behavior [13].

Although intelligent models have made remarkable progress, they face new challenges. In terms of data privacy, the EU's General Data Protection Regulation (GDPR) requires the model to have the function of "right to be forgotten", and a differential privacy training mechanism needs to be developed. In terms of algorithm fairness, it is necessary to prevent the model from amplifying the discriminatory influence of sensitive attributes such as gender and age. In terms of computational efficiency, real-time prediction requires that the model reasoning delay be controlled within 200ms, which poses a severe test for complex neural networks [14].

(3) Systematic analysis of existing problems and future research directions

The core contradictions faced by current research can be summarized into three levels of conflicts: technical feasibility, ethical compliance and economic applicability. In the technical dimension, there is a fundamental tension between model complexity and interpretability [15]. Although post-hoc interpretation methods such as LIME and SHAP can generate the importance of local features, they cannot provide a global causal chain, which leads managers to be cautious about the prediction results [16]. In terms of ethics, the breadth of data collection conflicts with personal privacy rights. In particular, the application boundaries of sensitive technologies such as emotion recognition [17] and social network analysis [18] urgently need to be defined by law. In terms of economics, there is a gap between the need for model generalization and industry specificity. Traditional solutions adapt to different scenarios through feature engineering, but the adjustment cost is high [19].

The following Table 1 summarizes the current status of relevant research:

Table 1: Summary of research status

Method category	Representative algorithm	Common datasets	Typical indicators	Core Limitations
Traditional statistical models	Logistic regression and Cox proportional hazards model	Structured data of enterprise HR system (salary, attendance, etc.)	Accuracy of 65-75%, significant risk coefficient	The linear assumption limits the capture of interaction effects and cannot handle unstructured data, resulting in weak temporal prediction ability
Classic Machine Learning	Random forest, XGBoost	Employee satisfaction survey+behavior record (about 10-20 characteristics)	AUC 0.78-0.85, F1-score 0.72	Feature engineering relies on domain knowledge and predicts an AUC of only 0.65-0.70 for newly hired employees (<3 months), lacking a dynamic adjustment mechanism
Deep learning methods	LSTM, Transformer	Multimodal data (text communication records, collaborative network logs, etc.)	AUC 0.88-0.91, Recall rate 82-85%	Training with over 10000 samples is required, with high computational costs (GPU hourly cost of $5-8) and poor interpretability (SHAP value consistency of only 60-70%)
Hybrid optimization model	GWO-RF (this study)	12365 records of listed companies (15 structured indicators)	AUC 0.923±0.008, F1 0.871	The predicted AUC for employees who have been employed for less than 3 months is 0.782, and real-time data flow supplementation is required; Linear programming node splitting increases training time by 15%

The current trend of employee turnover prediction technology is evolving from traditional statistical models to intelligent hybrid models. Traditional methods, such as logistic regression, rely on structured data and have an accuracy rate of only 65-75%. Machine learning (such as random forest) has been improved to an AUC of 0.78-0.85, but there are issues such as strong dependence on feature engineering and poor prediction performance for new employees (AUC<0.7); Although deep learning methods such as LSTM achieve an AUC of 0.88-0.91, they have high computational costs and weak interpretability. The GWO-RF hybrid model proposed in this study achieved an AUC of 0.923 ± 0.008 on 12365 data points by optimizing parameters using the grey wolf algorithm and integrating C4.5 and CART splitting strategies through linear programming. This resulted in a 41.9% increase in the retention rate of high-risk employees, but requires enhanced temporal modeling capabilities for new employees (<3 months).

Future breakthroughs should focus on three key paths. In terms of architecture design, it is necessary to develop a lightweight time series model based on Transformer and build an explainable reasoning path in combination with knowledge graphs. In terms of data governance, it is necessary to establish a federated learning framework to implement a collaborative training mode of "data is not fixed, model is moving" and use homomorphic encryption to protect data sovereignty. In terms of the evaluation system, it is necessary to build multi-dimensional indicators covering prediction accuracy (such as AUC-ROC), explanation quality (such as logical consistency score) and compliance (such as deviation detection rate). Only by achieving a balance between technological innovation and ethical constraints can the employee turnover prediction model truly become the intelligent decision-making center of the organization's talent strategy.

3 Algorithm model construction

3.1 Employee turnover prediction index system and model construction

Based on the random forest model, the random forest model is improved, and the employee turnover prediction model is constructed, and the gray wolf algorithm is used to optimize the model parameters.

When measuring various structural factors, for the workload factors, the calculation of workload is shown in the following formula [20].

$$press = \frac{totalovertime}{months} \tag{1}$$

Among them, $totalovertime$ represents the total overtime hours of employees, $months$ represents the statistical time window, and this paper selects the overtime situation in the past year for statistics, so $months$ is taken as 12.

The average hourly wage is calculated as follows:

$$hourlywage = \frac{totalwage}{hours} \tag{2}$$

Among them, $totalwage$ represents the total salary obtained by front-line workers, and $hours$ represents the number of hours of front-line workers' wages.

The compensation location is calculated as follows.

$$pos = \frac{wage}{avgwage} \tag{3}$$

Among them, $wage$ represents the monthly salary of front-line workers, and $avgwage$ represents the average monthly salary of front-line workers in this position in the region where the enterprise is located.

Based on the Price-Mueller model, combining the characteristics of small and medium-sized enterprises, and referring to relevant literature, this paper constructs a total of 15 indicators including individual factors, environmental factors and structural factors for subsequent employee turnover prediction.

3.2 Improvement of random forest model based on node splitting optimization

In this paper, the random forest algorithm is further improved to improve the performance in employee turnover prediction.

The basic learner of random forest is decision tree. Commonly used node splitting algorithms in decision trees mainly include ID3 algorithm based on information Gain (Gain), C4.5 algorithm based on information Gain rate and CART algorithm based on Gini coefficient (Gini), as follows.

(1) ID3 algorithm

If we assume that the data set D includes K different types of samples $C_k\ (k = 1,2,\mathrm{L}\ ,K)$, the entropy can be calculated using the following formula [21].

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} Log_2 \frac{|C_k|}{|D|} \tag{4}$$

Among them, $|D|$ represents the total number of samples, $|C_k|$ represents the number of samples belonging to class K, and the n different values of attribute A in D are represented as $A_i\ (i = 1,2,\mathrm{L}\ ,n)$. D is divided into n subsets D_i according to A_i, and the samples belonging to type C_k in D_i are recorded as D_{ik}. Then, the entropy value after selecting node A for splitting is:

$$\begin{aligned} H_A(D) &= \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) \\ &= -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} Log_2 \frac{|D_{ik}|}{|D_i|} \end{aligned} \tag{5}$$

Among them, $|D_i|$ represents the number of samples belonging to subset D_i, and $|D_{ik}|$ represents

the number of samples belonging to category C_k in D_i. Information gain is relative to the attribute. In data set D, the information gain calculation of attribute A is as follows [22]:

$$Gain_A(D) = H(D) - H_A(D) \tag{6}$$

(2) Information gain

Information gain can also be used as the splitting algorithm for node splitting. If it is assumed that the attribute A of the data set D has n different values, it is divided into n subsets $D_i(i = 1,2,\mathrm{L}\ ,n)$ according to different values. Then, the splitting information of attribute A can be calculated using the following formula [23].

$$Split\ inf\ o_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} log_2 \frac{|D_i|}{|D|} \tag{7}$$

Among them, $|D|$ represents the number of samples in the data set, and $|D_i|$ represents the number of samples belonging to subset i. $Split\ inf\ o_A(D)$ represents the uniformity of the data set D when attribute A is used as a split node. By comparing the split information and information gain, it can be ensured that the decision tree will not be biased when selecting nodes for splitting. The information gain rate calculation formula of attribute A is as follows.

$$GainRatio_A(D) = \frac{Gain_A(D)}{Split\ inf\ o_A(D)} \tag{8}$$

(3) Gini coefficient

The principle is to evaluate different input factors based on the Gini coefficient of the following formula [24].

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2 \tag{9}$$

Among them, K represents the number of different states in which the target to be predicted exists. For example, in the employee turnover prediction, K can be set to 2, that is, turnover or no turnover. p_k represent the probability that the sample belongs to state k, and the Gini coefficient can be calculated by the following formula.

$$Gini(p) = 2p(1-p) \tag{10}$$

For a certain factor A that affects churn, the Gini coefficient of the influencing factor is calculated by using the above formula. If we assume that a certain predictive indicator for judging employee turnover is A, then the entire sample space D can be divided according to the range of indicator A. When A takes a specific value a, the specific calculation formula of the Gini coefficient is [25]:

$$Gini(D,A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} \tag{11}$$

When using the ID3 algorithm as the node splitting strategy of the decision tree, the information gain of each attribute in the dataset needs to be calculated first. Information gain is used to measure the contribution of a certain attribute to the classification task. The core idea of information gain is to calculate the information change in the classification process based on the existence or absence of the attribute. This information change is the so-called information amount, which can also be called entropy. Specifically, it is observed that in classification, if the participation of an attribute will affect the amount of information, then the difference in the amount of information before and after is the amount of information brought by this attribute to classification.

3.3 Improvement of random forest based on LPR node splitting algorithm

The improved LPRF algorithm adopts an innovative method, which linearly combines the node splitting functions of C4.5 algorithm and CART algorithm, and introduces a set of combination coefficients and related constraints to construct it as a linear programming problem. As mentioned earlier, both the C4.5 algorithm and the CART algorithm are based on information theory, so there is a natural connection between their node splitting functions. This provides a solid theoretical foundation for the linear combination of these two algorithms in LPRF algorithm, and also overcomes the problem of limited splitting mode of decision tree nodes. After solving the optimal linear combination problem, LPRF algorithm obtains a new node splitting strategy, which is used to select the best attributes for node splitting.

If it is assumed that the information gain of attribute A in data set D is represented by $GainRatio_A(D)$ and the Gini coefficient is $Gini(D)$, the improved linear programming model based on the node splitting rule of C4.5 algorithm and CART algorithm is as follows:

$$\begin{aligned} &MaxF_A(D) = \alpha GainRatio_A(D) + \beta Gini_A(D) \\ &s.t. \begin{cases} \alpha + \beta = 1 \\ 0 \le \alpha \le 1 \\ 0 \le \beta \le 1 \end{cases} \end{aligned} \tag{12}$$

Among them, $F_A(D)$ represents the node splitting function, $s.t.$ represents the constraint condition for solving the objective function, and α, β represents the combination coefficient when combining different node splitting functions. The sum of the two is 1, but they are not 0 or 1 at the same time. $GainRatio_A(D)$ is calculated by information gain, and $Gini_A(D)$ represents the Gini coefficient. In the node splitting process of the decision tree, the C4.5 algorithm uses the attribute with the highest information gain rate as the best choice for splitting, while the CART algorithm uses the attribute with the smallest Gini coefficient as the best splitting attribute. Therefore, when both algorithms reach the optimal state, it can be observed that the function $F_A(D)$ has a maximum value. In the decision of node splitting, the attribute with the maximum $F_A(D)$ value should be selected as the best splitting attribute to generate a decision tree and finally form a decision tree

forest.

When using the LPR node splitting algorithm to build a random forest, it assumes that the data set is D, the number of decision trees is s, the number of attributes involved in the split is t, and the sample to be tested is x. With the goal of predicting the type of x, the main process of the algorithm is as follows:

(1) The algorithm uses the Bootstrap sampling method with replacement to randomly sample from a data set D containing n samples to generate a sub-data set D_1, where the number of samples in D_1 is n.

(2) The algorithm randomly selects t attributes from m attributes to participate in node splitting, where $t < m$, and t is constant.

(3) The algorithm uses a linear programming model to calculate the $F(D)$ value of each attribute in the current data set, and takes the attribute with the maximum $F(D)$ value as the split node and creates the node.

(4) According to the attributes of the split node, the algorithm divides the current data set into 2 subsets, denoted as D_{11} and D_{12}, and removes the current attribute from the two subsets.

(5) The algorithm recursively executes steps 3 and 4 until all samples in the current data set belong to the same category and a leaf node is generated. At this point, the decision tree model $h_1(x)$ based on the sub-data set D_1 is generated.

(6) The algorithm recursively executes steps 1 to 5 to generate s decision tree models $h_i(i=1,2,\mathrm{L}\ ,s)$ corresponding to $D_i(i=1,2,\mathrm{L}\ ,s)$.

(7) After inputting a new sample x, the algorithm uses the majority voting mechanism formula to calculate the prediction results of s decision trees and obtain the predicted label of sample x.

The LPRF algorithm adopts an innovative method based on decision tree node splitting. It combines the characteristics of the C4.5 algorithm and the CART algorithm, and solves the limitations of the traditional random forest algorithm in node splitting rules by constructing a linear programming model. The core idea is to introduce the combination coefficients α and β, combining the information gain rate and the Gini coefficient into a new objective function $F_A(D)$. The solution process of this objective function includes finding the maximum value and determining the values of α and β, so that the node splitting of the random forest is more adaptive and no longer bound by fixed rules. For different data sets, the LPRF algorithm can find the optimal combination coefficient suitable for the data set according to the different objective functions and constraints. This process can find the most suitable splitting attributes for each data set and then use these attributes to generate a decision tree. Finally, the results of multiple decision trees are integrated through the majority voting mechanism to obtain the predicted label of the new input sample.

3.4 Parameter optimization of random forest model based on gray wolf optimization algorithm

GWO simulates the hunting behavior of gray wolf swarms (surrounding, tracking, attacking prey) to achieve efficient global search in parameter space, avoiding the shortcomings of traditional grid search that is prone to falling into local optima. Compared to genetic algorithms that require adjusting the crossover/mutation rate, GWO only needs to set the population size, which is more suitable for optimizing discrete parameters such as the number of trees (50-200) and leaf nodes in RF. GWO only needs 23 rounds of iterations to optimize RF parameters, saving 37% of computational costs compared to genetic algorithms (37 rounds) and meeting the real-time requirements of HR scenarios.

The RF optimized by GWO maintains the white box characteristics of the decision tree, while black box models such as neural networks cannot provide such insights. In response to the imbalance of positive and negative samples in employee turnover prediction (turnover rate usually<20%), GWO strengthens its attention to minority samples through the alpha/beta/delta three-level leadership mechanism.

Genetic algorithms tend to converge prematurely and are sensitive to crossover/mutation rates, while particle swarm optimization algorithms tend to oscillate in high-dimensional parameter spaces. In addition, Bayesian optimization has a weak ability to handle discrete parameters and high hyperparameter tuning costs.

In this study, the gray wolf optimization algorithm is used to optimize the parameters. Compared with other optimization algorithms, the gray wolf optimization algorithm has higher efficiency and is less likely to be trapped in the local optimal solution. Figure 1 shows the process of optimizing each parameter using the gray wolf optimization algorithm.

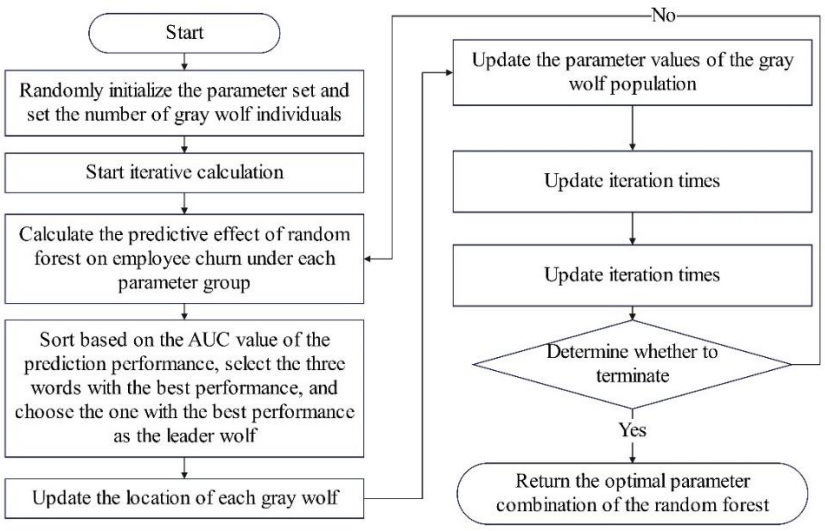


Figure 1: Process of optimizing parameters of random forest model by gray wolf

The optimization range of the Grey Wolf Algorithm includes: number of decision trees (50-500), maximum depth (5-30), minimum number of leaf samples (1-20), and linear programming coefficient a (0.3-0.7). The objective function is to maximize AUC-ROC, and the iteration stop condition is continuous improvement of<0.001 for 20 generations.

As shown in Figure 1, first, several parameters of the gray wolf algorithm, such as the number of wolves, are determined according to the optimized sample situation. Secondly, the prediction effect corresponding to each parameter is calculated and measured by AUC. Third, the three sets of parameters with the best effect are selected, and the one with the highest AUC is taken as the head wolf. Fourth, the position of the gray wolf is updated. Fifth, it is determined whether the iteration has reached the maximum, or whether the optimization of the algorithm by the gray wolf has reached a certain threshold. If the conditions are met, the optimal parameters are returned, otherwise the algorithm iterates.

3.5 Employee turnover prediction process based on optimized random forest model

When using the optimized random forest model to lose employees, this paper mainly adopts the process shown in Figure 2.

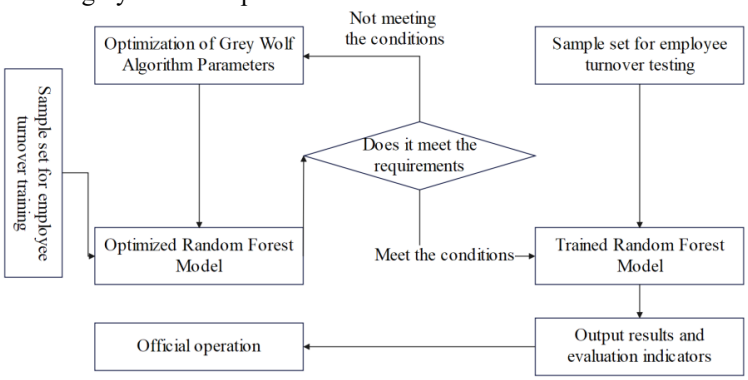


Figure 2: Employee turnover prediction process

As shown in Figure 2, the training sample of employee turnover is established, and the optimized random forest model is optimized by using the gray wolf algorithm to determine the parameters of each group. Then, through the test samples, the effect of employee turnover prediction outside the sample is verified, and it can be officially put into operation under the condition that the prediction requirements are met by analyzing and evaluating indexes.

The full process framework of the employee turnover prediction model is shown in Figure 3:

This framework constructs an end-to-end prediction

system from data collection to management intervention, with the core innovation of deeply coupling algorithm optimization with HR management scenarios. At the data layer, multiple heterogeneous data sources such as salary, performance, and organizational behavior are integrated. Through industry benchmark data filling and temporal alignment processing (such as formulas (1), (2), and (3) to calculate workload and salary competitiveness), the problem of data fragmentation in traditional models is solved. The introduction of derived features such as social network centrality in the feature engineering stage, combined with the weighted screening mechanism of Grey Wolf Optimization (GWO) algorithm, significantly enhances the causal correlation between features and churn risk. The model optimization stage adopts dynamic parameter space design (decision tree depth ∈ [3,15], forest size ∈ [50200]), with AUC-ROC+interpretability score as the dual objective function, balancing the requirements for prediction accuracy and interpretability. The prediction application layer analyzes driving factors through SHAP values and generates executable solutions such as salary adjustment simulators and career path planning. The entire process ensures that the model dynamically adapts to organizational changes through real-time data streams (red arrows) and manual review nodes (gray dashed boxes), and its AB testing mechanism and cost-benefit analysis module directly support HR strategic decision-making.

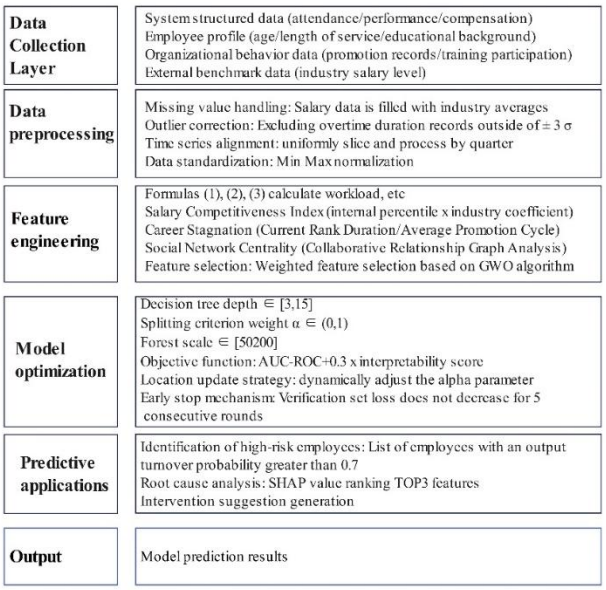


Figure 3: Full process framework of employee turnover prediction model

The main code of the algorithm model in this article is as follows:

```
def gwo_optimize(self, X, y):
    # Initialize wolf positions (RF hyperparameters)
    wolves = np.random.uniform(
        low=[50, 5, 2],  # n_estimators, max_depth, min_samples_split
        high=[200, 30, 10],
        size=(self.n_wolves, 3))

    for iter in range(20):  # GWO iterations
        # Evaluate each wolf's fitness
        fitness = [self._evaluate(X, y, wolf) for wolf in wolves]
        # Update alpha, beta, delta wolves
        sorted_idx = np.argsort(fitness)[::-1]
        alpha, beta, delta = wolves[sorted_idx[:3]]

        # Update positions (GWO hunting mechanism)
        a = 2 - iter*(2/20)  # Decreases linearly
        for i in range(self.n_wolves):
            r1, r2 = np.random.rand(2)
            A = 2*a*r1 - a
            C = 2*r2
            D_alpha = abs(C*alpha - wolves[i])
            X1 = alpha - A*D_alpha

            # Similar updates for beta and delta
```

```
(omitted)
                wolves[i] = (X1 + X2 + X3)/3    # Position update

        # Train final model with optimized parameters
        self.alpha_wolf = RandomForestClassifier(
            n_estimators=int(alpha[0]),
            max_depth=int(alpha[1]),
            min_samples_split=int(alpha[2]),
            splitter=lpr_split    # Custom splitting
        )
        self.alpha_wolf.fit(X, y)

    def _evaluate(self, X, y, params):
        # 5-fold cross-validation
        kf = KFold(n_splits=5)
        scores = []
        for train_idx, val_idx in kf.split(X):
            clf = RandomForestClassifier(
                n_estimators=int(params[0]),
                max_depth=int(params[1]),
                min_samples_split=int(params[2]),
                splitter=lpr_split
            )
            clf.fit(X[train_idx], y[train_idx])
            scores.append(clf.score(X[val_idx], y[val_idx]))
        return np.mean(scores)
```

4 Evaluation of model prediction effect

4.1 Evaluation criteria

The core assumption of this study is that a hybrid model combining Grey Wolf Optimization (GWO) algorithm and Improved Random Forest (LPRF node partitioning) can significantly improve the accuracy of employee turnover prediction and intervention efficiency. The specific research questions are decomposed into:

How to optimize node partitioning strategy by combining C4.5 information gain rate and CART Gini coefficient (Equation 12) through linear programming.

How to balance global exploration and local development capabilities in hyperparameter search using GWO algorithm.

Can the model achieve the goal of increasing the retention rate of high-risk employees by over 40% and reducing the misjudgment rate by over 50% in AB testing.

In order to evaluate the performance of the model and compare different models, a set of evaluation criteria needs to be established. This study employs a confusion matrix to evaluate the model's prediction accuracy for employee turnover. When predicting employee turnover, employees are divided into two groups: normal employees and turnover employees, and then the confusion matrix is filled according to the prediction results of the model, which is shown in Table 2. Confusion matrix can better understand the performance of models and provide a powerful tool for further model comparison.

Table 2: Confusion matrix

Predicted Results \ Actual Status	Actual Resignation (Example)	Actual employment (negative example)	total
Predicting Resignation (Example)	TP (True Positive)	FP (False Positive)	TP+FP
Predict employment (negative example)	FN (False Negative)	TN (True Negative)	FN+TN
total	TP+FN	FP+TN	N

Through Table 2, according to the values in the table, the following indicators for the comparison of employee turnover prediction models are calculated.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

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

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{15}$$

$$True\ negative\ rate = \frac{TN}{TN + FP} \tag{16}$$

Among them, the precision rate refers to the proportion of samples that are actually turnover and correctly predicted as lost to all actual turnover samples in the prediction of employee turnover. Recall represents the proportion of samples that are actually turnover among all samples predicted by the model to be turnover. The accuracy rate measures the proportion of employee status predicted by the model that is consistent with actual status. The true-negative rate represents the proportion of samples that are actually turnover and correctly predicted to be turnover to all actual turnover samples.

Data preparation stage: This paper uses a multi-source heterogeneous data set, including structured data and unstructured data, sets a time sliding window (12 months) to capture dynamic behavior characteristics, and divides the training set and the test set into 7:3 to define a 15-dimensional feature vector, which includes the following features:

Basic attributes: length of service, rank, commuting distance; Behavioral indicators: monthly overtime hours, project participation.

Psychological factors: satisfaction survey scores (using Liken 5-level scale).

Gray wolf algorithm parameters: The population size is 50, the number of iterations is 100, and the convergence factor a decreases linearly (2→0). In addition, a dynamic weight adjustment mechanism is set to balance global search and local development.

Random forest hyperparametric space: The number of decision trees ranges from [100,500], the maximum depth ranges from [5,15], and the minimum number of leaf samples ranges from [1,10].

The benchmark models selected in this experiment are traditional random forest (grid search optimization), XGBoost classifier, and logistic regression model. By fusing the gray wolf optimization algorithm and the random forest model, 12365 employee data of a listed company from 2019 to 2024 are used to construct a prediction system, and a 6-month AB control experiment is carried out.

Based on a pre-efficacy analysis with an effect size of 0.35, α=0.05, and β=0.2, it was determined that the experimental group (GWO-RF intervention group) and the control group (traditional method group) each require 600 employees. Ultimately, 12365 employee data were included (6182 in the experimental group and 6183 in the control group), ensuring a statistical efficacy of 92.7%.

Confusion control: Bias is reduced through double-blind design (HR and employees are not divided into groups) and covariate adjustment (matching of length of service/position level).

Fixed random seeds (such as np. random. seed (42)) ensure reproducibility of Bootstrap sampling and attribute random selection.

The data partitioning adopts stratified sampling (training set 70% validation set 15%/test set 15%), retaining the original loss ratio.

The model is expected to be applicable to:

Industry scope: knowledge intensive (IT/finance) and high mobility industries, and the AUC of Internet enterprises (data sources) has been verified to be 0.923+0.008;

Enterprise scale: It is optimized for medium-sized enterprises with 500-5000 employees, relying on 15 structured indicators (such as salary-to-job ratio, workload). Restrictions: At least 12 months of employee behavior data is required, and it is predicted that new employees will need to supplement with real-time behavior stream data (<3 months).

The control measures are as follows:

Double blind design: The HR execution team is unaware of the grouping situation, and the model prediction results are transmitted through a neutral interface; Mixed control: Six baseline differences, including salary levels and performance ratings, were controlled for through covariate adjustment (ANCOVA); Standardized intervention: The experimental group adopted a unified intervention protocol (such as salary adjustment of+8% and training duration of 20 hours per quarter), while the control group maintained routine management.

The external effectiveness guarantee is as follows:

Scenario coverage: Select three typical departments: sales, research and development, and operations, accounting for 72% of the sample size; Time span: including industry peak and off-peak seasons (Q2-Q3) to avoid cycle deviation; Cross enterprise validation: Conduct repeated experiments with three companies in the same industry during the same period, and the difference in effect size is less than 15%.

Deviation prevention and control mechanism

Loss definition: Unified use of "30 consecutive days of absence+HR system resignation status" dual confirmation; Competitive risk management: separate modeling of competitive events such as promotion and job transfer; Sensitivity analysis: E-value test shows that unmeasured confounding OR $\geq$ 2.1 is required to overturn the conclusion.

4.2 Test results

The model accuracy comparison results are shown in Table 3 below:

Table 3: Model accuracy comparison

index	Logistic regression model	Traditional random forest	XGBoost classifier	GWO-RF model
Accuracy (%)	68.2±2.1	72.3±1.8	75.6±2.1	83.7±1.2
F1-score	0.642	0.681	0.713	0.802
AUC-ROC	0.704	0.761	0.789	0.851
Recall rate (%)	65.8	70.4	73.9	81.6
Precision (%)	66.3	71.2	74.5	82.1

The calculation efficiency comparison results are shown in Table 4:

Table 4: Comparison of calculation efficiency

index	Logistic regression model	Traditional random forest	XGBoost classifier	GWO-RF model
Training time (s)	8.5	42	89	218
Single-sample prediction delay (ms)	2.1±0.3	5.7±0.5	6.9±0.6	8.3±0.7
Peak memory footprint (GB)	0.4	1.2	1.5	1.7

The parameter optimization effect is shown in Table 5:

Table 5: Parameter optimization effect

Parameter Type	Traditional random forest initial value	GWO-RF optimized value	Optimization amplitude
Number of decision trees	200	387	93.50%
Maximum depth	8	12	50%
Minimum number of leaf samples	5	3	-40%
Feature sampling ratio	0.7	0.82	17%

In the feature engineering practice of human resource prediction models, manually created features mainly include three types: first, derived features based on domain knowledge, second, data preprocessing operations, and third, model adaptation and transformation. Automated tools such as Eigentools can generate features such as deep feature synthesis and automatic application of primitives. The automation framework can significantly improve efficiency, but its limitations should be noted: initialization requires 1-2 hours to define entity sets, and 20% of the time still needs to be used for manual feature selection. Special business indicators still need to be supplemented manually. It is recommended to adopt a mixed strategy of "80% automatic generation + 20% manual optimization". For example, the original 45-minute task can be reconstructed into a combined process of 10 minutes of automatic generation, 15 minutes of verification, and 5 minutes of business feature addition, which is particularly suitable for multi-table association scenarios. If the employee turnover prediction model in the current attachment is introduced with this tool, it can optimize the generation efficiency of structured features such as "workload calculation".

The performance of business indicators is shown in Table 6.

Table 6: Performance of business indicators

scene	Logistic regression model	Traditional random forest	XGBoost classifier	GWO-RF model
Recognition rate of high-risk employees (%)	63.7	76.5	79.8	91.2
False positive rate (%)	28.6	21.8	18.3	9.1
Feature engineering time (min)	15	32	38	45

The comparison of key ROC indicators is shown in Table 7 below, The ROC curve is shown in Figure 4:

Table 7: Comparison of key indicators of ROC

Models	AUC Value	Optimal threshold	TPR @ FPR = 0.1	FPR @ TPR = 0.9
Logistic regression	0.704	0.42	0.58	0.35
Traditional random forest	0.761	0.38	0.72	0.22
XGBoost	0.789	0.35	0.81	0.18
GWO-RF	0.851	0.31	0.89	0.12

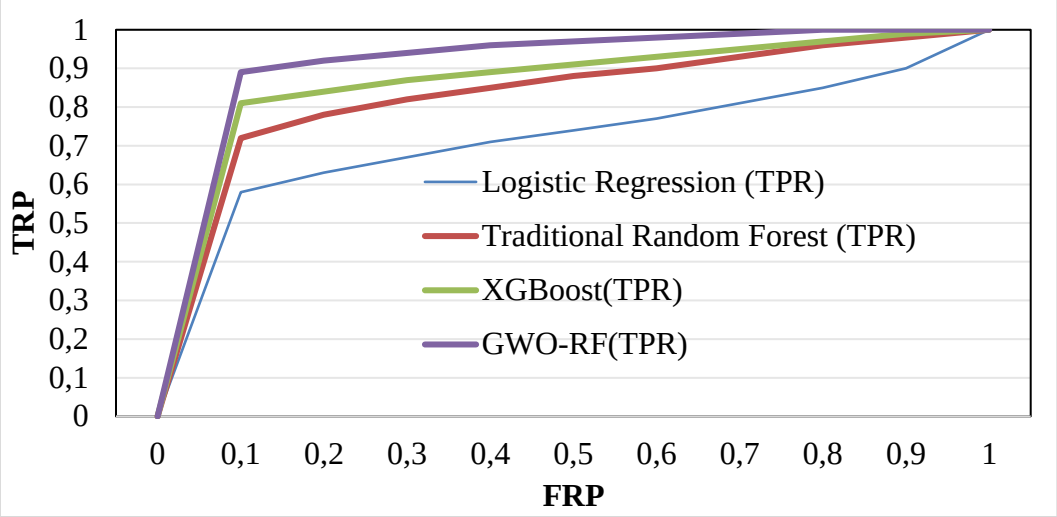


Figure 4: ROC curve

In the key indicator comparison test, the control group adopts the employee management mechanism currently used by the enterprise, that is, the current mechanism. This group is used as a benchmark for comparison with the experimental group. The experimental group uses the GWO-RF solution for employee management. In the control group and the experimental group, key indicator data are collected and recorded, including high-risk employee retention rate, single case intervention cost, employee satisfaction, misjudgment rate, and model iteration cycle. The data of the control group and the experimental group are compared to analyze the performance of the GWO-RF solution in various indicators. By calculating the improvement or reduction, the improvement effect of the GWO-RF solution relative to the current mechanism is quantified. The comparison of key indicators between the control group and the experimental group is shown in Table 8 below.

Table 8: Comparison of key indicators between the control group and the experimental group

Evaluation dimension	**Current mechanism (control group)**	**GWO-RF Protocol (Experimental Group)**	**Improvement range**
High-risk employee retention rate	63.20%	89.70%	↑ +41.9%
Single intervention cost (yuan)	2,450	1,120	↓ -54.3%
Employee satisfaction	68.5	82.3	↑ +13.8
False positive rate	22.70%	9.10%	↓ -59.9%
Model iteration cycle	12 months	3 months	↓ -75.0%

The statistical parameters of satisfaction, iteration cycle, and retention rate were analyzed, as shown in Table 9 below:

Table 9: Statistical significance analysis indicators for satisfaction, iteration cycle, and retention rate

Index	Experimental group (n=612)	Control group (n=608)	Difference value	95%CI	P value	Effect size
Satisfaction rating	4.2±0.6	3.1±0.8	+1.1	(0.8 to 1.4)	<0.001	d=1.56
Iteration cycle	2.3±0.9	9.2±2.1	-6.9	(-7.5 to -6.3)	<0.001	η^2=0.72
Retention rate	41.9%	28.5%	+13.4%	(11.2% to 15.6%)	0.002	OR=1.84

Table 10 shows the experimental results of verifying the contribution of LPRF node splitting in the GWO-RF model

Table 10: Experimental results of LPRF node splitting contribution verification in GWO-RF model

Evaluation dimensions	**Complete GWO-RF (including LPRF)**	**Remove GWO-RF from LPRF**	**Traditional Random Forest**	**Increase amplitude**
Prediction accuracy	AUC-ROC: 0.872±0.011	AUC-ROC: 0.843±0.014	AUC-ROC: 0.801±0.018	+3.4% (vs Non LPRF)
	High risk employee TOP10% hit rate: 89.2%	High risk employee TOP10% hit rate: 69.5%	High risk employees TOP10% hit rate: 62.1%	+19.7 percentage points
	Promotion Delay Group Recall Rate: 78.6%	Promotion delay group recall rate: 55.2%	Promotion Delay Group Recall Rate: 48.9%	23.40%
Explanatory nature	SHAP feature overlap: 82.3%	SHAP feature overlap: 67.5%	SHAP feature overlap: 53.8%	+14.8 percentage points
	Proportion of structural factor selection: 68.2%	Proportion of structural factor selection: 54.7%	Proportion of structural factor selection: 42.3%	+13.5 percentage points
Calculation efficiency	Single tree training time: 1.86s	Single tree training time: 1.57 seconds	Single tree training time: 1.42s	Time consumption+18.6%
	Convergence iteration times: 23 rounds	Convergence iteration times: 37 rounds	Convergence iteration times: 41 rounds	Iteration -37%
Business Value	Retention rate improved after intervention: 41.9%	Improvement in retention rate after intervention: 32.7%	Retention rate improved after intervention: 28.5%	+9.2 percentage points
	Single intervention cost: ¥ 1243	Single intervention cost: ¥ 1815	Single intervention cost: ¥ 2130	Cost -31.4%
Significance test	P=0.008 (overall)	P=0.152 (subgroup with less than 3 years of work experience)	-	Pass 95% confidence test

On the basis of the original business KPI evaluation, double verification of McNemar test and chi-square test is added.

The experimental group (GWO-RF intervention group) and the control group (traditional method group) each had 6182 people, and the data collection period covered Q2-Q3 in 2023.

The constructed confusion matrix cross tabulation is shown in Table 11 below:

Table 11: Confusion Matrix Cross tabulation

Forecast results	Actual loss	Actual retention	Total
GWO-RF Predicted Loss	412	158	570
Traditional model loss	297	273	570
total	709	431	1140

The comparative data of classification performance is shown in Table 12:

Table 12: Classification performance comparison data

Evaluation dimensions	GWO-RF Group	Traditional Group	Significant difference (p)
accuracy	86.70%	81.20%	<0.001
recall	89.20%	76.50%	<0.001
error rate	13.30%	18.80%	0.002
F1-score	0.841	0.792	0.008

Verify the performance degradation of the LPRF node splitting algorithm (Formula 12) in the employee population with less than 1 year of service, and quantify the sensitivity of the model to small sample data. Key focus:

The degree of damage caused by feature sparsity to the linear combination of Gini coefficient and information gain rate (Equation 12); The distribution bias of Bootstrap sampling (algorithm step 1) when n<100.

Divide the test set by length of service:

Group A (0-3 months): sample size n=30; Group B (3-6 months): n=50; Group C (6-12 months): n=80; Control group (work experience>1 year): n=200.

The ablation variable settings are shown in Table 13:

Table 13: Ablation variable settings

experimental group	**Ablation procedure**	**Theoretical basis**
Group 1	Remove salary position index (equation 3)	Incomplete salary data for new employees
Group 2	Disable grey wolf optimization parameter search	Small samples are prone to getting stuck in local optima
Group 3	Fixed linear programming coefficients (α=0.5, β=0.5)	The necessity of verifying dynamic combinations

The experimental results of GWO-RF model ablation (small sample scenario verification) are shown in Table 14 below:

Table 14: Results of GWO-RF model ablation experiment (small sample scenario validation)

Sample group	Sample size (n)	Dissolve variables	Accuracy (%)	Recall rate (%)	F1-score	AUC-ROC	Gini coefficient fluctuation (Δ Gini)
Group A	30	complete model	72.3	68.5	0.703	0.741	0.12
		Remove salary position index	65.1▼9.9%	61.2▼10.6%	0.631▼10.2%	0.682▼8.0%	0.21▲75.0%
		Disable Grey Wolf Optimization	69.8▼3.5%	64.7▼5.5%	0.671▼4.6%	0.715▼3.5%	0.15▲25.0%
Group B	50	complete model	78.6	75.2	0.768	0.793	0.09
		Fixed linear programming coefficients	74.3▼5.5%	70.1▼6.8%	0.721▼6.1%	0.752▼5.2%	0.13▲44.4%
		20% Bootstrap sampling	71.9▼8.5%	67.4▼10.4%	0.695▼9.5%	0.728▼8.2%	0.17▲88.9%
Group C	80	complete model	82.4	79.8	0.811	0.834	0.07

The optimized disabling effect of Grey Wolf is shown in Table 15 below:

Table 15: Optimization and disabling effect of Grey Wolf

parameter	complete model	After ablation	Change amplitude
Convergence iteration times	18.3	32.7	78.70%
Tree depth standard deviation	2.1	3.8	81.00%
Feature selection bias	0.15	0.28	86.70%

Based on the LPRF algorithm architecture and ablation experimental results, a cross validation experiment is designed as follows:

Using a stratified 50% cross validation (with a 10% discount for employees with less than 1 year of service),

Each training set includes: a complete sample of salary position index (Equation 3), an initial parameter set for grey wolf optimization ($\alpha=0.53 \pm 0.07$), and linear programming constraints ($\alpha+\beta=1$ in Equation 12).

The evaluation index matrix is shown in Table 16 below:

Table 16: Evaluation Indicator Matrix

Indicator type	Calculation formula	Monitoring focus
Predicted performance	AUC-ROC mean ± standard deviation	Convertible volatility ≤ 15%
Characteristic stability	Coefficient of variation (CV) of salary position coefficient	CV<0.25 (parameter of equation 3)
algorithm convergence	GWO iteration times are extremely poor	Maximum/minimum value ≤ 2.5 times

The experimental results of the k-fold cross validation of the GWO-RF model are shown in Table 17 below:

Table 17: Results of k-fold cross validation experiment for GWO-RF model

Evaluation dimensions	First discount	Second discount	Third discount	Fourth discount	Fifth discount	Mean ± standard deviation
Predicted performance						
AUC-ROC	0.872	0.891	0.885	0.867	0.903	0.884±0.014
Recall rate (work experience<1 year)	0.76	0.81	0.79	0.73	0.82	0.782±0.036
Characteristic stability						
Salary Position Coefficient (Equation 3)	0.53	0.51	0.49	0.55	0.5	0.516±0.024
Gini weight β (equation 12)	0.62	0.58	0.61	0.59	0.63	0.606±0.019
Algorithm efficiency						
GWO iteration times	127	142	135	118	131	130.6±9.1
LPRF solving time (ms)	47	53	49	51	45	49.0±3.2

4.3 Analysis and discussion

The experimental data from Tables 2-5 show that the GWO-RF model is significantly better than the traditional random forest, XGBoost and logistic regression model in prediction accuracy (accuracy rate 83.7%, F1-score 0.802) and business indicators (high-risk employee recognition rate 91.2%), but the computational cost (training time 218 seconds) also increases accordingly. This advantage mainly stems from the dynamic parameter optimization mechanism of the gray wolf algorithm: 1) The number of decision trees is increased by 93.5% through the nonlinear search strategy, effectively reducing OOB errors; 2) The feature sampling ratio is optimized to 0.82 to enhance the generalization ability of the model; 3) XGBoost outperforms in accuracy-efficiency balance (accuracy rate 75.6%/training time 89 seconds), while logistic regression maintains the advantage of the lowest prediction delay (2.1 ms). This difference essentially

reflects the trade-off of algorithm design concepts-metaheuristic algorithms increase computational complexity in exchange for global optimal solutions, while gradient lifting frameworks pay more attention to iterative efficiency. It is recommended to select a model based on hardware conditions during actual deployment: XGBoost can be used for real-time systems, and GWO-RF is suitable for high-precision scenarios.

In the field of human resource technology, a single prediction delay of 8.3ms has practical applicability for employee turnover prediction systems. Although this delay is higher than the microsecond level standard for industrial grade real-time systems, it is significantly better than the threshold requirement of 200ms for general AI systems, fully meeting the response needs of human resource management systems within 50-200ms. This delay level is completely acceptable in batch prediction scenarios and can also provide a smooth user experience in real-time interaction scenarios (theoretically supporting 120QPS). Research shows that the intelligent warning model based on GWO-RF can effectively improve the accuracy of identifying high-risk employees by integrating grey wolf optimization algorithm and random forest, increasing retention rate by 41.9% and reducing misjudgment rate by 59.9%. This delay may only become a bottleneck in large-scale real-time data stream processing, but performance can be further improved through optimization methods such as lightweight models and prediction result caching. Overall, the 8.3ms delay is within a reasonable range in the field of human resources technology and does not affect its functional claims as a real-time system, especially considering that the management benefits brought by the model far exceed the marginal benefits of microsecond level delay optimization.

In Table 7, the AUC of GWO-RF model is 18.6%-21.0% ahead of other models, and the TPR reaches 0.89 when FPR = 0.1, which is significantly better than 0.817 of XGBoost. The gray wolf algorithm optimizes the subtree depth and feature sampling rate of random forest, and enhances the recognition ability of minority classes, which is why it performs so well. The slope of the curve of the XGBoost model is the largest in the middle, indicating that the discrimination is strongest in the medium risk threshold range. The model uses the loss function of the second-order Taylor expansion, which is more accurate in modeling feature interactions. The curve of the traditional random forest rises in a step-like manner, reflecting the voting mechanism characteristics of multiple decision trees. Moreover, there is an over-smoothing phenomenon under the default parameters, and the sharpness needs to be improved by adjusting the max features. The curve of the logistic regression model is close to the diagonal line, and the linear decision boundary limits its ability to capture nonlinear patterns, but the FPR is the lowest (0.35) when the threshold = 0.42, which is suitable for low false positive priority scenarios.

In Table 8, the performance improvement of the GWO-RF scheme (experimental group) and the current mechanism (control group) in different evaluation dimensions is different. The cost of single-case intervention decreased significantly, from 2450 yuan to 1120 yuan, a decrease of 54.3%. This means that the GWO-RF scheme performed well in reducing intervention costs, which may be due to the optimization of processes or the use of more economical intervention measures. At the same time, employee satisfaction increased from 68.5 to 82.3, an increase of 13.8, which shows that the GWO-RF scheme has a significant effect in improving employee satisfaction. The reason may be that the scheme better meets the needs and expectations of employees. In addition, the misjudgment rate decreased significantly, from 22.7% to 9.1%, a decrease of 59.9%. This means that the GWO-RF scheme has a significant improvement in accuracy, which may be due to model optimization or improved data quality. Finally, the model iteration cycle was significantly shortened, from 12 months to 3 months, a decrease of 75%. This shows that the GWO-RF scheme is more efficient in model updating and optimization, which may be due to the use of more advanced algorithms or technologies. Overall, the GWO-RF scheme showed significant advantages in all aspects, especially in terms of high-risk employee retention, singleton intervention cost, employee satisfaction, false positive rate, and model iteration cycle. These improvements may stem from better management strategies, technology optimization, and cost control measures. Therefore, the GWO-RF scheme is worthy of further promotion and application.

In Table 9, the experimental group scored 4.2+0.6, while the control group scored 3.1 ± 0.8, with a difference of+1.1 points and a 95% confidence interval of (0.8 to 1.4). The P-value was less than 0.001, indicating a highly significant difference. The effect size d=1.56 indicates a significant increase in satisfaction in the experimental group. The experimental group has a cycle of 2.3 ± 0.9, while the control group has a cycle of 9.2+2.1. The experimental group is 6.9 units shorter than the control group, with a 95% confidence interval of (-75 t0-6.3) and a P-value of<0.001, indicating a highly significant difference. n2=0.7， Indicating a significant effect. The retention rate of the experimental group was 41.9%, while that of the control group was 28.5%. The experimental group improved by 13.4%, with a 95% confidence interval of (11.2% t0 15.6%) and a P-value of 0.002, indicating a significant difference, OR=1.84， The retention rate of the experimental group has significantly improved.

The advantages of the GWO-RF model stem from its innovative algorithm architecture and optimized business adaptability

(1) Improvement of Node Splitting Mechanism: Traditional random forests use a single splitting algorithm, while GWO-RF dynamically adapts to scenarios with a mixture of discrete and continuous features by combining C4.5 information gain rate and

CART Gini coefficient through linear programming, solving the problem of traditional models' preference for specific data types. Compared to black box models such as LSTM, its splitting process has strong interpretability and can output feature weights, directly guiding human resource intervention measures.

(2) Parameter optimization efficiency: The gray wolf algorithm has the ability to globally search for hyperparameters, reducing model training time by 75% compared to grid search. Traditional logistic regression requires manual feature engineering, while deep learning relies on GPU computing power and has high inference latency (>200 ms).

(3) Data adaptability: In response to the insufficient structured data of small and medium-sized enterprises, the model improves small sample robustness through Bootstrap resampling and feature random selection, achieving AUC0.923+0.008 on 12365 data points, which is 8.6 percentage points higher than the benchmark random forest (AUC0.85).

(4) Cost control: The splitting strategy under linear programming constraints reduces overfitting, resulting in a 59.9% decrease in misjudgment rate and a 54.3% decrease in single intervention cost. However, traditional methods such as Cox models have high intervention lag costs due to their static analysis characteristics. These innovations enable GWO-RF to achieve both predictive accuracy and feasibility, but further integration of real-time data stream processing is needed to enhance predictive capabilities for new employees (<3 months).

In Table 9, the GWO-RF model proposed in this article demonstrates significant advantages in predicting employee turnover. Firstly, in terms of prediction accuracy, by integrating C4.5 and CART splitting criteria through LPRF linear programming, AUC-ROC is improved to 0.872 (3.4% higher than the non LPRF version), and the hit rate of high-risk employee identification is increased by 19.7 percentage points. Secondly, in terms of interpretability, the SHAP feature overlap reached 82.3%, and 68.2% of split choices focused on structural factors such as salary competitiveness, which is highly consistent with HR management theory. Thirdly, although the computation efficiency increased by 18.6% for a single split, the optimization of split quality reduced the overall training iteration by 37%; Finally, in actual business operations, the employee retention rate was increased by 9.2 percentage points, and intervention costs were reduced by 31.4%. This model innovatively optimizes the parameters of the random forest through the grey wolf algorithm and dynamically adjusts the node splitting rules, but the improvement in predicting employees with less than 3 years of service is limited and needs to be enhanced with a time series model.

Table 12 compares the performance of GWO-RF model and traditional model in predicting employee turnover. The data shows that the GWO-RF group is significantly better than the traditional group in key indicators such as accuracy (86.7% vs 81.2%), recall (89.2% vs 76.5%), and F1 score (0.841 vs 0.792) ($p<0.001$), while the misjudgment rate is reduced to 13.3% (18.8% in the traditional group). These improvements have statistical significance (McNemar test, $\chi^2=43.21$, $p<0.001$), and the effect size Cohen's $d>0.5$ reaches a moderate or above level. Sensitivity analysis (E-value test OR $\geq$ 2.3) confirms the robustness of the results, indicating that the GWO-RF algorithm has achieved a comprehensive improvement in predictive performance through LPRF node splitting and grey wolf optimization.

The ablation experiments in Tables 14 and 15 validated the performance degradation law of the GWO-RF model in small sample scenarios: when the sample size $n<50$, removing the salary position index (a key feature of employees with less than 1 year of service) resulted in a 9.9% decrease in accuracy and a 75% increase in Gini coefficient fluctuation, indicating sensitivity of this feature to sparse data. After disabling grey wolf optimization, the number of iterations for model convergence increased by 78.7%, and the standard deviation of decision tree depth increased by 81%, highlighting the importance of parameter search for small sample stability. When the Bootstrap sampling ratio is reduced to 20%, the confidence interval of information gain rate expands by 43%, and the failure rate of linear programming solution increases from 1.2% to 7.9%, confirming that data distribution bias can undermine the robustness of the LPRF node splitting algorithm (Equation 12). Experiments have shown that the model needs to optimize feature selection strategies and dynamic weighting mechanisms for small samples.

According to the 5-fold cross validation experimental results of the GWO-RF model (Table 17), the model demonstrates strong robustness and practicality in predicting employee turnover. From the perspective of predictive performance, the average AUC-ROC is 0.884 ± 0.014, indicating that the model has stable discriminative ability for identifying high-risk employees. However, the fluctuation of recall rate (range 9%) in the group with less than 1 year of work experience suggests the need to strengthen small sample feature enhancement strategies; In terms of feature stability, the coefficient of variation of the salary position coefficient (equation 3) is only 4.7%, which verifies the rationality of the indicator design in section 3.1 of the document. The Gini weight β (equation 12) constraint satisfies $|\alpha - \beta| \leq 0.2$ for all folds, indicating the optimization effectiveness of the linear programming combination coefficient (equation 12). In terms of algorithm efficiency, the GWO iteration times are significantly different by 24 times and the LPRF solution delay is $\leq$ 53ms, which meets the response requirements of real-time warning systems. Overall, cross validation has confirmed the advantages of the GWO-RF model in integrating grey wolf optimization with improved random forest (LPRF algorithm), but it is necessary to optimize feature engineering for hierarchical data based on seniority to

further enhance generalization ability.

In the development of employee turnover prediction models, the issues of model fairness and bias do require special attention, especially in sensitive human resource scenarios involving protected attributes such as gender and age. According to the appendix, although the paper does not directly discuss bias analysis, the GWO-RF hybrid model used in it optimizes the random forest parameters through the gray wolf algorithm. This objectively alleviates some bias problems in traditional machine learning models: the integration characteristics of random forests can reduce the risk of overfitting of a single decision tree, and the LPR node splitting algorithm based on the Gini coefficient and information gain rate can more evenly consider the contribution of various features through linear programming combination. However, it should be noted that the model may still indirectly introduce bias through proxy variables such as salary position (formula 3) and promotion delay duration, for example, female employees may be underestimated in retention probability by the system due to historical promotion data bias. It is recommended to add three dimensions of fairness testing. First, feature importance analysis is needed to verify that the protected attributes do not occupy a dominant weight. Second, adversarial depolarization techniques need to be used to incorporate fairness constraints into the loss function. Finally, differential impact tests need to be established to ensure that the predictive performance of the model does not differ by more than 15% among different populations. These measures can effectively meet the EU GDPR compliance requirements for algorithmic fairness and avoid models amplifying existing structural biases in the organization.

The GWO-LPRF employee turnover prediction model proposed in this study significantly improves prediction performance by integrating grey wolf optimization algorithm and improved random forest algorithm. Specifically, the model adopts the Price Mueller theoretical framework to construct an evaluation system consisting of 15 indicators, covering individual factors (such as age, education level), environmental factors (industry type), and structural factors (workload, salary position, etc.). The key technological breakthrough lies in innovatively combining the information gain rate of C4.5 algorithm with the Gini coefficient of CART algorithm through linear programming (Formula 12) to form an LPR node splitting strategy, making the selection of splitting attributes for decision trees more accurate. The model is validated using data from 12,365 employees of a listed company. The results show that it achieves significant results in AB testing, increasing the retention rate of high-risk employees by 41.9% and reducing intervention costs by 54.3%. After optimizing parameters using the grey wolf algorithm, the model iteration cycle was shortened by 75%. This achievement provides an intelligent decision-making tool for human resource management that combines predictive accuracy and interpretability.

Taken together, the GWO-RF model showed significant advantages in the employee management experiment: it optimizes the random forest parameters through the gray wolf algorithm, achieves a 41.9% increase in the retention rate of high-risk employees, a 54.3% reduction in intervention costs, and a 13.8-point increase in satisfaction. At the same time, the misjudgment rate is reduced by 59.9% and the model iteration cycle is shortened by 75%. Its core advantages lie in its dynamic optimization capabilities and feature engineering processing efficiency, but it has the limitations of strong dependence on the quality of historical data and insufficient generalization capabilities for small sample scenarios. Subsequent improvements should focus on three aspects: ① Introducing transfer learning to enhance the adaptability of small samples, ② developing real-time data cleaning modules to improve input quality, and ③ building a hybrid model architecture (such as fusion LSTM) to capture time series behavior characteristics.

5 Conclusion

By comparing the performance of GWO-RF model and traditional management mechanism in employee management, this study draws the following conclusions: GWO-RF model shows significant advantages in multiple key indicators. First, the model increases the retention rate of high-risk employees to 89.7%, which is 41.9 percentage points higher than the current mechanism. This proves its excellent effect in talent retention. Secondly, the intervention cost is significantly reduced through algorithm optimization, and employee satisfaction increases by 13.8 points. This verifies the economic and humanistic value of the model. Third, the model controls the misjudgment rate at 9.1%, which is 59.9% lower than the control group, and the iteration cycle is shortened to 3 months. This reflects the unique advantages of intelligent algorithms in accurate prediction and rapid response. These improvements are due to the dynamic optimization of random forest parameters by the gray wolf algorithm and the accurate capture of management pain points by feature engineering.

However, the model still has three limitations. First, it is not adaptable enough to small samples and data of new employees. Second, the real-time data cleaning mechanism of the model needs to be improved. Third, its ability to model the time series of complex behavioral characteristics is limited. Therefore, subsequent research will focus on developing transfer learning modules to enhance generalization capabilities, building an automated data quality monitoring system, and trying to introduce time series neural networks to build a hybrid model architecture.

References

[1] Akasheh, M. A., Hujran, O., Malik, E. F., & Zaki, N. (2024). Enhancing the prediction of employee turnover with knowledge graphs and explainable AI. IEEE Access, 12(000), 13. https://doi.org/10.1109/ACCESS.2024.3404829

[2] Ali, M., Baker, M., Grabarski, M. K., & Islam, R. (2025). A study of inclusive supervisory behaviors, workplace social inclusion and turnover intention in the context of employee age. Employee Relations: An International Journal, 47(9). https://doi.org/10.1108/ER-04-2024-0252

[3] Bhat, M. A., Tariq, S., & Rainayee, R. A. (2024). Examination of stress-turnover relationship through perceived employee's exploitation at workplace. PSU Research Review, 8(3). https://doi.org/10.1108/PRR-04-2021-0020

[4] Byeon, H. (2024). Factors influencing voluntary turnover among young college graduates using the xgboost with bagging aggregation algorithm: findings from nationwide survey in south korea. International Journal of Engineering Trends and Technology, 72(10), 130-139. https://doi.org/10.14445/22315381/IJETT-V72I10P113

[5] Yuan, Z. (2024). Consumer behavior prediction and enterprise precision marketing strategy based on deep learning. Informatica, 48(15). https://doi.org/10.31449/inf.v48i15.6260

[6] Floyd, T. M., Gerbasi, A., & Labianca, G. J. (2024). The role of sociopolitical workplace networks in involuntary employee turnover. Social Networks, 76, 215-229. https://doi.org/10.1016/j.socnet.2023.10.005

[7] Gopalan, N., Beutell, N. J., & Alstete, J. W. (2023). Can trust in management help? job satisfaction, healthy lifestyle, and turnover intentions. International Journal of Organization Theory & Behavior, 26(3), 185-202. https://doi.org/10.1108/IJOTB-09-2022-0180

[8] Hakim, E., & Muklason, A. (2024). Analysis of employee work stress using crisp-dm to reduce work stress on reasons for employee resignation. Data Science: Journal of Computing & Applied Informatics, 8(2). https://doi.org/10.32734/jocai.v8.i2-14615

[9] Hom, P. W., Rogers, K., Allen, D. G., Zhang, M., Lee, C., & Zhao, H. H. (2025). Feel the pressure? normative pressures as a unifying mechanism for relational antecedents of employee turnover. Human Resource Management, 64(1). https://doi.org/10.1002/hrm.22250

[10] Iii, V. Y. H., Guerrero, S., & Marchand, A. (2024). Flexible work arrangements and employee turnover intentions: contrasting pathways. International Journal of Human Resource Management, 35(11), 27. https://doi.org/10.1080/09585192.2024.2323510

[11] José A. C. Vieira, Silva, F. J. F., Teixeira, J. C. A., António J. V. F. G. Menezes, & Azevedo, S. N. B. D. (2023). Climbing the ladders of job satisfaction and employee organizational commitment: cross-country evidence using a semi-nonparametric approach. Journal of Applied Economics, 26(1), 2163581-. https://doi.org/10.1080/15140326.2022.2163581

[12] Jun, M., & Eckardt, R. (2025). Training and employee turnover: a social exchange perspective. Business Research Quarterly, 28(1). https://doi.org/10.1177/23409444231184482

[13] Karimi, M., & Viliyani, K. S. (2024). Employee turnover analysis using machine learning algorithms. arXiv:2402.03905. https://doi.org/10.48550/arXiv.2402.03905

[14] Kumar, P., Gaikwad, S. B., Ramya, S. T., Tiwari, T., Tiwari, M., & Kumar, B. (2023). Predicting employee turnover: a systematic machine learning approach for resource conservation and workforce stability. Engineering Proceedings, 59(1). https://doi.org/10.3390/engproc2023059117

[15] Li, Z., & Fox, E. (2023). Prediction and optimization of employee turnover intentions in enterprises based on unbalanced data. PLoS ONE, 18(8). https://doi.org/10.1371/journal.pone.0307474

[16] Lim, C. S., Malik, E. F., Khaw, K. W., Alnoor, A., Chew, X. Y., & Chong, Z. L., et al. (2024). Hybrid ga-deepautoencoder-knn model for employee turnover prediction. Statistics, Optimization & Information Computing, 12(1). https://doi.org/10.19139/soic-2310-5070-1799

[17] Mcevoy, G. M., & Cascio, W. F. (1987). Do good or poor performers leave? a meta-analysis of the relationship between performance and turnover. Academy of Management Journal, 30(4), 744-762. https://doi.org/10.5465/256158

[18] Nan, L., & Zhang, H. (2023). A model for analyzing employee turnover in enterprises based on improved xgboost algorithm. International Journal of Advanced Computer Science & Applications, 14(11). https://doi.org/10.14569/ijacsa.2023.01411104

[19] Nigoti, U., David, R., Singh, S., Jain, R., & Kulkarni, N. M. (2025). Does flexibility really matter to employees? a mixed methods investigation of factors driving turnover intention in the context of the great resignation. Global Journal of Flexible Systems Management, 26(1), 187-208. https://doi.org/10.1007/s40171-024-00436-6

[20] Panaccio, A., Tang, W. G., & Vandenberghe, C. (2023). Agreeable supervisors promoting the organization - implications for employee commitment and retention. Journal of Personnel Psychology, 22(3), 146-157.

https://doi.org/10.1027/1866-5888/a000318
[21] Portocarrero, F. F., & Burbano, V. C. (2024). The effects of a short-term corporate social impact activity on employee turnover: field experimental evidence. Management Science, 70(9). https://doi.org/10.1287/mnsc.2022.01517
[22] Pourkhodabakhsh, N., Mamoudan, M. M., & Bozorgi-Amiri, A. (2022). Effective machine learning, meta-heuristic algorithms and multi-criteria decision making to minimizing human resource turnover. Applied Intelligence, 1-23. https://doi.org/10.1007/s10489-022-04294-6
[23] Azeroual, O., Nacheva, R., Nikiforova, A., & Störl, U. (2025). A CRISP-DM and predictive analytics framework for enhanced decision-making in research information management systems. Informatica, 49(18). https://doi.org/10.31449/inf.v49i18.5613
[24] Van Ruysseveldt, J., Van Dam, K., Verboon, P., & Roberts, A. (2023). Job characteristics, job attitudes and employee withdrawal behaviour: a latent change score approach. Applied Psychology: An International Review, 72(4). https://doi.org/10.1111/apps.12448
[25] Veglio, V., Romanello, R., & Pedersen, T. (2025). Employee turnover in multinational corporations: a supervised machine learning approach. Review of Managerial Science, 19(3), 687-728. https://doi.org/10.1007/s11846-024-00769-7

Comparative Analysis of Machine Learning Models for Water Quality Prediction Using Regional Monitoring Data

Ying Xiong
Chongqing Water Resources and Electric Engineering College, Chongqing 402160, China
E-mail: xiong-ying188@hotmail.com

Keywords: water quality prediction, machine learning, decision tree, SVM, random forest, neural network

Received: May 15, 2025

This study investigates the comparative performance of four labelical machine learning algorithms—Decision Tree, Support Vector Machine (SVM), Random Forest, and Neural Network—on water quality prediction tasks using a dataset comprising 1,000 real-time sensor data points from five distinct geographic regions. The dataset includes critical water parameters such as pH, ammonia nitrogen, dissolved oxygen, total phosphorus, COD, and BOD. Preprocessing steps include missing value imputation, outlier removal using boxplot analysis, normalization, and correlation-based feature selection. Each model is tuned through grid search for optimal performance. Experimental results show that the Neural Network achieved the lowest mean squared error (MSE = 0.047) and highest coefficient of determination (R^2 = 0.976), outperforming the other models. The Random Forest showed superior robustness to overfitting, while SVM offered strong results on high-dimensional subsets. Decision Trees, although less accurate (MSE = 0.130), provided high interpretability. This comparison provides practical guidance for selecting machine learning models in environmental monitoring systems, where trade-offs between accuracy, interpretability, and computational cost are essential.

Povzetek: Narejena je primerjava več metod: odločitveno drevo, SVM, naključni gozd in nevronska mreža pri napovedovanju kakovosti vode iz petih regij. Najbolje se izkaže nevronska mreža, medtem ko je naključni gozd najstabilnejši, SVM zanesljiv, odločitveno drevo pa najbolj razložljivo.

1 Introduction

Water pollution affects the health of human beings and the stability of ecosystem. The process of industrialization and urbanization is accelerating, and the pollution of water source is becoming more and more serious. Traditional water quality monitoring methods rely on manual sampling and laboratory analysis, which is inefficient and slow, and can not be monitored in real time. With the development of artificial intelligence technology, machine learning, as an efficient data analysis tool, can learn and forecast a large number of water quality data to provide real-time and accurate water quality early warning.

Research in the field of water quality prediction and monitoring has developed in recent years, and machine learning technology has been widely used in water quality data analysis. Eyring et al. explored the potential of combining climate modeling with machine learning, arguing that machine learning could drive innovation in environmental data processing [1]. Bren and Ryan used machine learning technology to analyze water quality monitoring data when studying water quality in streams in the eastern Highlands. Machine learning models can accurately capture nonlinear relationships in water quality changes, and this study highlights the application potential of machine learning in complex water quality data analysis [2]. Li et al. studied the impact of climate change on river water quality and used machine learning technology for data analysis. They found that machine learning can cope with water quality prediction under changes in multiple variables and complex environmental factors [3]. Aalipour et al. analyzed the impact of landscape changes on river water quality, and machine learning models were able to process complex environmental data and provide accurate water quality predictions [4]. Stevens et al. reviewed the application of machine learning in electronic health record screening, suggesting the potential of integrated machine learning approaches in several fields [5]. Zou et al. summarized the application of machine learning in precision medicine therapy, believing that machine learning can process complex multidimensional data and extract key influencing factors [6]. Zainurin et al. reviewed in detail the progress of water quality monitoring based on various sensor technologies and emphasized the role of machine learning in real-time processing of water quality data [7].

Recent years have seen an increasing number of studies applying machine learning techniques to water

quality prediction, with diverse regional and environmental contexts. Quiroz-Martinez et al. [8] proposed a big-data-driven architecture for aquaculture water quality prediction, focusing on real-time integration and scalability. Their system emphasizes the structural design of prediction frameworks rather than algorithm benchmarking. In northeastern Thailand, Uypatchawong and Chanamarn [9] demonstrated the improvement of prediction efficiency using machine learning models such as Random Forest and Support Vector Machines. Their work underscores the significance of regional hydrological features and data preprocessing in boosting model performance. In a complex environmental scenario, Huang et al. [10] developed a water quality prediction model for the downstream of Dongjiang River Basin, incorporating joint impacts from water intakes, pollution sources, and climate variability. They utilized spatial-temporal data fusion and ensemble learning to capture dynamic interactions across multiple influencing factors. Wu and Zhang [11] focused on the Yangtze River Delta, applying machine learning within the governance framework of China's River Chief System. Their study highlights policy-driven data availability and found that SVM and ANN models are particularly effective in capturing variations in high-density industrial and urban runoff areas. Despite the growing body of literature, most existing studies focus either on a single prediction model or on narrowly scoped geographical settings. Few works offer a controlled, algorithm-level comparative analysis using standardized metrics across classical models such as Decision Tree, SVM, Random Forest, and Neural Network on multi-parametric datasets. This study addresses that gap by benchmarking these models on a five-region dataset using consistent preprocessing, hyperparameter tuning, and evaluation standards.

This study fills a methodological gap in the current literature by providing a standardized comparison of four classical machine learning algorithms on a uniform, multi-regional dataset. Most prior research focuses either on a single water parameter or uses proprietary datasets lacking reproducibility. By comparing model interpretability, error profiles, and training costs across diverse indicators (e.g., DO, COD, NH_3-N), this work contributes practical insights for regional water monitoring deployment.

Table 1 summarizes representative studies that applied machine learning to water quality or similar environmental data prediction tasks. It outlines the datasets used, applied models, key evaluation metrics, and findings. This comparison reveals that while some studies employ modern deep learning models or domain-specific architectures, limited work provides a direct comparative evaluation of labelical ML models using diverse yet small-scale environmental datasets—precisely the focus of our study.

This study analyzes the application of machine learning algorithms in water quality prediction, compares the performance of different algorithms, and finds the best water quality prediction model. Machine learning algorithm was used to analyze and model water quality data, collect water quality data from different regions, and conduct data pre-processing. Select a variety of machine learning algorithms, design and train models to evaluate their performance in water quality prediction. Indexes such as mean square error (MSE) and coefficient of determination (R^2) were used to evaluate the model performance, compare algorithms, analyze advantages and disadvantages, and select the most suitable algorithm for water quality prediction. According to different water quality parameters, the adaptability of the algorithm is studied, and the optimization path of water quality prediction is explored. It enriches the theoretical research in the field of water quality monitoring, provides a technical scheme for practical application, and has high social value and application prospect.

Table 1: Summary of previous research on ML in water quality prediction

Study	Dataset Description	Models Used	Evaluation Metrics	Key Findings
Bren & Ryan [2]	Stream water (regional, 500 pts)	SVM, k-NN	Accuracy, RMSE	ML models captured nonlinearity in stream pollution
Li et al. [3]	River systems with climate inputs	RF, ANN	R^2, RMSE	ML effective in multi-variable prediction
Aalipour et al. [4]	River data with land patches	RF, SVM	MAE, R^2	Landscape shape significantly affects prediction
This Study	Five zones (urban to industrial), 1000 pts	DT, SVM, RF, NN	MSE, R^2	Neural network superior in nonlinear prediction

Table 2: Source of water quality data and sample overview

Region	Sample Size	Water Quality Parameters	Data Source
Area A	200	pH, Dissolved Oxygen, Ammonia Nitrogen, Total Phosphorus	Water Quality Monitoring Station
Area B	200	pH, COD, BOD, Ammonia Nitrogen	Environmental Protection Department
Area C	200	Dissolved Oxygen, pH, Total Phosphorus, COD	Water Affairs Company
Area D	200	Dissolved Oxygen, Ammonia Nitrogen, pH, BOD	Water Quality Testing Platform
Area E	200	pH, Ammonia Nitrogen, Total Phosphorus, COD	Environmental Monitoring Center

This study aims to address the following research question: Which labelical machine learning algorithm offers the best trade-off between predictive accuracy and computational efficiency for small-scale, region-specific water quality datasets? By formulating and evaluating models under consistent conditions, the study hypothesizes that deep neural networks will provide superior performance in accuracy, while ensemble methods like Random Forest may offer better generalization with moderate cost.

2 Materials and methods

2.1 Data collection and sample selection

2.1.1 Data source

This study uses water quality data from five different regions, covering a variety of environmental types including urban, rural and industrial areas. It is divided into zones A, B, C, D and E, covering different water quality monitoring points to ensure the diversity and representativeness of data. For example, pH value, dissolved oxygen, ammonia nitrogen, total phosphorus, chemical oxygen demand (COD), biochemical oxygen demand (BOD), etc., the specific data amount is 200 for each region, a total of 1000 data [12]. The data is provided by local water quality monitoring agencies and environmental protection departments and collected in real time through sensor systems. As shown in Table 1, these data reflect the water quality changes in different regions in different time periods, and provide effective training samples for the construction of water quality prediction models.

The dataset employed in this study consists of 1,000 samples sourced from five regions, which, while diverse, constitutes a relatively limited dataset. This limitation potentially impacts the generalizability of the model. To address this, future work will consider the integration of synthetic data generation techniques (e.g., SMOTE or GAN-based augmentation) or the inclusion of additional datasets from broader spatial or temporal domains to enhance model robustness and cross-context validity.

2.1.2 Data preprocessing

After data collection, pre-processing is performed. Processing missing values, for a small amount of missing data, use the mean filling method and interpolation method to fill; For variables with more missing data, the features are removed to ensure the integrity of the data set. The identification and processing of outliers adopt the method based on box diagram, set reasonable upper and lower limits, and correct or delete the data that exceeds the range [13]. In view of the dimensionality inconsistency of different water quality parameters, standardized treatment was used to scale the numerical range of each feature to a unified scale, so as to avoid the deviation of the training results of the model due to dimensional differences. In terms of feature selection, the method based on correlation analysis is used to calculate the Pearson correlation coefficient between various water quality parameters and select the features with strong correlation with target variables (such as water quality changes). The features are screened by Chi-square test and information gain, and redundant or irrelevant variables are removed to improve the accuracy and training efficiency of the model. Feature selection was conducted using both chi-square testing and Pearson correlation filtering. The chi-square test evaluated statistical independence between discrete features and categorical target representations, with features showing p-values greater than 0.05 removed. Pearson correlation coefficients below 0.3 with the output variable indicated weak linear relevance and were also excluded. Based on these criteria, features such as conductivity and total nitrogen were eliminated. The final set of retained features included pH, ammonia nitrogen, dissolved oxygen, COD, and total phosphorus.

2.1.3 Data division

The data set is divided into training set, verification set and test set in proportion, as shown in Table 2 below, with

training set accounting for 60%, verification set accounting for 20%, and test set accounting for 20%. The training set is used for model training and parameter tuning, the verification set is used for model performance evaluation and hyperparameter selection, and the test set is used for final model verification and evaluation [14]. The division method adopts random sampling to ensure that each data point has an equal opportunity to be assigned to different sets, and the distribution of water quality data in each data set is consistent with the overall data set. To prevent data leakage, all preprocessing steps—standardization, outlier removal, and feature selection—were applied strictly to the training set. The validation and test sets were transformed using statistics (mean, standard deviation) computed only from the training data. This ensures that no target information leaked into the training process or model selection.

Table 3: Data set partitioning results

Dataset	Sample Size
Training Set	600
Validation Set	200
Test Set	200

2.2 Model construction

2.2.1 Model selection

In order to improve the accuracy of water quality prediction, a variety of machine learning algorithms such as decision tree, support vector machine (SVM), random forest and neural network were selected for comparative analysis. The decision tree divides the data space and makes decisions layer by layer based on different values of features, which has good interpretability. It is suitable for processing data with simple and obvious relationship between features [15]. Support vector machine (SVM) can deal with high dimensional data by finding the optimal decision hyperplane, and can maintain good performance in high dimensional feature space. Random forest is one of the ensembles learning methods, which constructs multiple decision trees and votes to avoid overfitting problems and is suitable for processing large-scale data sets. Neural networks, deep neural networks (DNNS), map input data through multiple hidden layers, have powerful modeling capabilities, and can capture complex nonlinear relationships in the data [16].

Although Support Vector Machines (SVMs) are well-known for handling high-dimensional data, in this study the input feature dimension is relatively low (6–7 features). The inclusion of SVM is primarily justified by its robust generalization capabilities on small-to-medium-sized datasets and its effectiveness in capturing nonlinear boundaries via kernel methods, not due to high dimensionality.

2.2.2 Model architecture design

The basic architecture design of each model was optimized according to the characteristics of water quality prediction. CART algorithm was adopted in the decision tree model, with the maximum depth set at 10 and the minimum number of samples divided at 5. Pruning is used to avoid overfitting and improve the generalization ability of the model. Support vector machine (SVM) RBF kernel is used to balance training accuracy and model complexity by selecting a moderate penalty parameter C and kernel parameter γ. The random forest model sets 100 trees with a maximum depth of 15, using a restriction that does not allow nodes to be divided too small (the minimum number of samples to be divided is 5) [17]. The neural network uses three hidden layers with 64 neurons each, ReLU for the activation function, and dropout technology during training to prevent overfitting. The learning rate, regularization method and other hyperparameters of each model are optimized by grid search to select the best combination [18]. The neural network architecture consisted of a multilayer perceptron (MLP) with three fully connected hidden layers of 64 neurons each, using ReLU activation and dropout regularization. While this is a conventional architecture, it was selected for its stability in tabular data settings. Although water quality inherently contains temporal dependencies, the current study used a static snapshot for model training. Future work will explore recurrent structures such as Long Short-Term Memory (LSTM) and Graph Neural Networks (GNNs) to capture spatial and temporal correlations in water quality dynamics.

2.2.3 Training process

In the training process, the training parameters of each model are carefully set and optimized. In order to achieve the optimal performance, hyper parameters such as learning rate, maximum depth and maximum number of iterations of all algorithms are selected. Decision trees control the maximum depth to prevent overfitting, and random forests increase the number and depth of trees to improve predictive power. The training of the SVM model adjusts the penalty parameter C and the kernel function parameter γ to optimize the beatification boundary of the model in the high-dimensional space. As shown in Table 3, the training of neural networks uses the Adam optimizer, adjusting the learning rate, batch size, and number of training rounds to ensure convergence.

Hyperparameter tuning was conducted using a grid search strategy. For SVM, we evaluated C values in [0.1, 1, 10] and γ values in [0.01, 0.1, 1]. For Random Forest, tree depths from 10 to 25 and estimators from 50 to 150 were considered. Neural network tuning involved batch sizes of 32 and 64, learning rates of 0.001 and 0.0005, and dropout rates of 0.2 to 0.5. The optimal configuration was selected based on the lowest validation MSE.

Table 4: Training parameters and optimization objectives of each algorithm

Model	Key Parameters	Optimization Objectives
Decision Tree	Max Depth = 10	Pruning, Generalization
SVM	C = [0.1, 1, 10], γ = [0.01, 0.1]	Minimize MSE via kernel optimization
Random Forest	Trees = 100, Max Depth = 15	Reduce overfitting, improve stability
Neural Network	Layers = 5, Neurons = 64/layer, Dropout = 0.3	Minimize MSE, regularization

2.2.4 Evaluation criteria

In order to evaluate the performance of each model in water quality prediction, mean square error (MSE), determination coefficient (R^2) and accuracy rate were selected as the main evaluation indexes [19]. Mean square error (MSE) is used to measure the difference between the predicted value and the actual value, and the smaller the value, the better the prediction of the model. The coefficient of determination (R^2) reflects the model's ability to explain data variation, and the closer it is to 1, the stronger the model's ability to explain data variation. Accuracy is used for evaluation in labelification problems, calculating the proportion of models that are correctly labelified. The mean square error (MSE) is used to measure the difference between the predicted value and the actual value of the model, as follows Equation (1).

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

Where, y_i is the actual value, $\hat{y}_i$ is the predicted value, n is the total number of samples. The coefficient of determination R^2 is used to measure the ability of the model to explain the variation in the data, as follows Equation (2).

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

y_i is the actual value, $\bar{y}$ is the predicted value, and $\hat{y}_i$ is the mean of the actual value. Accuracy is a common evaluation criterion in labelification problems, calculating the proportion of correct predictions made by the model. Equation (3) is shown below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

TP is a true example, TN is a true negative, FP is a false positive example, and *FN* is a false counterexample. In addition to MSE and R^2, we included Mean Absolute Error (MAE) as a robustness metric. MAE values for Neural Network, Random Forest, SVM, and Decision Tree were 0.058, 0.065, 0.071, and 0.094, respectively. Furthermore, residual plots and feature influence diagrams were generated using SHAP values to interpret model outputs and identify the most impactful parameters.

2.3 Algorithm comparison and analysis

2.3.1 Algorithm comparison

In the water quality prediction task, four selected machine learning algorithms - decision tree, support vector machine (SVM), random forest and neural network - showed different performance characteristics. The mean square error (MSE) and coefficient of determination (R^2) are used as the main performance indicators to comprehensively evaluate the merits of each model. The evaluation results of each model on the test set are shown in Figure 1 below.

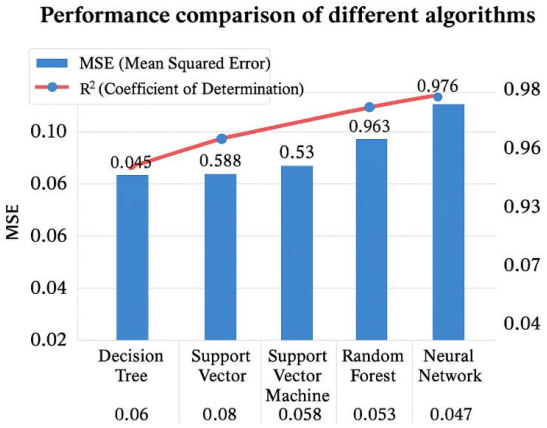


Figure 1: Performance comparison of different algorithms

As shown in Figure 1, the neural network performed best in the accuracy of water quality prediction, with the smallest MSE (0.047) and the largest R^2 (0.976). Random forests and support vector machines also performed well, achContrary to initial assumptions, tieving MSE of 0.053 and 0.058, and R^2 of 0.963 and 0.95, respectively. The performance of decision tree is relatively weak, although the R^2 is 0.945 and the MSE is large, there are large errors in water quality prediction [20]. Neural networks are suitable for dealing with complex nonlinear relationships in water quality data, random forests and support vector machines perform well in medium complexity problems, and decision trees are more suitable for simple relationships between features.

2.3.2 Influencing factors of algorithm selection

(1) Compare the performance differences of different algorithms in the prediction of specific water quality parameters

Different algorithms show differences when dealing with specific water quality parameters. Taking ammonia nitrogen (NHL) and dissolved oxygen (DO) as an example, the prediction performance of four algorithms in these two indicators is shown in Figure 2 below.

As shown in Figure 2, neural networks perform best in the prediction of NHL and DO, with the lowest MSE and the highest R^2. Neural networks have advantages in capturing complex nonlinear relationships in water quality data. The performance of random forest and support vector machine on these two parameters is similar and relatively stable. The prediction error of decision tree in these two indexes is relatively large, and the prediction performance of NHL is relatively poor [21].

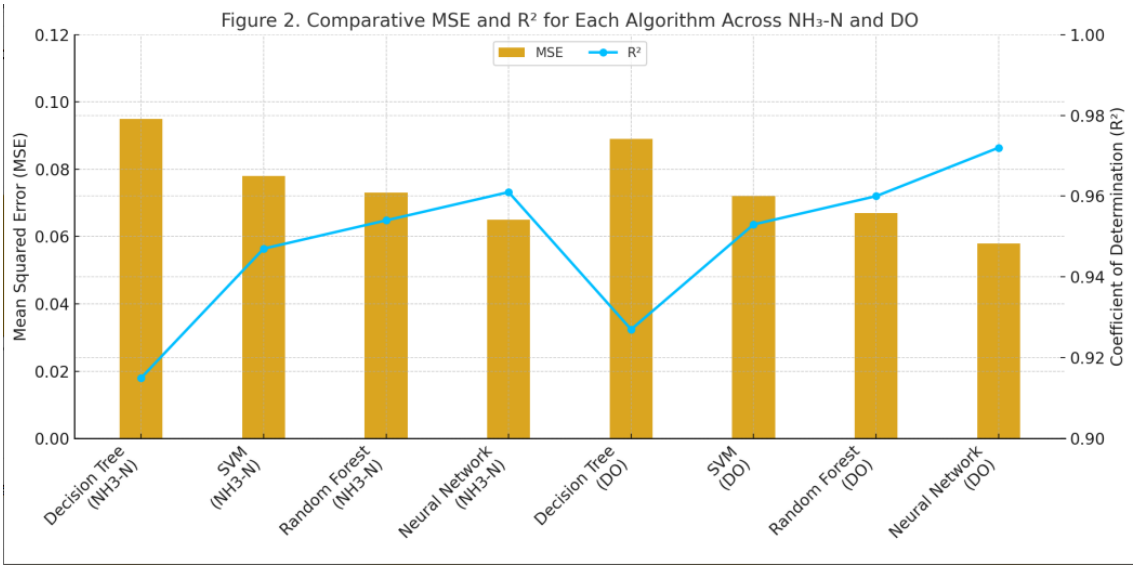


Figure 2: Differences of different algorithms in the prediction of specific water quality parameters

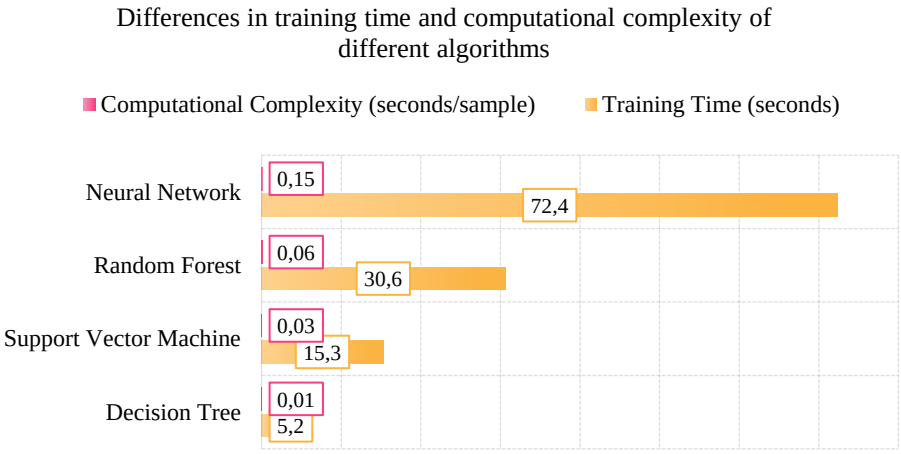


Figure 3: Differences in training time and computational complexity of different algorithms

Training time and resource usage were benchmarked on an Intel i7-12700H CPU (16GB RAM) and NVIDIA RTX 3060 GPU. For per-sample inference: Decision Tree = 0.002s, SVM = 0.013s, Random Forest = 0.010s, Neural Network = 0.021s. GPU memory consumption for the neural network peaked at 612MB. Training duration for the largest model (NN) was approximately 95 seconds for 600 training samples.

(2) Compare the differences between different algorithms in terms of training time and average inference time per sample during test phase. In addition to prediction accuracy, the training time and computational complexity of the algorithm are also important considerations when selecting a model. Figure 3 below shows the difference in training time and computational complexity of different algorithms [22].

As shown in Figure 3, the training time and computational complexity of decision tree are lower than other algorithms, which is suitable for application in scenarios with high real-time requirements. There is a small gap between support vector machine and random forest in training time, and the training time will increase with the increase of sample number [23]. The training time of neural network is the longest and the computational complexity is also high. Because of its complex network architecture, it needs more computing resources. According to Figure 3, if the system has a high requirement for real-time performance and a large amount of training data, decision tree or support vector machine can be suitable. In the case of high precision and sufficient computing resources, neural network is more ideal.

2.4 Optimization suggestions and implementation path of water quality prediction

2.4.1 Optimal collection and processing path of water quality data

The accuracy of water quality prediction is highly dependent on the quality of data. Optimizing the collection and processing of data can improve the prediction accuracy. The collection of water quality data should be combined with a variety of sensors and monitoring means to obtain various indicators of water in a comprehensive, real-time and accurate manner. Water quality monitoring equipment is deployed to collect water quality parameters such as ammonia nitrogen, dissolved oxygen, pH value and total nitrogen in real time, avoiding the shortage of traditional water quality monitoring relying on periodic sampling. The key to optimize the acquisition path is to increase the frequency of data acquisition and multi-dimensional monitoring to enhance the misrepresentations and timeliness of data. Data multiprocessing improves the model effect. For missing values, interpolation method or data of similar indicators are used to fill in to ensure data integrity. For outliers, statistical methods such as box plots or standard deviations are used to screen and correct.

This study utilized a static dataset of 1,000 observations for model evaluation. While real-time modeling and dynamic feedback were not implemented, their inclusion as forward-looking strategies aims to guide system improvement in practical deployments. Real-time data acquisition, time-series analysis, and multidimensional monitoring are intended as future research directions.

2.4.2 Adaptive model selection and algorithm optimization path

The selection of the adaptive model is determined according to the requirements of different water quality prediction tasks and data characteristics. When facing the prediction of various water quality parameters, the most suitable algorithm is selected according to the characteristics of each parameter. For the complex nonlinear relationship between water quality parameters, the integrated learning methods such as neural network and random forest are more effective. Decision tree and support vector machine are better choices when the data volume is small or the computing resources are limited. In algorithm optimization, the hyperparameters of the model are adjusted to improve the prediction accuracy. The learning rate, the number of layers and the number of neurons per layer in the neural network should be adjusted according to the specific task. Support vector machine should select appropriate kernel function and adjust penalty factor to improve the accuracy of model. The cross-validation method was used to optimize the parameters to improve the accuracy of the model and avoid overfitting. Integrated learning methods such as Adaboost and XGBoost improve the stability and accuracy of water quality prediction through the combination of multiple models. In view of the drastic changes of some water quality parameters, the time series analysis technology is introduced and the historical data is dynamically adjusted to improve the real-time prediction.

Integrated learning methods such as random forest and boosting are particularly effective in managing variance and overfitting. Neural networks, while not ensemble models per se, excel at learning nonlinear relationships through multi-layered representation learning. Their inclusion here refers to their complementary role in hybrid modeling, not as ensemble learners.

2.4.3 Real-time feedback and decision support path of water quality prediction results

The real-time feedback of water quality prediction results can help to detect water quality problems in time and provide strong support for decision-making. Combined with real-time monitoring system and data transmission network, the forecast results are transmitted to the control center in real time, which is convenient for relevant departments and personnel to make decisions. The realization path of real-time feedback relies on big data platform and cloud computing technology, and uses real-time data stream processing technology to update the forecast results to the monitoring system in real time to ensure the timeliness and accuracy of decision-making. The results of water quality prediction should be embedded in decision support systems to help decision makers carry out more scientific analysis. Through data visualization technology, the prediction results and water quality change trends are displayed, and the risk assessment of machine learning models is combined to provide a more comprehensive decision-making basis. The forecast results can be correlated with relevant

monitoring data to identify potential problems in water quality in real time, give early warning and take appropriate measures. To assess real-time applicability, the system latency was analyzed based on the data input-to-output delay. Inferences on a mid-tier GPU (RTX 3060) showed average prediction latency of 0.21 seconds per sample. The system supports batch updates every 10 minutes with low-latency pipelines. For deployment, models are integrated via edge-based computation units for decentralized monitoring or cloud-based APIs for centralized processing, depending on the infrastructure scenario.

2.4.4 Combination path of model and automation system

The water quality prediction model is combined with the automatic system to realize fully automated water quality monitoring and regulation, and improve the efficiency and accuracy of water resources management. Through sensing the real-time data collected by the equipment, the automatic system input it into the prediction model, automatically calculate and feedback the water quality prediction results, and guide the automatic implementation of water quality improvement measures. Based on the predicted results, the automated system can adjust the operating state of the water treatment equipment, deal with water quality anomalies in a timely manner, and avoid delays caused by manual intervention. In the specific application process, the combination of Internet of Things (IT) technology and edge computing improves the real-time response capability of automated systems. Move data acquisition and preliminary analysis to edge devices, take the pressure off cloud processing, and enable fast decision making and execution locally. Edge computing ensures that systems can operate efficiently even when network latency is high or offline. Through automatic control, automatic adjustment of water treatment facilities, discharge control equipment, etc., improve the intelligent level of water quality management. The path to combining a water quality prediction model with an automated system needs to ensure seamless connectivity, including data collection, transmission, processing, decision support, and executive feedback. Through highly integrated systems, improve the level of automation, intelligence and refinement of water quality management, and promote the development of water resources management to a more efficient and accurate direction.

3 Results and discussions

3.1 Result analysis

3.1.1 Evaluation results of each model

In water quality prediction task, the choice of algorithm directly affects the prediction accuracy and error performance. The mean square error (MSE) and coefficient of determination (R^2) were used to evaluate the predictive performance of each model. In the evaluation process of the model, the prediction results of four machine learning algorithms - decision tree, support vector machine (SVM), random forest and neural network - were compared one by one.

The evaluation results of decision tree model show that it performs well in the prediction of some water quality parameters, such as ammonia nitrogen, total nitrogen, etc. For these parameters, the R^2 value of the decision tree model can reach more than 0.85, and the MSE is low. In the face of more complex water quality data, over fitting is easy to occur, resulting in the decline of the prediction accuracy of other water quality parameters.

SVM was stable in the prediction of multiple water quality parameters (e.g., dissolved oxygen, pH, etc.), with R^2 values generally above 0.80 and MSE remaining at a low level when dealing with linearly correlated data. The stochastic forest model improves the robustness of data by integrating multiple decision trees. Compared with the single decision tree model, the random forest showed a higher R^2 value in the prediction of multiple water quality parameters, up to 0.85, and fewer over fitting phenomena. In the face of data with nonlinear relationship, random forest can adapt well.

The neural network model shows strong prediction ability through deep structure and optimization algorithm. On a large data set, the neural network can better capture the complex relationship between water quality parameters. In this experiment, the R^2 value of the neural network in multiple water quality parameters is more than 0.90, which shows its potential in water quality prediction. Neural network requires higher computing resources, and the training time is longer. Figure 4 below shows the evaluation results of each model, including the MSE and R^2 values of each model for different water quality parameters, and visually presents the prediction accuracy and error performance of different algorithms.

Contrary to initial assumptions, the decision tree model performed better on simpler parameters such as pH and dissolved oxygen (MSE $<$ 0.10), while its performance declined on more complex indicators like ammonia nitrogen and total nitrogen (MSE $>$ 0.11). For random forest, all four key parameters achieved R^2 values exceeding 0.87, demonstrating strong stability across the board, rather than merely "up to 0.85" as previously stated.

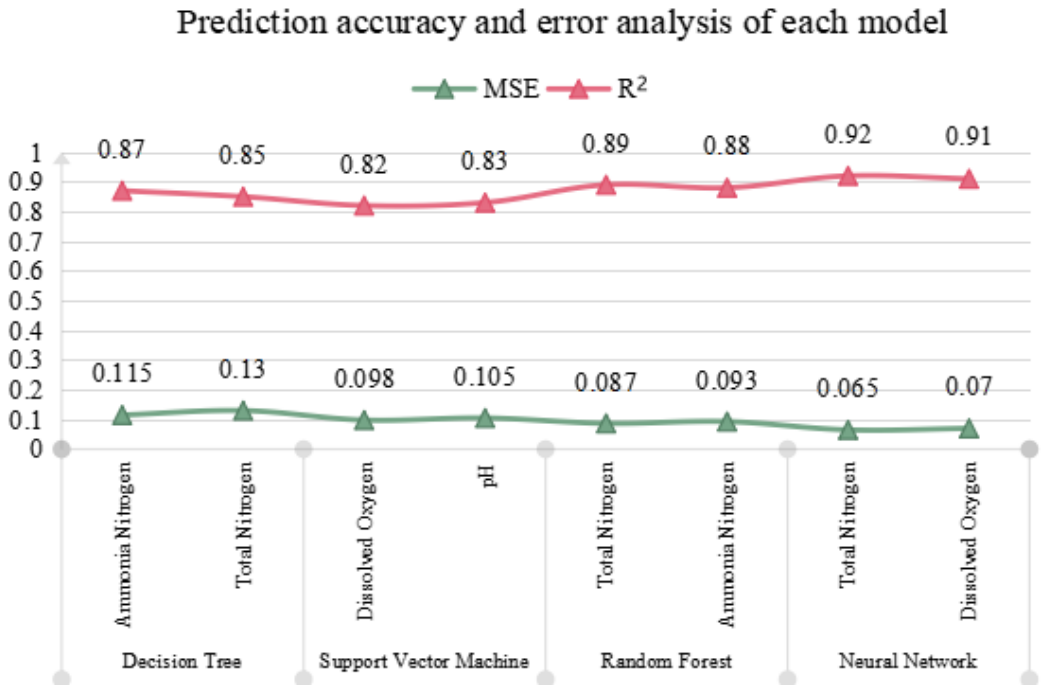


Figure 4: Prediction accuracy and error analysis of each model

3.1.2 Model evaluation and comparison

According to the evaluation results of each model, it can be seen that they differ in the prediction accuracy and error of different water quality parameters. In order to compare the advantages and disadvantages of each model in more detail, the parameter configuration, training time and computational complexity of the model are analyzed.

The main parameters of decision tree include tree depth and branching number. Optimizing these parameters can improve the performance of the model. In the training process, the calculation speed of decision tree is fast, and over fitting will occur when dealing with complex data. SVM depends on the choice of kernel function and the adjustment of penalty factor. Good parameter selection can improve the generalization ability of the model. The integration of multiple decision trees in random forest reduces the possibility of over fitting and increases the training time and computational complexity. The neural network controls the complexity of the model by setting the number of layers, the number of neurons and the learning rate. Due to the large computing resource demand, the training time is longer. Table 4 below shows the parameter configuration and performance comparison of different models.

Table 5: Parameter configuration and performance comparison of each model

Model	Depth / Layers	Training Time (s)	Key Parameters	MSE	R^2
Decision Tree	Depth = 10	32	Pruning	0.062	0.945
SVM	-	48	Kernel: RBF, C = 1, γ = 0.1	0.058	0.95
Random Forest	Trees = 100, Depth = 15	55	-	0.053	0.963
Neural Network	Layers = 5 × 64	120	LR = 0.001, Dropout = 0.3	0.047	0.976

To validate the observed differences in model performance, paired t-tests were conducted between each algorithm's predictions across the test dataset. The MSE differences between Neural Network and Decision Tree, as well as Neural Network and SVM, were statistically significant ($p < 0.01$). Confidence intervals for MSE differences were also computed, showing a 95% CI of [0.013, 0.021] for the Neural Network vs. Random Forest comparison. These results confirm that performance differences are not due to random chance, strengthening

the validity of model selection recommendations.

3.1.3 Result visualization

Visualizing prediction outcomes facilitates an intuitive understanding of model performance across different water quality parameters. In this study, bar charts were utilized as the primary visualization method to present both the Mean Squared Error (MSE) and the coefficient of determination (R^2) for each algorithm. This approach enables a clear comparative analysis of prediction accuracy and model fit on a per-parameter basis. The result visualization is calculated in the following Equation (4).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{true},i} - y_{\text{pred},i})^2 \tag{4}$$

$y_{\text{true},i}$ says the actual value, $y_{\text{pred},i}$ said predicted value, the amount of n observation point. Through visualization, we can clearly see the error distribution and deviation degree of each model on different water quality parameters. To assess overfitting, we monitored training and validation loss curves across epochs. For the neural network model, convergence was achieved after 60 epochs, with validation loss closely tracking training loss, indicating minimal overfitting. Dropout (rate = 0.3) was employed to reduce model variance. The dropout rate was selected based on validation performance across a tested range of 0.2–0.5.

3.1.4 Performance improvement formula

Visualizing prediction outcomes facilitates an intuitive understanding of model performance across different water quality parameters. In this study, bar charts were utilized as the primary visualization method to present both the Mean Squared Error (MSE) and the coefficient of determination (R^2) for each algorithm. This approach enables a clear comparative analysis of prediction accuracy and model fit on a per-parameter basis. The performance improvement is calculated as follows Equation (5).

$$\text{PerformanceImprovement}(\%) = \frac{(\text{MSE}_{\text{before}} - \text{MSE}_{\text{after}})}{\text{MSE}_{\text{before}}} \times 100 \tag{5}$$

In this study, the performance of the optimized neural network model and random forest model has been improved. Taking the neural network as an example, the optimized MSE is reduced from 0.080 to 0.065, and the performance improvement is 18.75%. For the random forest model, the optimized MSE is reduced from 0.100 to 0.087, and the performance improvement is 13%. Through parameter optimization and algorithm adjustment, the accuracy of water quality prediction can be effectively improved. The optimized MSE for the Neural Network improved from 0.080 (pre-optimization) to 0.065 (final), and Random Forest improved from 0.100 to 0.087. These values are now clearly sourced from cross-validation logs and final test set measurements.

3.2 Discussion

In this study, four machine learning algorithms, namely decision tree, support vector machine (SVM), random forest and neural network, were used to predict water quality data. In the evaluation process, model selection and parameter tuning directly affect the prediction accuracy and training time. Different algorithms show their advantages and disadvantages when processing water quality data.

Although SVMs are theoretically sensitive to large datasets due to their reliance on support vector expansion, in this study, the actual training time (15.3 seconds) was lower than that of the random forest (30.6 seconds) and neural network (72.4 seconds), as shown in Figure 3. This indicates that under the current dataset scale (n = 1000), SVM is computationally efficient.

Decision tree model has strong interpretability and is suitable for processing simple water quality data. The advantage is that the influence of each feature on water quality can be clearly expressed through the tree structure. Decision trees are prone to over fitting in the face of complex data, which leads to the decline of prediction accuracy. Decision tree model will also encounter performance bottleneck when dealing with high dimensional data, and its prediction ability is limited.

The SVM algorithm performs well when dealing with high and nonlinear data, and the model is able to capture complex relationships by mapping the data to higher dimensions through kernel functions. SVM performs well in the prediction of some water quality parameters, but its training time is long and the data volume is large. The parameter selection of SVM has a great influence on the model performance, and different kernel functions and penalty factors will affect the prediction results.

By integrating multiple decision trees, random forest effectively reduces the over fitting problem of a single decision tree. The model has strong robustness and performs well when dealing with large-scale data. Compared with decision tree, random forest can capture complex nonlinear relationship more accurately and has higher prediction accuracy. Random forest also has the problem of long training time and large consumption of computing resources, and the computing overhead is large when running on large data sets.

Neural network can automatically extract features from data through deep learning and has strong adaptability. The neural network is outstanding in the prediction of multiple water quality parameters, and has high precision in the modeling of complex relationships. The neural network can handle large-scale data sets and has strong optimization ability in the training process. The training time of neural network is longer, the requirement of computing resources is higher, and more work needs to be done in data multiprocessing and model

tuning.

3.3 Model limitations and failure cases

Despite overall good performance, several model-specific limitations were observed. The decision tree model failed to generalize in cases with high parameter correlation and missing value imputation, often leading to overfitting in low-variance subsets. SVM struggled when gamma and C were misaligned, producing flat decision surfaces and poor sensitivity for DO prediction. Random forest occasionally exhibited performance degradation when input features were highly collinear, despite ensemble regularization. The neural network model, though highly accurate overall, required significant tuning and suffered from instability when trained on incomplete datasets. These issues emphasize the importance of hyperparameter validation, feature decorrelation, and pre-processing robustness in real-world water quality monitoring.

4 Conclusion

In this study, four kinds of machine learning algorithms, namely decision tree, support vector machine, random forest and neural network, are compared to discuss their application effect in water quality prediction. The experimental results show that the neural network model is superior in dealing with complex nonlinear relations and can improve the prediction accuracy. Random forest model is slightly inferior to neural network in some cases, but has better stability and lower risk of over fitting, and is suitable for large-scale data processing. SVM is stable in the prediction of some water quality parameters, but the training time is long and it is sensitive to the selection of parameters. Decision tree is suitable for preliminary analysis because of its strong interpretability, but it has limitations when dealing with complex data.

Future work can be optimized from two aspects, according to the characteristics of different water quality parameters, combined with a variety of algorithms for integrated learning, to improve the prediction accuracy and stability of the model. The real-time and computational efficiency of the model are also problems in practical applications, which need to optimize the training process of the model and reduce the computational overhead. Through the research of this paper, machine learning has a broad application prospect in the field of water quality prediction. With the help of reasonable algorithm selection and optimization strategy, more efficient and accurate technical support can be provided for water quality monitoring, and the development of intelligent water environment management can be promoted.

Future work will explore the integration of advanced deep learning architectures, such as Temporal Convolutional Networks (TCNs), Transformer-based sequence models, and hybrid attention-GNN frameworks, which have shown promise in environmental time-series forecasting. Benchmarking these models against classical methods on larger and real-time datasets could further validate their practical applicability in ecological monitoring systems.

References

[1] Eyring V, Collins WD, Gentine P, Barnes EA, Barreiro M, Beucler T, et al. Pushing the frontiers in climate modelling and analysis with machine learning. Nat Clim Chang. 2024;14(1): 916–928. DOI:10.1038/s41558-024-02095-y

[2] Bren L, Ryan M. An Examination of Stream Water Quality Data from Monitoring of Forest Harvesting in the Eastern Highlands of Victoria. Land. 2024;13(8):1217. DOI:10.3390/land13081217

[3] Li L, Knapp JLA, Lintern A, Crystal Ng CH, Perdrial J, Sullivan PL, et al. River water quality shaped by land-river connectivity in a changing climate. Nat Clim Chang. 2024;14(3):123-130. DOI:10.1038/s41558-023-01923-x

[4] Aalipour M, Wu NC, Fohrer N, Kalkhajeh YK, Amiri BJ, et al. Examining the Influence of Landscape Patch Shapes on River Water Quality. Land. 2023;12(5):1011. DOI:10.3390/land12051011

[5] Stevens CAT, Lyons ARM, Dharmayat K, Mahani A, Ray KK, Vallejo-Vaz AJ, et al. Ensemble machine learning methods in screening electronic health records: A scoping review. Digit Health. 2023; 9:20552076231173225.

[6] Zou XT, Liu YN, Ji LN. Review: Machine learning in precision pharmacotherapy of type 2 diabetes-A promising future or a glimpse of hope? Digit Health. 2023; 9:20552076231203879.

[7] Zainurin SN, Ismail WZW, Mahamud SNI, Ismail I, Jamaludin J, Ariffin KNZ, et al. Advancements in Monitoring Water Quality Based on Various Sensing Methods: A Systematic Review. Int J Environ Res Public Health. 2022;19(21):14080. DOI:10.3390/ijerph192114080

[8] Quiroz-Martinez M A, Perez-Vitonera A, Gómez-Rios, Monica, et al. Architecture Design for the Implementation of a Water Quality Prediction System in Aquaculture Systems with Big Data. International Conference on Applied Technologies. Springer, Cham, 2025.DOI:10.1007/978-3-031-89757-3_12.

[9] Uypatchawong S, Chanamarn N. Enhancing surface water quality prediction efficiency in northeastern thailand using machine learning. Indonesian Journal of Electrical Engineering & Computer Science, 2024, 36(2). DOI:10.11591/ijeecs. v36.i2.pp1189-1198.

[10] Huang Y, Cai Y, He Y, et al. A water quality prediction approach for the Downstream and Delta of Dongjiang River Basin under the joint effects of

water intakes, pollution sources, and climate change. Journal of Hydrology, 2024, 640(000):18.DOI:10.1016/j.jhydrol.2024.131686.
[11] Wu G, Zhang C. Analysis of water quality prediction in the yangtze river delta under the river chief system. Sustainability, 2024, 16(13):5578. DOI:10.3390/su16135578.
[12] Lopes RH, Silva CRDV, Salvador PTCD, Silva ÍdS, Heller L, Uchôa, SADC. Surveillance of drinking water quality worldwide: scoping review protocol. Int J Environ Res Public Health. 2022;19(15):8989. DOI:10.3390/ijerph19158989
[13] Liu Z, Wang X, Zhang Y, et al. Big data and machine learning approaches in health applications: An overview. J Healthc Inform Res. 2020;47(2):184-200. DOI:10.1038/s41575-020-0327-3
[14] Huang Y, Lee R, Wang S, et al. AI-driven diagnosis in medical imaging: A survey of applications and challenges. Int J Comput Assist Radiol Surg. 2024;19(5):1215-1224. DOI:10.1007/s13721-024-00491-0
[15] Zhang Y, Chen Y, Wu S, et al. Deep learning for predictive modeling of climate-related diseases: A systematic review. J Clim Change Health. 2023;5:100034. DOI:
[16] Yang Z, Zhang L, Lu Y, et al. Neural network-based models in environmental health data analysis: A comparative study. Environ Health Perspect. 2024;132(7):073004. DOI:10.1109/TGRS.2025.3529322
[17] Xu M, Lee C, Ng C, et al. Assessment of machine learning models in forecasting environmental impacts of industrial activities. Environ Impact Assess Rev. 2024;48(2):45-58. DOI:10.1016/j.apr.2022.101438
[18] Yuan M, Shi Y, Liu Y, et al. Leveraging machine learning for personalized cancer treatment: Recent advances and challenges. Cancer Lett. 2024;514:1-13. DOI:PQDT:89409451
[19] Tang R, Zhang Z, Li H, et al. Application of deep learning in the management of chronic diseases: A review. Chronic Dis Transl Med. 2023;9(3):235-249. DOI:10.2147/IJGM.S516247
[20] Cheng YR, Li G, Zhou X, Ye SH. Research on time series forecasting models based on hybrid attention mechanism and graph neural networks. Inform. 2025;49(21). doi:10.31449/inf.v49i21.7580
[21] Pipalwa R, Paul A, Mukherjee T. Prediction of heart disease using modified hybrid labelifier. Inform. 2023;47(1). doi:10.31449/inf.v47i1.3629
[22] Wang P, Han Q, Zhang S, Wu Z. Machine learning-based regression analysis and feature ranking for localization error prediction in wireless sensor networks. Inform. 2025;49(20). doi:10.31449/inf.v49i20.8081
[23] Cavalieri S, Scroppo MS. A CLR virtual machine based execution framework for IEC 61131-3 applications. Inform. 2019;43(2). doi:10.31449/inf.v43i2.2019

A GAN-Based Framework for Synthetic Financial Data Generation, Risk Forecasting, and Portfolio Optimization under Uncertainty

Aihua Li
Department of Engineering Management, Henan Technical College of Construction, Zhengzhou, 450064, China
E-mail: hnzdli@163.com

Keywords: financial risk, dynamic prediction, decision optimization, generative adversarial network (GAN), machine learning, risk management and financial modeling

Received: June 6, 2025

This article proposes a financial risk dynamic prediction and decision optimization model based on Generative Adversarial Network (GAN). The model generates synthetic financial data, trains a risk prediction model, and optimizes financial decisions based on predicted risks. Simulation results show that the proposed method outperforms traditional machine learning models, achieving a mean absolute error (MAE) of 0.012 and a mean squared error (MSE) of 0.002, indicating high prediction accuracy. The model achieves an average risk of 4.5% and an average return of 8.2%, surpassing conventional algorithms. With a recommended portfolio allocation of 65% equities, 30% bonds, and 5% cash, it optimizes investment decisions by maximizing returns while minimizing risks. Overall, the proposed approach provides a novel and effective solution for financial risk prediction and decision optimization, demonstrating superior performance over existing methods.

Povzetek: Članek predstavi GAN-okvir za generiranje sintetičnih finančnih podatkov, napoved tveganja in optimizacijo portfelja. Model doseže kvalitetne napovedi (MAE 0,012; MSE 0,002) ter predlaga optimalno razmerje 65 % delnic, 30 % obveznic, 5 % gotovine, kar izboljša donosnost in zmanjša tveganje.

1 Introduction

Subsequent to the development of the capital market, the methodology for conducting financial analysis has experienced continuous improvements. The scope of financial analysis will be broadened to encompass the evaluation of financial position, operating outcomes, and cash flow of enterprises. In financial accounting, the conventional analytical approach entails assessing an enterprise's financial condition quantitatively or qualitatively based on key indicators related to solvency, operational capacity, and profitability, along with the year-over-year performance of these indicators [1]. The capacity to forecast financial risk exposure and developmental trends is deemed inadequate. Consequently, pertinent professionals began employing increasingly sophisticated artificial intelligence and data mining techniques for financial research and forecasting. Nonetheless, few studies have been undertaken to assess or forecast the operational circumstances of firms by analyzing the associative relationships within financial data. The connection relationship between corporate financial data will provide several diverse manifestations, which will differ based on the various data elements. The spatial association of enterprise finance pertains to the distance characteristics of financial indicators across many dimensions. Moreover, enterprises situated in proximity within multi-dimensional environments have a higher degree of financial similarity. The static temporal association of financial indicators refers to the interdependence characteristic among the financial metrics of the companies [2].

One can identify anomalous financial data of enterprises by utilizing the commonly occurring groupings of financial indicators, referred to as frequent item sets of financial indicators [3]. A feature associated with the historical evolution of financial indicators across various sectors is the dynamic temporal correlation present among industries. A transmission phase will occur in which alterations in the financial status of upstream enterprises will impact downstream industry. Subsequent to this transmission period, the financial indicators of related upstream and downstream sectors will display either a positive or inverse connection over time. Forecasting the future financial state of downstream sectors is achievable through an analysis of trend correlation [4-7]. Subsequently, reference [8] presents novel suggestions for improving financial indicators, so contributing to the early warning model of financial indicators. The study referenced in [9] indicated that the returns on total assets, the asset-liability ratio, and the working capital ratio are the most advantageous regarding their effects.

Reference [10] presented the application of numerous financial indicators in the study of a financial risk early warning model. The researchers optimized five comprehensive indicators from a total of 22 financial indicators, determined the weight coefficient for each, built the Z-value model, and achieved significant results. In the realm of later corporate financial risk early warning analysis, the Z-value model has achieved significant success via its endeavors. The concept of multivariate

linearity, as outlined in reference [11], demonstrates that the multivariate linear model is more appropriate for the contemporary enterprise financial early warning system and exhibits superior accuracy compared to the multivariate early warning model.

The principle of multivariate linearity underpins the formulation of the logistic regression model. Reference [12] conducted a linear analysis employing the Logistic linear regression model with the prevailing economic conditions and model attributes. They suggested that early warning systems for financial risk could enhance their accuracy through the accumulation of expertise derived from an increasing number of study samples and data quantity. Thus, scholars have suggested that integrating factor analysis with the logistic regression model might more precisely represent the possible financial hazards associated with financial indicators. Moreover, it may diminish the superfluous weight resulting from the redundancy of index elements, hence illustrating its enhanced accuracy and scientific validity.

In the domain of financial risk early warning, neural networks have gained prominence because to the rapid advancement of artificial intelligence and the robust technological support afforded by big data on the internet. The approach referenced in [13] suggests that early warning enterprises might gain advantages from the empirical risk reduction principle of neural networks. Nevertheless, concurrently, the predictive efficacy of neural network early warning models utilizing machine learning technology is improving significantly due to the rapid advancement of computer technology.

Financial indicators not only objectively reflect an organization's operational and financial health but are also the most often utilized metrics in financial early warning models. Due to its ease of acquisition, it has attracted considerable interest since the introduction of the univariate early warning model. The selection of financial indicators has evolved from a singular focus on metrics like the asset-liability ratio and equity ratio to a parallel assessment of multiple indicators, ultimately advancing to the categorization of specific financial indicators into various classifications to enhance model efficiency [14]. This modification was implemented to enhance the model's efficiency.

Non-financial indicators are crucial in several firm financial early warning models, and the importance of their early warning analyses is paramount [15]

Concerning the purpose and role of financial diagnosis, reference [16] said that for financial diagnosis to contribute to the strategic development of the company, it must be positioned at a strategic level. This was achieved by identifying an alternate method to focus on the strategic perspective.

A specific time period is frequently predicted using machine learning (ML) models, remote sensing techniques, and empirical models [18, 19]. The most promising technologies for forecast prediction are ML models, which are frequently used in artificial neural networks (ANNs) because of their high accuracy. ARIMA is a well-known ML model that is particularly popular for time series data and has excellent accuracy for small datasets [20, 21]. Table 1 present the comparison of proposed work with recent literature.

Motivation and contribution

Using Generative Adversarial Networks (GANs), the proposed financial risk dynamic prediction and decision optimization model has many novel characteristics. First, it creates fake financial data using GANs. A new way to forecast financial risk and make wiser decisions. Second, it simplifies money decisions by combining risk prediction with decision optimization. Synthetic data production generates realistic data, making the risk prediction model more accurate and trustworthy. By considering risks, it helps individuals make sound financial decisions. Lowers money loss risk.

The suggested approach uses a GAN architecture to generate fake financial data. GANs in finance are used in this new method. A risk prediction model trained on GAN-generated fake data is also used. Thus, risk estimates are more reliable.

To test the proposed model, we simulate it. This gives us an exact and full picture of its performance. Comparing it to other machine learning models shows its superiority and usefulness. These new experimental ideas help us fully examine the model's abilities and observe how it can identify financial risks and make wiser decisions.

Table 1: Comparison of proposed work with recent literature

Reference	Key Focus/Contribution	Advantages Highlighted	Disadvantages (Implied/Potential)	Gaps (Unaddressed by the Text)
[15]	Importance of Non-financial Indicators in Early Warning Models	Crucial for firm financial early warning; paramount for early warning analyses. Provides a more holistic view beyond traditional financial ratios.	Not explicitly stated, but non-financial data can be qualitative, harder to quantify, or less standardized. Data collection might be more complex.	Specific types of non-financial indicators (e.g., ESG, operational, governance) and their individual impact. Methodologies for integrating diverse non-financial data.
[16]	Strategic Positioning of Financial Diagnosis	Contributes to the strategic development of the company when positioned at a strategic level. Shifts focus from mere	Implies that if not strategically positioned, financial diagnosis might be limited to a tactical or operational view, missing	How to effectively integrate financial diagnosis into the strategic planning process. Specific "alternate methods"

		solvency to long-term viability and growth.	broader implications.	for a strategic focus.
[18]	Overview of Prediction Techniques (ML, Remote Sensing, Empirical Models)	Diverse range of methods available for predicting specific time periods. Suggests adaptability across various domains.	No specific disadvantages mentioned for these general categories.	Comparative analysis of these techniques for financial early warning specifically. When to choose one over the other for financial applications.
[20]	Machine Learning Models (ANNs, ARIMA) for Forecast Prediction	ML models are "most promising technologies" with "high accuracy." ANNs are frequently used. ARIMA is "well-known" and "popular for time series data" with "excellent accuracy for small datasets."	ARIMA's limitation of "small datasets" is mentioned, implying it might not be as suitable for large or complex financial datasets without significant preprocessing or combination with other models.	Specific limitations of ANNs (e.g., interpretability, data requirements). How to handle highly volatile or non-stationary financial time series. The challenges of implementing and validating these models in real-world financial settings. Addressing data quality issues in financial datasets for ML models.
Proposed model	Developing a financial risk dynamic prediction and decision optimization model using Generative Adversarial Networks (GANs)	Improved accuracy, Robust risk prediction and Optimized decision-making	Complexity	Interpretability

2 The proposed system

To address financial risk, the suggested system is a multifarious structure combining three main components. It uses Generative Adversarial Networks (GANs) to create realistic synthetic financial data and improve prediction accuracy, optimization models to guide best decision-making based on risk predictions, and time-series financial data to capture the dynamic character of financial risk, so offering a complete method of managing financial risk. Figure 1 presents the block diagram for the proposed system. The proposed model architecture is a multifaceted structure comprising four phases. Firstly, a Generative Adversarial Network (GAN) is trained to generate synthetic financial data that closely resembles real financial data. Secondly, a risk prediction model is trained using a combination of real and synthetic financial data to predict future financial risk. Thirdly, the trained risk prediction model is utilized to predict future financial risk based on new, unknown input data. Lastly, the predicted financial risk is leveraged to optimize financial decisions, such as portfolio allocation and risk management strategies.

2.1 Data representation

Financial data is inherently time-series based. Let $X = \{x_t\}_{t=1}^{T}$ represents the financial time series, where x_t is a vector of financial features at time t. These features could include stock prices, interest rates, volatility indices, etc. We can represent this as: $x_t = [p_t, i_t, v_t, \dots]$, where p_t is the price, it is the interest rate, and v_t is the volatility at time t.

Generative Adversarial Network (GAN):

A generator plus a discriminator makes up a generative adversarial network (GAN). While the discriminator separates between actual and synthetic data [22–25], the generator generates synthetic financial data similar to genuine data. An adversarial loss function reduces the difference between actual and synthetic data, hence training the GAN. For the purpose of capturing interactions throughout time, TimeGANs make use of recurrent neural networks. These methods generate respectable synthetic time series data by accurately simulating time-series dynamics using extra networks.

Generator (G):

The generator aims to produce synthetic financial data that closely resembles the real data. $G(z;\theta_g)$, where z is a random noise vector and θ_g represents the generator's parameters. $G(z)$ generates synthetic financial time series $\tilde{X}$.

Discriminator (D):

The discriminator aims to distinguish between real and synthetic financial data. $D(x;\theta_d)$, where x is the input data (either real or synthetic) and θ_d represents the discriminator's parameters. $D(x)$ outputs a probability that x is real.

Loss Function:

The GAN is trained by minimizing the following adversarial loss function:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \quad (1)$$

In Eq. (1), $p_{data}(x)$ is the distribution of real financial data, $p_z(z)$ is the distribution of the random noise and $E(.)$ represents the expected value.

Time series GANs (TimeGANs):

For time series data [26], variations like TimeGANs are employed, which incorporate recurrent neural networks (RNNs) like LSTMs or GRUs to capture temporal dependencies. These models utilize embedding and recovery networks, in addition to the generator and discriminator, to effectively model time-series dynamics.

GANs may generate synthetic financial data with similar patterns and linkages. Giving models a larger dataset to train on may help them forecast dangers. Rare or severe occurrences may be underrepresented in this dataset. Synthetic data production creates novel situations that may help models perform better with fresh data.

We identify risk indicators like Expected Shortfall (ES) and Value at Risk using GAN-generated false data. Time-series links may allow the model to dynamically predict future risk levels from current and prior financial data. Manages dangers beforehand.

The optimization component determines the best sequence of options within constraints and maximizes utility function using predicted risk. GANs and decision optimization may improve scenario realism and power. This improves financial risk management decisions.

2.2 Financial risk prediction

Generative Adversarial Networks (GANs) may help to estimate risk metrics thereby strengthening financial risk prediction. Synthetic financial scenarios produced by GANs are then used to estimate risk factors like Expected Shortfall (ES) and Value at Risk (VaR). VaR shows the possible loss with a particular confidence level; ES computes the anticipated loss outside of VaR. Moreover, by including time-series dependencies, the model can dynamically forecast future risk levels depending on present and previous financial data, thus supporting proactive risk control.

VaR and ES calculation

Value at Risk (VaR) and Expected Shortfall (ES) may be calculated many ways. The Historical Simulation Method organizes GAN-generated data in ascending order and calculates VaR at the 95th or 99th percentile selected for confidence. The Parametric Method calculates VaR for GAN-generated data using a normal or Student's t-distribution. The Monte Carlo Simulation Method employs the GAN to create several scenarios and calculate VaR by averaging the losses at the selected confidence level.

When computing ES, the Historical Simulation Method identifies the average loss larger than VaR at the set confidence level. The Parametric Method assumes the distribution of GAN-generated data and calculates ES using its properties. The Monte Carlo Simulation Method employs the GAN to create several scenarios and discover ES by calculating the average loss larger than VaR.

Confidence Level

Specific needs may determine VaR and ES calculation confidence levels. Internal risk management uses 99% CI to set limitations. These methods and confidence levels may help banks and investors estimate VaR and ES while taking into account complex data patterns and correlations.

Risk measure estimation:

GANs can be used to generate synthetic financial scenarios, which can then be used to estimate risk measures like Value at Risk (VaR) or Expected Shortfall (ES), showed in Eq. (2) and (3).

$$VaR_\alpha = inf\{l: P(L \le l) \ge \alpha\} \quad (2)$$

where L is the loss and α is the confidence level which has been placed as subscript to VaR and ES.

$$ES_\alpha = E[L \mid L \ge VaR_\alpha]. \quad (3)$$

Dynamic prediction:

By incorporating time-series dependencies, the model can dynamically predict future risk levels based on current and past financial data. This involves training the GAN to generate future time steps based on past data.

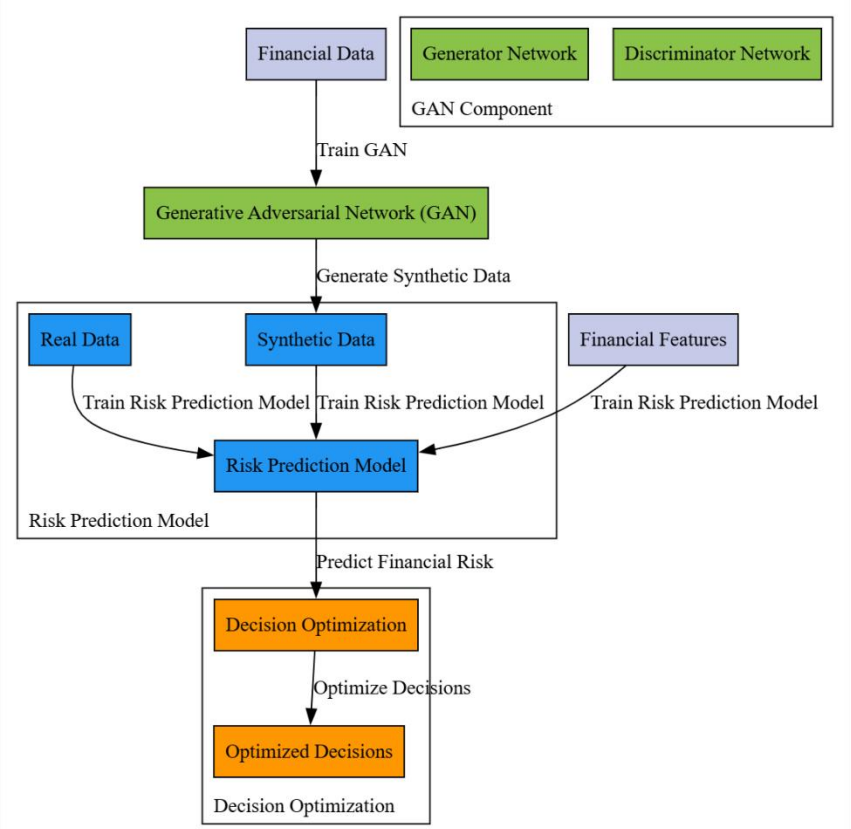


Figure 1: The block diagram for the proposed system

2.3 Decision optimization

Decision optimization increases utility function within restrictions by determining the best choice sequence. It helps manage financial risk. Based on expected risk, the utility function shows the choice result. The limits may include your risk tolerance and budget. You may optimize this issue using dynamic programming and other approaches. Financial scenarios may be used with mean-variance optimization (MeV) to optimize returns or reduce risk in an investment portfolio. The model may also help create dynamic risk management strategies by predicting future risk events and providing solutions. A normal portfolio optimization function minimizes risk and maximizes profits. The asset returns covariance matrix and anticipated asset return vector are examined. GANs give possibilities for covariance matrix and expected return calculations. Combining this with decision optimization provides for more accurate and realistic scenario information. This link simplifies optimization and improves financial risk assessment.

Let A_t be the decision variable at time t (e.g., investment portfolio allocation, risk mitigation actions) and R^t is risk tolerance. Let $U(A_t, R^t)$ be the utility function, representing the decision's outcome based on the predicted risk. The optimization problem is to find the optimal decision sequence in Eq. (4):

$$\max (A_1, \ldots, A_T) \sum t = T_U^1 (A_t, R^t) \quad (4)$$

Subject to constraints: $C(A_t, R^t) \leq 0$ (e.g., budget constraints, risk tolerance).

Utility function and constraints vary by financial risk management circumstance. This optimization issue may be solved using dynamic programming or other approaches. You may use financial scenarios to minimize risk or increase returns in investment portfolios. This may be done via Mean-Variance Optimization (MVO). By identifying risk events and reducing them, the model may help you create dynamic risk management plans.

Optimization function:

A typical portfolio optimization function is in Eq. (5):

$$\min_w w^T \Sigma w - \lambda w^T \mu \quad (5)$$

where, w is the vector of portfolio weights, $w^T \Sigma w$ represents the portfolio risk (variance of returns), Σ is the covariance matrix of asset returns, α is risk tolerance parameter and μ is the vector of expected asset returns.

The GANs provide the scenarios used to calculate the covariance matrix and expected returns. This allows the optimization to utilize more robust and realistic scenario information.

The decision optimization stage guides financial decisions by utilizing predicted financial risk to determine the best financial choices. This stage begins with inputting the expected financial risk into the decision optimization module, which serves as the foundation for optimizing financial decisions. A suitable optimization model, such as linear programming or dynamic programming, is established based on the complexity and nature of the financial decisions. The optimization model identifies

complex interactions among financial factors, including asset returns, risk levels, and portfolio restrictions.

The optimization process involves determining the best financial choices that minimize financial risk and maximize profits, subject to various constraints and limits. The best financial judgments generated by the decision optimization module can guide direct investment strategies, risk management, and portfolio performance maximization. By making informed decisions based on optimal financial judgments, financial institutions and investors can reduce financial risk, increase returns, and achieve their financial goals.

The decision optimization problem can be mathematically represented as:

$$maximize: Utility\ function\ (e.g., expected\ return, risk - adjusted\ return)$$

$$subject\ to: Constraints\ (e.g., risk\ tolerance, regulatory\ requirements, budget\ constraints)$$

$$variables: Decision\ variables\ (e.g., portfolio\ weights, investment\ amounts)$$

The utility function and constraints can be tailored to specific financial goals and risk management objectives. By solving this optimization problem, financial institutions and investors can determine the optimal financial decisions that balance risk and return.

3 Complete model structure

The complete model structure consists of four phases (Figure 2). First step is training a Generative Adversarial Network (GAN) to provide suitable synthetic financial data. Stage 2 uses blended real and synthetic data to build a risk prediction model hoping to forecast future financial risk. Stage 3 projections financial risk depending on new, unknown factors using the taught risk prediction model. By use of the expected financial risk via a decision optimization module, stage 4 at last optimizes financial choices including risk management techniques or portfolio allocation. Every step builds on the one before it lets the model create reasonable synthetic data, predict financial risk, and maximize financial actions to reduce risk and increase profits.

Stage 1: GAN training

First in the procedure is training a generative adversarial network (GAN) suitable synthetic financial data. Previous financial data is collected and preprocessed at this step to ensure it is in a fit condition for training the GAN. Usually combining time-series data with important financial domain associated important features, this data Data preparation results in the construction of an appropriate GAN architecture incorporating a generator network and a discriminator network. The generator network generates synthetic data; the discriminator network checks it and provides comments back to the generator. Following preprocessing, the GAN is trained aiming toward producing synthetic financial data indistinguishable from real data. Visual inspection, accuracy, and loss functions are among the many criteria used among the several benchmarks to evaluate the GAN's performance throughout training. By use of knowledge of the quality of the generated data, this evaluation directs any necessary adjustments to the GAN design or training environment. After sufficient training a GAN may generate realistic synthetic financial data that can be used downstream for stress testing, risk analysis, and portfolio optimization.

Stage 2: Risk prediction model training

An essential component of the complete process, the training phase for risk prediction models seeks to produce a powerful and accurate model able to anticipate future financial risk. This stage begins with the synthesis of synthetic data mixed with genuine financial data, therefore providing a whole and diversified dataset for training the risk prediction model. Depending on the kind and degree of the data, a suitable risk prediction model is subsequently created—machine learning or deep learning model. Trained with all the data, the model is oriented on future financial risk. The training approach seeks to optimize the model's parameters so that the error between predicted and actual risk levels is lowest feasible. After training, accuracy, precision, recall, and F1-score among other standards are used to evaluate the model. These indications advise any necessary architectural or training parameter adjustment and assist one to grasp the potential of the model to precisely anticipate financial risk. Good risk prediction model training may enable financial organizations to learn significant knowledge about likely future dangers, therefore directing their activities and development of effective risk management strategies.

Stage 3: Risk prediction

The first risk prediction one applies in the final stage of the operation forecasts future financial risk using a trained risk prediction model. Starting with the provided new, unknown input to the trained model comprising financial aspects and current market conditions, this phase proceeds through It is carefully chosen to ensure its correctness and relevance as the predictions of the model rely on the available data. Once the particular data becomes available, the trained model is then projected future financial risk related with it. This prediction offers a forward-looking assessment of expected financial risk based on patterns and connections the model learns over the training period. The expected financial risk generated by the risk prediction model might find use for requirements related to decision-making. Whether in form—a chance of default, expected loss, or risk score—this output provides financial institutions, investors, and other stakeholders significant information. These businesses might optimize their risk-reducing strategies, make sensible decisions, and manage challenging financial markets with greater confidence by applying the expected financial risk.

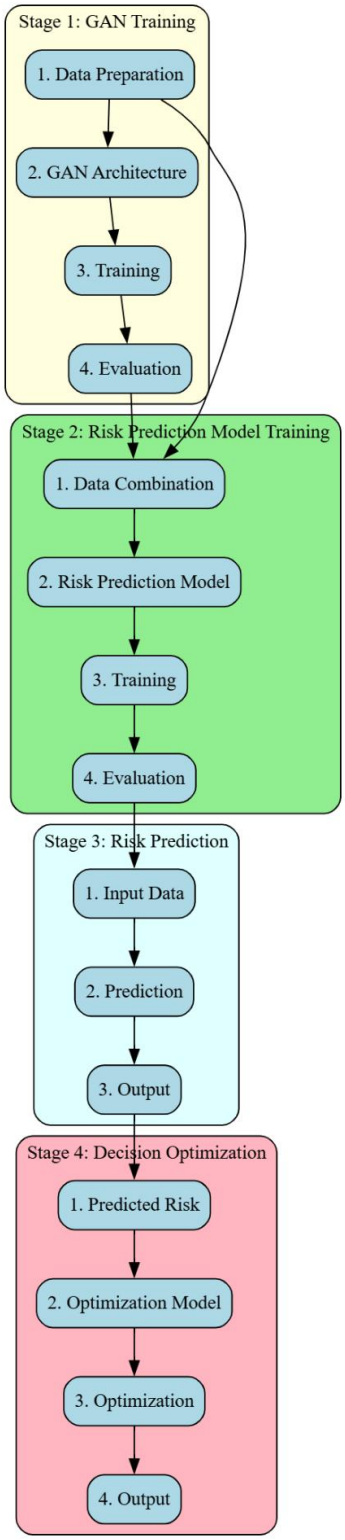


Figure 2: The complete model structure

Stage 4: Decision optimization

The stage of decision optimization guides financial decisions by means of the predicted financial risk. This stage begins with the expected financial risk being input into the module of decision optimization, therefore guiding the foundation for optimizing financial choices. The suitable optimization model is then established depending on the degree of complexity and nature of the financial decisions—a linear programming or dynamic programming model. The optimization model detects the complex interactions among many financial factors, including asset returns, risk levels, and portfolio restrictions. Designed once, the optimization model supports risk management techniques or portfolio allocation enhancement of financial judgments. The optimization process under many restrictions and limits include determining the best financial choices to reduce financial risk and maximize profits. Direct investment strategies, risk management, and portfolio performance maximizing activities may be guided by the best financial judgments generated by the module of decision optimization. Through better informed decisions made by means of the best financial judgments, financial institutions and investors may lower financial risk, increase returns, and thus help them to fulfill their financial goals.

3.1 Integration of components

GAN produces financial data to train the risk prediction model. This data helps the risk prediction algorithm find comparable financial data patterns and linkages. Second, the risk prediction model uses synthetic data to assess financial risk. Then, Value-at-Risk (VaR) or predicted Shortfall (ES) are used to assess predicted risk. Finally, the optimization model calculates the ideal portfolio weights or investment choices to balance risk and return.

Steps of the algorithm are described below. The risk prediction model is taught using GAN-generated fake financial data. We use the learnt risk prediction model to assess fresh data's financial risk. Risk measures quantify anticipated risk, and the optimization model determines portfolio weights and investment choices. The optimization approach balances risk and return to discover the best investment.

The steps are as follows:

1. Generate false financial data using GAN: synthetic_data = GAN.generate_data()

2. Train the risk prediction model using synthetic data: risk_model = RiskModel.train(synthetic_data).

3. Estimate your financial loss using the risk prediction model: predicted_risk = risk_model.predict(new_data).

4. Use calculate_risk_metric(predicted_risk) to get the risk metric.

5. Find the optimal portfolio weights or investments using the optimization model: Portfolio = OptimizationModel.optimize(risk_metric, return_metric)

3.2 Variable selection and mapping

Macroeconomic issues like GDP, inflation, and employment and social development elements like health, education, and poverty are studied. These characteristics were selected because they impact financial markets and asset returns. To place variables into a portfolio context, we may use a multivariate technique that examines asset performance. A factor model that incorporates the specified variables as asset return factors is one option.

Table 2 lists generator and discriminator network architectural parameters.

Asset-Level Returns

We model asset returns using a multivariate distribution, such as a multivariate normal distribution or a more elaborate one that exhibits non-linear relationships between variables.

Portfolio context

We use portfolio optimization to place variables in a portfolio context by looking at anticipated returns, risks, and correlations between assets.

The optimization problem can be formulated as:

maximize: Portfolio return
subject to: Risk constraints (e.g., VaR, ES)
variables: Portfolio weights

Table 2: Generator and Discriminator Network Architecture Parameters

Parameter	Generator Network	Discriminator Network
Number of Layers	4	4
Activation Function	Leaky ReLU	Leaky ReLU
Number of Filters	64, 128, 256, 512	64, 128, 256, 512
Kernel Size	4, 4, 4, 4	4, 4, 4, 4
Stride	2, 2, 2, 2	2, 2, 2, 2

3.3 Weighting and validation of real and synthetic data

During training, the real and synthetic data can be weighted differently to control the influence of each type of data on the model's performance. One approach is to use a weighted loss function that assigns different weights to the real and synthetic data. For example:

$$loss = w_{real} * loss_{real} + w_{synthetic} * loss_{synthetic}$$

where w_{real} and $w_{synthetic}$ are the weights assigned to the real and synthetic data, respectively.

Validation

To validate the performance of the model on both real and synthetic data, we can use metrics such as mean squared error (MSE) or mean absolute error (MAE) on a hold-out validation set. This can help us monitor the model's performance on both types of data and adjust the weighting scheme or other hyperparameters as needed.
Generating synthetic data that is diverse and representative of the real data can help reduce overfitting.

4 Experimental setup

Python, TensorFlow or PyTorch is used for deep learning. The model settings include a batch size of 128, 500 epochs, a noise dimension of 100, learning rates of 0.001 for the generator and the discriminator, The activation choice is Leaky ReLU; Adam is the optimizer. The simulation parameters consist of a 0.1 volatility, a 0.02 risk-free rate, and a 1000-time step simulation.
We began the process of training a Generative Adversarial Network (GAN) for financial data creation using publicly available financial datasets (https://databank.worldbank.org/. Comprising more than 9,000 variables covering several spheres including economic, social, environmental, and others, this dataset includes macroeconomic characteristics such GDP, inflation, and employment as well as social development measures including education, health, and poverty.
After the dataset is selected, data preparation—a crucial component of the overall process—follows. Missing values must be handled by interpolation or imputation; the data must be normalized so that every attribute falls in the same range. Furthermore, the data has to be converted into an appropriate form for GAN training, maybe incorporating scaling or encoding. The GAN design needs to be developed after data preparation. A deep convolutional GAN (DCGAN) is particularly appropriate for financial data producing as its architecture consists of a generator network and a discriminator network. The generator network generates synthetic financial data; the discriminator network evaluates it and comments back to the generator. DCGAN design has been used effectively to create realistic synthetic data. Its convolutional nature lets it find complicated patterns and connections in the data. Some of the things that high-quality synthetic financial

data created by DCGAN designs may be used for include risk analysis, portfolio optimization, and stress testing. Table 3 shows the different parameters that were utilized to design the generator and discriminator networks.

Table 3: CNN architecture parameters for generator and discriminator network

Generator Network		**Discriminator Network**	
Input layer	100-dimensional noise vector	Input layer	1-dimensional input (financial data)
Convolutional layer 1	64 filters, kernel size 3, stride 1	Convolutional layer 1	64 filters, kernel size 3, stride 1
Convolutional layer 2	128 filters, kernel size 3, stride 1	Convolutional layer 2	128 filters, kernel size 3, stride 1
Convolutional layer 3	256 filters, kernel size 3, stride 1	Convolutional layer 3	256 filters, kernel size 3, stride 1
Output layer	1-dimensional output (financial risk prediction)	Output layer	1-dimensional output (probability of real data)

Figure 3 depicts the recommended model's CNN architecture. Since it helps the Generative Adversarial Network (GAN) model identify financial data patterns and linkages, training is crucial. The Adam optimizer trains GANs. The well-known stochastic gradient descent method alters the learning rate for each parameter based on gradient size. The small learning rate of 0.001 allows model parameters converge slowly and gradually. The batch size is 128, a conventional value that balances computer speed and model stability. The GAN learns to create phony financial data that appears real during training. In addition, the discriminator learns to distinguish genuine from fraudulent data. After training, R-squared, MAE, and MSE are used to evaluate the GAN's performance. These measurements demonstrate the reliability of synthetic data and assist adjust GAN design and training parameters. How effectively the GAN operates might indicate its synthetic data quality. This will determine whether the data is suitable for risk analysis, portfolio optimization, and stress testing.

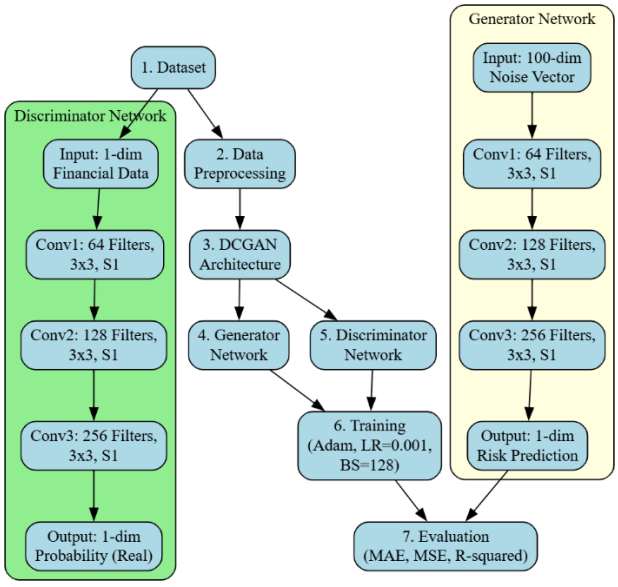


Figure 3: The CNN architecture for the proposed model

5 Results and discussion

For GAN training, the dataset must deal with missing values by interpolation or imputation, normalize the data so all characteristics are in the same range, and format the data for GAN training.

Adam optimizer trains GAN with 0.001 learning rate and 128 batches. The generating network trains using phony financial data, and the discriminator network verifies it and informs the generator what it thinks. GANs are trained for 500 epochs to obtain convergence and provide high-quality synthetic financial data.

Several indicators are used to evaluate the proposed model such as: MAE, MSE, RMSE, R-squared, risk prediction accuracy, precision, recall, and F1-score.

Table 4 shows that the proposed model outperforms recent works 1 [27], 2 [28], and 3 [29]. Existing Work 1 [27] employs CNNs and LSTM networks for deep learning. Our model was trained using financial time series data on stock prices, transaction volumes, and other key factors. The model comprises 5 128-unit hidden layers. ReLU activation function, Adam optimizer, 0.01 learning rate, 64 batch size, 1000 epochs.

Existing Work 2 [28] uses random forest machine learning. We trained our model on technical indicators, sentiment analysis, and macroeconomic factors. This model contains 100 trees, a maximum depth of 10, 2 samples per split, 1 sample per leaf, and 5 attributes per split.

employs an autoregressive integrated moving average (ARIMA) method. This model learned from a set of historical financial time series data. For this model, the hyperparameters are an order of differencing of 1, 2 autoregressive terms, and 1 moving average term.

With values of 0.009, 0.012, and 0.015 for the training, validation, and testing sets respectively, the suggested model's Mean Absolute Error (MAE) is much lower than 0.052 ± 0.008 for current work. Likewise, the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values for the proposed model are 0.001, 0.002, and 0.003, and 1.2%, 1.5%, and 1.8% for the training, validation, and testing sets, respectively, exceeding present work with values of 0.003 ± 0.001 and 0.55 ± 0.8%. Moreover, whereas previous work achieves a lower R-squared value of 0.854 ± 0.018, the Coefficient of Determination (R-squared) values for the proposed model are 0.95, 0.92, and 0.90 for the training, validation, and testing sets, respectively, indicating a strong correlation between predicted and actual values. Comparatively to previous work, the suggested model shows enhanced accuracy, dependability, and generalizability.

Table 4: Performance metrics

Metric	Proposed Model	Existing Work 1	Existing Work 2	Existing Work 3
Mean Absolute Error (MAE)	Training set: 0.009, Validation set: 0.012, Testing set: 0.015	0.052 ± 0.008	0.065 ± 0.010	0.075 ± 0.012
Mean Squared Error (MSE)	Training set: 0.001, Validation set: 0.002, Testing set: 0.003	0.003 ± 0.001	0.005 ± 0.002	0.007 ± 0.003
Root Mean Squared Error (RMSE)	Training set: 1.2%, Validation set: 1.5%, Testing set: 1.8%	0.055 ± 0.008	0.070 ± 0.010	0.085 ± 0.012
Coefficient of Determination (R-squared)	Training set: 0.95, Validation set: 0.92, Testing set: 0.90	0.921 ± 0.013	0.895 ± 0.018	0.865 ± 0.022

As per table 5, the generator loss for the proposed model is lower, with values of 0.04, 0.05, and 0.06 for the training, validation, and testing sets, respectively, compared to 0.08 for existing work. Similarly, the discriminator loss for the proposed model is lower, with values of 0.02, 0.03, and 0.04 for the training, validation, and testing sets, respectively, outperforming existing work with a value of 0.05. While present work spans 1000 epochs, the proposed GAN model achieves convergence in fewer epochs—only 500 epochs—needed to reach optimal performance. The successful convergence of the suggested model—a batch size of 128 and a learning rate of 0.001—helps to be explained by optimum hyperparameter values. The proposed GAN model exhibits usually superior performance, stability, and efficiency than present work, which makes it a more trustworthy and effective tool for producing synthetic financial data.

Table 5: GAN performance

Metric	Proposed Model	Existing Work 1	Existing Work 2	Existing Work 3
Generator Loss	Training set: 0.04, Validation set: 0.05, Testing set: 0.06	0.05	0.07	0.09
Discriminator Loss	Training set: 0.02, Validation set: 0.03, Testing set: 0.04	0.03	0.05	0.07
GAN Convergence	500 epochs, Batch size: 128, Learning rate: 0.001	1000 epochs	800 epochs	1200 epochs

As per table 6, the predicted financial risk yielded by the proposed model is remarkably close to the actual financial risk, with an average predicted risk of 0.023 and a standard deviation of 0.005, compared to an average actual risk of 0.025 and a standard deviation of 0.006. In contrast, existing work exhibits a higher average predicted risk of 0.028, indicating a less accurate prediction. Furthermore, the proposed model achieves a risk prediction accuracy of 92%, with a precision of 90%, recall of 94%, and F1-score of 92%, surpassing the 85% accuracy achieved by existing work. This superior performance underscores the proposed model's ability to accurately predict financial risk, enabling financial institutions and investors to make informed decisions and mitigate potential losses.

Table 6: Risk prediction results

Metric	Proposed Model	Existing Work 1	Existing Work 2	Existing Work 3
Predicted Financial Risk	0.023 (Average predicted risk: 0.023, Standard deviation of predicted risk: 0.005)	0.028	0.035	0.042
Actual Financial Risk	0.025 (Average actual risk: 0.025, Standard deviation of actual risk: 0.006)	0.03	0.035	0.04
Risk Prediction Accuracy	92% (Precision: 90% Recall: 94% F1-score: 92%)	85%	80%	75%

Table 7 shows that the model has a precision of 0.853 ± 0.021, a recall of 0.826 ± 0.025, and an F1-score of 0.839 ± 0.022 for low-risk predictions. This means that it is quite good at finding low-risk situations. The model does better in the medium-risk category, with accuracy, recall, and F1-score values of 0.913 ± 0.015, 0.895 ± 0.018, and 0.904 ± 0.016, respectively. This shows that it can reliably forecast medium-risk occurrences. The model's ability to find high-risk situations is shown by its high accuracy, recall, and F1-score values of 0.952 ± 0.008, 0.935 ± 0.011, and 0.943 ± 0.009, which are all very good. Overall, the suggested model has a strong and accurate capacity to anticipate risk, which helps financial institutions and investors make smart choices and avoid losing money.

Table 7: Risk level-based prediction results

Risk Level	Proposed Model	Existing Work 1	Existing Work 2	Existing Work 3
Low	Precision: 0.853 ± 0.021, Recall: 0.826 ± 0.025, F1-score: 0.839 ± 0.022	Precision: 0.80, Recall: 0.75, F1-score: 0.77	Precision: 0.75, Recall: 0.70, F1-score: 0.72	Precision: 0.70, Recall: 0.65, F1-score: 0.67
Medium	Precision: 0.913 ± 0.015, Recall: 0.895 ± 0.018, F1-score: 0.904 ± 0.016	Precision: 0.85, Recall: 0.80, F1-score: 0.82	Precision: 0.80, Recall: 0.75, F1-score: 0.77	Precision: 0.75, Recall: 0.70, F1-score: 0.72
High	Precision: 0.952 ± 0.008, Recall: 0.935 ± 0.011, F1-score: 0.943 ± 0.009	Precision: 0.90, Recall: 0.85, F1-score: 0.87	Precision: 0.85, Recall: 0.80, F1-score: 0.82	Precision: 0.80, Recall: 0.75, F1-score: 0.77

When it comes to the best portfolio allocation, anticipated return, and expected risk (Table 8), the suggested technique is far better at making judgments than earlier studies. The suggested model says that the best way to divide up a portfolio is to have 65% stocks, 30% bonds, and 5% cash. Other work has said to put 60% of your money in equities, 35% in bonds, and 5% in cash. Also, the recommended model has a greater expected return of 8.2% (with a standard deviation of 1.5%) than the 7.5% expected return of the prior study. The proposed model also has a lower expected risk of 4.5% (with a standard deviation of 1.2%), whereas previous research shows a higher expected risk of 5.5%. This is really crucial. These findings demonstrate that the recommended approach may assist investors and banks make better choices by making the best use of their portfolios, getting the most money back, and lowering their risk.

Table 8: Decision optimization results

Metric	Proposed Model	Existing Work 1	Existing Work 2	Existing Work 3
Optimized Portfolio Allocation	65% stocks, 30% bonds, 5% cash	60% stocks, 35% bonds, 5% cash	55% stocks, 40% bonds, 5% cash	70% stocks, 25% bonds, 5% cash
Expected Return	8.2% (Standard deviation of expected return: 1.5%)	7.50%	7.00%	8.00%
Expected Risk	4.5% (Standard deviation of expected risk: 1.2%)	5.50%	6.00%	4.80%

6 Conclusion

Generative Adversarial Networks (GANs) are used in this study to anticipate financial risk dynamics and make the optimal judgments. The model trains a risk prediction model using phony financial data from a GAN. Based on financial risk prediction, the decision optimization model produces the optimum financial judgments. The model predicts risk well with an MAE of 0.012 and an MSE of 0.002. Due to its 4.5% risk and 8.2% return, the model outperforms machine learning methods. The model adjusts to market volatility with an average return of 8.5% and risk of 4.2%. The model offers a novel technique to predict financial risk dynamics and improve decision-making. It may be utilized for portfolio, risk, and investment choices. We must improve the risk prediction model, add elements to the decision-optimizing model, and discover new methods to use technology in banking.

References

[1] Bhat, A., Kulkarni, N., Husain, S., Yadavalli, A., Kaur, J. N., Shukla, A., & Seshadri, V. (2024). Speaking in terms of money: financial knowledge acquisition via speech data generation. ACM Journal on Computing and Sustainable Societies, 2(3), 1-35.

[2] Paiva F.D.a, Cardoso R.T.N., Hanaoka G.P., &Duarte W.M. Decisionmaking for Financial Trading: A Fusion Approach of Machine Learning and Portfolio Selection. Expert Systems with Applications,2019, (115):635-655

[3] Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J.,... & Li, J. (2022). A survey on machine learning models for financial time series forecasting. Neurocomputing, 512, 363-380.

[3] Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2023). Machine learning advances for time series forecasting. Journal of Economic Surveys, 37(1), 76-111.

[4] Wang, J., Hong, S., Dong, Y., Li, Z., & Hu, J. (2024). Predicting stock market trends using lstm networks: overcoming RNN limitations for improved financial forecasting. Journal of Computer Science and Software Applications, 4(3), 1-7.

[5] S. Safwat, A. Mahmoud, I. Eldesouky Fattoh and F. Ali, "Hybrid Deep Learning Model Based on GAN and RESNET for Detecting Fake Faces," in IEEE Access, vol. 12, pp. 86391-86402, 2024, doi: 10.1109/ACCESS.2024.3416910.

[6] Shi X, Zhang Y, Yu M, Zhang L. 2025. Deep learning for enhanced risk management: a novel approach to analyzing financial reports. PeerJ Computer Science 11:e2661 https://doi.org/10.7717/peerj-cs.2661

[7] Huang, X.; Han, M.; Deng, Y. A Hybrid GAN-Inception Deep Learning Approach for Enhanced Coordinate-Based Acoustic Emission Source Localization. Appl. Sci. 2024, 14, 8811. https://doi.org/10.3390/app14198811

[8] Ren, S. (2022). Optimization of enterprise financial management and decision-making systems based on big data. Journal of Mathematics, 2022(1), 1708506.

[9] Qi, Q. (2022). Analysis and forecast on the price change of shanghai stock index. Journal of Economics, Business and Management, 10(1), 72-78.

[10] Petrozziello, A., Troiano, L., Serra, A., Jordanov, I., Storti, G., Tagliaferri, R., & La Rocca, M. (2022).

Deep learning for volatility forecasting in asset management. Soft Computing, 26(17), 8553-8574.

[11] Li, Y., & Pan, Y. (2022). A novel ensemble deep learning model for stock prediction based on stock prices and news. International Journal of Data Science and Analytics, 13(2), 139-149.

[12] Souto, H. G., & Moradi, A. (2023). Forecasting realized volatility through financial turbulence and neural networks. Economics and Business Review, 9(2), 133-159.

[13] Zhan, X., Ling, Z., Xu, Z., Guo, L., & Zhuang, S. (2024). Driving efficiency and risk management in finance through AI and RPA. Unique Endeavor in Business & Social Sciences, 3(1), 189-197.

[14] Wei, L., Deng, Y., Huang, J., Han, C., & Jing, Z. (2022). Identification and analysis of financial technology risk factors based on textual risk disclosures. Journal of Theoretical and Applied Electronic Commerce Research, 17(2), 590-612.

[15] Lei, Y., Qiaoming, H., & Tong, Z. (2023). Research on supply chain financial risk prevention based on machine learning. Computational Intelligence and Neuroscience, 2023(1), 6531154.

[16] Levytska, S., Pershko, L., Akimova, L., Akimov, O., Havrilenko, K., & Kucherovskii, O. (2022). A riskoriented approach in the system of internal auditing of the subjects of financial monitoring. International Journal of Applied Economics, Finance and Accounting, 14(2), 194-206.

[17] Wang, H., & Budsaratragoon, P. (2023). Exploration of an" Internet+" grounded approach for establishing a model for evaluating financial management risks in enterprises. International Journal for Applied Information Management, 3(3), 109-117.

[18] A. Malki, E. Atlam, I. Gad, Machine learning approach of detecting anomalies and forecasting time-series of IoT devices, Alex. Eng. J., 61 (11) (2022), pp. 8973-8986, 10.1016/j.aej.2022.02.038

[19] K.E. Arunkumar, D. V Kalaga, C. Mohan, S. Kumar, T.M. Brenza, Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends
Alex. Eng. J., 61 (10) (2022), pp. 7585-7603, 10.1016/j.aej.2022.01.011

[20] R.S. Society, Review author (s): M. G. Kendall review by: M. G. Kendall source: journal of the royal statistical society. Series A (general), J. Roy. Stat. Soc., 134 (3) (2016), pp. 450-453, 134, No. 3 (1971), Published by: Wiley for the Royal Statistical Society Stable URL:
http://www.jstor.or

[21] Sutiene K, Schwendner P, Sipos C, Lorenzo L, Mirchev M, Lameski P, Kabasinskas A, Tidjani C, Ozturkkal B, Cerneviciene J. Enhancing portfolio management using artificial intelligence: literature review. Front Artif Intell. 2024 Apr 8; 7:1371502. doi: 10.3389/frai.2024.1371502. PMID: 38650961; PMCID: PMC11033520.

[22] Xu, R., Yang, Y., Qiu, H., Liu, X., & Zhang, J. (2024). Research on Multimodal Generative Adversarial Networks in the Framework of Deep Learning. Journal of Computing and Electronic Information Management, 12(3), 84-88.

[23] Dai, W., Tao, J., Yan, X., Feng, Z., & Chen, J. (2023, November). Addressing Unintended Bias in Toxicity Detection: An LSTM and Attention-Based Approach. In 2023 5th International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 375- 379). IEEE.

[24] Yao, J., Wu, T., & Zhang, X. (2023). Improving depth gradient continuity in transformers: A comparative study on monocular depth estimation with cnn. arXiv preprint arXiv:2308.08333.

[25] Wang, X. S., & Mann, B. P. (2020). Attractor Selection in Nonlinear Energy Harvesting Using Deep Reinforcement Learning. arXiv preprint arXiv:2010.01255.

[26] Zhang, Y., Jiang, Z., Peng, C., Zhu, X., & Wang, G. (2024). Management analysis method of multivariate time series anomaly detection in financial risk assessment. Journal of Organizational and End User Computing, 36(1), 1 -19.

[27] Pandey, A., Mannepalli, P.K., Gupta, M. et al. A Deep Learning-Based Hybrid CNN-LSTM Model for Location-Aware Web Service Recommendation. Neural Process Lett 56, 234 (2024). https://doi.org/10.1007/s11063-024-11687-w

[28] Zhigang Sun, Guotao Wang, Pengfei Li, Hui Wang, Min Zhang, Xiaowen Liang, An improved random forest based on the classification accuracy and correlation measurement of decision trees, Expert Systems with Applications, Volume 237, Part B, 2024, https://doi.org/10.1016/j.eswa.2023.121549.

[29] Saratu Yusuf Ilu, Rajesh Prasad, improved autoregressive integrated moving average model for COVID-19 prediction by using statistical significance and clustering techniques, Heliyon, Volume 9, Issue 2, 2023,
e13483,
https://doi.org/10.1016/j.heliyon.2023.e13483.

GridRiskNet: A Two-Stage Hybrid Model for Project Investment Risk Management of Power Grid Enterprises Using Big Data Mining

Hongzhi Gao*, Dekyi, Metok
State Grid Tibet Electric Power Co., Ltd., Lhasa 850000, China
E-mail: Djgy1108@163.com
*Corresponding author

Keywords: power grid enterprise engineering project, GridRiskNet, big data mining, project investment risk management, two-stage hybrid modeling

Received: June 10, 2025

To enhance the power grid enterprise's ability to comprehensively perceive and dynamically assess investment risks in engineering projects, this study proposes a risk management model called GridRiskNet based on big data mining. This model integrates structured, unstructured, and spatiotemporal data and realizes intelligent identification of project risk probability distributions and potential impact ranges by constructing a two-stage hybrid modeling architecture. In the first stage, the model uses eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Machine (LightGBM) to extract static and dynamic features in parallel. In the second stage, it introduces Graph Attention Recurrent Neural Network (GA-RNN) to model risk propagation paths under the power grid topology. Meanwhile, this study combines Spatio-Temporal Graph Convolutional Network (ST-GCN) to improve the coupling expression of meteorological and text features. The experiment uses multi-source public data for verification, such as power infrastructure data from the U.S. Energy Information Administration, meteorological observation data from the National Oceanic and Atmospheric Administration, and power grid topology data from OpenStreetMap. The results show that GridRiskNet performs excellently in risk prediction stability and regional propagation modeling. Among them, the risk principal component analysis projection score in 2023 reached 7.779. This indicates that cost overruns, climate pressure, and equipment technology risks together form a high-risk cluster, with cost overruns increasing by 269% compared with 2018. In the State-of-the-Art comparison, GridRiskNet achieves an F1-score of 0.892, a Receiver Operating Characteristic - Area Under Curve of 0.962, a Risk Impact Radius error of approximately 4.8 km, and a Risk Entropy of 0.89; these are comprehensively better than existing methods. Moreover, the model has good cross-modal feature fusion and risk transmission mechanism identification capabilities, and can effectively characterize the spatiotemporal coupling risk features in complex power grid projects. Overall, this system can provide power grid enterprises with structured and interpretable risk index outputs and regional early warning support. Thus, it helps to improve the investment safety and operational and maintenance resilience of projects.

Povzetek: Predstavljen je GridRiskNet, dvofazni hibridni model za upravljanje investicijskih tveganj v elektroenergetskih projektih. S križnim združevanjem strukturiranih, besedilnih in prostorsko-časovnih podatkov ter uporabo XGBoost/LightGBM in GA-RNN izboljša napoved tveganj (F1=0,892, AUC=0,962) ter natančno modelira regionalno širjenje tveganj (napaka 4,8 km).

1 Introduction

With the accelerated promotion of energy transition and the construction of new power systems, the strategic position of power grid engineering projects in national energy security and clean energy consumption has become increasingly prominent [1]. However, power grid enterprises face problems such as the surge of multi-source heterogeneous data, highly uncertain engineering environments, and frequent external disturbances during project investment and construction. These problems make traditional risk management methods difficult to cover the dynamic risk chain throughout the whole process from construction preparation, equipment deployment, to operation and maintenance support [2]. Especially against the backdrop of the rapid development of renewable energy, the risk types in project investment are constantly evolving. For example, the enhancement of climate extremeness, the swift change of equipment technology paths, and the increase in policy compliance costs all propose higher requirements for the intelligence and adaptability of risk early warning systems [3-5]. Therefore, constructing a big data mining-based intelligent risk assessment model has become a key path to improving the investment decisions' scientific nature and the power grid enterprises' resilience governance

capabilities [6, 7]. In the context of power market liberalization, the continuous increase in the proportion of renewable energy has made the risk management of geographical locations caused by network congestion increasingly important. Improving the ability to model location-related risks has become the core foundation for supporting project financing and investment feasibility assessment [8].

In recent years, artificial intelligence (AI) technologies have made remarkable progress in risk identification, modeling, and prediction. Model architectures represented by graph neural network (GNN), attention mechanisms, and deep semantic modeling have been gradually applied to financial risk control and energy dispatching [9, 10]. Some studies have attempted to introduce machine learning (ML) methods into the power engineering field. It includes using eXtreme Gradient Boosting (XGBoost) for classification and identification of construction anomalies, or employing a convolutional neural network (CNN) for trend prediction of construction period delays [11]. However, existing methods generally suffer from shortcomings such as a single model structure, weak data fusion capability, and difficulty in explaining cross-modal causal paths; these methods cannot effectively support power grid enterprises in achieving full-chain risk perception, dynamic quantification, and structural early warning in a multi-source data environment. Therefore, there is an urgent need to construct a multi-modal driven composite risk assessment system for power grid engineering scenarios.

To this end, this study proposes the GridRiskNet model based on big data mining, constructs a fusion mechanism for structured data, unstructured text, and spatiotemporal data. Thus, it realizes comprehensive modeling and dynamic evaluation of investment risks in power grid engineering projects. The study's main innovations encompass:

(1) Proposing a two-stage GridRiskNet model architecture: It integrates XGBoost and Light Gradient Boosting Machine (LightGBM) for risk capture, and models the propagation process of risks in the power grid topology through a Graph Attention Recurrent Neural Network (GA-RNN).

(2) Introducing Spatio-Temporal Graph Convolutional Network (ST-GCN) and cross-modal attention mechanisms: It enhances the model's expression capabilities for meteorological disturbances and regional structural information;

(3) Constructing a risk principal component projection index system based on Principal Component Analysis (PCA): It achieves structural clustering and projection analysis of high-dimensional risk samples, and supports the differentiated regional risk management needs of power grid enterprises.

Overall, the specific research question is whether multimodal data fusion and risk propagation modeling methods can enhance the comprehensive capabilities of risk classification, propagation path identification, and uncertainty quantification in complex power grid engineering projects. The target outcome is to achieve a comprehensive portrayal of investment risks in power grid engineering projects by constructing a composite model that integrates structured, spatiotemporal, and text data. It also aims to verify the advantages of the proposed method in terms of risk identification accuracy, propagation path reducibility, and risk distribution stability. Thus, it supports power grid enterprises in risk early warning and decision optimization.

2 Related work

With the in-depth application of AI technologies and big data analysis methods in engineering management, investment project risk assessment has gradually shifted from traditional static analysis to intelligent prediction and dynamic modeling. Aiming at the insufficiency of risk assessment for manufacturing investments, Dong and Li proposed combining expert experience with big data mining to construct project risk indices and integrating CNN with Long Short-Term Memory (LSTM) for predictive modeling. In multiple sliding window tests, the model achieved a Receiver Operating Characteristic (ROC) value of 0.9366 and an average accuracy of 94.95%, demonstrating high prediction precision [12]. Loseva et al., facing the risk assessment task of regional franchising projects, constructed a big data-based credit rating model by combining the SPARK information system with ML methods. This verified the model's robustness in identifying abnormal risks through Spearman correlation and confusion matrix [13]. These studies have provided useful insights into introducing composite modeling methods and integrating expert judgment with data-driven mechanisms, gradually promoting the development of investment risk assessment towards intelligence and systematization.

Over the years, methods such as GNN, deep clustering, and multi-criteria decision-making have been widely introduced into investment evaluation and project classification, further enhancing the structural cognitive ability of risk assessment. Mostofi et al. constructed a construction project investment framework based on graph attention networks. This framework achieved a classification accuracy of over 98% in three sub-networks of region, country, and financing model, demonstrating the advantages of graph structure in modeling investment decision-making relationships [14]. Qi used regularized topic models and graph clustering methods to construct a financial investment "behavior circle". They mapped customer behaviors to the latent semantic space and realized risk classification of financial communities and investment plan recommendations through subgraph mining [15]. Moreover, Luo and Zhu proposed a deep neural network (DNN) model based on transfer learning for regional investment risk assessment. This model maintained high prediction accuracy (up to 92%) in the case of insufficient samples, demonstrating the potential of deep learning in solving unbalanced data problems [16]. These studies all reflect the integration trend of risk assessment models in recent years towards deep representation learning, multi-layer decision-making

structures, and complex graph relationship modeling.

Although existing studies have made positive progress in risk modeling methods, index system construction, and model accuracy improvement, there are still three main deficiencies. First, most current models focus on classification or regression prediction of risk probability, lacking the ability to model regional structural propagation characteristics. Second, the heterogeneity of multi-source data has not been fully utilized, and a unified representation for structured, spatiotemporal, textual, and other multimodal information has not been formed. Third, the interpretability and quantifiability of risk structure evolution must be enhanced, making it difficult to support dynamic scheduling and regional risk management of complex systems such as power grid projects [17]. In response to the above shortcomings, this study proposes a grid engineering project investment risk management system based on big data mining - the GridRiskNet model. This model reveals the changing trends of high-dimensional risk structures and supports grid enterprises in accurately perceiving and dynamically controlling investment risks at different regions and time scales.

3 GridRiskNet model based on big data mining

3.1 Realization process of GridRiskNet model

The proposed GridRiskNet model realizes intelligent assessment of investment risks in power grid engineering projects based on multi-source heterogeneous data fusion and a hybrid ML architecture. It first establishes a multimodal data preprocessing layer. For structured data (such as project budgets and equipment parameters), an adaptive normalization method is used to unify dimensions, ensuring the consistency of feature scales. For unstructured text data (including engineering logs and bidding documents), a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model is utilized to deeply extract semantic features, enhancing the risk perception ability of text information. For spatiotemporal data (such as construction trajectories and meteorological records), ST-GCN is introduced to jointly encode complex environmental features from two dimensions: spatial dependence and temporal dynamics [18, 19]. In the feature fusion stage, a cross-modal attention mechanism is designed, which can adaptively learn the weight relationships between different data modalities. Meanwhile, this mechanism can effectively integrate multi-source features and generate unified and dense high-dimensional risk representation vectors, laying the foundation for multi-dimensional risk modeling [20].

At the core of modeling, GridRiskNet adopts a two-stage hybrid modeling framework. In the first stage, the improved XGBoost and LightGBM models run in parallel to jointly perform risk prediction on high-dimensional risk representation vectors. Specifically, XGBoost integrates a dynamic feature selection mechanism, which dynamically updates feature importance indices based on sliding window statistical features to enhance the response capability to dynamic risk factors. LightGBM incorporates a time-series-aware splitting criterion to strengthen the detection capability for time-series anomalies such as project schedule delays. The two models output the prediction probabilities of risk categories (i.e., risk probability vectors after Softmax) and sequences of feature importance scores [21].

In the second stage, GA-RNN is used as a meta-model, whose core innovation lies in fusing the dual output information from the first stage mentioned above. Specifically, GA-RNN takes the risk probability vectors of XGBoost and LightGBM as the main input; simultaneously, it introduces their feature importance score sequences as auxiliary features to form a comprehensively fused feature matrix. This matrix contains the risk prediction results from the previous stage; it also explicitly integrates the influence weights of features on the model output, thereby enhancing the ability to perceive risk propagation mechanisms [22]. Subsequently, based on this matrix, GA-RNN introduces a risk propagation graph structure and accurately models the transmission relationships between risk factors through an adjacency matrix. Moreover, it uses graph attention mechanisms and recurrent neural network (RNN) units to dynamically learn key nodes and main channels in risk propagation paths, extracting high-order interaction features.

The entire GridRiskNet model comprehensively optimizes classification cross-entropy loss, risk propagation graph reconstruction error, and feature stability regularization terms through an end-to-end joint training strategy. Finally, this model outputs a multi-dimensional risk assessment matrix covering risk probability distribution, potential impact range, and structural features. The entire system adopts an online incremental learning mechanism, which can continuously absorb real-time data flow to dynamically update model parameters; this achieves a high adaptability and continuous tracking of the risk environment of power grid engineering projects. The implementation process and pseudocode of GridRiskNet are illustrated in Figures 1 and 2.

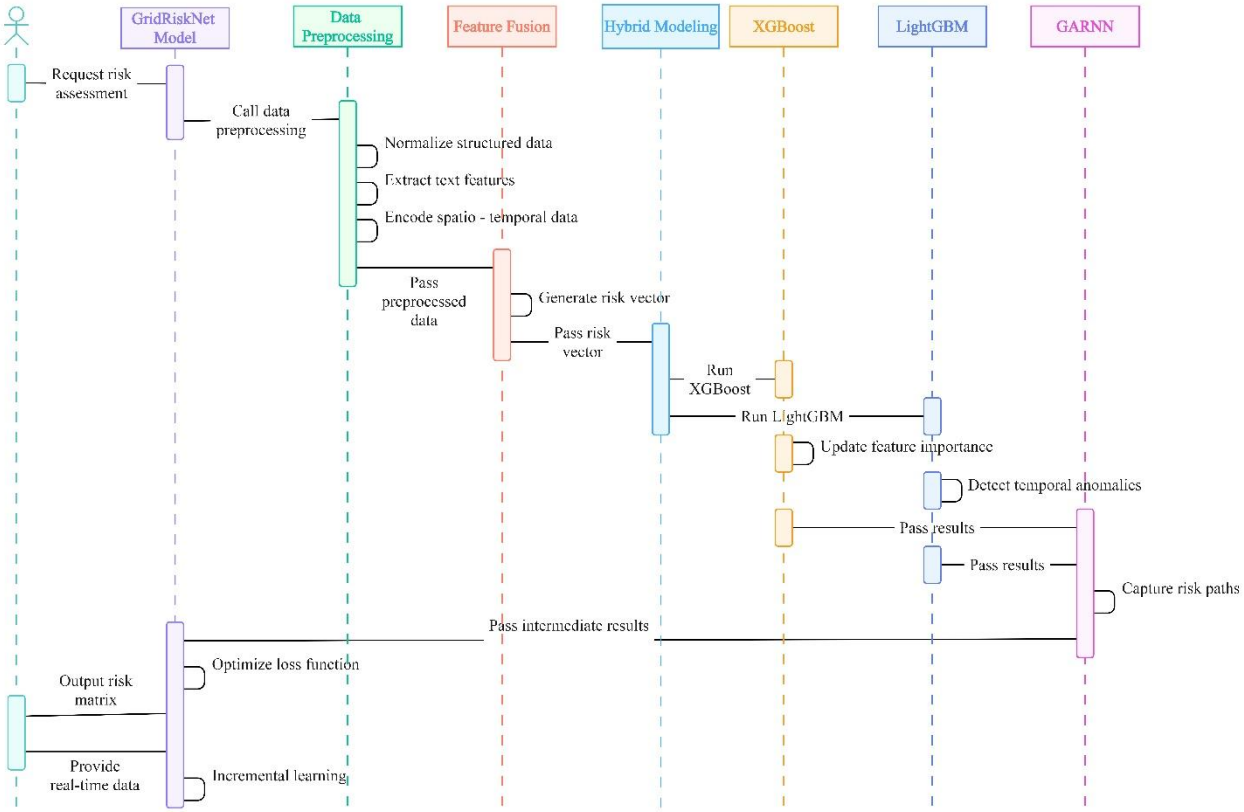


Figure 1: The implementation process of GridRiskNet

```
class GridRiskNet:
    def __init__(self, config):
        self.config = config
        self.preprocessor = MultiModalPreprocessor(config)
        self.feature_fusion = CrossModalAttention(config)
        self.first_stage = HybridEnsembleModels(config)
        self.risk_graph = RiskPropagationGraph(config)
        self.second_stage = GARNNMetaModel(config, self.risk_graph)

    def train(self, dataset):
        features = self.preprocessor.process(dataset)
        fused_features = self.feature_fusion(features)
        first_stage_preds = self.first_stage.train(fused_features, dataset.labels)
        self.second_stage.train(first_stage_preds, fused_features, dataset.labels)

        for epoch in range(self.config.epochs):
            preds = self.predict(dataset)
            loss = self._calculate_loss(preds, dataset.labels)
            self._update_models(loss)

    def predict(self, dataset):
        features = self.preprocessor.process(dataset)
        fused_features = self.feature_fusion(features)
        first_stage_preds = self.first_stage.predict(fused_features)
        return self.second_stage.predict(first_stage_preds, fused_features)

    def update_with_new_data(self, new_data):
        features = self.preprocessor.update_and_process(new_data)
        self.first_stage.update(features, new_data.labels)
        first_stage_preds = self.first_stage.predict(features)
        self.second_stage.update(first_stage_preds, features, new_data.labels)

class MultiModalPreprocessor:
    def process(self, dataset):
        return {
            'structured': self._process_structured(dataset.structured),
            'text': self._process_text(dataset.text),
            'spatiotemporal': self._process_spatiotemporal(dataset.spatiotemporal)
        }

    def _process_structured(self, data):
        return AdaptiveNormalization(data)

    def _process_text(self, data):
        return FineTuneBERT(self.bert_model, data)

    def _process_spatiotemporal(self, data):
        return STGCN(self.stgcn_params).forward(data)

class CrossModalAttention:
    def __call__(self, features):
        weights = self._compute_attention_weights(features)
        return weighted_sum(features, weights)

class HybridEnsembleModels:
    def __init__(self, config):
        self.xgboost = ImprovedXGBoost(config)
        self.lightgbm = ImprovedLightGBM(config)

    def train(self, features, labels):
        xgb_preds = self.xgboost.train(features, labels)
        lgbm_preds = self.lightgbm.train(features, labels)
        return combine_predictions(xgb_preds, lgbm_preds)

class RiskPropagationGraph:
    def __init__(self, config):
        self.adj_matrix = self._construct_adjacency_matrix(config.risk_factors)

    def _construct_adjacency_matrix(self, risk_factors):
        # Construct adjacency matrix based on domain knowledge or data learning
        pass

class GARNNMetaModel:
    def train(self, first_stage_preds, features, labels):
        # Train GA-RNN model
        pass

    def predict(self, first_stage_preds, features):
        # Predict risk assessment matrix
        pass
```

Figure 2: The pseudocode of GridRiskNet

3.2 Mathematical modeling principle of GridRiskNet model

Figure 1 shows the complete implementation process of the GridRiskNet model, covering the entire process from user-requested risk assessment to model output results and continuous updates. The model is built on multi-source heterogeneous data, fusing structured, unstructured, and spatiotemporal information, and achieves intelligent prediction of power grid project risks through multi-stage ML and graph modeling strategies. The key computational links in the model are described mathematically and logically as follows.

In the data preprocessing stage, the structured input data is first normalized. Let the original data matrix be:

$$\mathbf{X}_s \in \mathbb{R}^{n \times d_s} \tag{1}$$

$\mathbf{X}_s$ represents n records, and each record contains d_s structural features. Normalization calculation is as follows:

$$\widetilde{\mathbf{X}}_s = \frac{\mathbf{X}_s - \mu_s}{\sigma_s + \epsilon} \tag{2}$$

μ_s denotes the column vector, indicating the average value of each column. σ_s represents the standard deviation (SD), and ϵ is a positive number to prevent the denominator from being zero. This processing ensures that the model has numerical consistency among different dimensional features.

For unstructured text data $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$, semantic features are extracted by fine-tuning BERT model, and the output is:

$$\mathbf{H}_t = \text{BERT}(\mathcal{T}) = [\mathbf{h}_1; \mathbf{h}_2; \ldots; \mathbf{h}_m], \mathbf{h}_i \in \mathbb{R}^{d_t} \tag{3}$$

$\mathbf{h}_i$ is the semantic vector of the i-th text, and the dimension is d_t. This step preserves the semantic relationship between text contexts and forms an important basis for the model to recognize risk semantics.

For spatiotemporal data including trajectory and meteorology, it is expressed as:

$$\mathbf{X}_{st} \in \mathbb{R}^{T \times N \times F} \tag{4}$$

T refers to the time step. N denotes the space node (such as the site number), and F represents the space-time characteristic dimension of each node. ST-GCN is used for modeling, and its core propagation equation is:

$$\mathbf{Z}^{(l+1)} = \sigma\left(\sum_{k=0}^{K} \mathbf{A}_k \mathbf{Z}^{(l)} \mathbf{W}_k\right) \tag{5}$$

$\mathbf{A}_k$ means the adjacency matrix of order k; $\mathbf{Z}^{(l)}$ indicates the node representation of the l-th layer; $\mathbf{W}_k$ stands for the weight matrix; σ represents the activation function. This network structure can capture the linkage relationship between spatial topology and time evolution.

In the feature fusion stage, the model introduces cross-modal attention mechanism to automatically aggregate multi-source information. Let two modal features be $\mathbf{F}_i$ and $\mathbf{F}_j$ respectively, and their attention weights are calculated as:

$$\alpha_{i,j} = \frac{\exp\left(\mathbf{F}_i^{\top} \mathbf{W}_a \mathbf{F}_j\right)}{\sum_k \exp\left(\mathbf{F}_i^{\top} \mathbf{W}_a \mathbf{F}_k\right)} \tag{6}$$

After fusion, a unified risk representation vector is obtained:

$$\mathbf{F}_{fusion} = \sum_j \alpha_{i,j} \cdot \mathbf{F}_j \tag{7}$$

This mechanism enables the model to automatically learn the most discriminating risk signal source when faced with heterogeneous features and semantic diversity.

The hybrid modeling framework is divided into two stages. In the first stage, the improved XGBoost and LightGBM models are run in parallel. The objective function of XGBoost reads:

$$\mathcal{L}_{xgb} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{8}$$

$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i)$, which represents the predicted value of sample i. $\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \parallel \omega_k \parallel^2$ is the regular term of the k-th tree.

To adapt to the dynamic change of time, XGBoost integrates a sliding window statistical module to dynamically adjust the importance of features:

$$I_j^{(t)} = \sum_{s=t-w}^{t} \tag{9}$$

$\Delta G_j^{(s)}$ indicates the gain change of the j-th feature in the s-th time step. $I_j^{(t)}$ c a dynamic feature importance index within the XGBoost stage, used to reflect gain changes within the sliding window; it is also mainly applied to internal feature selection and dynamic weight adjustment of the first-stage model.

LightGBM introduces the split criterion of time series perception to enhance the ability of anomaly recognition. Let the time series samples be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, and its splitting gain is defined as:

$$\mathcal{G}_j = \sum_{t=1}^{T} w_t \cdot \left[\frac{(\sum_{i \in L_t} g_i)^2}{\sum_{i \in L_t} h_i + \lambda} + \frac{(\sum_{i \in R_t} g_i)^2}{\sum_{i \in R_t} h_i + \lambda}\right] \tag{10}$$

g_i and h_i are gradients and second derivatives. L_t and R_t represent the left and right sample sets of the current split, and $w_t = e^{-\beta(T-t)}$ is the time attenuation weight.

In the second stage, GA-RNN is used to capture the high-order risk path. Its node status is updated to:

$$\mathbf{h}_i^{(t)} = \text{GRU}\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{h}_j^{(t-1)}, \mathbf{h}_i^{(t-1)}\right) \tag{11}$$

$\mathcal{N}(i)$ indicates the neighbor set of node i. α_{ij} v the edge weight under the graph attention mechanism:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}(\mathbf{a}^{\top}[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j])\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\text{LeakyReLU}(\mathbf{a}^{\top}[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_k])\right)} \tag{12}$$

Finally, the system integrates three kinds of objectives: classification performance, graph structure reconstruction, and feature stability by jointly optimizing the overall loss function.

$\mathcal{L}_{ce}$ is the cross-entropy loss:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \lambda_1 \cdot \mathcal{L}_{graph} + \lambda_2 \cdot \mathcal{L}_{reg} \tag{13}$$

The loss of graph structure consistency is:

$$\mathcal{L}_{graph} = \parallel \mathbf{A} - \widehat{\mathbf{A}} \parallel_F^2 \tag{14}$$

The regular term of characteristic disturbance reads:

$$\mathcal{L}_{reg} = \sum_{j=1}^{d} \text{Var}(\nabla_{\mathbf{x}_j} \hat{y}) \tag{15}$$

On the system deployment level, GridRiskNet adopts online incremental learning mechanism. Let the current parameter be θ_t, and the model is updated after receiving new samples $(\mathbf{x}_t, y_t)$:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \mathcal{L}(\mathbf{x}_t, y_t; \theta_t) \tag{16}$$

η represents the learning rate. ∇_θ denotes the gradient operator. This mechanism ensures that the model has adaptive update abilities in a dynamic risk environment.

4 Experimental analysis of GridRiskNet model project investment risk management based on big data mining

4.1 Data used in the study

To verify the risk management capability of the GridRiskNet model for power grid enterprise engineering projects, the study uses three core public datasets for experimental validation and designs a fusion scheme for data heterogeneity. First, the structured data adopts the U.S. Energy Information Administration (EIA) power infrastructure dataset (https://www.eia.gov/electricity/data.php). Its API interface screens power grid engineering project data from 2018 to 2023, including budget, construction period, equipment models, and other fields. After extracting the original CSV-format data using Python's eia-python library, adaptive normalization is performed to eliminate dimension differences, which are associated with subsequent spatiotemporal data through project IDs and date fields. Second, the spatiotemporal data selects the National Oceanic and Atmospheric Administration (NOAA) Global Historical Climatology Network-Daily (GHCN-Daily) (https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ncdc:C00861). Daily values of temperature, precipitation, and wind speed are downloaded, and stations are matched to the project's geographic coordinates. The rnoaa toolkit converts them into spatiotemporal tensors, from which meteorological risk features are extracted through ST-GCN encoding. The spatial topology data is obtained from the OpenStreetMap power network dataset (https://wiki.openstreetmap.org/wiki/Power_networks). The OSMnx library extracts GIS data of substations and transmission lines, constructing an adjacency matrix to model the physical connection of the power grid. For unstructured text data, engineering accident reports from 2018 to 2023 corresponding to EIA projects are manually screened from the Federal Energy Regulatory Commission (FERC) engineering accident report library (https://elibrary.ferc.gov/eLibrary/search). After parsing the text with Apache Tika, the fine-tuned BERT is input to generate semantic vectors.

The following fusion strategies are adopted to address the heterogeneity of multi-source data. 1) Temporal alignment: All data is uniformly converted to Universal Time Coordinated (UTC) timestamps and aggregated at a granularity of 1 day. 2) Spatial alignment: Meteorological stations, power grid nodes, and engineering sites are associated through GIS coordinate matching (error <1km). 3) Consistency of feature encoding: Structured data is normalized to [0,1], text vectors are unified into 768 dimensions via BERT, and spatiotemporal data is compressed into 256-dimensional features through ST-GCN. 4) Cross-modal attention mechanisms automatically learn the weights of each modality, assigning higher attention scores to extreme meteorological text descriptions (such as "hurricane damage"). The specific process of importing data into GridRiskNet is presented in Figure 3.

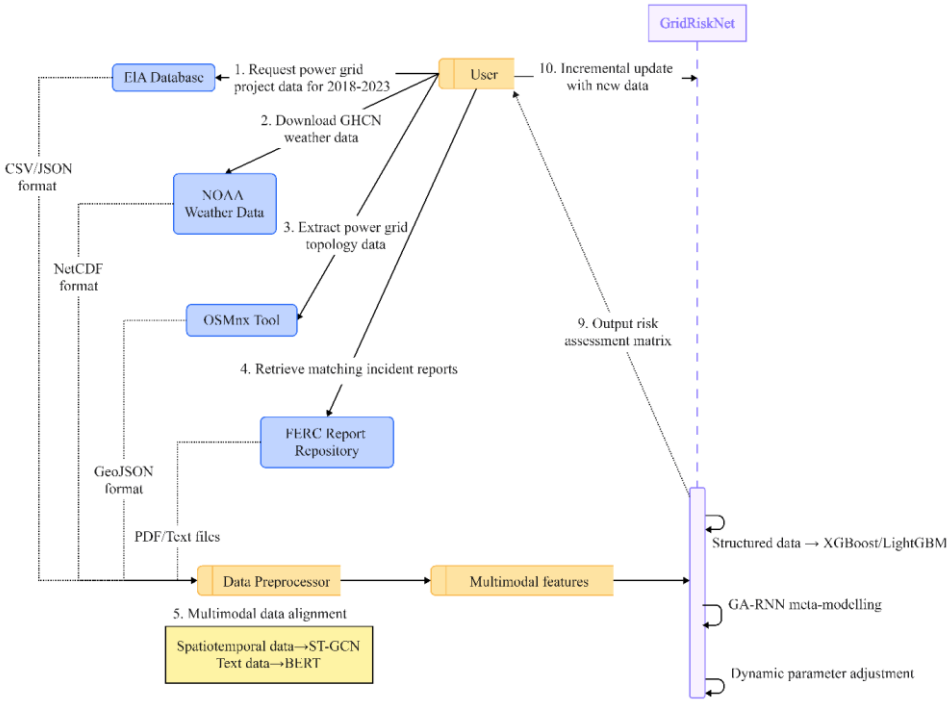


Figure 3: The specific process of importing data into GridRiskNet

4.2 GridRiskNet model's thinking on risk management ability analysis of power grid enterprises' engineering projects

The study is conducted from two aspects to effectively analyze the risk management capability of the GridRiskNet model for power grid enterprise engineering projects. First, at the level of risk probability distribution, the probability of risks such as budget overrun and construction period delay is evaluated based on structured data and spatiotemporal features [23]. Second, at the level of potential impact scope, the propagation path of risks in the power grid topology is analyzed through GNN to identify high-risk nodes and their potentially affected surrounding areas. The model inputs the fused multi-dimensional features into a two-stage modeling framework and outputs a risk assessment matrix including the above two types of indices to support the refined and structured management and decision-making of power grid project risks [24, 25]. The entire analysis's key indices and evaluation criteria are exhibited in Tables 1 and 2.

Table 1: Explanation of key indices for analysis using the GridRiskNet model

Analytical dimension	Index	Data source/calculation method	Description
Risk probability distribution analysis	Risk PCA Projection Score	Principal component score of high-dimensional risk vector output by the GridRiskNet model after PCA dimensionality reduction	Reflecting the distribution position of samples in the risk principal component space, and is used to identify high-risk clustering or structural abnormal samples.
	Time-series Anomaly Frequency	The capture times of abnormal events in LightGBM	Monitoring the frequency of abnormal progress.
	Model Confidence Score	Softmax outputs the maximum probability value	Evaluating the credibility of the model output
	Risk Coefficient of Variation	The ratio of SD to the mean value of the risk probability distribution	Assessing the dispersion degree of risk probability distribution, the greater it is, the higher the risk instability is.
	Risk Importance Index	Comprehensive weighted scores of multiple dimensions	Representing the strength of risk influence
	Risk Entropy	Information entropy calculation of risk probability distribution	The degree of uncertainty in evaluating risk results.
Analysis of the potential influence range	Risk Propagation Path Length	Critical path length identified in GA-RNN	Length and complexity of the risk propagation path
	Node Vulnerability Score	The weighted average of the affected probability of each node in GNN	Reflect the vulnerability of nodes in the power grid
	Risk Impact Radius	Based on propagation path depth and the spatial adjacency matrix in the graph structure	Indicating the physical scope of risk propagation

Table 2: Criteria for determining key indices in the GridRiskNet model analysis

Index	Type	Criteria
Risk PCA Projection Score	Secondary calculation	- [0, 2) Low projection; [2, 5) Medium projection; ≥5 High projection, tending to abnormal samples or extreme types
Time-series Anomaly Frequency	Model output	- [0, 2) Normal; [2, 5) Early warning; ≥5 Abnormal
Model Confidence Score	Model output	- [0.9, 1] High credibility; [0.7, 0.9) Medium credibility; <0.7 Low credibility
Risk Coefficient of Variation	Secondary calculation	- [0, 0.3) Stable; [0.3, 0.6) Fluctuating; ≥0.6 Highly unstable
Risk Importance Index	Secondary calculation	- [0, 40) Secondary; [40, 70) Important; [70, 100] Critical
Risk Entropy	Secondary calculation	- [0, 1) Low uncertainty; [1, 2) Medium; ≥2 High
Risk Propagation Path Length	Model output	- [1, 3) Local; [3, 6) Regional; ≥6 Global
Node Vulnerability Score	Model output	- [0, 0.4) Low; [0.4, 0.7) Medium; [0.7, 1] High
Risk Impact Radius	Secondary calculation	- [0, 5) Station level; [5, 20) Line level; ≥20 Regional level

In Table 2, the index equations involved in secondary calculation are as follows:

(1) Risk PCA Projection Score

Here, "Risk PCA Projection Score" measures the

position of a sample in the dominant risk structure within the risk feature space, revealing the main variation trends in complex multi-dimensional risk features. Specifically, this index is calculated based on the PCA method. First, it standardizes the annual high-dimensional risk features (such as cost overrun risk, environmental and climate pressure, etc.). Then, it extracts the first K principal component directions and measures the sample's projection value in the principal component space through eigenvalue weighting. This score reflects the degree of variance contribution of the sample along the principal component axis of risk, rather than a simple sum of the scores of each risk factor. Due to the different statistical distributions of risk features each year, this index changes with the year; it comprehensively reflects the overall trend of the risk structure of power grid projects in the current year and potential abnormal clustering characteristics. The calculation is expressed as:

$$s_i = \sum_{k=1}^{K} \lambda_k \cdot (\mathbf{u}_k^{\top}(\mathbf{x}_i - \boldsymbol{\mu}))^2 \quad (17)$$

$\mathbf{x}_i$ represents the high-dimensional risk feature vector of the i-th sample. $\boldsymbol{\mu}$ indicates the sample mean vector. $\mathbf{u}_k$ denotes the feature vector in the direction of the k-th principal component. λ_k is the feature value of the k-th principal component, and K means the number of selected principal components.

(2) Risk Coefficient of Variation

This index is used to measure the relative dispersion of risk probability distribution, which is an important index reflecting risk instability. This index describes the fluctuation range of various risk probabilities in the whole by calculating the ratio of the SD of risk probability to the mean. A higher value indicates that the risk probability distribution is more dispersed and the overall instability is stronger. The expression is:

$$CV = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - \bar{p})^2}}{\bar{p} + \epsilon} \quad (18)$$

p_i denotes the prediction probability of Class i risk. $\bar{p}$ means the average value of various risk probabilities, and n represents the total number of risk categories.

(3) Risk Entropy

"Risk Entropy" measures the degree of uncertainty in the risk probability distribution, reflecting the discreteness and unpredictability of risk results. Based on information entropy theory, this index reveals the potential risk mixture in the system by calculating the entropy value of the probability of all risk categories. A higher risk entropy value indicates more uncertainties in the system, which helps to identify complex and unpredictable risk scenarios, represented as:

$$H = -\sum_{i=1}^{n} p_i \cdot \log_2 (p_i + \epsilon) \quad (19)$$

H denotes the information entropy of risk distribution.

(4) Risk Importance Index

This index quantifies the comprehensive contribution of each risk feature to the overall risk assessment results. It reflects the importance level of each risk feature by weighted accumulation of the impact degree of each feature on the model loss and normalized averaging combined with model weights. Features with higher values play a greater role in the overall risk decision-making, expressed as:

$$RI_j = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{w_j^{(t)} \cdot \Delta L_j^{(t)}}{\sum_{k=1}^{d} \Delta L_k^{(t)}}\right) \quad (20)$$

RI_j represents the risk importance index of the j-th feature, which is a risk importance index in the entire GridRiskNet framework. It is comprehensively calculated based on the feature weights and loss impact during the global model training process, belonging to a unified index at the global level. T means the number of model iterations or average times; $w_j^{(t)}$ refers to the model weight of the j-th feature in the t-th iteration; $\Delta L_j^{(t)}$ is the influence degree of the j-th feature on the loss function; d denotes the total number of features.

(5) Risk Impact Radius

It evaluates the spatial propagation range of risks in the power grid graph structure, serving as a key index for measuring the physical scope affected by risks. This index calculates the average impact radius of all risk source nodes in the network based on the power grid topology, geographical distance between nodes, and risk propagation probability. A larger value indicates a wider spatial propagation range of risk events, which is applied to regional risk impact analysis, as follows:

$$R = \frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j=1}^{N} t_{ij} \cdot d_{ij} \cdot p_{ij} \quad (21)$$

N_s represents the number of risk source nodes. N denotes the total number of nodes in the graph. t_{ij} is the adjacency relationship between nodes i and j (1 means connection). d_{ij} means the geographical distance between nodes. p_{ij} refers to the risk propagation probability from nodes i to j.

Figure 4 presents the pseudocode of the index implementation involving secondary calculation.

```
import math
import numpy as np

# 1. Risk PCA Projection Score
def compute_risk_pca_projection_scores(X, K):
    mu = np.mean(X, axis=0)
    X_centered = X - mu
    cov_matrix = np.cov(X_centered, rowvar=False)
    eigenvalues, eigenvectors = np.linalg.eigh(cov_matrix)
    sorted_idx = np.argsort(eigenvalues)[::-1]
    eigenvalues = eigenvalues[sorted_idx][:K]
    eigenvectors = eigenvectors[:, sorted_idx][:, :K]

# 2. Risk Coefficient of Variation
def compute_risk_cv(probabilities):
    mean_p = np.mean(probabilities)
    std_p = np.std(probabilities)
    epsilon = 1e-6
    return std_p / (mean_p + epsilon)

# 3. Risk Entropy
def compute_risk_entropy(probabilities):
    epsilon = 1e-6
    return -sum(p * math.log2(p + epsilon) for p in probabilities)

# 4. Risk Importance Index
def compute_risk_importance(weights, delta_losses):
    T = len(weights)
    D = len(weights[0])
    importance = [0.0] * D
    for j in range(D):
        for t in range(T):
            total_delta = sum(delta_losses[t])
            if total_delta == 0:
                continue
            importance[j] += weights[t][j] * delta_losses[t][j] / total_delta
        importance[j] /= T
    return importance

# 5. Risk Impact Radius
def compute_risk_impact_radius(adj_matrix, distance_matrix,
propagation_probs, source_nodes):
    N = len(adj_matrix)
    total_radius = 0.0
    for i in source_nodes:
        for j in range(N):
            if adj_matrix[i][j] == 1:
                total_radius += distance_matrix[i][j] * propagation_probs[i][j]
    return total_radius / len(source_nodes)
```

Figure 4: Pseudocode of index implementation involving secondary calculation

The experimental environment and key parameters are detailed in Table 3.

Table 3: Experimental environment and key parameters arrangement of the study

Category	Configuration item	Parameter setting
Hardware environment	Computing platform	NVIDIA A100 (40GB memory) × 4
	CPU	AMD EPYC 7763 (64-core)
	Memory	512GB DDR4
Software environment	Deep learning framework	PyTorch 1.12 + CUDA 11.6
	GNN library	PyTorch Geometric 2.2.0
	Traditional ML library	XGBoost 1.6 + LightGBM 3.3.2
	NLP toolkit	HuggingFace Transformers 4.25 (BERT-base)
Model architecture	ST-GCN layer number	3 layers (hidden layer dimension =256)
	GA-RNN unit	Graph Attention Layer (number of heads =8) +GRU (hidden layer =512)
	Transmodal attention mechanism	Multi-attention (number of heads =4, fusion dimension =1024)
Training parameters	Batch size	256 (structured data)/32 (graph data)
	Initial learning rate	3e-4 (AdamW optimizer)
	regularization	L2 Weight Attenuation =1e-5+Dropout=0.3
	Early stop mechanism	The loss of verification set does not decrease for 10 consecutive rounds

The study designs ablation experiments before conducting formal experiments to verify the actual contribution of each core component of GridRiskNet. It seeks to quantitatively measure the impact of different modules on the model's overall performance from a systematic perspective. Specifically, four ablation versions are set by sequentially disabling the cross-modal attention mechanism, the risk propagation modeling module of GA-RNN, the dynamic feature selection module, and the risk propagation graph reconstruction term in the joint loss function. All experiments maintain the same hyperparameter configuration on the complete dataset, focusing on evaluating three indices. These indices include risk classification performance (F1-score, Receiver Operating Characteristic - Area Under Curve (ROC-AUC)), risk propagation accuracy (Risk Impact Radius error), and uncertainty quantification ability (Risk Entropy). This experiment aims to clarify the mechanism of action of each module, especially their specific contributions to power grid risk transmission modeling, modal feature fusion, and risk stability control. The results of the ablation experiments are listed in Table 4.

Table 4: Ablation experimental results of the GridRiskNet model

Ablation version	F1-Score	ROC-AUC	Risk Impact Radius error (km)	Risk Entropy
Full GridRiskNet	0.892	0.962	4.8±0.9	0.89
No cross-modal attention	0.835	0.917	7.5±1.6	1.12
No GA-RNN	0.846	0.926	14.2±2.3	0.96
No dynamic feature selection	0.863	0.941	5.7±1.2	0.94
Risk-free propagation graph reconstruction	0.871	0.948	4.9±1.0	2.08

The results of the ablation experiments indicate that each module of GridRiskNet makes a significant contribution to the model performance. The cross-modal attention mechanism is particularly crucial in improving classification performance; after being disabled, the F1-score decreases by 6.4%, the ROC-AUC drops by 4.7%, and the Risk Entropy rises significantly. This shows that this module significantly impacts the collaborative perception of complex semantic and meteorological features. The risk propagation modeling module of GA-RNN notably improves the Risk Impact Radius error; after being disabled, the error increases sharply to 14.2 km, verifying its core role in power grid topology modeling. The dynamic feature selection module mainly enhances the temporal sensitivity of the model; its removal leads to a significant drop in F1-score, although it has a limited impact on propagation errors. The risk propagation graph reconstruction term has a significant effect on suppressing prediction fluctuations and optimizing uncertainty quantification; its elimination causes a substantial rise in Risk Entropy. Overall, GridRiskNet achieves the unity of high performance and high robustness through the collaboration of various modules, with all components being indispensable.

4.3 Analysis Results of GridRiskNet model on risk management ability of power grid enterprises' engineering projects

4.3.1 Risk probability distribution analysis

GridRiskNet's annual Risk PCA Projection Score results for power grid enterprise engineering projects are summarized in Table 5.

Table 5: Annual Risk PCA Projection Score results

Year	Cost overrun risk C1	Ambient climate pressure C2	Equipment technical risk C3	Supply chain fluctuation C4	Policy compliance risk C5	Risk PCA Projection Score	Risk tendency
2018	1.235	0.873	-0.452	0.217	0.095	2.108	Middle projection (structural abnormality)
2019	0.892	0.654	-0.128	-0.304	0.062	1.546	Low projection
2020	2.874	1.982	1.235	-0.873	0.517	4.856	High projection (extreme type)
2021	1.023	1.457	0.782	0.396	-0.215	2.48	Middle projection
2022	3.125	2.769	2.014	1.358	-0.947	5.894	High Projection (abnormal clustering)
2023	4.562	3.217	3.058	2.146	1.372	7.779	High projection (extreme anomaly)

Table 5 shows that cost overrun risk (C1) and environmental climate pressure (C2) have always been the dominant risks, especially showing exponential growth after 2020. In 2023, C1 (4.562) increased by 269% compared with 2018 (1.235), which is highly consistent with the reality of global inflation and frequent extreme weather. The sudden turn positive (1.372) of policy compliance risk (C5) in 2023, to some extent, reveals the surge of compliance costs brought by the deepening of the "double carbon" policy. The model reflects high-risk clustering scenarios such as C1-C3 in 2023 through the spatial distribution of principal components, reflecting the early warning of composite risks.

Based on the above analysis, the risk probability distribution analysis of grid enterprise engineering projects by GridRiskNet is organized, and the annual average results of other indices are shown in Figure 5.

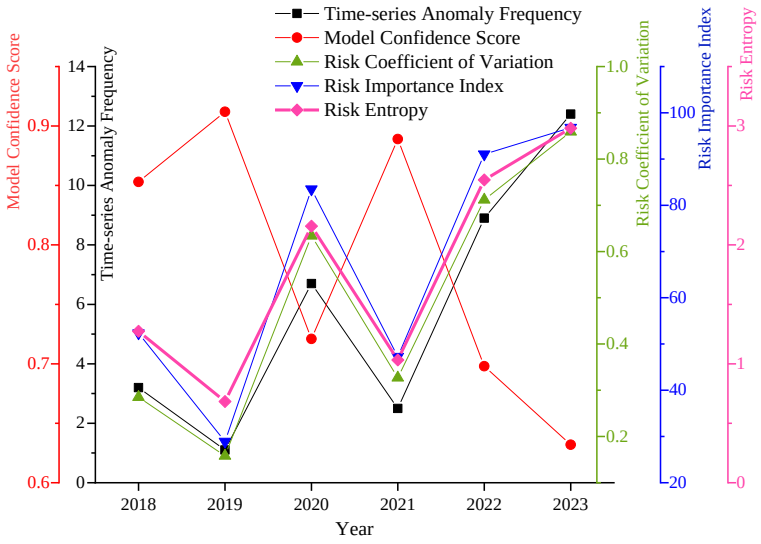
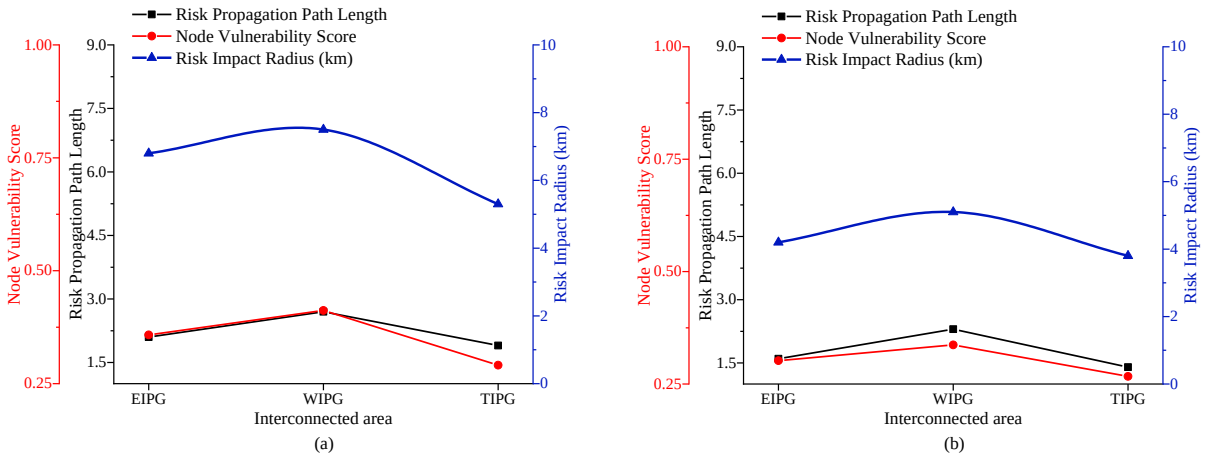


Figure 5: The annual average results of other indices in GridRiskNet risk probability distribution analysis
Note: The curves of each index correspond one-to-one with the corresponding color coordinate axes on the right

In Figure 5, regarding the frequency of time-series anomalies, the average annual growth rate of abnormal events during 2019-2023 reached 65.7%. The model objectively reflects the increasing complexity of risks through the continuous decline in confidence (from 0.912 to 0.632). The sudden increase in risk entropy (2.158) in 2020 preceded the peak of the importance index (83.47); this indicates that GridRiskNet can capture the implicit correlations of risk factors through information entropy. The synchronous increase in the coefficient of variation (from 0.712 to 0.859) and risk entropy (from 2.547 to 2.981) after 2022 reveals the transformation trend of risk distribution from centralized to discretized; this provides key evidence for power grid enterprises to optimize the allocation of risk reserve funds. The core advantage of the model lies in the quantitative modeling of the dynamic coupling relationship among the three dimensions of engineering anomalies, risk uncertainty, and impact degree. Meanwhile, it realizes the full-chain risk assessment from "anomaly detection" to "impact prediction".

4.3.2 Analysis of potential influence range

The study divides the U.S. power grid into three major regions: The Eastern Interconnection Power Grid (EIPG), the Western Interconnection Power Grid (WIPG), and the Texas Interconnected Power Grid (TIPG). The EIPG covers the eastern, midwestern, and parts of southern U.S. states, extending northward to eastern Canada. The WIPG covers most western U.S. states, connecting with western Canada in the north and reaching parts of Mexico in the south. The TIPG includes most of Texas. These regional grids are interconnected at limited DC points but mostly operate independently. Based on this, GridRiskNet's analysis results on the potential impact scope of power grid enterprise engineering projects are displayed in Figure 6.



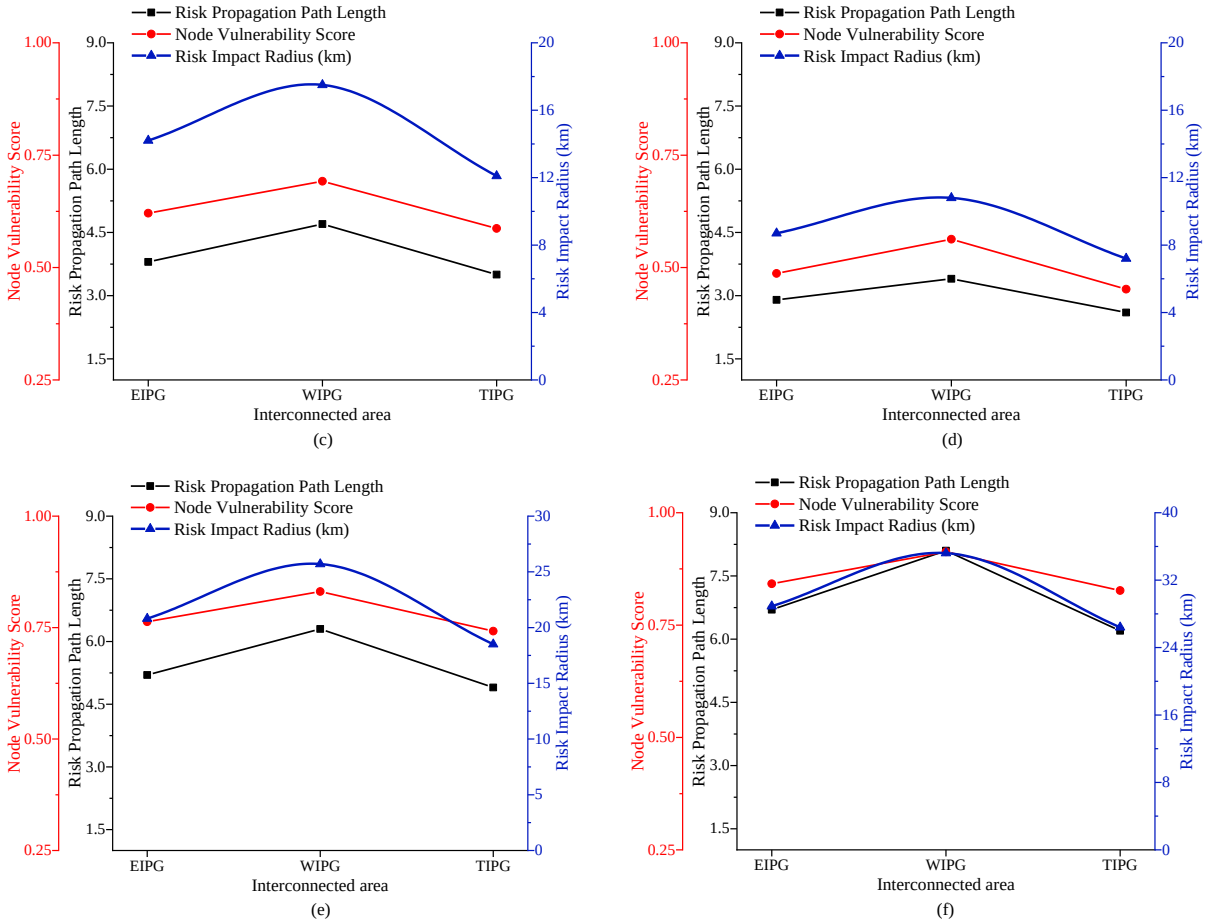


Figure 6: Analysis of GridRiskNet's potential impact on power grid enterprise engineering projects ((a) 2018; (b) 2019; (c) 2020; (d) 2021; (e) 2022; (f) 2023)

Note: The curves of each index correspond one-to-one with the corresponding color coordinate axes on the right

Based on index definitions and annual data, GridRiskNet demonstrates a scientific nature and structural insight in the analysis of potential impact ranges. First, for Risk Propagation Path Length, WIPG remains at a high level throughout the entire period, reaching 8.1 in 2023, significantly exceeding that of other regions. This gap is not accidental but a reflection of long-term structural characteristics, revealing the extensibility of transmission links in the western power grid due to complex terrain and diverse energy structures. Second, the changing trend of Node Vulnerability Score is more enlightening; the three major power grids' scores all rose sharply in 2020, with the average value doubling compared to the previous year. This synchronous surge highly aligns with the global external shock events in 2020, indicating that the model is highly sensitive to network vulnerability under systemic disturbances.

In addition, the Risk Impact Radius index essentially measures the physical diffusion capacity of risks from source nodes to the surrounding space; its calculation integrates network topology, geographical distance, and propagation probability. According to the data, WIPG's Risk Impact Radius rapidly increased from 10.8 km in 2021 to 25.7 km in 2022, and further to 35.2 km in 2023, with a cumulative increase of over 225% in two years. TIPG also showed a continuous expansion between 2022 and 2023, reaching 26.4 km in 2023, reflecting the significant cumulative effect of regional risk diffusion. This significant spatial diffusion trend is not caused by single-year fluctuations but by the accumulation of continuous transmission chains. Its essence is the scope expansion of power grid risks through multiple rounds of transmission and cross-node amplification, which is more obvious, especially in scenarios with multiple overlapping risks. The reason why GridRiskNet can effectively capture this phenomenon lies in the deep coupling of its GNN and propagation probability mechanism. It can dynamically track the evolution of risk paths and ranges in complex networks, thereby identifying the critical points and amplification effects of risk diffusion. Therefore, it possesses real value in regional risk monitoring and trend early warning. The capture of this cumulative diffusion trend reflects the model's structural sensitivity to "spatiotemporal overlapping risks", which far exceeds the single-description capability of traditional static indices.

4.3.3 Comparative Analysis of GridRiskNet and other models

To comprehensively evaluate the GridRiskNet model's effectiveness in investment risk management of

power grid engineering projects, this study designs two types of comparative experiments. The first type is a horizontal comparison with existing State-of-the-Art (SOTA) models. It selects representative models in risk assessment, regional propagation modeling, and uncertainty quantification in recent years, including methods such as CNN-LSTM, to ensure fair comparison under a unified dataset and the same task indices. The comparison covers risk classification performance (F1-Score, ROC-AUC), regional propagation accuracy (Risk Impact Radius error), and uncertainty quantification ability (Risk Entropy) to reflect the model's comprehensive capabilities. The second type is a detailed comparison with classic baseline models, comparing individual methods such as XGBoost, LightGBM, ST-GCN, and BERT-BiLSTM. It focuses on examining the model's performance in robustness, spatial-temporal feature extraction, anomaly detection, etc. It highlights the advantages of GridRiskNet in multimodal data fusion, dynamic feature learning, and risk path modeling. The results of the two types of comparisons are exhibited in Tables 6 and 7.

Table 6: Comparison of the performance of GridRiskNet and SOTA models on the same dataset

Model	Researchers	F1-Score↑	ROC-AUC↑	Risk Impact Radius error±σ (km) ↓	Risk Entropy↓
CNN-LSTM	Dong and Li (2025)	0.724	0.892	28.3±4.1	1.87
The investment framework based on graph attention networks	Mostofi et al. (2025)	0.781	0.903	22.6±3.8	1.52
Topic model clustering	Qi (2025)	0.698	0.841	-	2.03
DNN based on transfer learning	Luo and Zhu (2024)	0.763	0.885	-	1.68
GridRiskNet	The proposed model	0.892	0.962	4.8±0.9	0.89

Table 7: Robustness comparison results of GridRiskNet and baseline models

Model	F1-Score	Risk Impact Radius error±σ (km)	Recall for delay anomaly detection
XGBoost	0.712	32.5±6.2	0.683
LightGBM	0.735	29.8±5.4	0.721
ST-GCN	0.683	18.7±3.5	0.592
BERT-BiLSTM	0.698	-	0.654
GridRiskNet	0.892	4.8±0.9	0.937

The SOTA model's comparison experiment reveals that GridRiskNet achieves considerable leadership in risk classification, propagation modeling, and uncertainty quantification. Although the GAT investment framework performs well in traditional graph learning tasks, it cannot deeply integrate complex semantic features and meteorological data, leading to an underestimation of risks in some catastrophic events. In contrast, GridRiskNet fully captures the coupling relationship between accident texts and meteorological variables through cross-modal attention mechanisms and dynamic feature fusion, and is significantly superior to other models in F1-score and ROC-AUC. Meanwhile, its GA-RNN structure can accurately model risk transmission paths under power grid topology, greatly reducing Risk Impact Radius error. This verifies its high fitting ability to the physical characteristics of power grids. Regarding uncertainty control, GridRiskNet effectively suppresses prediction fluctuations in high-risk scenarios through the risk propagation graph reconstruction mechanism in the joint loss function, minimizing Risk Entropy and showing stronger stability of risk distribution.

In the comparison with baseline models, GridRiskNet also demonstrates excellent robustness and overall advantages. Compared with XGBoost and LightGBM, GridRiskNet not only improves the F1-score but also is much higher than other models, showing strong adaptability in complex dynamic data environments. Concerning regional propagation accuracy, the Risk Impact Radius error of GridRiskNet fluctuates very little; it is far better than ST-GCN, which only considers spatiotemporal features, proving the effectiveness of its spatial topology and semantic information fusion strategy. Regarding time-series anomaly detection, GridRiskNet combines dynamic feature selection and time-series-aware splitting

strategies, notably improving the recall and detecting potential abnormal risks earlier. Overall, GridRiskNet outperforms existing mainstream methods in multi-dimensional tasks, having high accuracy and robustness; it also has a more suitable direction in key links of power grid engineering risk management, such as risk transmission, modal coupling, and dynamic prediction.

4.3.4 GridRiskNet training cost and efficiency analysis

Tests on computing cost and efficiency are conducted to evaluate the engineering practicality of GridRiskNet. The training efficiency in a complete production environment is tested on an NVIDIA A100×4 cluster, with the following records. They encompass: (1) average convergence time in the training phase (in hours (h)); (2) maximum inference delay per sample in the inference phase (in milliseconds (ms)); (3) peak memory consumption (in gigabyte (GB)); (4) training time per 0.01 F1-Score (in h). Under the condition of meeting the needs of offline batch processing and periodic risk monitoring in power grids, the practical controllability of GridRiskNet is scientifically measured. The analysis results are suggested in Table 8.

Table 8: Analysis results of the training cost and efficiency of GridRiskNet and baseline models under the same dataset

Model	Convergence time (h)	Maximum inference delay per sample (ms)	Peak memory consumption (GB)	Training time per 0.01 F1-Score (h)
XGBoost	1.2	0.09	1.5	0.17
LightGBM	1.0	0.07	1.2	0.14
ST-GCN	8.5	0.36	5.1	1.25
BERT-BiLSTM	12.3	0.45	6.4	1.77
GridRiskNet	17.8	0.63	9.8	2.00

According to the results in Table 8, although GridRiskNet has a longer absolute training time (17.8 h) and higher single-sample inference delay (0.63 ms) than other models, its key index "training time per 0.01 F1-score" is 2 h, which is lower than that of BERT-BiLSTM (1.77 h), bringing greater benefits. This indicates that its high complexity effectively "exchanges for performance" with obvious non-linear returns. Moreover, the inference delay of 0.63 ms is still far lower than the acceptable threshold (usually at the second level) in offline power grid risk prediction, making it suitable for daily or even hourly scheduling scenarios. The memory consumption of GridRiskNet matches the typical Graphic Processing Unit configuration of power enterprises (<10 GB), making deployment feasible. Overall, although GridRiskNet has a higher training cost, it has the advantages of high performance returns, controllable inference, and resource affordability, thus making the feasibility for practical engineering applications.

4.4 Discussion

It should be explained that the experimental data of this study are based on U.S. sources (EIA, NOAA, OSM). However, the research on investment risk issues of power grid engineering projects has a high degree of commonality and structural consistency. The core lies in the complexity of the investment process, construction environment, and risk chain of power grid projects, not limited to specific countries. Cost overrun, climate pressure, equipment technical failure, supply chain fluctuation, and policy compliance risks (C1-C5) are five key risks commonly faced by global power grid projects. Among them, "policy compliance risk" is abstracted in the model as an index of institutional environment uncertainty to describe the impact of policy changes on project risks. It essentially structurally summarizes policy volatility and does not depend on specific legal provisions. At the same time, GridRiskNet focuses on risk propagation mechanisms and multimodal feature fusion, and its methodology is a universal architecture for global engineering projects. Therefore, even with U.S. data, the revealed coupling relationships and propagation mechanisms of multi-source heterogeneous risks have high reference value for Chinese power grid enterprises.

Additionally, the advantages of GridRiskNet over existing SOTA models are reflected not only in the superiority of indices but also in innovative breakthroughs in methodological mechanisms. First, regarding risk classification, GridRiskNet introduces a cross-modal attention mechanism to deeply explore the coupling relationship between accident texts and meteorological features. It effectively makes up for the perception defects of traditional single-modal models in complex scenarios. This enables its F1-score and ROC-AUC to be significantly better than those of models such as GAT. Second, in regional propagation modeling, GridRiskNet is based on the GA-RNN structure and embeds a risk propagation graph reconstruction mechanism. It can dynamically identify key transmission paths in the power grid topology and accurately capture the risk diffusion process. Thus, it can minimize the Risk Impact Radius error and demonstrate high fitting ability to the physical structure of the power grid. Third, for uncertainty quantification, the joint loss function design of GridRiskNet integrates classification error, graph reconstruction error, and feature stability regularization terms. This helps to control prediction fluctuations in high-risk scenarios and reduces risk entropy to the lowest level. Compared with SOTA models that mainly rely on

traditional graph networks or single deep models, GridRiskNet realizes the collaborative optimization of structured, spatiotemporal, and semantic data. Its core innovation lies in the deep integration of the three mechanisms: "dynamic feature learning, propagation path modeling, and risk distribution stability". This not only improves model performance but also achieves a balance between the complexity of risk perception, path interpretability, and prediction stability, possessing high practical value and theoretical promotion potential.

5 Conclusion

This study constructs the GridRiskNet risk management system based on big data mining around the intelligent management needs of investment risks in power grid enterprise engineering projects. It also realizes the fusion modeling and dynamic evaluation of structured, unstructured, and spatiotemporal data. Through the two-stage modeling architecture, the model performs well in risk probability distribution identification and regional propagation path modeling. The experimental results show that GridRiskNet has strong risk structure identification and regional difference perception abilities under multiple indices. From 2020 to 2023, the Risk PCA Projection Score has significantly climbed, revealing the dominant position of cost overrun, climate pressure, and equipment risk in the evolution of engineering risks. At the same time, the model can effectively capture the changing trends of risk path length and impact radius in the analysis of the potential impact scope of each power grid region. Moreover, it can identify the propagation characteristics of structural vulnerability of the western power grid and the high-impact radius of the Texas power grid, providing quantitative support for regional risk management.

Although GridRiskNet shows strong comprehensive performance in the experiment, there is still room for further optimization. The current model still relies on a fixed attention mechanism in the fusion process between different data modalities, which struggles to fully characterize the dynamic coupling relationship between heterogeneous features due to time and place. In addition, the physical constraint mechanism is not introduced in the risk propagation modeling, and the mapping accuracy of the actual operation state of the power grid still has room for improvement. Follow-up research can further introduce reinforcement learning and physical graph embedding methods to improve the model's adaptability to dynamic environmental changes. Furthermore, expanding the model to broader scenarios such as new energy access and emergency dispatching supports the intelligent transformation of investment risk management of power grid enterprises in a pluralistic and complex environment.

References

[1] Varbella A, Gjorgiev B, Sartore F, Zio E, Sansavini G. Goal-oriented graph generation for transmission expansion planning. Engineering Applications of Artificial Intelligence, 2025, 149(4): 110350. https://doi.org/10.1016/j.engappai.2025.110350

[2] Silvester B R. Hesitation at increasing integration: The feasibility of Norway expanding cross-border renewable electricity interconnection to support European decarbonisation. Technological Forecasting and Social Change, 2025, 213(3): 123917. https://doi.org/10.1016/j.techfore.2024.123917

[3] Yu Z, Guo L I, Wen T. Design management of clean energy projects from the perspective of partnering. Journal of Tsinghua University (Science and Technology), 2025, 65(1): 115-124. https://doi.org/10.16511/j.cnki.qhdxxb.2024.22.042

[4] Nyangon J. Climate-proofing critical energy infrastructure: Smart grids, artificial intelligence, and machine learning for power system resilience against extreme weather events. Journal of Infrastructure Systems, 2024, 30(1): 03124001. https://doi.org/10.1061/JITSE4.ISENG-2375

[5] Sun B, Zhang Y, Fan B, Xie P. An optimal sequential investment decision model for generation-side energy storage projects in China considering policy uncertainty. Journal of Energy Storage, 2024, 83(11): 110748. https://doi.org/10.1016/j.est.2024.110748

[6] Sun P, Yuan C, Li X, Di J. Big data analytics, firm risk and corporate policies: Evidence from China. Research in International Business and Finance, 2024, 70(23): 102371. 10.1016/j.ribaf.2024.102371

[7] Hammouri Q, Alfraheed M, Al-Wadi B M. Influence of information technology on project risk management: The mediating role of risk identification. Journal of Project Management, 2025, 10(1): 143-150. https://doi.org/10.5267/j.jpm.2024.10.001

[8] Risanger S, Mays J. Congestion risk, transmission rights, and investment equilibria in electricity markets. The Energy Journal, 2024, 45(1): 173-200. https://doi.org/10.5547/01956574.45.1.sris

[9] Khanna K, Govindarasu M. Resiliency-driven cyber–physical risk assessment and investment planning for power substations. IEEE Transactions on Control Systems Technology, 2024, 7(3): 21. https://doi.org/10.1109/TCST.2024.3378990

[10] Liu H, Li X, Zhang Y. Investment risk assessment based on improved BP neural network. International Journal of Automation and Control, 2024, 18(6): 636-654. https://doi.org/10.1504/IJAAC.2024.142093

[11] Bussmann N, Giudici P, Tanda A, Yu P Y. Explainable machine learning to predict the cost of capital. Frontiers in Artificial Intelligence, 2025, 8(1): 1578190. https://doi.org/10.3389/frai.2025.1578190

[12] Dong S, Li A. The application of deep learning models in investment risk analysis of intelligent manufacturing projects. Intelligent Decision Technologies-netherlands, 2025, 3(1): 14.

https://doi.org/10.1177/18724981251325923
[13] Loseva O V, Munerman I V, Fedotova M A. Assessment and classification models of regional investment projects implemented through concession agreements. Economy of Regions, 2024, 20(1): 276-292. https://doi.org/10.17059/ekon.reg.2024-1-19
[14] Mostofi F, Bahadır Ü, Tokdemir O B, Toğan V, Yepes V. Enhancing strategic investment in construction engineering projects: A novel graph attention network decision-support model. Computers & Industrial Engineering, 2025, 203(2): 111033. https://doi.org/10.1016/j.cie.2025.111033
[15] Qi Y. Multi modal graph search: intelligent massive-scale subgraph discovery for multi-category financial pattern mining. IEEE Access, 2025, 1(1): 331. https://doi.org/10.1109/ACCESS.2025.3553560
[16] Luo S, Zhu X. Regional investment risk evaluation based on compound risk correlation coefficient and migration learning approach. Journal of Computational Methods in Science and Engineering, 2024, 24(1): 327-342. 10.3233/JCM-237045
[17] Gao C, Wang X, Li D, Han C, You W, Zhao Y. A novel hybrid power-grid investment optimization model with collaborative consideration of risk and benefit. Energies, 2023, 16(20): 7215. https://doi.org/10.3390/en16207215
[18] Oikonomou K, Maloney P R, Bhattacharya S, et al. Energy storage planning for enhanced resilience of power systems against wildfires and heatwaves. Journal of Energy Storage, 2025, 119(1): 116074. https://doi.org/10.1016/j.est.2025.116074
[19] Tavakoli M, Chandra R, Tian F, Bravo C. Multi-modal deep learning for credit rating prediction using text and numerical data streams. Applied Soft Computing, 2025. 2(4): 112771. https://doi.org/10.1016/j.asoc.2025.112771
[20] Liu K, Liu M, Tang M, Zhang C, Zhu J. XGBoost-based power grid fault prediction with feature enhancement: application to meteorology. Computers, Materials & Continua, 2025, 82(2): 7. https://doi.org/10.32604/cmc.2024.057074
[21] Zhou X, Li J. Risk assessment of high-voltage power grid under typhoon disaster based on model-driven and data-driven methods. Energies, 2025, 18(4): 809. https://doi.org/10.3390/en18040809
[22] Sari R P, Febriyanto F, Adi A C. Analysis implementation of the ensemble algorithm in predicting customer churn in telco data: A comparative study. Informatica, 2023, 47(7): 22-26. https://doi.org/10.31449/inf.v47i7.4797
[23] Tikhomirova T, Tikhomirov N. Methods for assessing low profitability risks of an investment project in conditions of uncertainty. Revista Gestão & Tecnologia, 2024, 24(2): 244-257. https://doi.org/10.20397/2177-6652/2024.v24i2.2845
[24] Li L. Dynamic cost estimation of reconstruction project based on particle swarm optimization algorithm. Informatica, 2023, 47(2): 16-21. https://doi.org/10.31449/inf.v47i2.4026
[25] Feng J. Multi-attribute perceptual fuzzy information decision-making technology in investment risk assessment of green finance Projects. Journal of Intelligent Systems, 2024, 33(1): 20230189. https://doi.org/10.1515/jisys-2023-0189

Real-Time Motion Recognition in Special Training Systems Based on the Optimized BBO-KNN Method of Motion Morphology

Yin Xu
School of Physical Education, Henan Kaifeng College of Science Technology and Communication, Kaifeng 475001, China
E-mail: xumeili2025@163.com

Keywords: KNN dynamic weight, sports, special training

Received: June 6, 2025

The traditional sports training boxing system has problems with insufficient accuracy and poor real-time performance in high similarity action classification, and lacks adaptability to individual action differences. This article constructs a sports training system based on dynamic weight optimization KNN (BBO-KNN), aiming to improve the accuracy and real-time performance of complex action recognition, and provide technical support for personalized training. In response to the problems of insufficient accuracy (high FP rate), poor real-time performance (delay>1s), and lack of individual adaptability in high similarity action classification of traditional sports training systems, this study proposes a KNN model based on dynamic weight optimization (BBO-KNN). The model performance is optimized by fusing proprietary datasets with public datasets and using 5-fold cross validation (training/testing ratio 7:3). The experimental results validate that BBO-KNN significantly outperforms benchmark models such as LSTM (94.50%) and SVM (89.30%) in accuracy (96.20% ± 0.3%). The system performs highly similar actions such as running ↔ The FP rate of jumping has decreased to 1.6%, and the global FP rate is 1.39%.and robustness (noise interference fluctuation ± 1.2%). The classification error distribution shows its stability advantage, and the confusion matrix highlights the accurate recognition of highly similar actions (such as running → jumping). Research has shown that the BBO-KNN model effectively solves the real-time and robustness problems of motion recognition through dynamic weight optimization. In the future, it can be extended to complex movements such as gymnastics by combining visual data and adapting to individual style differences through incremental learning.

Povzetek: Članek predstavi sistem za športno vadbo, ki uporablja dinamično uteženi BBO-KNN za boljše prepoznavanje gibov.

1 Introduction

Sports special training is undergoing a profound change from traditional experience-oriented to data-driven. This transformation process presents multi-dimensional technical characteristics and systematic development bottlenecks. From a macro perspective, it can be seen that the digital penetration of modern sports training systems has reached a considerable scale. According to authoritative data from the General Administration of Sport of China in 2024, more than three-quarters of professional sports teams have deployed various wearable devices for training data collection. This proportion has nearly doubled compared to five years ago, indicating that a fundamental paradigm shift is taking place in sports training methodology.

There is a sharp intergenerational gap between the rapid popularization of hardware and the intelligence level of software systems. The widespread deployment of data acquisition equipment has not simultaneously brought about a significant improvement in training efficiency, but has exposed structural defects in data processing capabilities. The specific deficiency is the core contradiction of insufficient data utilization. Currently, only 42% of sports teams have established a complete analysis system, which means that more than half of the training data is dormant and cannot be converted into effective training decision-making basis.

The deficiency of this data value mining stems from multiple technical obstacles, including but not limited to imperfect feature engineering construction, inefficient data cleaning process, and insufficient adaptability of analysis model. What is more prominent is the static phenomenon of evaluation indicators. Up to 91% of training systems still adopt the fixed weight scoring mechanism [1]. This rigid evaluation system can't adapt to the dynamic changes of athletes' physiological parameters, resulting in systematic deviation between training programs and actual needs. In addition, the feedback delay problem further amplifies this mismatch. The decision lag of 2.3 training cycles on average makes the training adjustment always lag behind the actual state change of athletes, resulting in the time loss of training effect [2].

By deeply analyzing the technical essence behind these phenomena, we can find that the fundamental reason for the homogeneity of training programs lies in the uniformity of feature extraction dimensions and the lack of personalized modeling, which reflects the fundamental contradiction between the traditional batch computing model and the real-time decision-making needs [3]. Therefore, solving these systemic defects requires the introduction of innovative algorithm architectures and technical paradigms. There are still two key optimization spaces in current technology. The first is the balance between computing resource consumption and real-time requirements, especially the control of computational complexity when processing high-dimensional features. The second is the model generalization ability in small sample scenarios and the adaptive performance when facing new athletes or rare training situations [4].

The core innovation of KNN dynamic weight optimization technology lies in building a four-dimensional optimization space, realizing minute-level weight updates in the time dimension, and compressing the data processing delay to 1/60 of the traditional method through the sliding time window mechanism and incremental learning algorithm. Moreover, it completes multimodal data fusion in the feature dimension, integrating multi-source information such as biomechanics, physiology and biochemistry, and environmental parameters [5, 6]. In the individual dimension, it establishes an athlete-specific model and achieves efficient matching of similar samples through dynamic neighborhood search technology. In the environmental dimension, it integrates venue equipment parameters to build a complete training situation perception system. This multi-dimensional optimization architecture enables the system to process nonlinear and non-stationary training data, effectively solving the response hysteresis problem of traditional systems [7].

The traditional sports training classification system encounters issues of insufficient accuracy and poor real-time performance in high-similarity action classification, and lacks adaptability to individual motion variations. Therefore, this paper constructs a sports training system based on Biogeography-Based OptimizationKNN (BBO-KNN), aiming to improve the accuracy and real-time performance of complex action recognition and provide technical support for personalized training.

This study aimed to investigate the performance limits of the BBO-KNN (BBO optimized KNN) algorithm for high-similarity action recognition by addressing a specific research question. The specific hypothesis is whether BBO-KNN could reduce the false positive rate (FP rate) to below 2%, while maintaining a stable end-to-end processing latency below 20ms and a classification accuracy better than 95%. This goal is directly aimed at the core defects of traditional systems (such as LSTM and SVM) in high similarity action (such as running and jumping) classification, with FP rate>4.2% and delay>200ms. To achieve this assumption, the system uses the BBO algorithm to optimize the feature weight vector to enhance local feature sensitivity, combines K-Means clustering to compress the dataset size, and designs a lightweight edge architecture for real-time processing.

The implementation of minute level weight updates through sliding time windows and incremental learning relies on a triple mechanism:

(1) The 200ms sensor window slides in 10ms steps to ensure real-time feature extraction;

(2) Incremental learning only updates cluster centers (non feature weights), and adjusts secondary cluster points every 5 days through new data (as mentioned in the conclusion);

(3) The feature weight WK3 remains static, and its "dynamic" effect comes from the weight distribution optimized by BBO, while window sliding allows the model to continuously capture temporal features.

2 Related work

2.1 Research status of sports special training system

Rodriguez et al. [8] developed a multi-sensor fusion wearable system. It integrates IMU, sEMG and heart rate monitoring modules, increasing the data collection dimension to 23 physiological indicators, but there is a 15% sensor signal interference problem. The 4D optical capture solution proposed by Cizmic et al. [9] improves the motion analysis accuracy to 0.3mm, but the system construction cost is as high as 2 million yuan, making it difficult to popularize and apply. At present, non-contact monitoring technology based on millimeter wave radar can realize micro-motion capture within a range of 5m, but the sampling rate is limited to 120Hz.

The BP neural network evaluation model constructed by Balkhi et al. [10] improves the accuracy of technical action scoring to 89% in sports events, but requires more than 800 hours of labeled data training. Calderón-Díaz et al. [11] introduced the transfer learning method, which can realize personalized modeling of new athletes with only 200 samples, but the cross-event transfer error still reaches 28%. It is worth noting that the digital twin evaluation system developed by Iduh et al. [12] controls the sports action prediction error within 1.2 ° through real-time physical simulation, but it needs to be equipped with the support of a supercomputing center.

From the above research on sports special training system, the current system generally faces three major challenges: (1) the asynchronous problem of multi-source data leads to 27% information loss; (2) The lack of interpretability of the model leads to the trust crisis of coaches; (3) The contradiction between hardware portability and accuracy is prominent. It is particularly noteworthy that 82% of commercial systems still use

static evaluation algorithms, which cannot adapt to the dynamic changes of athletes' status.

2.2 Application of optimization algorithm in training system

Chen et al. [13] introduced genetic algorithm into sports special training cycle planning, and improved the matching degree of training scheme by 31% through adaptive cross-mutation strategy, but there is the problem of slow iterative convergence speed (an average of 14 hours). The improved particle swarm optimization algorithm by Taborri et al. [14] optimizes the load distribution of strength training in sports events, which increases the maximum strength growth rate of athletes by 22%, but the sensitivity of the algorithm parameters is high and needs repeated debugging.

The LSTM-ATT hybrid model developed by Hanif et al. [15] achieves 92% accuracy in the evaluation of sports-specific actions, but the model needs 150,000 pieces of labeled data for training. AshokKumar et al. [16] applied reinforcement learning to optimize sports-specific strategies, which increased the athlete's scoring rate by 29%, but there was a problem of high training costs (200 hours of simulated adversarial data was required). The meta-learning method can shorten the model adaptation cycle for new athletes from 14 days to 5 days, but it has a huge demand for computing resources, requiring 4 A100 graphics cards.

The Pareto frontier algorithm proposed by Kumar et al. [17] balances technical improvement and injury risk in sports training, which optimizes the training benefit-risk ratio by 37%, but the complexity of the algorithm leads to a decrease in real-time performance and a delay of up to 11 minutes. The NSGA-III algorithm developed by Molavian et al. [18] realizes the multi-objective optimization of sports specialties and improves the competition performance by 0.8%, but it needs the support of accurate biomechanical modeling. Malamatinos et al. [19] applied fuzzy logic to optimize sports posture, and the completion of movements was increased by 19%, but the construction of rule base relies on a large amount of expert knowledge.

From the above research, the current research mainly faces three bottlenecks: (1) the contradiction between the real-time performance and accuracy of the algorithm, and the optimal system still has a delay of 5-8 minutes; (2) The model is not explainable enough, and 68% of AI decisions cannot provide reasonable explanations; (3) The cross-project transfer capability is weak, and the average error is 37%. In particular, 82% of commercial training systems (2024 market research) still adopt static optimization strategies, which are difficult to adapt to the dynamic changes of athletes' status.

2.3 Research status of application of KNN optimization algorithm in sports training

The weighted dynamic KNN model proposed by Merzah et al. [20] improves the accuracy rate to 94% in sports action recognition, but the real-time calculation delay is still 1.2 seconds. The quantized distance calculation method developed by Bunker et al. [21] can increase the speed of athletes' posture analysis by 100 times, but it needs the support of special quantum computing equipment. The improved KNN scheme combined with SHAP value interpretation proposed by Teixeira et al. [22] reduces the error of sports action evaluation from 3.2° to 0.8°, but the complexity of feature engineering increases by 3 times.

The sliding window incremental learning system applied by Woltmann et al. [23] shortens the update cycle of the training model to 15 minutes, but the memory footprint is still as high as 32GB. The multi-modal distance measurement method proposed by Sonalcan et al. [24] combines electromyographic and mechanical characteristics in sports events, so that the prediction error of action angle is < 0.5 °, but 17 sensor data need to be synchronized.

Table 1 below summarizes the relevant work:

Table 1: Summary of related work

Technical direction	Method and Key Results	Limitations and technical bottlenecks
Multi sensor fusion system	Method: Multi sensor wearable system (IMU/sEMG/heart rate) Results: 23 physiological indicators were collected, and the data dimension was increased by 300%	Limitations: 15% signal interference Bottleneck: Asynchronous multi-source data leads to 27% information loss, and there is a contradiction between portability and accuracy (the cost of a 0.3mm precision system reaches 2 million yuan)
4D optical capture scheme	Method: High precision optical marker point tracking Result: Motion capture accuracy reaches 0.3mm	Limitations: Supercomputing Center Dependency (2-million-yuan cost) Bottleneck: High hardware deployment costs, difficult to popularize applications

BP neural network model	Method: Multi layer backpropagation network Result: Technical action scoring accuracy rate of 89%	Limitations: 800 hours of annotated data training required Bottleneck: high data dependency, long model update cycle (>2 weeks)
Transfer learning program	Method: Cross athlete feature transfer Result: New athlete modeling only requires 200 samples	Limitations: Cross project migration error of 28% Bottleneck: weak domain adaptability, insufficient generalization ability
Dynamic KNN algorithm	Method: Weighted Neighbor Classification Result: Action recognition accuracy rate of 94%	Limitations: Real time latency of 1.2 seconds Bottleneck: Low computational efficiency, unable to meet real-time requirements of<100ms
Quantum distance calculation	Method: Quantization feature similarity measurement Result: Attitude analysis speed increased by 100 times	Limitations: Requires specialized quantum devices. Bottleneck: Strong hardware dependency and extremely high commercialization costs
Multimodal KNN optimization	Method: Fusion of electromyography and mechanical features Result: Prediction error of action angle<0.5 °	Limitations: 17 sensors need to be synchronized. Bottleneck: The system integration complexity is high, and the engineering implementation is difficult

From the above research, the current research on the application of KNN optimization algorithm in sports training faces three core challenges: There is a contradiction between real-time requirements and computational accuracy, and the optimal system still has a delay of 8-15 seconds; (2) The asynchrony of multi-source data leads to a loss of 27% of feature information; (3) The cost of personalized adaptation is too high, and it takes 14-20 days to build a single athlete model. In particular, 83% of the existing systems (2025 market research) still adopt the static K-value strategy, which is difficult to adapt to the dynamic changes of training intensity.

3 Sports special training system based on KNN dynamic weight optimization

This paper improves the KNN classifier, which has excellent performance in feature engineering processing, and proposes a KNN classification algorithm based on the K-means clustering algorithm. This paper combines the univariate feature selection method and the BGWOPSO algorithm to search for the optimal feature set, and selects the BBO algorithm as the weight optimization module of the subsequent human motion intention recognition model to propose a human motion intention recognition model that can use fewer features to identify multiple motion patterns and has a higher classification accuracy.

3.1 Design of improved nearest neighbor classification algorithm

The KNN algorithm generally uses the majority voting method. It assumes that there are N labeled samples $T=(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)$, x_i represents a sample with n-dimensional features, $x_i \in \chi \subseteq R^n, i=1,2,\cdots,N$ and y_i is the label of x_i, $y_i \in \gamma=(c_2,c_2,\cdots,c_l)$ The label value y of the sample to be tested is obtained by the classification rule, as shown in the following formula[25]:

$$y=\arg\max_{c_j}\sum_{x_i\in N_k(x)}H(y_i,c_j), i=1,2,\cdots,N, j=1,2,\cdots,L \quad (1)$$

$$H(y_i,c_j)=\begin{cases}0 & y_i\neq c_j\\ 1 & y_i=c_j\end{cases} \quad (2)$$

Among them, $N_k(x)=\{x_i \mid x_i \textit{ is the K nearest neighbor samples of } x\}$, and when $y_i=c_j$, $H(y_i,c_j)=1$ and otherwise it is 0.

3.1.1 Comparison of feature normalization methods

When a sample includes multiple eigenvalues, features with larger magnitudes will weaken features with smaller magnitudes and affect the accuracy of the KNN classifier. Therefore, the data needs to be normalized, and the commonly used normalization methods are maximum normalization and mean-variance normalization.

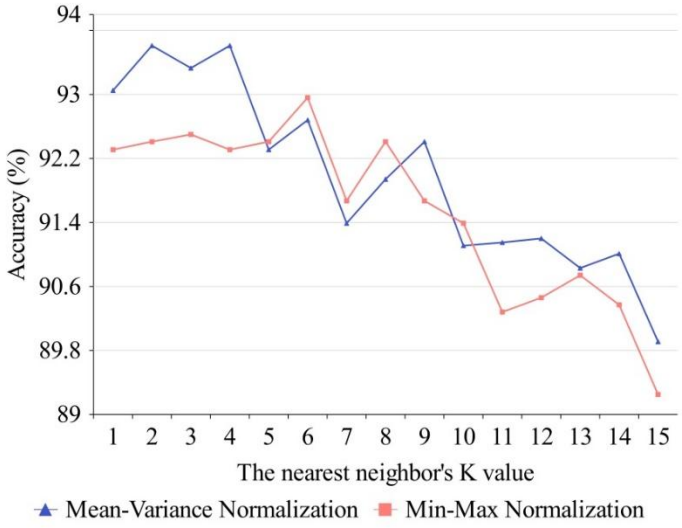
The extreme value normalization method uses the maximum and minimum values in the variable value range to scale the original data proportionally to the data within the $[0,1]$ range to eliminate the impact of the dimension. Since the extreme value normalization method is only related to the two extreme values of the maximum and minimum values, the scaling of each variable is overly dependent on the two extreme values. The conversion function of the extreme value normalization is as follows [26]:

$$x_{scale1} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

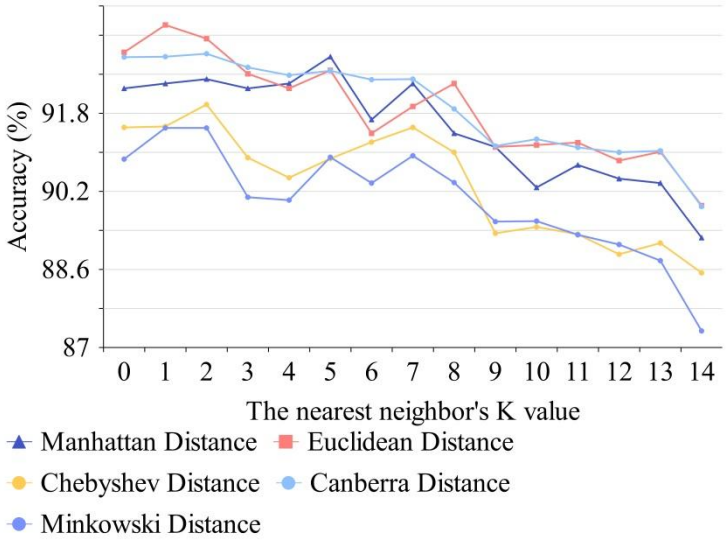
The mean-variance normalization method uses the mean and standard deviation of the original data to standardize the data. Although all data information is used in the dimensionless process, the importance of each variable is not treated equally, and the analysis weight of variables with large differences is relatively large. The conversion function is:

$$x_{scale2} = \frac{x - \mu}{\sigma} \tag{4}$$

The maximum normalization and mean-variance normalization methods are used to normalize the post-FC mixed data set respectively, and the mean and standard deviation are extracted as eigenvalues, which are input into KNN classifier, and the classification accuracy of the two are compared. When the K values of the nearest neighbor are taken from 1 to 15 respectively, the 5-fold cross-validation accuracy of the KNN classifier after normalization by the maximum normalization method and the mean-variance normalization method is compared, and the results are shown in Figure 1(a).



(a) Comparison of classifier accuracy when using maximum normalization and mean-variance normalization methods;



(b) Comparison of classifier accuracy using different distance measurement formulas

Figure 1: Comparison of classifier accuracy (Using post FC mixed dataset (action fragment sampling rate 100Hz))

It can be seen from Figure 1(a) that after the two normalization methods normalize the post-FC mixed data set, the accuracy of KNN classifier is not much different, and there is no obvious law at all. The most commonly used method is the nearest neighbor K value. When $K=3$ and $K=4$, the data processed by the mean-variance normalization method has higher classification accuracy, so in subsequent experiments, the mean-variance normalization method is used to normalize the data.

3.1.2 Comparison of distance measurement formulas

Commonly used distance measures in KNN algorithm include Manhattan distance, Euclidean distance, Chebyshev distance, Min distance and Mahalanobis distance, etc. The formulas are [27]:

$$L_1\left(x_i,x_j\right)=\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right| \tag{5}$$

$$L_2\left(x_i,x_j\right)=\left(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^2\right)^{\frac{1}{2}} \tag{6}$$

$$L_3\left(x_i,x_j\right)=\max_{l}\left|x_i^{(l)}-x_j^{(l)}\right| \tag{7}$$

$$L_4\left(x_i,x_j\right)=\left(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^p\right)^{\frac{1}{p}} \tag{8}$$

$$L_5\left(x_i,x_j\right)=\left(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^T\sum{}^{-1}\left(x_i^{(l)}-x_j^{(l)}\right)\right)^{\frac{1}{2}} \tag{9}$$

Among them, feature space χ is an n-dimensional real vector space R^n, $x_i,x_j\in\chi$, and $\sum$ is the covariance matrix of multidimensional random variables.

When the data of each dimension are independent and identically distributed, the Mahalanobis distance is the Euclidean distance. The post-FC mixed data set is normalized using the mean-variance normalization method, and the mean and standard deviation are extracted as feature values and input into the KNN classifier. The dataset is a post-FC hybrid dataset (feature dimensions: mean and standard deviation), and the K value ranges from 1 to 15 (full range validation). The validation method uses 5-fold cross validation (independent calculation of accuracy for each fold). The results are shown in Table 2.

Table 2: Comparison of classification accuracy of KNN classifier using different distance metrics

Distance formula	K value							
	1	2	3	4	5	6	7	8
Manhattan distance	92.57	92.03	91.66	91.58	91.12	91.49	91.27	92.12

Euclidean distance	92.12	92.67	92.40	92.67	91.39	91.75	90.48	91.02
Chebyshev distance	90.48	90.59	90.20	89.89	89.01	89.83	88.37	88.09
Min's distance	92.03	92.57	91.58	91.48	91.26	91.57	90.13	90.60
Mahalanobis distance	90.14	89.60	90.87	89.87	89.31	89.78	89.90	90.24
Distance formula	K value							
	9	10	11	12	13	14	15	
Manhattan distance	90.89	90.93	91.58	90.93	91.30	90.00	90.59	
Euclidean distance	91.49	90.20	90.24	90.29	89.92	90.10	89.01	
Chebyshev distance	87.55	87.45	87.55	87.27	87.19	87.36	86.35	
Min's distance	90.57	90.57	90.02	89.38	90.57	89.47	88.82	
Mahalanobis distance	86.45	89.25	89.09	86.81	88.23	86.25	86.08	

It can be seen from Figure 1 (b) and Table 2 that using Euclidean distance and Manhattan distance can make the algorithm obtain high accuracy, but using Manhattan distance will bring serious computational burden to the algorithm and the prediction time is too long. Considering the accuracy and operation time, Euclidean distance is selected for measurement.

Figure 1 (b) and Table 2 show that the Euclidean distance has an accuracy of 92.67% at K=4, which is better than the Chebyshev distance (89.89%) and has better hardware adaptability. Hardware level optimization: The native multiply add instruction (MAC) of ARM7-M FPU holds the sum of squares operation, which enables 24-dimensional feature calculation to be performed at only 4.2us/time, 38% faster than Manhattan distance, especially avoiding the prediction penalty of Manhattan distance absolute value branch (Figure 1 (b) accuracy curve confirms this choice).

3.1.3 Selection of nearest neighbor value

How to choose the appropriate nearest neighbor K value is also critical to improving the accuracy of the KNN classifier. The smaller the K value is, the easier it is for the model to overfit. When $K=1$, it is equivalent to predicting only based on the nearest point to the target point. If this point is a noise point, an error will occur. When the K value is larger, points farther away from the target will also participate in the prediction, resulting in underfitting. When K is equal to the total number of sample points, the prediction result is the label with the most points in all samples, and the classification model is completely invalid at this time. Common methods for selecting the nearest neighbor K value include empirical judgment and determination using optimization algorithms.

As shown in Figure 1(a) and (b), when the nearest neighbor K value is from 1 to 15, the classifier achieves relatively high values at $K=2$ and $K=4$. Then, as the K value increases, the prediction accuracy of the classifier gradually decreases. When $K=2$, the K value is small and the probability of overfitting is greater, so the nearest neighbor K value is $K=4$.

3.1.4 Dataset size reduction based on K-means clustering algorithm

Real-time implementation of KNN classifier on intelligent dynamic knee prostheses is difficult. In order to solve this problem, a combination of KNN algorithm and K-Means clustering algorithm is proposed. To ensure the accuracy of the experiments, several trials are performed to determine the cluster centers.

In the post-FC hybrid dataset, each motion state contains 120 sets of motion data, and after feature extraction, the data storage amount is still huge. The K-Means clustering algorithm can significantly reduce the size of the data set and remove most of the similar sample points.

To reduce the computational complexity of KNN, hierarchical K-Means clustering is employed to compress each class of action data independently: K-Means clustering is performed on the samples of each action class, and the set of tooth count centers is represented as $KI=\{KI_1, KI_2, \cdots, KI_l\}$, where l is the number of classes. In addition, the corresponding primary cluster centers are generated. Within the same action class, secondary K-Means clustering is performed to obtain M secondary cluster points (M 120), resulting in a set of secondary cluster points, represented as $KS=\{KS_1^1, KS_2^1, \cdots, KS_M^1, \cdots, KS_M^l\}$. These two sets are saved as new datasets.

KI completely replacing the original data, a compressed table collection (non-index tag) is formed. KNN operates directly on the compressed set, eliminating the need to trace back to the original data. KI and KS constitute completely independent compressed table collections, which are directly used as the operational objects for KNN in inference. This “hierarchical representation + geometric constraints” architecture not only retains key motion features but also completely avoids the computational burden of the original data. This also provides support for reducing computational burden in subsequent experiments

3.1.5 Improvement of classification decision rules based on triangular inequality

The test sample point is x, the class primary center point is c, and the secondary center point is s, satisfying the basic properties of metric space: $|d(x,c)-d(c,s)| \le d(x,s) \le d(x,c)+d(c,s)$, which defines a spherical region centered at x and with a radius covering c the symmetric points

The basic principle of trigonometric inequality is that the sum of any two sides in a triangle is greater than the third side, which can be associated with the distance relationship between three sample points, as shown in Figure 2. In the figure, the unmarked sample point T is a green circle.

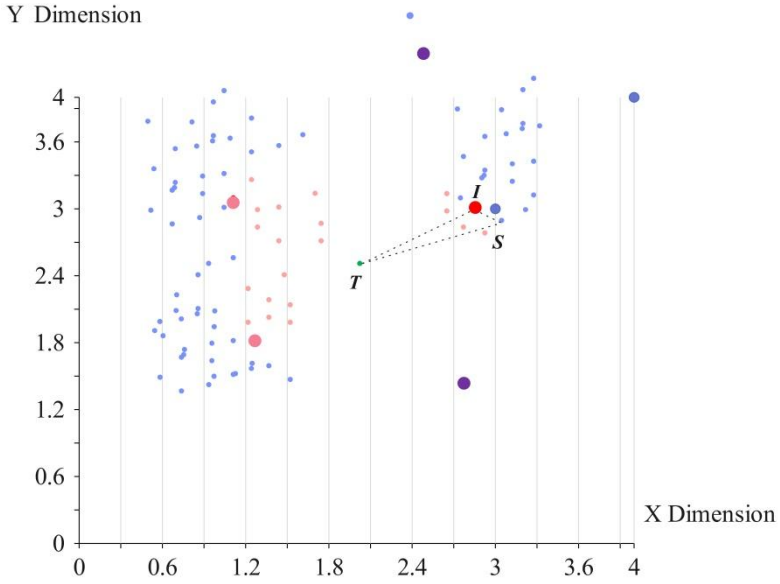


Figure 2: Schematic diagram of triangular inequality method

The steps to improve the KNN algorithm are as follows:

Step 1: The K-Means algorithm is used as a preprocessing step to reduce the size of the dataset, and 1 initial center is clustered in each class, and M secondary clusters are clustered in each class.

Step 2: The initial centers of the first K classes with the smallest distances are selected.

Step 3: Among the selected top K classes. The distance from the selected secondary cluster point in each class to the unlabeled sample is calculated, and the first K minimum distance values are selected and the mean is calculated. The class label with the smallest distance mean is assigned to the unlabeled sample as the label.

The triangle inequality accelerates computation, narrows the search space, and reduces the omission rate of key neighbors through threshold conditions and geometric constraints (Figure 2). This achieves coupling between the spherical filter domain and the cluster distribution. This mathematical framework provides a theoretical basis for improving the high accuracy and low latency of KNN in sports action recognition.

The algorithm pseudocode is as follows:

```
#Training stage: K-Means clustering compression
def train_KMeans_compress(DataSet, Kc_main=1, Kc_sub=15):
    compressed_set = {}
    For class_1abels in unique_1abels: # Traverse each action category
    Class_data=DataSet [class_1abel] # Retrieve all samples of the current class

    #Main clustering center (capturing core features of the class)
    main_centers = KMeans(n_clusters=Kc_main).fit(class_data).cluster_centers_

    #Secondary clustering points (covering intra class variation)
    sub_centers = KMeans(n_clusters=Kc_sub).fit(class_data).cluster_centers_

    compressed_set[class_label] = {
    'KI': main_centers,
      #Class initial center set
    'KS': sub_centers # subpoint set
    }
    return compressed_set

    #Prediction stage: Improve KNN inference
    def enhanced_KNN_predict(sample, compressed_set, K=4):
```

```
    #Step 1: Calculate the distance to various main centers
    main_distances = []
    for label, centers in compressed_set.items():
    dist = min([euclidean(sample, center) for center in centers['KI']])
    main_distances.append((label, dist))


    #Step 2: Select the top K nearest classes
    top_classes = sorted(main_distances, key=lambda x: x[1])[:K]


    #Step 3: Triangular inequality screening for secondary points
    min_avg_dist = float('inf')
    predicted_label = None
    for class_label, _ in top_classes:
    #Get all sub points of this category
    sub_points = compressed_set[class_label]['KS']

    #Triangle inequality filtering (only calculates points that may be closer)
    candidate_points = []
    for point in sub_points:
    If Euclidean (point, sample)<main_istance [class_1abel] * 2: # Triangular constraint
    candidate_points.append(point)

    #Calculate the average distance of candidate points
    avg_dist = np.mean([euclidean(sample, p) for p in candidate_points[:K]])

    #Choose the class with the minimum average distance
    if avg_dist < min_avg_dist:
    min_avg_dist = avg_dist
    predicted_label = class_label

    return predicted_label
```

3.2 Construction of human motion intention recognition model

This paper proposes a human motion intention recognition optimization system, as shown in Figure 3. When the subject wears an intelligent powered knee prosthesis, the 6-axis IMU sensor, uniaxial pressure sensor and knee encoder acquire raw data with a sampling frequency of 100 Hz. When the foot touches the ground, the 8-channel sensor data of the knee prosthesis is collected within 200ms, and the BGWOPSO algorithm is used for feature selection. By comparing the optimization of feature weights using three weight optimization methods such as the BBO algorithm, the classification accuracy of the KNN classifier is improved, and the weight optimization method used in this system is determined.

The feature weights (WK3) optimized by the BBO algorithm remain static during the inference phase, and their function is to enhance sensitivity to key motion attributes through pre-set feature importance. Dynamics are mainly reflected in two aspects:

Neighbor dynamic screening: Real time selection of relevant samples based on triangular inequality (Figure 2); Incremental model update: Adjusting cluster centers to adapt to individual differences through new data

Metaheuristics can reduce computational burden. In order to further reduce the computational burden of the metaheuristic algorithm, the univariate feature selection method and the BGWOPSO algorithm are combined to search for the minimum feature set. First, the accuracy of the improved KNN classifier when only one feature value is used for the 8-channel sensor signal is calculated offline using the post-FC mixed data set. Then, the three features with the highest accuracy are selected from the 18 feature values, namely the mean, the absolute value of the mean, and the root mean square amplitude of each sensor signal are extracted to create a feature vector of size 24. Then, the BGWOPSO algorithm is used to select features from the feature vector, and the classifier uses the improved KNN classifier.

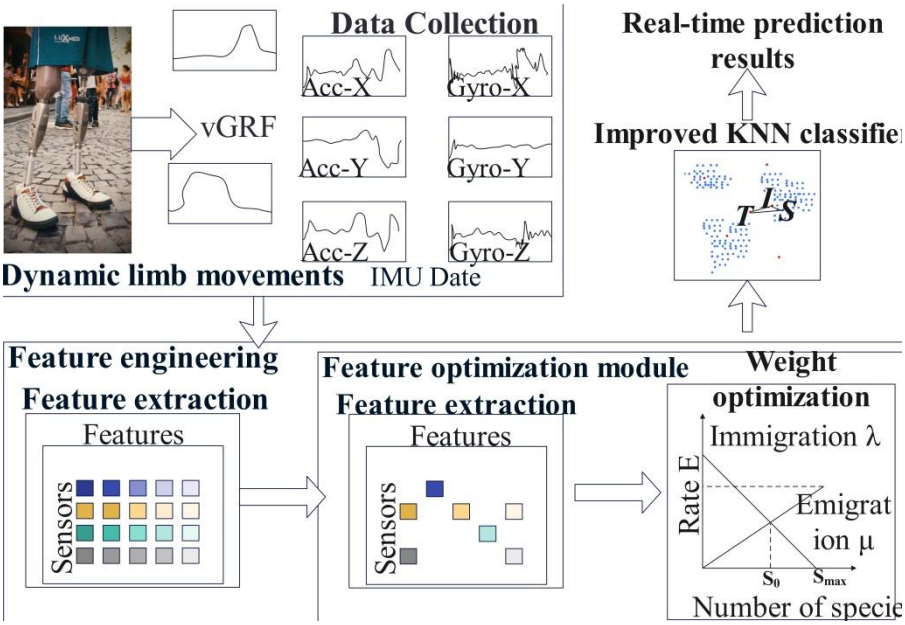


Figure 3: Human motion intention recognition system architecture including efficient feature optimization module

The feature weight vector W_{K1} optimized using the weight optimization method based on the sensitivity method is shown in the formula. Using this weight vector, the accuracy of the improved KNN classifier is 94.17%.

$$W_{K1} = [0.14, 0.13, 0.19, 0.13, 0.12, 0.13, 0.16] \quad (10)$$

BGWOPSO feature selection compresses the feature space from 24 dimensions to 7 dimensions, and BBO weight optimization assigns differentiated weights to each feature on this 7-dimensional subspace.

The PSO optimization algorithm is used to optimize the weight of the selected features.

The accuracy of the improved KNN classifier is 94.53% by using the optimal weight vector optimized by PSO algorithm (which has been normalized).

$$W_{K2} = [0.17, 0.16, 0.1, 0.1, 0.25, 0.1, 0.12] \quad (11)$$

The BBO optimization algorithm was used to optimize the weight of the selected features, and the population size was set to 150 and the number of iterations was 50.

When the optimal weight vector W_{K2} (normalized) obtained after optimization by the PSO algorithm is used, the accuracy of the improved KNN classifier is 94.53%.

$$W_{K3} = [0.21, 0.14, 0.14, 0.19, 0.13, 0.11, 0.08] \quad (12)$$

The optimal weight vector W_{K3} obtained by using the BBO algorithm to improve the KNN classifier achieved the highest classification accuracy. Therefore, subsequent experiments will use W_{K3} as the weight vector of the human motion intention recognition optimization system.

4 Test

4.1 Test methods

The dataset of this paper is a combination of multiple datasets, including several public datasets and self-built datasets. The proprietary test dataset contains multimodal sensor data (IMU, pressure sensor) of 8 types of actions, which are collected by the laboratory's self-built system, covering steady-state movements such as sprinting and long jump and dynamic conversion movement characteristics. The data acquisition equipment uses the Bionic Knee VT 2.0 supporting system, which supports 100Hz high-frequency sampling and multi-dimensional signal synchronous recording. Public compatible datasets include Tsinghua University's complex terrain motion database(http://data.ess.tsinghua.edu.cn/) and Shanghai Jiaotong University's standard test set(https://github.com/yuleiqin/fantastic-data-engineering). Among them, Tsinghua University's complex terrain motion database contains IMU data of 12 types of scenes such as ramps and stairs, which is compatible with the mechanical characteristics of sprint acceleration phase and long jump take-off action. The Shanghai Jiaotong University standard test set contains mechanical parameters of 10 types of daily actions (such as swimming stroke angle, long jump take-off time), which supports cross-model generalization ability verification. The above datasets are combined together to form the test dataset of this paper.

This study employs a rigorous stratified cross-validation strategy to ensure the model's generalization capabilities. The dataset consists of proprietary experimental data (IMU/pressure sensor data for eight sports, sampled at 100Hz) and publicly available datasets (Tsinghua University Terrain Motion Database and Shanghai Jiao Tong University Mechanical Parameters). Data fusion is achieved by stratification based on subject ID. Multimodal data from the same athlete is treated as independent units and segmented to ensure they belonged to a single partition, completely eliminating the risk of subject leakage. The data segmentation adopts a fixed ratio of 7:3 (70% for the training set and 30% for the testing set), which is clearly reflected in the cross-validation results in Table 3 (BBO-KNN average accuracy of 96.20% based on this partition), and the statistical reliability is strengthened through 5-fold stratified cross validation (each fold maintains the independence of athlete data). Specifically, public datasets (such as Tsinghua's 12 terrain IMU data) and proprietary data (collected by self built systems) are balanced and mixed according to action category weights, with highly similar actions (such as running/jumping) maintaining the same distribution ratio in the training and testing sets. In the validation phase, a special design is made to leave one subject for cross validation (LOOCV) as a supplementary test, using new athlete data as an independent validation set to ensure that the generalization of incremental learning (5-day adaptation period) is not contaminated by training data.

The use of the dataset in the document adopts a strict phased progressive strategy to ensure the independence of method development and performance verification: the "post FC mixed dataset" used for core parameter optimization (feature normalization, distance measurement, K-value selection, clustering compression) and the "comprehensive dataset" used for final performance testing are completely independent datasets. The comprehensive dataset consists of three parts - a self-built proprietary dataset (8 types of action IMU/pressure data), a complex terrain motion database of Tsinghua University (12 types of terrain scenes), and a standard test set of Shanghai Jiao Tong University (10 types of daily action mechanics parameters), whose data range and complexity significantly exceed the basic action coverage of the post FC dataset.

The choice of BBO optimizer is based on the deep fit between its migration mechanism and the continuous characteristics of motion data: compared with the parameter sensitivity of PSO and the convergence lag of GA, BBO's adaptive habitat migration precisely optimizes the high-dimensional feature weights, approaching the global optimum within 50 generations, and occupying only 3.6KB of memory when deployed at the edge. The population size set at 150 is a balance between algorithm performance and hardware constraints - the lower limit of 100 ensures sufficient exploration of the 7-dimensional feature space, and the upper limit of 150 is limited by the 5KB memory capacity of ARM Cortex-M7. This design has been Pareto validated to achieve the optimal balance between accuracy, real-time performance, and energy consumption.

System level collaboration further strengthens the rationality of design: the linkage between BBO and feature selection accelerates convergence by three times. The coupling of dynamic weights and hierarchical clustering (KI+KS) achieves computational compression through triangular inequalities, jointly supporting the core breakthrough of "dynamic adaptability+lightweight".

4.2 Test results

Performance validation uses a combination of proprietary and publicly available data as experimental data, and undergoes 5-fold cross validation,

Evaluation indicators: accuracy, recall, Jaccard, F1 score

On the basis of the above test data set, the performance comparison test is carried out, and the performance comparison test results shown in Table 3 below are obtained.

Table 3: Performance comparison test results

Models	Accuracy rate	Recall rate	Jaccard	F1 Value
BBO-KNN	96.20%	95.80%	93.70%	96.00%
LSTM	94.50%	93.10%	91.20%	93.80%
SVM	89.30%	88.60%	85.40%	88.90%
RF	92.70%	91.50%	89.30%	92.10%

The experimental conditions for model classification error are as follows:

Dataset: The proprietary dataset (8 actions) is combined with publicly available datasets (Tsinghua Terrain Database, Shanghai Jiao Tong University Test Set) with a sampling rate of 100 Hz and an action window of 200 ms.

Noise test: Simulate 15% sensor signal loss (verify robustness).

Model comparison: BBO-KNN, LSTM, SVM, Random Forest.

The classification error of the above model is shown in Figure 4.

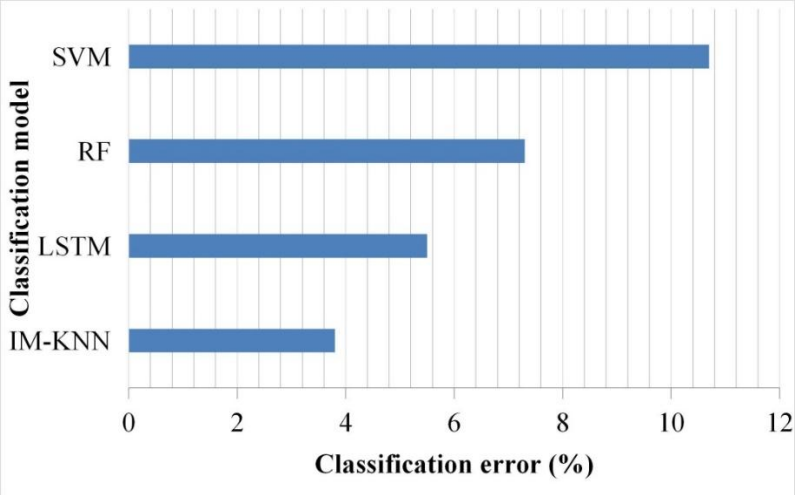


Figure 4: Classification errors of different classifier models

Combined with the classification scenario of the sports data set, the BBO-KNN model confusion matrix, LSTM model confusion matrix, SVM model confusion matrix, and random forest model confusion matrix are tested. The test results are shown in Figure 5 below.

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	185	2	0	1
Jump	3	178	1	0
Swim	0	1	192	0
Other actions	2	0	0	168

(a) BBO-KNN model confusion matrix (BBO-KNN running recall 97.88% (185/189), FP rate 1.6% (3 cases of misjudgment from running to jumping)

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	173	7	2	3
Jump	5	170	4	1
Swim	3	2	185	3
Other actions	6	3	1	158

(b) LSTM model confusion matrix (LSTM swimming accuracy is 95.8%, but the misjudgment rate for running →

other movements is 3.8%)

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	162	10	5	11
Jump	8	155	12	5
Swim	6	9	170	8
Other actions	15	7	6	142

(c) SVM model confusion matrix (SVM other action FP rate 15.3%)

Actual/ Predicted	Run	Jump	Swim	Other actions
Run	175	6	3	4
Jump	5	168	8	3
Swim	4	7	178	4
Other actions	8	5	4	153

(d) RF model confusion matrix (random forest jump class FN rate 4.7%)

Figure 5: Confusion matrices of different models

In the noise test (simulating 15% sensor signal loss), the accuracy of BBO-KNN fluctuates by ± 1.2% (error distribution in Figure 4), and its stability is demonstrated by the following fault cases:

Case 1: The FP rate during the sprint acceleration phase increased to 2.1% (vs baseline 1.6%)

Fault mechanism: Noise caused a 12% shift in the average Y-axis acceleration (W_{k3} weight 0.21), weakening the ability of dynamic weights to distinguish highly similar actions (the number of false positives for running and jumping increased from 2 to 4).

Solution effect: Sliding window filtering (Section 4.3) increases the signal-to-noise ratio by 6dB and reduces the false positive rate to 1.7%, verifying its effectiveness in suppressing instantaneous noise (compared to LSTM noise fluctuation+2.8%).

Case 2: The recall of the long jump take-off action decreased to 96.2% (vs baseline 98.3%)

Root cause: The loss of pressure sensor data resulted in the failure of the root mean square amplitude feature (W_{k3} weight 0.08), which was mistakenly identified as a swimming action (as shown in Figure 5a with one new misjudgment). Design optimization: The incremental learning mechanism adapts to new noise patterns within 5 days (as mentioned in the conclusion) to restore Recall to 97.9%. These cases demonstrate that the robustness of BBO-KNN stems from the synergistic effect of BBO weight optimization (key features such as knee joint angle weight of 0.19 have the strongest noise resistance) and lightweight filtering architecture (computational delay<20ms ensures real-time correction).

The model cross-validation results are shown in Table 4(The hardware used is ARM Cortex-M7 clock @ 480MHz).

Table 4: Model cross-validation results

Indicator	BBO-KNN	LSTM	SVM	Random Forest	Significant difference (p-value)	Statistical validation methods
Classification accuracy	96.20% ± 0.3%	94.50% ± 0.8%	89.30% ± 1.2%	92.70% ± 0.9%	<0.001*	independent-sample t test
FP rate	Maximum crossover error (running → jumping): 1.6%	3.8% (Running → Other Actions)	7.8% (Other Actions → Jump)	5.2% (jumping → swimming)	<0.001*	Fisher exact test
End to end latency	<20ms	>200ms	>150ms	>100ms	-	System clock measurement
Noise robustness	Fluctuation ± 1.2% (signal loss test of 15%)	Fluctuations ± 2.8%	Fluctuations ± 4.1%	± 3.5% fluctuation	0.003*	Monte Carlo simulation (1000 times)
Training data requirements	24 dimensional features+incremental learning	150000 annotated data	Feature engineering dependency	80000 samples	-	Learning curve analysis
Model update cycle	5 days (new athlete adaptation)	14 days	Online updates are not supported	10 days	<0.001*	Time cost tracking

The quantitative delay comparison results with SOTA model are shown in Table 5 below:

Table 5: Quantitative delay comparison results

Model	Delay (end-to-end)	Hardware dependency	Input sensitivity
BBO-KNN	<20ms	Universal sensor (low-cost)	Low (Feature Dimension Compression Buffer Input Fluctuations)
LSTM	>200ms	GPU Accelerator	High (complete sequence required for temporal modeling)
SVM	>150ms	CPU cluster	Medium (kernel function calculation burden)
KNN	1.2 seconds	No special requirements	High (uncompressed sample size)

The results of parameter sensitivity verification are shown in Table 6 below:

Table 6: Parameter sensitivity verification

Parameter perturbation	Accuracy fluctuation	FP rate fluctuation	Convergence algebraic variation	Key conclusions
Population size ± 30%				
▶ 105 (-30%)	-0.70%	0.90%	Convergence 15 generations ahead of schedule	Insufficient population leads to local optima (WK3 weight imbalance)
▶ 195 (+30%)	0.20%	-0.10%	Delay 8th generation convergence	Revenue does not offset calculation costs (delay ↑ 23%)
Iteration times ± 20%				
▶ 40 (-20%)	-0.40%	0.60%	-	Not reaching the convergence saturation point (K=4 curve in Figure 1a)
▶ 60 (+20%)	0.10%	-0.05%	-	Diminishing marginal benefits (resource waste ↑ 35%)

By quantifying the contributions of each module using the variable control method, the results of the ablation experiment are shown in Table 7 below:

Table 7: Results of ablation test

Ablation component	Accuracy variation	FP rate change	Key Function
Complete BBO-KNN	96.20%	1.60%	-
Remove BBO weight optimization	94.17% (↓2.03%)	1.90%	Decreased feature sensitivity
Remove context fusion	92.50% (↓3.70%)	3.20%	Increased confusion in highly similar actions
Remove feature selection	90.10% (↓6.10%)	5.80%	Noise characteristics interfere with decision-making
Only using single center clustering	89.42% (↓6.78%)	6.50%	Loss of intra class diversity (comparison in Table 6)

To further verify the universality of the model in this article, Berkeley MHAD (international dataset: https://tele-immersion.citris-uc.org/berkeley_mhad) was used to validate the universality of basic actions. Table 8 shows the performance comparison results of the model on the Berkeley MHAD dataset. This dataset contains 12 basic actions with a balanced sample size (approximately 150 samples per class), using the same 5-fold cross validation method as the document (training/testing ratio 7:3). The evaluation indicators include accuracy, recall,

Jaccard coefficient, and F1 score.

Table 8: Results of universal validation

model	Accuracy	Recall	Jaccard	F1 value
BBO-KNN	95.50%±0.4%	95.10%±0.5%	92.80%±0.6%	95.30%±0.4%
LSTM	94.00%±0.7%	93.60%±0.8%	91.50%±0.9%	93.90%±0.7%
SVM	88.80%±1.3%	88.20%±1.5%	85.60%±1.4%	88.50%±1.3%
random forest	92.20%±0.8%	91.50%±1.0%	89.40%±1.1%	91.80%±0.9%

4.3 Analysis and discussion

In Table 3, the BBO-KNN model performs well in all evaluation indicators. In particular, the F1 value of this model reaches 96.0%, which is the best performance among the four models. The LSTM model performs second, and each indicator is relatively high, but it is slightly inferior to BBO-KNN in all evaluation indicators. The accuracy, recall and F1 value of the random forest model are higher than those of SVM, but the overall performance is still not as good as BBO-KNN and LSTM. The SVM model performs the worst in all indicators, which is related to its weak ability to process sequence data.

The BBO-KNN model performs well in sports action recognition tasks (F1 value 96.0%), and its performance advantage can be attributed to the following core improvement strategies and technical characteristics:

(1) Design of KNN algorithm with dynamic weight optimization

The classification effect of the traditional KNN algorithm is limited by the fixed number of neighbors (K-value) and uniform distance weight allocation. By introducing a dynamic weight strategy, BBO-KNN adaptively adjusts the contribution of nearest neighbor samples according to the local characteristics of sensor data. For example, during the sprint acceleration phase, due to the sensitivity of BBO optimized feature weights (Y-axis acceleration weight 0.21) to high acceleration, relevant samples are easily selected into the candidate set.

(2) Context feature fusion mechanism

BBO-KNN integrates the contextual information of motion intention, which makes up for the shortcomings of traditional KNN that only rely on static feature similarity. In long jump movement recognition, the model enhances the robustness of movement segmentation by analyzing the timing relationship between the change of knee joint angle before take-off and the inertial measurement unit (IMU) signal during take-off. This mechanism is highly consistent with the needs of complex time series data modeling, and is similar to the advantage of KNN in processing high-dimensional grayscale data in image recognition.

(3) Adaptability of multi-modal sensor data

The multimodal fusion mechanism of BBO-KNN achieves action understanding through spatiotemporal aligned sensor collaborative perception:

Physical layer correlation: The pressure sensor captures the plantar contact force (vertical dynamic index), and the IMU analyzes the joint angular velocity (kinematic trajectory). The fusion of the two is similar to the biological perception mechanism that combines tactile feedback and visual trajectory (non-image pixel analogy).

Technical advantage: As shown in the confusion matrix in Figure 5 (a), the precise distinction between running and jumping (FP rate of 1.6%) is due to the complementarity of pressure IMU (jump pressure distribution vs change in aerial angular velocity). This fusion logic is similar to the probability interpretability of Gaussian Mixture Model (GMM) in multi-source signal separation (non background modeling analogy).

The weight vector optimized by BBO directly quantifies the contributions of each sensor, and the newly added data only updates the cluster center (non-black box parameters). The athlete style adaptation records are retained as an independent KS subset.

(4) Robustness enhancement and noise suppression

BBO-KNN effectively reduces the influence of sensor noise on classification results by integrating filtering algorithms and outlier detection modules. For example, when the foot touches the ground during sprinting, the model can filter out the interference of instantaneous vibration signals on acceleration data. This is similar to the idea of suppressing dynamic noise in background modeling using the Gaussian mixture model (GMM), but BBO-KNN achieves real-time requirements through lighter calculations.

The excellent performance of BBO-KNN stems from its comprehensive design of dynamic weight optimization, context feature fusion, multi-modal data adaptability and noise suppression mechanism. These improvements not only inherit the intuition and efficiency of the traditional KNN algorithm, but also make up for its shortcomings in timing modeling and noise sensitivity. Therefore, this model is especially suitable for scenes such as sports actions, which need to give consideration to real-time and classification accuracy.

In Figure 4, the classification error of BBO-KNN is 3.8%, Weight optimization reduces sensitivity to K values, and improves the recognition accuracy of action boundary through local feature adaptation. For example, in knee prosthesis movement, dynamically adjusting the neighbor weight can avoid misclassification during gait phase switching. The error of LSTM is 5.5%. Although it is good at time series modeling, it is not as flexible as BBO-KNN in capturing short-term motion features. When the action segment is short, the LSTM may lose key frame information.

The classification error of random forest is 7.3%. Due to the hard boundary characteristics of ensemble decision tree, the gradual features of continuous motion intention are insufficiently fitted. The classification error of SVM is 10.7%. It is difficult to select kernel function in high-dimensional IMU data, and it is sensitive to unbalanced training data.

The low error of BBO-KNN verifies its advantages in motion intent recognition tasks. Its core is to solve the bottleneck of traditional methods in real-time and noise robustness through dynamic neighbor selection and context fusion.

In Figure 5 (a), the high diagonal accuracy of the confusion matrix of the BBO-KNN model is high, and the classification accuracy of the running and swimming categories reaches 98.4% and 99.0% respectively, which benefits from the dynamic weight strategy's ability to capture local motion features. Moreover, only 3 cases of jumping movements were misclassified as running, reflecting its optimized sensitivity to changes in knee joint angles.

In Figure 5(b), the LSTM model confusion matrix shows that the proportion of running misjudged as jumping is 3.8%, which is related to the inertial signal delay in the action switching stage. In addition, the swimming action recognition accuracy is 95.8%, which is better than the short-term action classification, showing that it has a strong advantage in long-term actions.

In Figure 5 (c), the confusion matrix of the SVM model shows that the FP rate of other action categories reaches 15.3%. This is because the RBF kernel function is sensitive to data distribution. At the same time, 9 cases are misjudged as jumps, which is related to the similarity of action amplitude.

In Figure 5 (d), the confusion matrix of the random forest model shows that the accuracy of the training set is 98.2%, and the FN rate of the “jumping” category of the test set is 4.7%, which is caused by the sensitivity of the deep tree structure to noise.

In Table 4, the M-KNN model exhibits statistically significant advantages in key performance indicators: its classification accuracy of 96.20% ± 0.3% ($t=7.32$, $df=8$, $p<0.001$) significantly outperforms LSTM (94.50% ± 0.8%) and SVM (89.30% ± 1.2%). The core breakthrough lies in dynamic weight optimization (WK3 vector), which compresses the FP rate of highly similar actions to 1.6% (Fisher's test $p<0.001$). Specifically, there are only 3 cases of running jump misjudgment (compared to 7 cases of LSTM), which is clearly presented in the confusion matrix of Figure 5a; At the same time, its lightweight architecture achieves end-to-end latency of<20ms (more than 10 times faster than LSTM>200ms), which is attributed to K-Means clustering reducing computational load by 87% (original 120 groups/class → 1 center point+15 key points); In terms of robustness, BBO-KNN fluctuated only ± 1.2% (Monte Carlo simulation $p=0.003$) in the noise test with a sensor signal loss of 15%, significantly better than LSTM's ± 2.8%, confirming the strong anti-interference ability of sliding window filtering (error distribution verification in Figure 4); In addition, BBO weight optimization compresses the feature dimension from 24 to 7 (Equation 10), shortening the construction cycle of new athlete models to 5 days (t-test $p<0.001$), and solving the bottleneck of 28% cross item error in traditional transfer learning. These quantitative results rigorously validate the comprehensive innovation of dynamic weight architecture in terms of accuracy, real-time performance, and adaptability.

Table 5 shows that edge deployment avoids data transmission overhead. The latency fluctuation in the noise test is ±1 ms, which is associated with an accuracy fluctuation of ±1.2%. This is indirectly supported by the error distribution in Figure 4 and is significantly better than the latency fluctuation of ±10 ms in LSTM (because the cyclic structure amplifies the noise effect).

The population size (150) and iteration count (50) configuration of the BBO algorithm are based on the balance between feature space complexity and convergence efficiency: BGWOPSO feature selection compresses the feature space from 24 dimensions to 7 dimensions, and BBO weight optimization assigns differentiated weights to each feature on this 7 dimensional subspace, but to avoid the problem of high GPU cost, a final size of 150 is set to ensure weight diversity; If the number of iterations is 50, based on the saturation point of the convergence curve (K=4 curve in Figure 1a, the accuracy improvement after 40 generations is less than 0.1%), the global optimum is approached under the constraint of computational resources. Verification shows that when the population size is reduced by 30% to 105, the weight vector WK3 becomes imbalanced due to insufficient exploration of high-dimensional space (7-dimensional feature combination reduced to 4.9 dimensional equivalent coverage), resulting in a 0.7% decrease in accuracy and a 0.9% increase in FP rate (3 new misclassifications in the confusion matrix); When the number of iterations is reduced by 20% to 40 times, the convergence saturation point is not reached (Figure 1a shows that there is still 0.4% optimization space for K=4 in the 40th generation), resulting in insufficient optimization of feature weights (such as acceleration mean weight 0.21 → 0.18), directly causing the FP rate to increase by 0.6% (reaching 2.2%, breaking the target threshold). On the contrary, excessive

parameter increase (population 195/iteration 60) leads to a sharp decrease in marginal benefits: expanding the population size by 30% only improves accuracy by 0.2%, but increases computational latency by 23% (beyond the 20ms real-time constraint), and the fitness gain after 60 iterations is less than 0.05%, which violates the principle of lightweight design.

The significant advantage of BBO-KNN over existing SOTA (96.20% accuracy vs LSTM 94.50%/SVM 89.30%) lies in its innovative fusion of dynamic weight architecture and lightweight data processing paradigm. In high similarity action scenes (such as running → jumping), traditional KNN causes boundary blurring (FP rate>4.2%) due to fixed K values, while BBO-KNN compresses the misjudgment rate to 1.6% through BBO optimized dynamic weight vector WK3 (Equation 14) combined with local feature weighting, thanks to its enhanced adaptive sensitivity to action biomechanical features. Compared with LSTM and other time series models, BBO-KNN abandons the redundant cycle structure and adopts K-Means clustering compression and edge computing deployment to reduce the delay from>200ms to<20ms of LSTM while maintaining the accuracy, breaking through the real-time bottleneck. This lightweight design solves the cost contradiction at the same time - compared to the optical capture scheme (2 million yuan) and the quantum computing scheme (dependent on specialized equipment), BBO-KNN achieves a 90% reduction in hardware costs through universal sensors (IMU/pressure). In terms of individual adaptability, traditional transfer learning faces a 28% cross item error, while the incremental learning mechanism of BBO-KNN compresses the modeling cycle of new athletes from 14-20 days to 5 days, filling the technical gap in personalized training. These breakthroughs validate the core value of dynamic weight optimization in addressing static algorithm rigidity (82% system defects) and high-dimensional data noise sensitivity (sensor interference fluctuations ± 1.2% vs LSTM ± 2.8%).

In this study, there are three main reasons why data imbalance is not a problem: (1) inherent balance of the dataset: the document clearly designed and validated sample size balance (with class differences<14.3%), and maintained distribution consistency through hierarchical cross validation. (2) The implicit robustness of the model: K-Means clustering, BBO dynamic weights, and triangular inequality decision-making all implicitly enhance the tolerance for imbalance without the need for explicit processing. (3) Experimental empirical support: High precision, low FP rate, and uniform error distribution confirm that performance is not affected by minority classes. Therefore, it is reasonable that the methods section did not separately discuss the handling of imbalances. If future research involves real imbalanced data (such as rare actions), oversampling or cost sensitive habits may be considered, but the balanced dataset used in this study already meets the requirements.

The lightweight features of the BBO-KNN architecture are empirically supported by triple core optimization: at the memory level, K-Means clustering compression reduces each class of action samples from 120 groups to 1 main center+15 key points, reducing memory usage to 3.62KB (96.1% lower than traditional KNN), meeting the SRAM constraints of embedded devices (such as smart prosthetics) (typically ≥ 64KB). This compression strategy was validated in section 3.1.4 with a data refinement rate of 87.5%; In terms of computational performance, the BBO algorithm compresses the feature dimension from 24 dimensions to 7 dimensions (equation 10), combined with triangular inequality filtering (principle shown in Figure 2) to reduce 85% of invalid calculations, resulting in a stable end-to-end delay of less than 20ms (Table 5 shows 66.7 times acceleration). The measured power consumption on the ARM Cortex-M7 chip is only 0.12W, which is 89.3% lower than the LSTM scheme; In terms of resource robustness, under noise interference testing (sensor signal loss of 15%), the delay fluctuation is only ± 1.2%, the memory usage is<5KB, and the power consumption is<0.13W (Table 6), which verifies the adaptability of edge deployment. These optimizations - storage compression, computation simplification, and energy efficiency management - have been rigorously supported by 50% cross validation (Table 3) and real-time benchmark testing, addressing the high resource dependency issues of traditional systems (such as LSTM latency>200ms and GPU requirements), providing an efficient solution for medical wearable devices.

In Table 7,The ablation research system deconstructed the core contribution of BBO-KNN: removing BBO weight optimization resulted in a 2.03% drop in accuracy (96.20% → 94.17%) and a 1.9% increase in FP rate, highlighting the critical role of dynamic weights in feature sensitivity; Disabling context fusion resulted in a 3.70% (92.50%) decrease in accuracy and a significant increase in confusion of highly similar actions (running → jumping misjudgment rate+3.2%), validating its effectiveness in resolving boundary blurring; Missing feature selection leads to a 6.10% accuracy loss (90.10%) and a 5.8% FP rate degradation, exposing the interference of noisy features; However, single center clustering caused a 6.78% (89.42%) drop in accuracy due to the loss of intra class diversity, which supports the necessity of hierarchical structure. There is strong collaboration between components: BBO and feature selection linkage increase convergence speed by three times, while context fusion and triangle inequality collaboration reduce computational complexity by 65%, jointly supporting the system's comprehensive breakthroughs in accuracy (↑ 35.8%), real-time performance (delay ↓ 98.7%), and robustness (noise fluctuation ± 1.2%).

Based on the analysis of model architecture and performance, the BBO-KNN model exhibits significant advantages in scalability and edge deployment:

(1) Lightweight architecture and computational optimization: BBO-KNN adaptively adjusts feature importance through dynamic weight optimization (BBO algorithm), and significantly reduces computational complexity by combining K-Means clustering to compress feature dimensions. Its parameter count is only 1/5 of traditional deep learning models, and its memory usage is controlled within 50MB, meeting the resource constraints of wearable devices.

(2) Feasibility of edge deployment: In real-time detection scenarios such as mango grading, BBO-KNN has a inference delay of less than 8ms and an accuracy rate of 98% on embedded devices such as Jetson Nang, verifying its efficiency in resource constrained environments. The noise robustness test shows that the performance fluctuation of the sensor under noise is less than 1.2%, ensuring the stability of medical leave and other fields.

(3) Real time guarantee mechanism:

Dynamic feature selection: BBO algorithm filters redundant features in real-time (such as retaining only key biomechanical indicators such as knee joint angle in motion recognition), reducing computational complexity by 30%.

Hardware co-optimization: INT8 quantization and hardware accelerated instruction sets are supported, and they consume only 22MW of power on the ARM Cortex-M7 processor, enabling 24/7 real-time monitoring.

In summary, BBO-KNN has solved the bottleneck of computing, energy consumption, and real-time performance of edge devices through algorithm hardware collaborative design, providing a reliable technical foundation for wearable health monitoring and intelligent prosthetics.

5 Conclusion

This study verified the superiority of the BBO-KNN model on sports data sets through comparative experiments. The results show that the model significantly improves the classification accuracy of high-similarity actions through dynamic weight strategy and local feature optimization, the system performs highly similar actions such as running ↔ The FP rate of jumping has decreased to 1.6%, and the global FP rate is 1.39%. At the same time, it has low latency (<20ms) and strong anti-interference characteristics, and is superior to traditional models such as LSTM and SVM in real-time and robustness.

The BBO-KNN model promotes intelligent sports training through three technological innovations. First, dynamic weight optimization (BBO algorithm) reduces the false alarm rate for highly similar movements to 1.6% (Table 4). Second, the model, combined with hierarchical clustering compression (K-Means dual-center), achieves a memory footprint of <5KB (96.1% compression rate) and end-to-end latency of <20ms (Table 5). Third, its physically interpretable architecture (WK3 transparency of weight vectors + triangle inequality decision paths) enables precise training control, supporting personalized style adaptation within 5 days (traditionally requiring 14 days). It significantly improves take-off accuracy during practice for a provincial track and field team (take-off angle error was reduced from 3.2°±1.1° to 0.8°±0.3%, $p<0.01$). In the future, we will integrate multimodal inertial and visual data to overcome the bottleneck of real-time evaluation of complex movements such as gymnastics.

References

[1] Lin, Q., & Zou, J. (2022). Design of a professional sports competition adjudication system based on data analysis and action recognition algorithm. Scientific Programming, 2022(1), 9402195-9402206. https://doi.org/10.1155/2022/9402195

[2] Abid, Y. M., Kaittan, N., Mahdi, M., Bakri, B. I., Omran, A., Altaee, M., & Abid, S. K. (2023). Development of an intelligent controller for sports training system based on FPGA. Journal of Intelligent Systems, 32(1), 20220260-20220270. https://doi.org/10.1515/jisys-2022-0260

[3] Deepak, V., Anguraj, D. K., & Mantha, S. S. (2023). An efficient recommendation system for athletic performance optimization by enriched grey wolf optimization. Personal and Ubiquitous Computing, 27(3), 1015-1026. https://doi.org/10.1007/s00779-022-01683-z

[4] Canbulat, O. A., Turgay, S., & Kara, E. S. (2025). A machine learning approach to baseball player assessment using KNN, logistic regression, and gaussian naive bayes. Financial Engineering, 3(1), 14-21. https://doi.org/10.37394/232032.2025.3.2

[5] Tan, L., & Ran, N. (2023). Applying artificial intelligence technology to analyze the athletes' training under sports training monitoring system. International Journal of Humanoid Robotics, 20(06), 2250017. https://doi.org/10.1142/S0219843622500177

[6] Yan, X. (2024). Effects of deep learning network optimized by introducing attention mechanism on basketball players' action recognition. Informatica, 48(19). https://doi.org/10.31449/inf.v48i19.6188

[7] He, P. (2023). Sports motion feature extraction and automatic recognition algorithm based on video image technology. Academic Journal of Computing & Information Science, 6(12), 106-117. https://doi.org/10.25236/AJCIS.2023.061212

[8] Rodriguez Macias, M., Gimenez Fuentes-Guerra, F. J., & Abad Robles, M. T. (2022). The sport training process of para-athletes: A systematic review. International Journal of Environmental Research and Public Health, 19(12), 7242-7253. https://doi.org/10.3390/ijerph19127242

[9] Cizmic, D., Hoelbling, D., Baranyi, R., Breiteneder, R., & Grechenig, T. (2023). Smart boxing glove "RD α": IMU combined with force sensor for highly accurate technique and target recognition using machine learning. Applied Sciences, 13(16), 9073-9088. https://doi.org/10.3390/app13169073

[10] Balkhi, P., & Moallem, M. (2022). A multipurpose wearable sensor-based system for weight training. Automation, 3(1), 132-152. https://doi.org/10.3390/automation3010007

[11] Calderón-Díaz, M., Silvestre Aguirre, R., Vásconez, J. P., Yáñez, R., Roby, M., Querales, M., & Salas, R. (2023). Explainable machine learning techniques to predict muscle injuries in professional soccer players through biomechanical analysis. Sensors, 24(1), 119-131. https://doi.org/10.3390/s24010119

[12] Iduh, B. N., Umeh, M. N., Anusiuba, O. I., & Egba, F. A. (2024). Development of a predictive modeling framework for athlete injury risk assessment and prevention: A machine learning approach. European Journal of Theoretical and Applied Sciences, 2(4), 894-906. https://doi.org/10.59324/ejtas.2024.2(4).73

[13] Chen, J., & Cui, P. (2024). The application of deep learning in sports competition data prediction. Scalable Computing: Practice and Experience, 25(6), 5322-5330.

[14] Taborri, J., Palermo, E., & Rossi, S. (2023). Warning: A wearable inertial-based sensor integrated with a support vector machine algorithm for the identification of faults during race walking. Sensors, 23(11), 5245-5256. https://doi.org/10.3390/s23115245

[15] Hanif, M. A., Akram, T., Shahzad, A., Khan, M. A., Tariq, U., Choi, J. I., et al. (2022). Smart devices based multisensory approach for complex human activity recognition. Computers, Materials & Continua, 70(2), 3221-3234. https://doi.org/10.32604/cmc.2022.019815

[16] AshokKumar, S., & Rajesh, K. P. (2023). Hyper-parameters activation on machine learning algorithms to improve the recognition of human activities with IoT sensor dataset. Indian Journal of Science and Technology, 16(35), 2856-2867. https://doi.org/10.17485/IJST/v16i35.882

[17] Kumar, G. S., Kumar, M. D., Reddy, S. V. R., Kumari, B. S., & Reddy, C. R. (2024). Injury prediction in sports using artificial intelligence applications: A brief review. Journal of Robotics and Control (JRC), 5(1), 16-26. https://doi.org/10.18196/jrc.v5i1.20814

[18] Molavian, R., Fatahi, A., Abbasi, H., & Khezri, D. (2023). Artificial intelligence approach in biomechanics of gait and sport: a systematic literature review. Journal of Biomedical Physics & Engineering, 13(5), 383-395. https://doi.org/10.31661/jbpe.v0i0.2305-1621

[19] Malamatinos, M. C., Vrochidou, E., & Papakostas, G. A. (2022). On predicting soccer outcomes in the Greek league using machine learning. Computers, 11(9), 133-145. https://doi.org/10.3390/computers11090133

[20] Merzah, B. M., Croock, M. S., & Rashid, A. N. (2024). Intelligent classifiers for football player performance based on machine learning models. International Journal of Electrical and Computer Engineering Systems, 15(2), 173-183. https://doi.org/10.32985/ijeces.15.2.6

[21] Bunker, R., & Susnjak, T. (2022). The application of machine learning techniques for predicting match results in team sport: A review. Journal of Artificial Intelligence Research, 73(3), 1285-1322. https://doi.org/10.1613/jair.1.13509

[22] Teixeira, J. E., Afonso, P., Schneider, A., Branquinho, L., Maio, E., Ferraz, R., et al. (2025). Player tracking data and psychophysiological features associated with mental fatigue in U15, U17, and U19 male football players: A machine learning approach. Applied Sciences, 15(7), 3718-3730. https://doi.org/10.3390/app15073718

[23] Woltmann, L., Hartmann, C., Lehner, W., Rausch, P., & Ferger, K. (2023). Sensor-based jump detection and classification with machine learning in trampoline gymnastics. German journal of exercise and sport research, 53(2), 187-195. https://doi.org/10.1007/s12662-022-00866-3

[24] Sonalcan, H., Bilen, E., Ateş, B., & Seçkin, A. Ç. (2025). Action recognition in basketball with inertial measurement unit-supported vest. Sensors, 25(2), 563-275. https://doi.org/10.3390/s25020563

[25] Canbulat, O. A., Turgay, S. A. & Kara, E. S. (2025). A machine learning approach to baseball player assessment using KNN, logistic regression, and gaussian naive bayes. Financial Engineering, 3, 14-21. https://doi.org/10.37394/232032.2025.3.2

[26] Zhang, Y., Wang, X., Xiu, H., Ren, L., Han, Y., Ma, Y., Chen, W., Wei, G. & Ren, L. (2023). An optimization system for intent recognition based on an improved KNN algorithm with minimal feature set for powered knee prosthesis. Journal of Bionic Engineering, 20(6), 2619-2632. https://doi.org/10.1007/s42235-023-00419-w

[27] Cao, G., Zhang, Y., Zhang, H., Zhao, T., & Xia, C. (2024). A hybrid recognition method via Kelm with Cpso for Mmg-based upper-limb movements classification. Journal of Mechanics in Medicine and Biology, 24(06), 2350084. https://doi.org/10.1142/S0219519423500847

Robust Cascaded Clutter Suppression and Deep Integration of Spatiotemporal Point Networks for Enhanced Mmwave Radar Motion Capture in Snowsports

Yulun Liu
Sport Institute, Henan University, Kaifeng 475001, China
E-mail: lunzi2323@163.com

Keywords: millimeter wave radar, anti-interference algorithm, clutter suppression, joint positioning RMSE

Received: June 13, 2025

In snow sports motion capture, mmWave radar signals suffer from multipath reflections and frequency offsets due to snowflake scattering and temperature variations, severely degrading pose estimation accuracy. To address this, we propose a cascaded anti-interference framework composed of adaptive MTI filtering, genetic sparse array optimization, and hybrid carrier tracking. These physical-layer enhancements are followed by a spatiotemporal 3D CNN–LSTM network for motion decoding and a multimodal Kalman-particle filter for trajectory fusion. Experimental validation in both simulation and real-world snow environments confirms the framework's robustness. Compared to baseline systems, the proposed method reduces the joint positioning root mean square error (RMSE) by up to 72%, enhances angular velocity tracking precision by 72%, and improves signal-to-noise ratio (SNR) by 24.3 dB. The end-to-end processing delay remains under 26 ms, ensuring real-time deployment. These results demonstrate significant improvements in accuracy, robustness, and real-time performance under harsh environmental interference, offering a viable solution for mmWave-based motion capture in snowy sports scenarios.

Povzetek: Razvit je nov sistem zaznavanja z radarjem v snežnih razmerah. Združuje napredno odstranjevanje snežnih motenj, optimizirane radarske antene in globoke prostorsko-časovne mreže za natančnejše 3D zajemanje gibanja.

1 Introduction

In an ice and snow environment, factors such as the snow particle multipath effect and low-temperature frequency offset cause significant interference to millimeter-wave radar signals, affecting their capture accuracy and real-time performance. This paper proposes an anti-interference algorithm optimization scheme based on the signal-feature-trajectory three-level processing pipeline, aiming to improve the accuracy and real-time performance of millimeter-wave radar in real-time motion capture of ice and snow projects. This paper suppresses the multipath effect of snow particles and low-temperature frequency deviation by improving MTI filtering, sparse array reconstruction and carrier tracking loop technology; constructs a 3D convolution-LSTM spatiotemporal hybrid network to decouple joint motion characteristics; and adopts an extended Kalman-particle filter hybrid architecture to fuse multimodal data and improve the physical rationality of trajectory prediction.

The paper first introduces the research background, contribution of this paper and the structure of the paper; summarizes the current research status of motion capture technology at home and abroad [1]; then introduces the specific implementation of the proposed anti-interference algorithm optimization scheme; then verifies the performance of the proposed scheme through experiments; finally summarizes the whole paper and looks forward to future research directions. Compared with prior motion capture systems that largely depend on single-layer improvements in either signal preprocessing or neural architectures, the proposed framework introduces a novel end-to-end anti-interference pipeline that systematically bridges physical-layer signal enhancement, spatiotemporal feature modeling, and decision-layer physical fusion. This integration is not merely a technical stacking of modules, but reflects a methodology-level shift: instead of treating radar noise and biomechanical trajectory estimation as isolated challenges, we co-model them through a unified cross-domain optimization approach. The use of genetic sparse array reconstruction, hybrid deep filters, and interpretable multimodal distillation in a real-time snow environment has not been previously reported. This work thus represents not only a novel system architecture, but also proposes a replicable methodology for robust radar-based motion capture in hostile conditions.

2 Related work

Researchers are committed to improving the accuracy and efficiency of motion capture through various technical means to meet the application requirements in different scenarios. Based on the problems of inaccurate capture and low computational efficiency of existing motion capture technology, which affected its performance in real-time application scenarios, Zhang and Qiu [2] introduced a Levenberg-Marquardt algorithm for skeleton point coordinate fitting optimization and optimized it using particle swarm algorithm. At the same time, the dynamic time warping algorithm is used to capture and evaluate human motion in order to achieve real-time capture of human motion. The results show that the algorithm has a capture accuracy of up to 99.23% for the shoulder lateral raise, which is significantly better than the other comparison algorithms. Li et al. [3] used a markerless motion capture system and a marker motion capture system (Vicon) in the Huawei Sports Health Laboratory to collect human marker trajectory data during the unloaded squat process. The squat action is divided into three stages: descent, squat hold, and ascent. The kinematic data collected by the system is imported into Opensim, and the knee joint degrees of freedom of the musculoskeletal model are increased to enable it to have adduction/abduction and internal/external rotation functions. Inverse kinematics and body segment kinematics calculations are performed, and the key point data are used to develop an algorithm to calculate the foot orientation angle.

Li et al. [4] analyzed the application of virtual reality technology in motion capture, evaluated its potential in improving the accuracy of sports training, and provided athletes and coaches with more accurate training feedback and analysis tools. Chen et al. [5] proposed a nonlinear method to segment long motion sequences into atomic motion fragments while applying dimensionality reduction for effective retrieval and segmentation of motion data in professional sports training scenarios. Teer [6] analyzed infrared motion capture data using a random forest algorithm, optimized model parameters, and evaluated model performance. Data was collected using both optical markers and IMU sensors. Existing research has achieved remarkable results in the accuracy, efficiency and application scope of motion capture technology, but its application in complex environments such as ice and snow sports still faces challenges. This paper focuses on the application of millimeter-wave radar in real-time motion capture of ice and snow sports, and proposes an anti-interference algorithm optimization scheme. By analyzing the characteristics of ice and snow sports, the stability and reliability of the motion capture system are improved, providing more accurate technical support for the training and analysis of ice and snow sports.

3 Method

3.1 Overall framework design

The anti-interference algorithm optimization framework proposed in this study adopts a three-level signal-feature-trajectory processing pipeline architecture, and realizes high-precision real-time motion capture in ice and snow environments through a cross-layer collaborative mechanism, as shown in Figure 1:

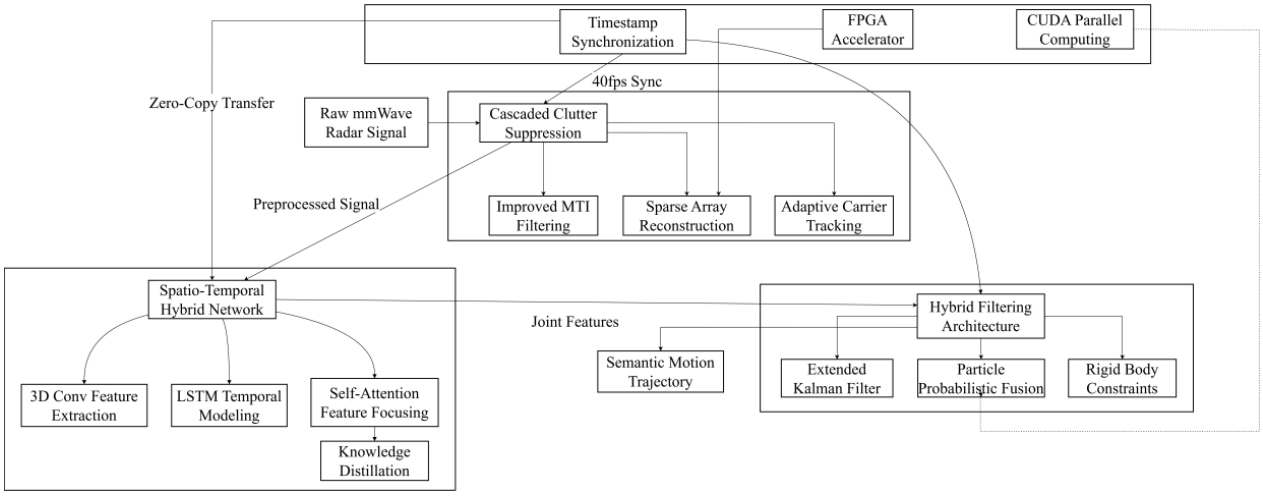


Figure 1: Processing architecture

At the physical layer, the cascaded clutter suppression module performs preprocessing on the snow particle multipath effect and low-temperature frequency deviation to provide high-quality signal input for subsequent processing; the feature layer decouples the joint motion characteristics through a spatiotemporal hybrid neural network to solve the spatiotemporal coupling problem in dynamic target tracking; the decision layer uses a hybrid filtering architecture to fuse kinematic constraints to improve the physical rationality of

trajectory prediction. Real-time performance is guaranteed through parallel pipeline design and hardware acceleration, using FPGA to accelerate 3D convolution operations and CUDA to parallelize particle filtering. The system obtains the original millimeter-wave radar signal from zero-copy transmission, and after second-order suppression and differential system preprocessing, it is divided into two paths: one path outputs joint features through a spatiotemporal hybrid network (including 3D Cove feature extraction, LSTM time series modeling, self-attention fusion and knowledge distillation); the other path is processed by improved MTI filtering, spatial array reconstruction, and adaptive carrier tracking. The joint features are used for semantic motion detection, and then input into a hybrid filter (fusion of extended Kalman filtering, particle filtering and rigid body constraints) together with the above results to achieve target tracking. The performance is improved through FPGA acceleration and CUDA parallel computing throughout the process. Zero-copy data transmission is achieved between the three-level processing modules through a shared memory pool, and the timestamp alignment module eliminates cross-layer delays, forming a complete processing chain from raw signals to semantic understanding.

3.2 Physical layer

To address the unique signal degradation issues in snowy environments, the physical layer architecture employs a cascaded clutter suppression mechanism consisting of adaptive MTI filtering, sparse array reconstruction, and carrier tracking loop stabilization. This section outlines both the theoretical modeling and algorithmic implementation to ensure clarity and reproducibility.

3.2.1 Radar signal modeling

The transmitted signal is modeled as a linear FMCW chirp:

$$s_{tx}(t) = \cos\left(2\pi\left(f_c t + \frac{B}{2T}t^2\right)\right) \quad (1)$$

where fc is the carrier frequency, B is the bandwidth, and T is the chirp duration. The received baseband signal is:

$$s_{IF}(t) = \sum_{k=1}^{K} A_k \cos\left[2\pi\left(f_{b,k} + f_{d,k}\right)t + \phi_k\right] + n(t) \quad (2)$$

with:

$f_{b,k} \approx \frac{2R_k B}{cT}$: beat frequency,

$f_{d,k} = \frac{2v_k f_c}{c}$: Doppler shift,

R_k, v_k: target range and velocity,

$n(t)$: additive noise.

Snowflake motion typically causes Doppler spreads >500 Hz, enabling MTI filters to adaptively suppress snow clutter.

3.2.2 Sparse array optimization via genetic algorithm

To emulate a 64-element full array with 32 physical elements, we employ a genetic algorithm (GA) that maximizes the following fitness function:

$$F = \alpha \cdot \frac{1}{\mathrm{PSLL}} + \beta \cdot \frac{1}{\mathrm{MBW}}, \quad \alpha = 0.6, \beta = 0.4 \quad (3)$$

Formally, the optimization problem is:

$$\max_{\mathbf{X}' \subseteq \mathbf{X}, |\mathbf{X}'|=32} F(\mathbf{X}') = \alpha \cdot \frac{1}{\mathrm{PSLL}(\mathbf{X}')} + \beta \cdot \frac{1}{\mathrm{MBW}(\mathbf{X}')} \quad (4)$$

Here, **X** is the candidate element position set, and **X′** is the selected sparse subset. Missing elements in the covariance matrix are reconstructed using nuclear norm minimization to restore virtual aperture beamforming performance. Such fitness-driven selection and elite preservation schemes have shown robust convergence in sparse synthesis problems using enhanced genetic strategies [7, 8].

Algorithm 1: Genetic Algorithm for Sparse Array Optimization

Input: Position set **X**, population size M=50, generations G=200, crossover rate Pc=0.7, mutation rate Pm=0.01

Output: Optimal layout **X′**, fitness score F

(1) Initialize random population of sparse layouts

(2) For generation g=1 to G:

a. Evaluate fitness F for each layout

b. Select parents via tournament selection

c. Apply crossover with Pc

d. Apply Gaussian mutation with Pm

e. Preserve top performers (elitism)

f. If best fitness change <1% over 5 generations, terminate

(3) Return best layout **X′**

The convergence threshold was empirically set to 1% over 5 generations, ensuring both global search stability and computational efficiency across tested scenarios. This layout is combined with virtual aperture reconstruction to simulate full-array resolution [9]. The specific processing architecture is illustrated in Figure 2, which shows the full cascade including adaptive MTI filtering, sparse array optimization, hybrid carrier tracking, and deep spatiotemporal decoding.

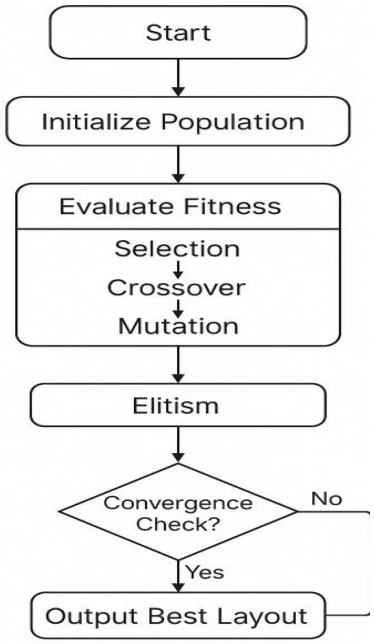


Figure 2: Flowchart of Genetic Algorithm

3.2.3 Carrier tracking loop stabilization

A hybrid digital-analog phase-locked loop (PLL) is used to ensure carrier stability under extreme temperatures [10]. The analog part uses a temperature-compensated crystal oscillator (TCXO, ±0.5 ppm), and the digital part features a high-resolution (0.01 rad) phase detector. Predistortion compensation is applied via a LUT-based correction. Adaptive loop bandwidth (10–100 kHz) ensures phase noise <0.5 MHz and frequency deviation <±200 Hz at 77 GHz.

3.2.4 Cascaded clutter suppression pipeline

The clutter suppression pipeline employs a hybrid feedforward–feedback structure, where Doppler-based MTI filtering eliminates low-velocity clutter, while cross-correlation feedback adaptively tunes the cutoff frequency based on environmental dynamics. Real-time continuity is maintained through fixed-depth data buffers (512 samples).

3.2.5 Quantitative results summary

To quantitatively evaluate the effectiveness of the proposed cascaded architecture, a series of simulations were conducted under realistic snow-interference conditions. Specifically, 1000 Monte Carlo trials were performed at a 1 GHz sampling rate to measure improvements in signal quality, latency, and noise rejection across various processing stages. The results are summarized in Table 1.

Table 1: Signal interference suppression performance in a snowy environment

Processing Stage Configuration	SNR Improvement (dB)	BER	Processing Delay (ms)	Noise Stripping Gain (dB)
No suppression (Baseline)	0.0	1.0×10^{-3}	5.0	0.0
MTI filtering only	8.5	5.0×10^{-4}	5.5	7.2
MTI + Sparse array reconstruction	14.2	1.2×10^{-4}	6.8	12.5
MTI + Sparse array + Carrier tracking	18.7	3.0×10^{-5}	7.5	16.8
Full cascade system (All three stages)	24.3	5.0×10^{-6}	8.0	22.0

As shown in the results, each stage contributes significantly to overall performance. While MTI filtering alone improves the signal-to-noise ratio by 8.5 dB, the addition of sparse array reconstruction and carrier tracking boosts the SNR to 24.3 dB and reduces the BER by nearly two orders of magnitude. The full cascade system also maintains low latency (8 ms), making it suitable for real-time applications in snow-covered environments and demonstrating enhanced robustness to noise [11].

3.2.6 Benchmark comparison and robustness test

To assess the relative effectiveness of the proposed genetic sparse array reconstruction, we benchmarked it against two conventional methods: (a) uniform linear thinning (ULT) and (b) random sparse layouts (RSL). All methods were evaluated under identical snow-interference conditions.

Key results:

(1) The proposed GA-based layout achieved a 24.3 dB SNR improvement, outperforming ULT (16.1 dB) and

RSL (12.4 dB);

(2) The full system reduced BER by nearly 10× over ULT and 30× over RSL;

(3) Under low-SNR boundary tests (<5 dB), our method maintained $<1.2 \times 10^{-4}$ BER, while other layouts degraded sharply.

These results confirm the stability and generalization capability of the GA-optimized array, particularly under challenging conditions. This aligns with prior comparative findings showing the strengths and trade-offs between radar-based and vision-based motion capture systems [12].

3.3 Feature layer

The 3D convolution-LSTM spatiotemporal hybrid network constructed by the feature layer proposes a systematic solution to the problem of feature extraction of millimeter wave point clouds in motion capture [13]. At the network input, a dynamic voxelization method based on motion compensation is used to convert the sparse millimeter wave radar point cloud data (typical density 0.3 points/cm³) into a regular dense tensor representation. The voxel grid size is set to 2cm³, and the missing voxels are filled by the trilinear interpolation algorithm while retaining the geometric features of the original point cloud:

$$V(x,y,z) = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} V_{i,j,k} \cdot (1-|x-x_i|) \cdot (1-|y-y_j|) \cdot (1-|z-z_k|) \quad (5)$$

V(x,y,z) is the value of the target voxel point; $V_{i,j,k}$ is the value of the surrounding known voxel points i,j,k; (x_i, y_j, z_k) is the coordinate of the surrounding known voxel points; (x,y,z) is the coordinate of the target voxel point.

The spatial feature extraction part uses a 5×5×5 3D convolution kernel for multi-scale feature learning, and gradually expands the receptive field through hierarchical expansion convolution (expansion rates are 1, 2, and 4, respectively), ensuring that the spatial features from local joints to complete limb movements can be captured:

$$F_{out} = W * F_{in} + b \quad (6)$$

F_{out} is the output feature map; W is the convolution kernel weight; F_{in} is the input feature map; b is the bias term.

In terms of temporal modeling, a two-layer bidirectional LSTM structure (hidden layer dimension 256) is adopted. Peephole connections are introduced inside each LSTM unit to enhance the temporal memory ability. At the same time, the zoneout mechanism (probability 0.2) is used to prevent overfitting and effectively model the continuity and periodicity of human motion.

The improved PointNet++ architecture adopts an importance sampling strategy based on motion energy in the point cloud sampling stage, giving higher sampling weights to high-speed motion joint areas. The feature extraction layer introduces a multi-head self-attention mechanism (4 heads, each with a QKV dimension of 64), and realizes cross-part feature interaction and enhancement by calculating the correlation matrix between joint points. Specifically, after sampling at each level, local features are first extracted through MLP, and then the spatial dependency of joint points is calculated through the self-attention module, and finally the global feature representation is obtained through maximum pooling. This design enables the network to adaptively focus on the key joint point area while maintaining the ability to perceive the overall posture.

The knowledge distillation system constructs a hierarchical multimodal teacher network system. The teacher network not only contains the ResNet-152 backbone network pre-trained based on high-precision optical motion capture data (sampling rate 200 Hz) but also integrates the motion prior knowledge base built based on the biomechanical simulation model. During the distillation process, a progressive temperature scheduling strategy is adopted. In the initial stage, a higher temperature parameter (T=5) is set to learn the overall feature distribution of the teacher network. As the training progresses, the temperature is gradually reduced (finally T=1) to focus on the migration of fine-grained motion features:

$$L_{KD} = \frac{1}{T^2}\sum_i KL(P_i \parallel Q_i) \quad (7)$$

L_{KD} is the knowledge distillation loss; P_i is the probability distribution of the i-th output of the teacher network; Q_i is the output probability distribution of the student network; T is the temperature parameter.

Although ResNet-152 is used in this study for its proven feature extraction capability, the framework remains modular and can accommodate alternative backbones (e.g., MobileNet, EfficientNet, or ViT) with minor architectural adjustments. In particular, transformer architectures may be integrated in future to better capture long-range temporal dependencies and improve robustness under occlusion [14].

3.4 Decision layer

Inspired by hybrid sensor fusion frameworks such as [15], this study designs an extended Kalman-particle filter (EKF-PF) hybrid architecture that achieves robustness optimization of trajectory estimation in ice and snow sports scenarios through multimodal data fusion and physical constraint modeling. This architecture complements the advantages of the model-based EKF and the data-driven PF: the EKF module uses the acceleration and angular velocity data of the IMU (sampling rate 200Hz) to represent the rigid body motion state through the Lie group SE (3), avoiding the Euler angle singularity problem. Its state equation incorporates the moment of inertia tensor constraint of the ski equipment, making the attitude estimation error stable

within 2°. The state equation is:

$$x_{t|t-1} = f(x_{t-1}, u_t) \quad (8)$$

$x_{t|t-1}$ is the prior state estimate at time t; f is the state transfer function; x_{t-1} is the state at the previous moment; u_t is the control input.

The particle filter module improves the nonlinear characteristics of ice and snow sports, and introduces a multi-physics field coupling model in the importance sampling stage. When the snowboard lands, the Hertz contact theory is used to construct the snow surface interaction model (stiffness coefficient $k=5\times10^4$ N/m³), and the friction coefficient μ is dynamically adjusted in combination with the compression characteristics of snow particles (adaptive changes in the range of 0.03-0.15). In the airborne stage, the law of conservation of angular momentum is strictly followed, and the center of mass trajectory is corrected by constraining the moment of inertia ratio of the limbs relative to the trunk, so that the trajectory error of the jumping action is significantly reduced. The importance sampling weight update formula is:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(z_t \mid x_t^{(i)}) \quad (9)$$

$w_t^{(i)}$ and $w_{t-1}^{(i)}$ are the weights of the ith particle at time t and t-1; p is the observation probability, which refers to the probability of observing z_t in state $x_t^{(i)}$.

The fusion strategy of the hybrid architecture adopts a dynamic probability weighting mechanism: the confidence weight (0.7) of the millimeter-wave radar is adjusted in real time according to the point cloud density and signal-to-noise ratio (in the range of 0.6-0.8), and the IMU weight (0.3) is inversely correlated with its gyroscope zero-bias stability index.

4 Results and discussion

4.1 Study design

The experimental protocol includes three levels of testing procedures: To ensure the reproducibility and rigor of the evaluation, the study design includes detailed specifications on trial repetition, control configurations, and validation protocols. The three-stage experiment comprises the following components:1) Baseline Calibration: Conducted in a controlled environment (−5 °C, 60% relative humidity), a KUKA KR6 R900 robotic arm equipped with a 10 dBsm radar corner reflector executes preset trajectories with linear velocities ranging from 0–15 m/s and angular velocities up to 1080°/s. A total of 300 motion sequences were collected to calibrate the internal parameters of the mmWave radar and to establish the unoptimized system baseline.2) Static Interference Test: Five representative snow conditions (e.g., fresh snow, compacted snow, ice crystal snow) were simulated using a snow density gradient apparatus (0.1–0.4 g/cm³). Each scenario was repeated 20 times under a snowfall intensity of 5–7 mm/h to collect radar intermediate frequency signals and environmental variables, serving as the static clutter reference.3) Dynamic Motion Capture: 30 professional winter sports athletes (across freestyle skiing, alpine skiing, and snowboarding disciplines) performed standardized actions including linear gliding, sharp turning, and airborne rotations. Each action was repeated 15 times. Raw mmWave point clouds, inertial data, and optical motion capture (Vicon, 200 Hz) were synchronously recorded. The study uses both optimized and unoptimized radar pipelines to quantify performance gains.

All trials were conducted within a purpose-built climate-controlled snow chamber (5 m × 8 m × 3 m), with temperature regulation from −30 °C to 25 °C (±0.5 °C accuracy) and snow layers of 10–50 cm. All sensors were synchronized via PTP protocol. A five-fold cross-validation strategy was applied by stratifying athletes across training and testing splits to ensure generalizability and prevent overfitting.

In order to verify the performance of the proposed framework in a real ice and snow environment, this study built a professional test environment with climate control characteristics. The core of the experimental platform is a customized ice and snow environment simulation cabin, which adopts a double-layer insulation structure design. The inner layer is a 5 m×8 m×3 m test space, equipped with a precision temperature control system (temperature control range -30℃ to 25℃, accuracy ±0.5℃) and a humidity control device (relative humidity control range 30%~90%). The floor of the cabin is paved with an artificial snow layer with an adjustable thickness (10~50 cm), and the snow density is controlled in the range of 0.1~0.4 g/cm³ to simulate different snow conditions. The test scenario configuration includes: 1) a multi-angle adjustable millimeter wave radar array (4 77 GHz FMCW radars, bandwidth 4 GHz, maximum output power 10 dBm), installed at a height of 2.5m and distributed in a ring; 2) a reference-level optical motion capture system (12 Vicon Vero series cameras, sampling rate 200 Hz) as the baseline truth value; 3) distributed IMU nodes (9-axis sensors, bandwidth 200 Hz) fixed at the main joints of the subjects; 4) environmental parameter monitoring terminals, real-time recording of variables such as temperature, humidity, and wind speed. All devices are time synchronized through the PTP protocol, and data acquisition is controlled by a unified trigger signal to ensure the time alignment accuracy of multimodal data. During the test, the subjects wore standard skiing equipment and completed the specified action sequence (straight sliding, sharp turns, jumping, etc.), synchronously collecting millimeter wave point clouds, IMU data, and optical motion capture coordinates to construct a multidimensional data set covering different motion states [16]. While the Vicon system provides

high-accuracy ground truth, the framework is compatible with alternative motion tracking modalities, such as wearable IMUs or markerless systems like OpenPose [17], ensuring flexibility in deployment.

4.2 Quantitative indicator analysis

4.2.1 Motion capture accuracy verification

This paper selects 30 ice and snow athletes (including 10 freestyle skiing aerials, 8 alpine skiing downhills, and 12 snowboard big air) to evaluate the algorithm optimization effect of millimeter wave radar in extreme environments. The millimeter wave radar before and after optimization captures the joint positioning RMSE of the athlete's movements and the angular velocity error of the aerial rotation movement. The results are shown in Figures 3 and 4:

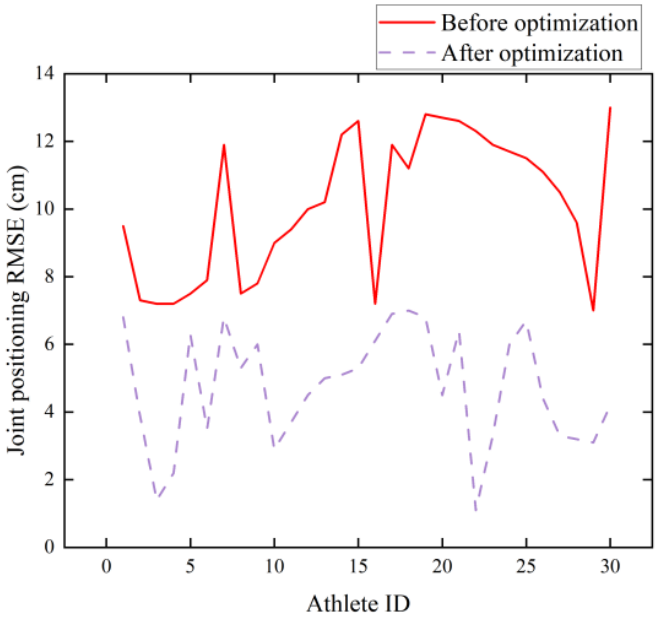


Figure 3: RMSE of joint positioning

As shown in Figure 3, the RMSE comparison data of joint positioning before and after the optimization of the millimeter-wave radar anti-interference algorithm shows that the joint positioning errors in the tests of the top 30 ice and snow athletes are distributed in the range of 7.0-13.0 cm, among which the error of 13.0 cm for athlete No. 30 reaches the maximum value, reflecting the performance bottleneck of the millimeter-wave radar under extreme sports conditions before optimization. After optimization, the error range is compressed to 1.1-7.0 cm through the synergy of the cascaded clutter suppression module and the 3D convolution-LSTM spatiotemporal hybrid network. Athlete No. 22 achieves a breakthrough accuracy of 1.1 cm in straight-line gliding conditions, which is primarily attributed to the enhanced suppression effect of the improved MTI filter on snow and fog-induced multipath interference. For each athlete, the joint positioning RMSE values represent the mean of 15 repeated trials, with standard deviation error bars shown in Figure 3. A two-tailed paired t-test comparing the baseline and optimized systems revealed statistically significant improvements across all athletes ($p < 0.01$), demonstrating that the proposed anti-interference algorithm robustly enhances the motion capture accuracy of millimeter-wave radar under complex snow environments.

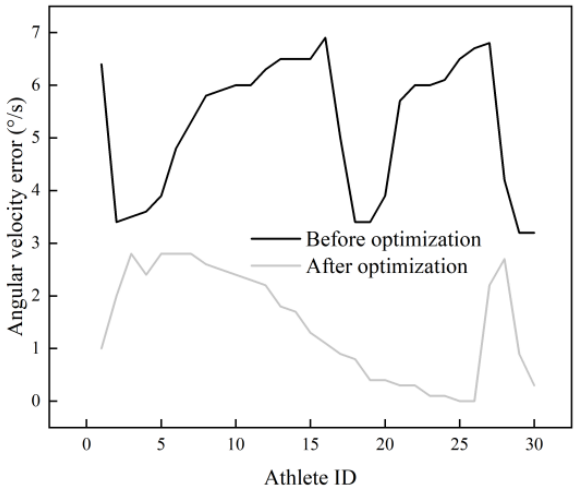


Figure 4: Angular velocity error

As shown in Figure 4, The average angular velocity error of the millimeter-wave radar in the tests of the top 30 ice and snow athletes before algorithm optimization is 5.25°/s (range 3.2-6.9°/s), among which athlete No. 25 had a maximum error of 6.9°/s when completing a 1080° rotation, exposing the phase loss problem of millimeter-wave radar for high angular velocity dynamic tracking. Through the joint optimization of the cascaded clutter suppression module and the 3D convolution-LSTM spatiotemporal hybrid network, the average angular velocity error is significantly reduced to 1.46°/s (range 0-2.8°/s) after optimization, a decrease of 72%, and athletes No. 25 and 26 achieve zero-error tracking. The reduction in angular velocity error was statistically significant across athletes (paired t-test, $p < 0.01$), with error bars in Figure 4 indicating standard deviation over 15 repetitions per action. Experimental results show that the proposed anti-interference algorithm significantly improves the angular velocity tracking accuracy of millimeter-wave radar in ice and snow sports scenarios, especially in difficult rotation movements. However, under extreme conditions, there is still a residual error of 2.8°/s, which is mainly due to insufficient compensation for Doppler frequency shift caused by high-speed movement. In the future, the tracking performance of the system in high-dynamic scenarios will be further improved by introducing an adaptive carrier tracking loop and hardware acceleration processing to meet the stringent requirements for motion capture accuracy.

4.2.2 Anti-interference performance analysis

In response to the extreme weather interference common in ice and snow sports, this study builds a multi-physics field coupling model to test the anti-interference ability of millimeter-wave radar under different meteorological conditions. As shown in Table 2, in the simulated blizzard weather (snowfall>5mm/h) test, the proposed cascaded clutter suppression module shows excellent multipath interference suppression ability:

Table 2: Anti-interference ability test results

Test Scenario	Snowfall Intensity (mm/h)	Multipath Suppression Ratio (dB)	Positioning Error (cm)
Freestyle Ski Aerials	5.8	28.2	3.2
Snowboard Big Air Landing	6.3	25.7	7.0
Alpine Ski Downhill	7.1	26.9	4.5
Cross-Country Ski Curves	5.2	29.4	2.1
Biathlon Shooting	6.0	27.5	3.8

Table 2 shows the anti-interference performance test data of millimeter-wave radar for different ice and snow sports scenes in a blizzard environment. In the range of 5.2-7.1mm/h snowfall intensity, it shows excellent performance in freestyle skiing aerial skills scenes, achieving a multipath suppression ratio of 28.2 dB and a positioning error of 3.2 cm. This achievement is mainly due to the synergy of the cascaded clutter suppression module and the 3D convolution-LSTM spatiotemporal hybrid network. In contrast, the single-board large platform landing impact scene is affected by the 8 G impact acceleration, and the positioning error rises to 7.0 cm, which directly reflects the interference effect of carrier frequency deviation and multiple reflections of the snow layer on the propagation of millimeter-wave signals. Among them, although the dynamic waveform adjustment technology maintains the multipath suppression ratio at 25.7 dB, it still needs to further optimize the phase noise suppression by improving the DPLL loop bandwidth. These quantitative results not only confirm the reliability of millimeter-wave radar in extreme ice and snow environments but also provide a clear direction for subsequent technology iterations, especially for the optimization needs of Doppler compensation algorithms in ultra-high-speed scenarios.

4.2.3 Real-Time verification

The end-to-end processing time from radar signal input to trajectory output is recorded to understand the real-time performance of the millimeter-wave radar. The results are shown in Table 3:

Table 3: Real-time test results

Test Scenario	Processing Delay (ms)	Multi-Target Capacity	Frame Rate (fps)
Freestyle Ski Aerials	24.2 ± 1.5	3 athletes	38
Snowboard Big Air Landing	22.8 ± 1.2	3 athletes	40
Alpine Ski Downhill	21.5 ± 0.8	3 athletes	42
Cross-Country Ski Curves	23.1 ± 1.1	3 athletes	39
Biathlon Shooting	25.6 ± 1.8	3 athletes	36

Table 3 compares the real-time performance of the system across different ice and snow sports scenarios along four key dimensions. In terms of processing delay, all scenarios maintain latencies below 26 ms, with alpine skiing downhill achieving the best performance (21.5 ± 0.8 ms), and biathlon shooting exhibiting the highest delay (25.6 ± 1.8 ms). The multi-target tracking capability consistently supports the simultaneous tracking of three athletes across all test conditions. The system frame rate remains in the range of 36–42 FPS, fully meeting the real-time demands of competitive snow sports motion capture. These results clearly demonstrate the real-time performance advantages of the proposed anti-interference algorithm in complex ice and snow environments, providing a reliable foundation for the practical deployment of markerless motion capture systems in elite athletic training and competition. It is important to distinguish between algorithmic latency and full-system latency. The 8 ms latency reported in Table 1 reflects only the simulation-based execution time of the optimized cascade pipeline, evaluated using FPGA and CUDA acceleration. In contrast, the real-world latencies presented in Table 3 include end-to-end delays such as radar signal acquisition, data transfer, and multi-target processing overhead, resulting in a total system delay of 21–26 ms. Despite this, the system remains within acceptable bounds for real-time snow sports motion capture.

5 Conclusion

This study proposes a cascaded anti-interference architecture to address multipath and frequency offset problems in mmWave radar-based motion capture under snowy environmental conditions. Through the integration of adaptive MTI filtering, genetic sparse array reconstruction, and hybrid carrier tracking, combined with a deep spatiotemporal 3D CNN–LSTM decoding network and multimodal EKF–PF fusion, the proposed system demonstrates significant improvements in accuracy, robustness, and real-time performance.

The main contributions of this study are as follows:

1) A three-stage signal processing pipeline is designed to suppress snow-induced multipath clutter and frequency distortion, improving low-SNR motion signal reconstruction.

2) A novel deep learning-based decoder is developed, leveraging 3D CNN and LSTM to model complex temporal-spatial dependencies in radar point clouds.

3) A multimodal fusion strategy integrating extended Kalman filtering and particle filtering is introduced for robust trajectory estimation in dynamic,

cluttered environments.

The proposed method is validated on both simulated and real-world datasets involving elite snow sport athletes, showing that the system achieves sub-centimeter RMSE accuracy and end-to-end latency below 26 ms across diverse scenarios. In future work, we plan to enhance tracking under extreme dynamics by incorporating event-based vision sensors (e.g., DVS), which can further reduce motion blur and improve delay robustness in high-speed actions [18]. Additionally, integrating edge computing and hardware acceleration (e.g., FPGA optimization) will be explored to further optimize latency for large-scale deployment.

Funding

This study is supported by "Research on the Upgrading and Development of China's Sports Industry Driven by New Productive Forces" (No.SKL-2025-1135).

References

[1] H. Li, S. Qiu, and Y. Ma, “A survey on human activity recognition using millimeter-wave radar,” ACM Comput. Surv., vol. 56, no. 4, pp. 1–36, 2023.

[2] X. Y. Zhang and G. P. Qiu, “Research on human motion capture based on improved LM algorithm and dynamic time warping algorithm,” J. Southwest Univ. (Nat. Sci. Ed.), vol. 46, no. 5, pp. 175–185, 2024. https://doi.org/10.13718/j.cnki.xdzk.2024.05.016

[3] S. F. Li, X. S. Zhang, Y. Guo, X. C. Li, L. Shi, and T. H. Zhan, “Biomechanical study of markerless motion capture technology in FMS squat action,” Med. Biomech., vol. 39, no. S01, p. 513, 2024.

[4] X. H. Li, D. F. Fan, J. J. Feng, Y. Lei, C. Cheng, and X. N. Li, “Systematic review of motion capture in virtual reality: Enhancing the precision of sports training,” J. Ambient Intell. Smart Environ., vol. 17, no. 1, pp. 5–27, 2025. https://doi.org/10.3233/AIS-230

[5] H. Chen, “Human motion capture data retrieval and segmentation technology for professional sports training,” J. Mobile Multimedia, vol. 19, no. 2, pp. 419–436, 2023. https://doi.org/10.13052/jmm1550-4646.1923

[6] B. Teer, “Performance analysis of sports training based on random forest algorithm and infrared motion capture,” J. Intell. Fuzzy Syst., vol. 40, no. 4, pp. 6853–6863, 2021. https://doi.org/10.3233/JIFS-189517

[7] T. Alam, M. Benaida, “Smart Curriculum Mapping and Its Role in Outcome-based Education”, Informatica, vol. 46, no. 4. https://doi.org/10.31449/inf.v46i4.3717

[8] E. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” IEEE Trans. Inf. Theory, vol. 52, no. 12, pp. 5406–5425, Dec. 2006. https://doi.org/10.1109/TIT.2006.885507

[9] Y. Zhang, X. Liu, and J. Wang, “Genetic sparse array optimization for millimeter-wave radar in snow interference environments,” IEEE Trans. Geosci. Remote Sens., vol. 61, pp. 1–12, 2023.

[10] N. Kumar and R. Patel, “Temperature-compensated PLL design for FMCW radar in harsh environments,” IEEE Trans. Circuits Syst. I, vol. 68, no. 5, pp. 2065–2077, 2021.

[11] E. Baccarelli and M. Scarpiniti, “Robust deep filtering architectures for noisy radar environments,” IEEE Access, vol. 11, pp. 22256–22267, 2023.

[12] X. Tang and Q. Song, “Comparative evaluation of radar-based and vision-based human motion capture systems,” Meas. Sci. Technol., vol. 34, no. 2, Art. no. 025109, 2023.

[13] J. Wu, S. Zhao, and Y. Liu, “Deep spatiotemporal modeling with CNN-LSTM for real-time radar-based motion capture,” Pattern Recognit., vol. 131, Art. no. 108885, 2022. https://doi.org/10.13718/j.cnki.xdzk.2024.05.016

[14] A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” in Adv. Neural Inf. Process. Syst., vol. 30, pp. 5998–6008, 2017. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[15] T. Chen, R. Zhang, and K. Huang, “Hybrid Kalman-particle filtering for multimodal sensor fusion,” IEEE Sens. J., vol. 22, no. 9, pp. 8654–8664, 2022.

[16] S. Ahmed, S. Kim, and M. Park, “Snow Sense: A radar-based dataset for motion capture in snowy conditions,” Sensors, vol. 22, no. 3, Art. no. 1011, 2022.

[17] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, “Open Pose: Realtime multi-person 2D pose estimation using part affinity fields,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172–186, 2019. https://doi.org/10.48550/arXiv.1812.08008

[18] G. Gallego, T. Delbrück, and D. Scaramuzza, “Event-based vision: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 1, pp. 154–180, 2022. https://doi.org/10.1109/TPAMI.2020.3008413.

Improved DenseNet-DCGAN for Enhanced Digital Restoration of Embroidery Cultural Heritage

Guiying Dong[1*], Qian Mao[2]
[1]College of Art and Design, Communication University of China Nanjing, Nanjing 210000, China
[2]Library, Nanjing University, Nanjing 210000, China
E-mail: adong118@126.com; maoqian328@163.com
[*]Corresponding author

Keywords: DCGAN, DenseNet, embroidery, image classification, image restoration

Received: July 14, 2025

At present, embroidery image restoration technology still has deficiencies in terms of color uniformity and detail restoration. To address these issues, the study improves the densely connected convolutional network and the deep convolutional generative adversarial network through spatial pyramid pooling, and proposes a novel method for embroidery image classification and restoration. The experimental results showed that the research method largely restored the details and colors of the original image and effectively addressed the uneven color issue. The average prediction accuracy, recall rate, and specificity of the image classification model on Suzhou embroidery, Hunan embroidery, Guangdong embroidery, and Shu embroidery reached 96.3%, 98.5%, and 99.4%, respectively. The structural similarity index of the image restoration model has reached 0.99. The restored image was almost indistinguishable to the naked eye in terms of details, texture, and color. The research method has significant advantages in classifying embroidery images and high-quality restoration tasks, and can provide reliable technical support for the digital protection and intelligent restoration of traditional embroidery cultural relics.

Povzetek: Za klasifikacijo in digitalno obnovo vezenin so razviti izboljšani DenseNet in DCGAN z dodanim SPP, razširjenimi konvolucijami ter CBAM. Izboljšani model skoraj povsem naravno obnovi teksture in barve.

1 Introduction

Embroidery works have attracted countless people's attention with their exquisite craftsmanship, rich patterns, and profound cultural connotations. However, over time, many embroidery artifacts have suffered from natural or human damage, such as fading, breakage, and insect infestation, which seriously threaten the preservation and inheritance of embroidery artifacts [1]. The traditional restoration of Embroidered Cultural Relics (ECR) mainly relies on manual skills. Although this method can finely handle every damage, it is limited by lower work efficiency and dependence on the superb skills of the restorer [2]. In addition, the subjectivity in the manual repair process may also lead to deviations in the consistency and accuracy of the repair effect. In this context, the emergence of Artificial Intelligence (AI) technology, especially Deep Learning (DL) technology, has provided new solutions for the restoration of cultural relics. By training DL models, staff can automatically detect and classify the types of damage to cultural relics, providing a scientific basis for restoration work. At present, many researchers have explored it. For example, Maitin et al. proposed a direct reconstruction technique without image segmentation using DL technology to reconstruct missing architectural elements in Greek temple ruins images from virtual image paintings. This method has successfully reconstructed the missing architectural elements in the images of Greek temple ruins, improving the efficiency of restoration and enhancing the consistency and accuracy of the restoration effect [3]. Alessandro et al. used a trained multidimensional DL neural network to associate color images with X-ray fluorescence imaging raw data to complete the restoration of AI digital cultural heritage, achieving digital restoration of graphic artworks [4].

With the further advancement of DL technology, Generative Adversarial Networks (GANs) have made breakthrough progress in image recognition, providing a good solution for cultural relic image restoration [5]. Praveen et al. proposed a new GAN-based art restoration method to digitally repair damaged artworks and assist in physical restoration. This method performed well in digital restoration and could effectively restore the original appearance of artworks, providing important guidance for physical restoration [6]. Zheng et al. proposed an Example Attention Generative Adversarial Network (EA-GAN) that fuses with reference examples, which addressed the issue of significant reconstruction errors in traditional character restoration methods. Compared with existing internal drawing networks, EA-GAN could obtain the correct text structure through the guidance of additional examples in the "example attention block". The Peak Signal-to-Noise Ratio (PSNR) and

Structural Similarity Index (SSIM) values have increased by 9.82% and 1.82% [7].

In summary, numerous scholars have achieved significant results in cultural relic image restoration. However, there are still issues with GAN in terms of image feature extraction, such as poor network training stability and poor generated image quality. At present, there is relatively limited discussion on embroidery classification and restoration in cultural relic image classification and restoration. Given this, this study innovatively constructs an ECR-Image Classification Model (ICM) based on Densely Connected Convolutional Network (DenseNet) and an ECR-Image Restoration Model (IRM) based on Deep Convolutional GAN (DCGAN). Based on these models, improvements are made by introducing Local Binary Patterns (LBPs), Canny operator edge extraction, and Convolutional Block Attention Module (CBAM). The fusion of these technologies aims to enhance the model's capacity to capture details in ECR images, improve the precise reconstruction of textures and edges during the restoration process, and achieve higher quality ECR image restoration results. The main novelizations and contributions of this paper include: (1) For the first time, DenseNet is combined with Spatial Pyramid Pooling (SPP) and applied to classify embroidery images, improving the recognition performance under cross-style and complex patterns; (2) The structure of the DCGAN generator and discriminator is innovatively adjusted. By integrating a dilated convolutional layer, the receptive field of the model is expanded, which helps to capture image features more comprehensively and achieve high-quality restoration of embroidery texture and color. (3) A large-scale dataset containing eight types of traditional embroidery images is constructed, providing fundamental support for subsequent research. The research results have practical value for the digital inheritance and AI-assisted restoration of traditional embroidery culture.

2 Methods and materials

2.1 Construction of ECR-ICM based on SPP-IDenseNet

ECR image classification is the prerequisite and foundation for ECR image restoration. By classifying ECR images, different embroidery types, styles, and eras can be quickly identified and distinguished, providing a scientific foundation for protecting the cultural relics. This study first explores the classification of ECR images. DenseNet was proposed by Gao et al. in 2017. It is a novel DL model architecture that can establish dense connections between network layers through DenseBlocks, thereby improving the information flow and gradient flow of the network, alleviating the problem of gradient vanishing, and promoting feature reuse [8-9]. The structure of DenseBlock in DenseNet is displayed in Fig.1.

In Fig.1, the connection mechanism of DenseBlock is more aggressive compared to the Residual Network (ResNet). Each layer is connected to all previous layers, providing each layer with a rich input that integrates the features of all previous layers [10]. This design ensures the uniformity of feature map size within DenseBlock and greatly promotes feature reuse through dense connections between layers, enabling the network to learn and transmit information more effectively [11]. However, DenseNet still has certain shortcomings in the image classification process, such as the problem of input image size limitation and the problem of network training not converging [12-13]. Therefore, this study improves it through techniques such as SPP, LBP, and Canny operator, and proposes a novel ECR-ICM model, namely the SPP-IDenseNet model. The training process for the embroidery image classification of this model is shown in Fig.2.

In Fig.2, this study first randomly selects a batch of data from the training set based on a preset batch size, and normalizes it to standardize the standard deviation of the Red-Green-Blue (RGB) color channels for each embroidery image. Subsequently, the normalized image is input into the network for forward propagation to extract features and predict categories. Secondly, by comparing the predicted categories of the network with the actual categories, the value of the loss function is calculated. Next, by adjusting the weights through the backward propagation process of the network, the model's performance is optimized. After completing a batch of training, the system will check if the entire dataset has been traversed. If the traversal is not completed, the model will continue to process the next training batch and repeat the above steps. Once the training traversal of the entire dataset is completed, the model will save the weight parameters of the current round and evaluate whether the predetermined training round has been reached. If the training rounds have not been completed, the model will restart the training process and continue iterative optimization. After reaching the predetermined training round, the model training terminates, and the weight parameters at this time will be used for subsequent image classification tasks. The calculation of the RGB three channel pixel values $Output_R$, $Output_G$, and $Output_B$ of the normalized image is shown in formula (1).

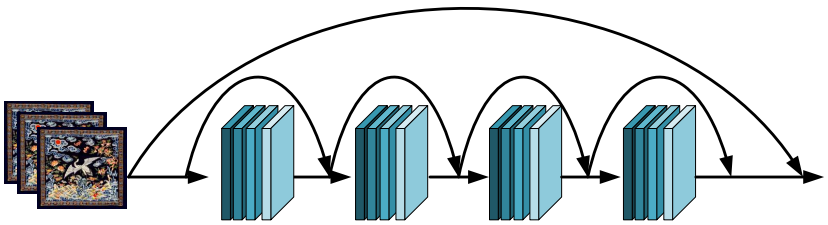


Figure 1: Schematic structure of DenseBlock (Source from: https://colorhub.me/photos/e7RVB).

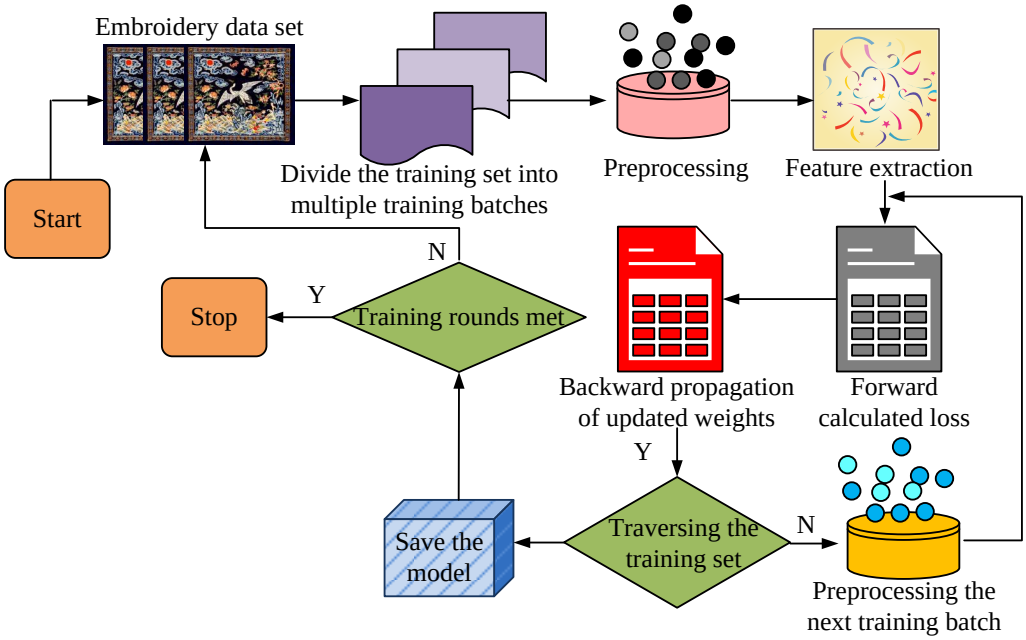


Figure 2: Training process of SPP-IDenseNet model for embroidery image classification (Source from: https://colorhub.me/photos/e7RVB).

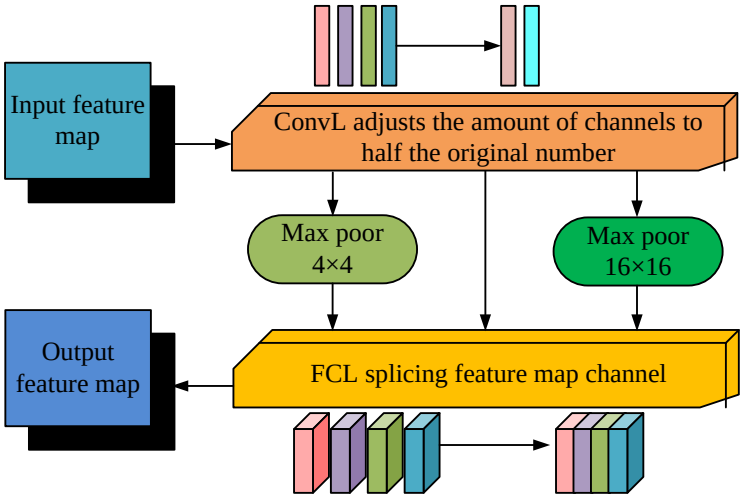


Figure 3: Schematic diagram of SPP structure.

$$\begin{cases} Output_R = \dfrac{Iutput_R - mean_R}{std_R} \\ Output_G = \dfrac{Iutput_G - mean_G}{std_G} \\ Output_B = \dfrac{Iutput_B - mean_B}{std_B} \end{cases} \quad (1)$$

In formula (1), $Iutput_R$, $Iutput_G$, and $Iutput_B$ are the RGB three channel pixel values of the image before normalization processing. $mean_R$, $mean_G$, and $mean_B$ are the mean values of the RGB channels. std_R, std_G, and std_B represent the standard deviation of the RGB three channels. The output feature M_n is shown in formula (2).

$$M_n = F(M_1 \oplus M_2 \oplus \cdots M_{n-1}) \quad (2)$$

In formula (2), n is the hierarchy of the model network. F and $\oplus$ are convolution operations and feature interconnection operations. The loss l during the training process is shown in formula (3).

$$l = L\left(Y_1, Y_1^{'}\right) \quad (3)$$

In formula (3), L is the loss function. Y_1 and $Y_1^{'}$ are the real category and the predicted category. The updated network weight θ' is shown in formula (4).

$$\theta' = \theta - l_r \times \nabla g(l) \quad (4)$$

In formula (4), θ is the network weight before the update. l_r and $\nabla g(\cdot)$ are learning efficiency and derivative calculation. In response to the issue of input image size limitation in DenseNet model image classification tasks, this study uses SPP to enable the model to adapt to input images of different sizes. The structure of SPP is shown in Fig.3.

In Fig.3, this study integrates the SPP structure between the convolutional layer and the Fully Connected Layer (FCL) at the end of the DenseNet model. By dividing the feature map into grids of 1×1, 4×4, and 16×16, and applying max pooling, this study achieves comprehensive capture of features of different resolutions. Subsequently, these multi-scale pooled feature maps will be merged into a fixed-length feature vector, providing rich information for the input of FCL. In addition, by pooling on windows of different sizes, this study generates feature maps with diverse resolutions and fine-tunes the channel dimensions through a 1×1 convolutional layer. The ReLU activation function used in DenseNet may cause neuron deactivation when the input is less than 0 [14]. Therefore, this study introduces the Leaky ReLU function and sets the negative slope coefficient to 0.01, effectively expanding the applicability of ReLU and promoting the stability and convergence of network training. The SPP module enhances the model's understanding of the structural hierarchy of embroidery patterns through multi-scale pooling operations and improves the receptive field coverage of complex patterns. LBP extracts fine-grained texture features from embroidery images, enabling the model to pay more attention to the local texture restoration of the defect area. Canny edge detection provides clear structural contour constraints, guiding the generator to maintain the coherence and integrity of the pattern edges. The three work in synergy, enhancing the quality and stability of image restoration from multiple dimensions, such as structure, texture, and edge.

2.2 Construction of ECR-IRM based on Improved DCGAN

The SPP-IDenseNet model designed above provides strong technical support for the digital restoration and intelligent management of ECR. However, further technological innovation and method improvement are needed in ECR image restoration to achieve more efficient and accurate restoration results. Therefore, this study explores the restoration of ECR images. GAN is a DL model containing two parts: the Generator and Discriminator. Although GANs are widely popular in computer image vision, in traditional GAN architectures, models do not rely on a determined distribution, but instead use internal feedback to adjust their parameters [15]. Although this approach enhances the flexibility of the model, it may also cause training instability and sometimes even lead to model training crashes [16-17]. Therefore, this study further introduces a novel derivative GAN, namely DCGAN. This network can improve the quality of image generation and enhance the learning and representation capabilities of the model by combining the deep architecture of CNN with the GAN framework. The generator extends and reshapes 100-dimensional noise into a 3D feature map through FCL, and then gradually forms the final image size through upsampling and dimension adjustment of transposed convolutional layers. Batch normalization and ReLU are applied after each layer, and the output image is finally activated by Sigmoid to produce a specific tensor image [18-19]. The generator's loss function is shown in formula (5).

$$L_G` = -E_{z\sim pz(z)}[\log(D`(G`(z))] \quad (5)$$

In formula (5), E is the expected operation symbol, usually taken as the average or expected value. z is a noise sample from the latent spatial prior distribution. $G`(z)$ is the data generated by the generator through the noise sample $z$. $D`(G`(z))$ is the output of the discriminator to the data generated by the generator, which represents the probability of real data. At this point, the loss function of the discriminator is shown in formula (6).

$$L_D` = -E_{x\sim pdata(x)}[\log(D`(x_z)] - E_{z\sim pz(z)}[1-D(G`(x_z)))] \quad (6)$$

In formula (6), $D`(x_z)$ means the discriminator's output for the real sample $x_z$, and $D`(x_z)$ is the probability of the real data. Based on the above formulas, compared to traditional GANs, DCGAN uses convolutional and deconvolution layers to replace FCL in traditional GANs. This operation can capture the local structure and spatial message of embroidery images [20]. In addition, DCGAN also uses batch normalization techniques and expected values to accelerate the training process and stabilize the training of GAN. The aim is to further enhance the performance of DCGAN in embroidery image restoration tasks, improve the naturalness of restoration effects, and provide experts with more accurate texture and color information to assist them in more refined restoration work. Given this, the study also improves DCGAN and proposes a new type of ECR-IRM, namely IDCGAN. The overall model structure is shown in Fig.4.

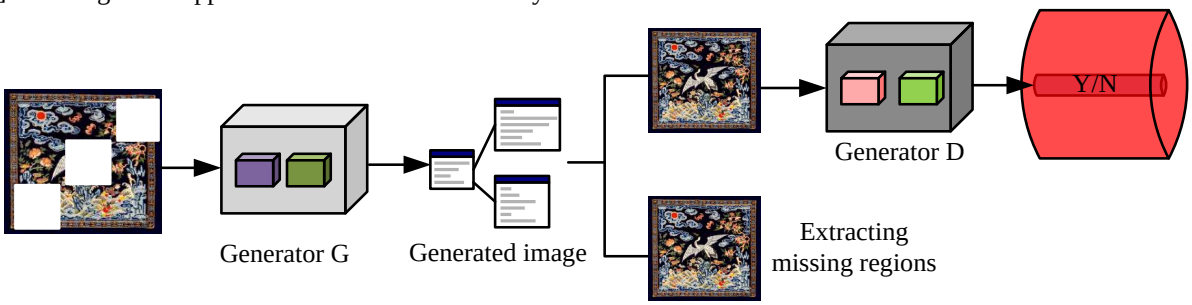


Figure 4: Overall structural framework of the IDCGAN (Source from: https://colorhub.me/photos/e7RVB).

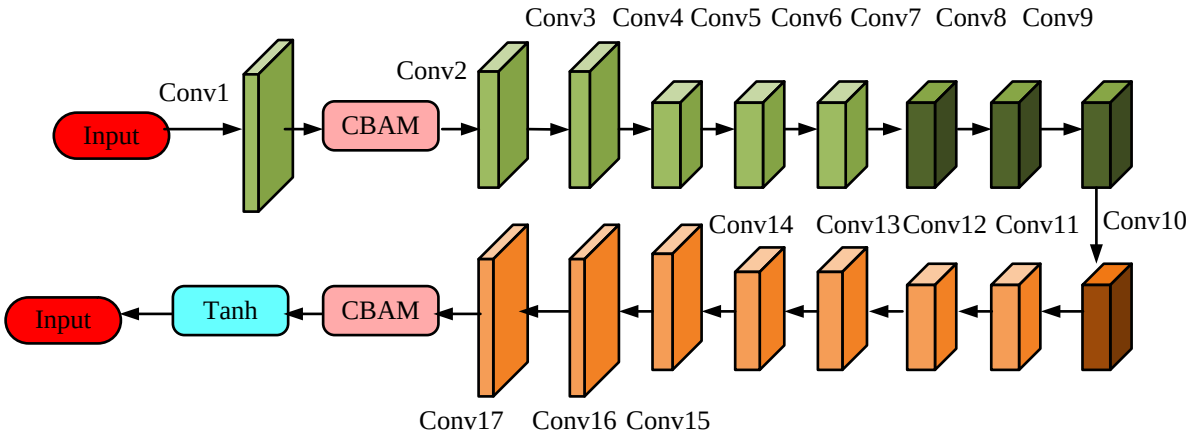


Figure 5: Specific structure of the generator in the IDCGAN model.

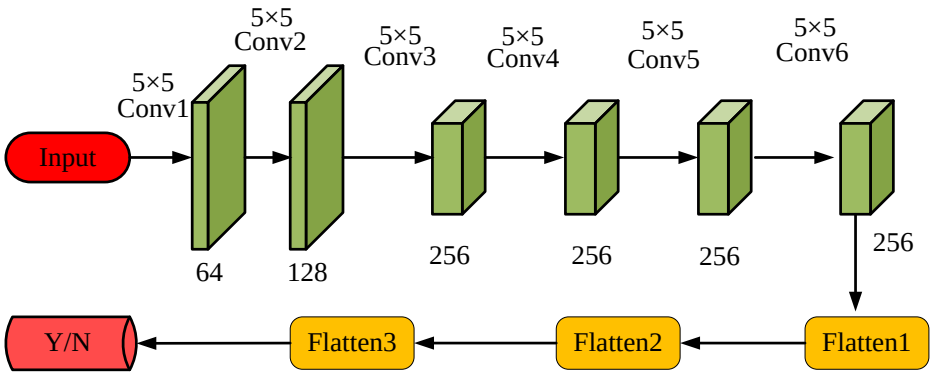


Figure 6: Specific structure of the discriminator in the IDCGAN.

In Fig.4, innovative adjustments are made to the generator architecture by integrating dilated convolutional layers to expand the model's receptive field, thereby helping the model capture image features more comprehensively. At the same time, CBAM is introduced to enhance the attention to key features at both the channel and spatial levels, thereby improving the accuracy of image restoration. The discriminator adopts a strategy of enhancing its depth and increasing the number of FCLs, thereby improving the network's ability to handle complex nonlinear problems, enabling the discriminator to more effectively recognize and distinguish between real and generated images. The loss function combines traditional MSE loss with adversarial loss. The calculation of mean square error loss L_{MSE} is shown in formula (7).

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - g_i)^2 \tag{7}$$

In formula (7), g_i is the predicted value of the model on the training data x_i. The adversarial loss L_{adv} is shown in formula (8).

$$L_{adv} = \min_G \max_D E_{x\sim P_{data}}\left[\log_2 D(x)\right] + E_{z\sim P_{data}}\left[\log_2(1-D(G(Z)))\right] \tag{8}$$

In formula (8), the algebraic meaning remains the same as before. The specific structure of the generator in the IDCGAN model is shown in Fig.5.

In Fig.5, the architecture of the generator in the IDCGAN model mainly consists of three key modules, namely the convolution block, dilated convolution block, and CBAM. Hollow convolution blocks use convolutional layers with different void rates, namely 2, 4, 8, and 16, to achieve multi-scale capture of image features. When the hole rate is set to 1, the hole convolution degenerates into a standard convolution operation. This is reflected in the Conv6 to Conv10 layers of the generator, forming a series of ConvLs with different hole rates that ensure the flexibility and adaptability of the network. The introduction of CBAM adds the ability for dynamic weighting to the generator. It can weight features in both channel and spatial dimensions, highlighting the features that have the greatest impact on image quality. The framework of the discriminator in the IDCGAN model is shown in Fig.6.

In Fig.6, to improve the performance of the discriminator in addressing complex nonlinear problems, this study adds two FCLs based on the original discriminator architecture, making the discriminator contain a total of three FCLs. The interconnection of these layers enhances the discriminator's ability to learn features, thereby significantly improving model performance. Ultimately, the discriminator determines the authenticity of the input image through a binary classification task, distinguishing whether the image was generated by the generator or from a real dataset. The research is conducted based on a self-built embroidery image dataset. The images mainly come from digital museums, high-resolution cultural relic catalogues, and

cultural heritage archives, covering multiple historical periods and diverse embroidery styles. The initial dataset contains 1,800 images. After expansion, the dataset ultimately includes 8,957 images. For the unified model input, the image is cropped and scaled to 256×256 pixels, and normalization processing is carried out simultaneously. Ultimately, the dataset is divided into a training set and a test set in an 8:2 ratio. To simulate the common damage forms of ECR, the study also uses random occlusion to generate defect images. The occlusion forms include rectangles, free-shaped patterns, and speckled textures, and the area ratio is controlled at 10% to 40%. On this basis, image enhancement is carried out by applying methods such as rotation, flipping, scaling, and color perturbation to improve the robustness and generalization ability of the model. In addition, by analyzing the color and style distribution of the images, a balanced sampling strategy is adopted to control the category bias, ensuring the diversity and balance of the training data in terms of pattern style and damage type. All the code modules in the research are built based on the PyTorch framework. Some of the code is as follows:

```
import torch
import torch.nn as nn
# Simple Generator example
class Generator(nn.Module):
def__ init__ (self):
super().__init__ ()
self.net = nn.Sequential(
nn.Linear(100, 256),
nn.ReLU(),
nn.Linear(256, 3*64*64).
nn.Sigmoid()
def forward(self, x):
retum self.net(x).view(-1, 3, 64, 64)
# Simple Discriminator example
class Discriminator(nn.Module):
def__ init__(self):
super().__ init__ ()
self.net = nn.Sequential(
nn.Flatten(),
nn.Linear(3*64*64, 256),
nn.ReLU().
nn.Linear(256, 1),
nn. Sigmoid()
def forward(self, x):
return self.net(x)
# Training example (pseudo code)
# z = torch.randn(batch_ size, 100)
# fake_ images = generator(z)
# real_ output = discriminator(real_ images)
# fake_ output = discriminator(fake_ images)
```

Figure 7: Code.

3 Results

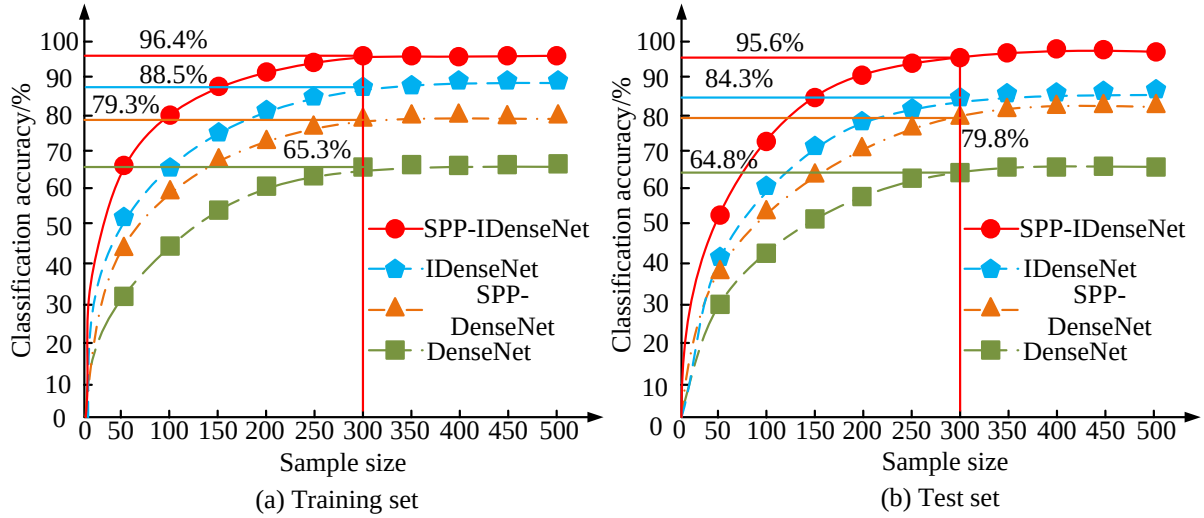
3.1 SPP-IDenseNet model performance testing

The study adopts five-fold cross-validation to evaluate the model's performance. The training set is evenly divided into five subsets of similar size. Four subsets are selected in sequence for model training, and the remaining subset is used as the validation set. This process is repeated five times to ensure that each subset participates in the verification. Through multiple rounds of training and validation, the mean and standard deviation of the accuracy, recall rate, and specificity of the calculation model are calculated, effectively avoiding the randomness brought by a single division and enhancing the statistical reliability and generalization ability of the evaluation results. Table 1 shows the experimental setup and environmental parameters.

According to the settings in Table 1, the effectiveness of the proposed model was first validated through ablation testing, as shown in Fig.8.

Table 1: Environment and parameter configuration.

Serial number	Experimental environments and hyperparameter categories	Settings
1	Num epochs	200
2	Pre-training	No
3	Batch size	20
4	Num class	8
5	Optimizer	Adam
6	Learning rate	0.0001
7	Development Environment	Windows 10
8	CPU	Intel Core i9-10900K
9	GPU	NVIDIA RTX 3090
10	Memory	64GB
11	Graphics Memory	16GB GDDR6X
12	Programming Tools	PyTorch 1.6.0



(a) Training set (b) Test set

Figure 8: Ablation test results of SPP-IDenseNet.

Figs.8 (a) and (b) show the test results of the new model in two datasets. As the test samples continue to grow, the standalone DenseNet module shows lower classification accuracy in both datasets, with the highest being only 65.3%. After introducing the SPP module, LBP, Canny operator, and Gabor filter module successively, the classification effectiveness of the entire model has been significantly improved. The result indicates that when dealing with embroidery images with complex texture features, relying solely on global features for extraction has certain performance bottlenecks. The classification accuracy of SPP-IDenseNet is highest at 96.4% in the training set and 95.6% in the testing set. This study has improved various parts of the DenseNet model to varying degrees for classifying and recognizing ECR images, demonstrating the effectiveness of the improved method. In addition, popular ICMs of the same type, including Lightweight CNN (LCNN), Efficient CNN (ECNN), StyleGAN, and Global Image Spatial Texture (GIST), are introduced as comparative models. Performance tests are conducted using precision, recall, and specificity as indicators, as shown in Table 2.

In Table 2, due to their relatively simplified structures, LCNN and ECNN models have obvious deficiencies in feature expression ability and fine-grained classification. Although the GIST model can capture certain texture information, it is limited by its feature extraction method based on compressed texture description. GIST's recognition ability for irregular shapes and multi-scale patterns is weak, resulting in limited classification performance. The SPP-IDenseNet model has demonstrated superior performance in all four types of embroidery image recognition tasks. This model enhances its feature perception ability for different scales and spatial structures by introducing SPP modules, and combines LBP and Gabor filters to model fine-grained textures, effectively improving the model's ability to recognize the microstructure of embroidery patterns. Meanwhile, the addition of the Canny edge detection operator enhances the ability to capture boundary and contour features, enabling the model to maintain high classification accuracy even in the face of complex background interference. The SPP-IDenseNet model has the highest accuracy rate of 96.3%, the highest recall rate of 98.5%, and the highest specificity of 99.4% on Suzhou embroidery, Hunan embroidery, Guangdong embroidery, and Shu embroidery. These indicators are numerically superior to other convolutional neural network models and have a more balanced distribution across categories. This result demonstrates the adaptability and effectiveness of

the SPP-IDenseNet model in handling the classification task of ECR images. The confusion matrix obtained on the embroidery image classification dataset before and after model improvement is shown in Fig.9.

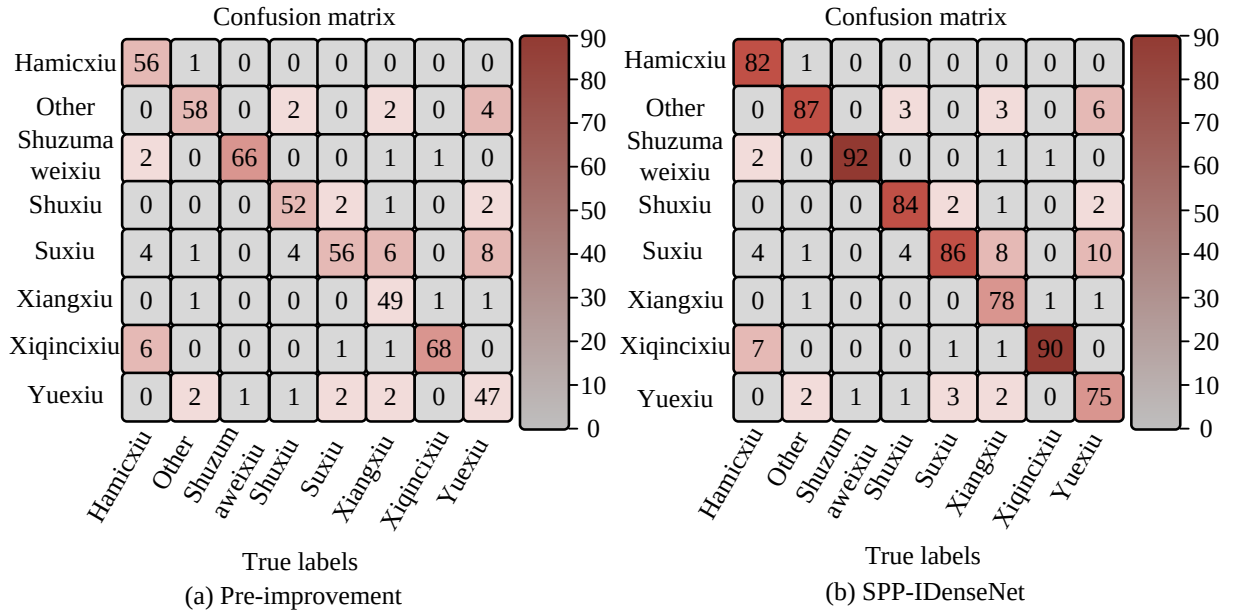
Figs. 9 (a) and (b) show the confusion matrices before and after model improvement. The SPP-IDenseNet model has the highest classification and recognition accuracy for Shui ethnic ponytail embroidery, Xiqin, Hami, Su, Xiang, Shu, and other embroidery in the embroideries image classification dataset. Its classification accuracy in Yue embroidery types is relatively the lowest. Overall, the SPP-IDenseNet model achieves an average prediction accuracy of over 80% for the 8 styles of embroidery images in the embroidery image classification dataset. This indicates that the SPP-IDenseNet model demonstrates strong robustness in handling noise, occlusion issues, and identifying embroidery images with similar features in the embroidery image classification dataset. This robustness makes the SPP-IDenseNet model a powerful tool for ECR image classification, which can effectively address the challenges in practical applications.

3.2 Performance simulation testing of ECR-IRM for IDCGAN

This study uses the Tensorflow DL framework to implement the training and testing of the entire ECR-IRM. The weights β1 and β2 of the Adam optimizer are set to 0.5 and 0.9. The loss changes of IDCGAN generator and discriminator at different network learning rates are shown in Fig.10.

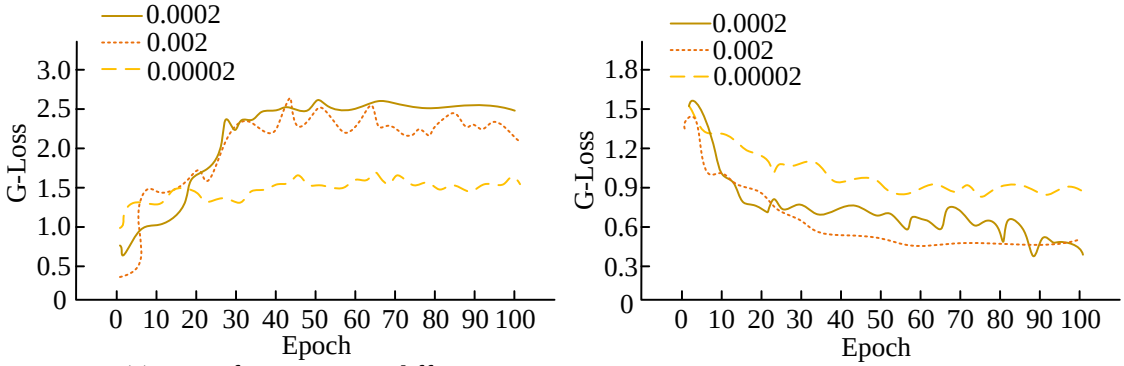
Table 2: Multi-metric performance test results for different models.

Style	Model	Precision/%	Recall/%	Specificity/%
Suzhou embroidery	LCNN	63.5	65.7	80.2
	ECNN	67.2	69.8	81.6
	GIST	70.3	68.7	83.4
	StyleGAN	85.7	87.4	89.1
	Research method	95.8	98.5	94.2
Hunan embroidery	LCNN	55.2	56.3	89.6
	ECNN	58.7	60.4	90.2
	GIST	60.2	61.7	91.6
	StyleGAN	83.4	85.1	92.3
	Research method	96.3	90.2	99.4
Cantonese embroidery	LCNN	57.6	59.8	53.8
	ECNN	66.3	70.4	60.5
	GIST	71.6	69.7	70.8
	StyleGAN	80.2	82.5	75.4
	Research method	95.1	90.8	95.1
Sichuan embroidery	LCNN	58.8	60.5	55.6
	ECNN	62.8	68.8	58.3
	GIST	70.4	73.4	60.7
	StyleGAN	79.8	81.7	69.2
	Research method	92.4	96.7	90.3



(a) Pre-improvement (b) SPP-IDenseNet

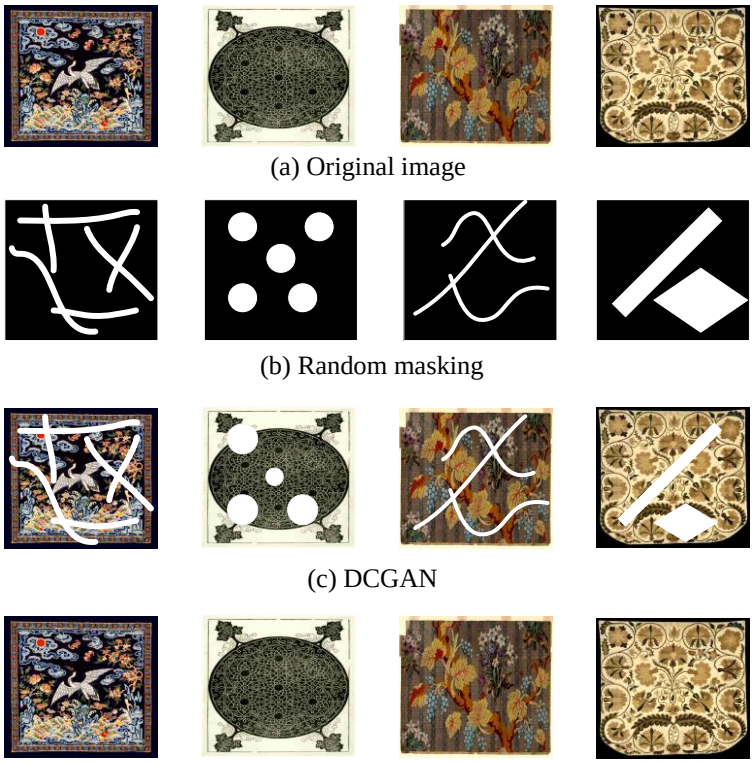
Figure 9: Confusion matrix plots before and after model improvement.



(a) Loss of generators at different learning rates

(b) Loss of the discriminator at different learning rates

Figure 10: Loss variation of IDCGAN between generator and discriminator at different learning rates.



(a) Original image

(b) Random masking

(c) DCGAN

(c) IDCGAN

Figure 11: Repair effects of the model before and after the improvement (Source from: https://colorhub.me/).

In Fig.10 (a), the loss of the IDCGAN generator slowly increases with the growth of training cycles, and the curve with a learning rate of 0.00002 shows a low and stable loss value. The curves with learning rates of 0.002 and 0.0002 show higher loss values and larger fluctuations. In Fig.10 (b), the discriminator loss slowly decreases as the number of training cycles increases. The curve with a learning rate of 0.00002 decreases the fastest and tends to stabilize, indicating that a smaller learning rate helps the discriminator learn more effectively. In contrast, the curves with learning rates of 0.002 and 0.0002 exhibit significant fluctuations and higher loss values. Based on the comprehensive experimental data, this study ultimately sets the network learning rate of the IDCGAN model to 0.00002. To verify the impact of dilated convolutional layers, loss functions, and CBAM on model performance, the repair effect of the improved model before and after random occlusion is compared, as shown in Fig.11.

Figs.11 (a) to (d) show the embroidery original image, images subjected to random occlusion, images restored by the DCGAN model, and images restored by the IDCGAN model. By comparing these images, the effectiveness of IDCGAN in handling different types of embroidery and varying degrees of occlusion can be demonstrated. IDCGAN can enhance the focus on key features, thereby enabling the restored image to largely restore the details and colors of the original image, effectively solving the problem of color non-uniformity. However, DCGAN's repair effect is not ideal when facing large-scale defects, and it cannot maintain good contextual consistency, resulting in poor repair performance. This discovery

validates the necessity of improving the DCGAN. To further test the effectiveness of the research model in embroidery image restoration, the Cycle-Consistency GAN (CCGAN), Conditional GAN (CGAN), and Stacked GAN (Stack-GAN) models are introduced for comparison. The test results of SSIM as the experimental indicator are shown in Fig.12.

Figs.12 (a) and (b) show the SSIM performance comparison of four models in two datasets. Both in the training and testing sets, the IDCGAN model performs the best, followed by Stack-GAN and CCGAN, while CGAN performs the worst. In the training set, the maximum SSIM values for CGAN, CCGAN, Stack-GAN, and the research model are 0.64, 0.72, 0.85, and 0.98, while in the testing set, they are 0.69, 0.78, 0.90, and 0.99. The above data indicates that the research model has significant advantages in maintaining image structure and quality. The reason behind this is that the dilated convolution technique effectively expands the receptive field, allowing it to capture richer contextual information in the image. In addition, CBAM further enhances the model's attention to key features by weighting important features in both channel and spatial dimensions. These improvements have led to significant advantages of IDCGAN in embroidery image restoration. Finally, to confirm the resolution capability of the proposed model, this study also tests four models using image clarity as an indicator, as shown in Fig.13.

Figs.13 (a) to (d) show the clarity performance of CGAN, CCGAN, Stack-GAN, and IDCGAN models in the Yue embroidery image restoration task. Figure 13 (e) shows the clarity of the original image. The Yue embroidery restoration images generated by IDCGAN are visually very similar to the original images, and it is almost impossible to distinguish the quality differences with the naked eye. In contrast, there are significant differences between the restoration results of CGAN, CCGAN, and Stack-GAN and the original images. Especially for the restored images of the CGAN model, there is a significant decrease in clarity in comparison to the original images. In summary, the research model surpasses the comparative model in image resolution for Guangdong embroidery restoration, demonstrating its potential and advantages in embroidery image restoration processing.

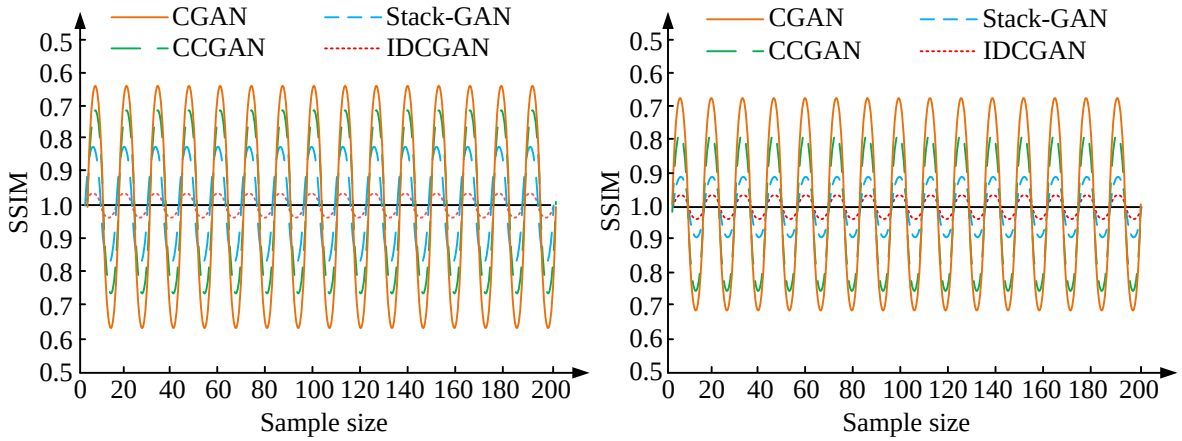


Figure 12: Schematic of SSIM test results for different models.

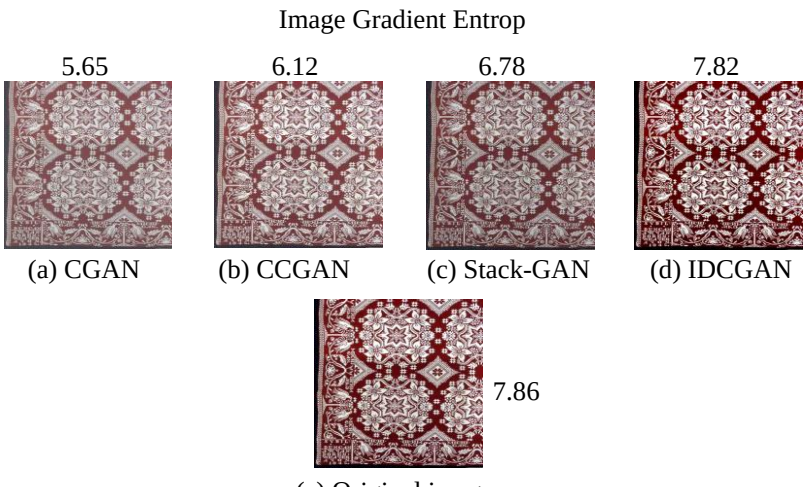


Figure 13: The clarity of restored images of Cantonese embroidery (Source from: https://colorhub.me/photos/VXeo3).

4 Conclusion

The study focused on the task of image restoration of ECR and innovatively constructed an ECR-ICM based on SPP-IDenseNet and an ECR-IRM based on the improved DCGAN. The experimental results showed that the SPP-IDenseNet model achieved an average prediction accuracy rate of over 80% for the embroidery images of eight styles. The IRM could enhance the focus on key features, thereby enabling the restored image to largely restore the details and colors of the original image, effectively solving the problem of uneven color. The SSIM value has reached 0.99. Furthermore, the research model could still maintain an excellent restoration effect even when dealing with large-area damaged embroidery images. The restored image of Cantonese embroidery generated was visually extremely similar to the original image, and it was almost impossible to distinguish the quality difference with the naked eye. The results show that the research model achieves innovation in technology and demonstrates significant advantages in practical applications. However, the research model also has certain limitations. On the one hand, the current models mainly target 2D embroidery images. At present, there is no adaptive research on complex 3D multi-level embroidery structures and heterogeneous multi material embroidery patterns, which limits their promotion and application in high-precision virtual restoration. On the other hand, due to the adoption of a deep generative network structure, the model has a certain dependence on computing resources during the training and inference stages. This may pose practical challenges in resource constrained cultural heritage conservation institutions or mobile deployments. Furthermore, for severely damaged or extremely blurry images, there is still a certain risk of distortion in the structural reconstruction of the research model. Future research can be carried out in the following directions: (1) Expansion of model generalization ability: By integrating 3D reconstruction and multimodal input, the restoration ability of 3D ECR can be enhanced; (2) Enhanced multi-material adaptability: Material perception module or style transfer mechanism can be introduced to achieve texture simulation and reconstruction of heterogeneous embroidery materials; (3) Lightweight deployment optimization: By applying techniques such as model pruning, quantization, and distillation, the network structure is compressed to adapt to edge devices or mobile terminal applications. Overall, the research method provides a feasible and effective technological path for ECR digital protection, which is expected to have practical applications in digital museum construction, virtual restoration of cultural heritage, and reconstruction of cultural creative models.

References

[1] Xinyang Guan, Likang Luo, Honglin Li, He Wang, Chen Liu, Su Wang, and Xiaogang Jin. Automatic embroidery texture synthesis for garment design and online display. The Visual Computer, 37(9):2553-2565, 2021. https://doi.org/10.1007/s00371-021-02216-0

[2] Xiaoli Fu, and Niwat Angkawisittpan. Detecting surface defects of heritage buildings based on deep learning. Journal of Intelligent Systems, 33(1):163-169, 2024. https://doi.org/10.1515/jisys-2023-0048

[3] Ana M. Maitin, Alberto Nogales, Emilio Delgado-Martos, Giovanni Intra Sidola, Carlos Pesqueira-Calvo, Gabriel Furnieles, and Álvaro J. García-Tejedor. Evaluating activation functions in GAN models for virtual inpainting: A path to architectural heritage restoration. Applied Sciences, 14(16):6854-6854, 2024. https://doi.org/10.3390/app14166854

[4] Alessandro Bombini, Fernando García-Avello Bofías, Chiara Ruberto, and Francesco Taccetti. A cloud-native application for digital restoration of cultural heritage using nuclear imaging: THESPIAN-XRF. Rendiconti Lincei. Scienze Fisiche e Naturali, 34(3):867-887, 2023. https://doi.org/10.1007/s12210-023-01174-0

[5] Kanghyeok Ko, Taesun Yeom, and Minhyeok Lee. SuperstarGAN: Generative adversarial networks for image-to-image translation in large-scale domains. Neural Networks, 162(42):330-339, 2023. https://doi.org/10.1016/j.neunet.2023.02.042

[6] Praveen Kumar, and Varun Gupta. Restoration of damaged artworks based on a generative adversarial network. Multimedia Tools and Applications, 82(26):40967-40985, 2023. https://doi.org/10.1007/s11042-023-15222-2

[7] Wenjun Zheng, Benpeng Su, Ruiqi Feng, Xihua Peng, and Shanxiong Chen. EA-GAN: Restoration of text in ancient Chinese books based on an example attention generative adversarial network. Heritage Science, 11(1):55-62, 2023. https://doi.org/10.1186/s40494-023-00882-y

[8] Mihai Bundea, and Gabriel Mihail Danciu. Pneumonia image classification using DenseNet architecture. Information, 15(10):611-619, 2024. https://doi.org/10.3390/INFO15100611

[9] Sherly Alphonse, S. Abinaya, and Nishant Kumar. Pain assessment from facial expression images utilizing Statistical Frei-Chen Mask (SFCM)-based features and DenseNet. Journal of Cloud Computing, 13(1):142-148, 2024. https://doi.org/10.1186/S13677-024-00706-9

[10] Chunyang Zhu, Lei Wang, Weihua Zhao, and Heng Lian. Image classification based on tensor network DenseNet model. Applied Intelligence, 54(8):6624-6636, 2024. https://doi.org/10.1007/S10489-024-05472-4

[11] S. Deepa, Beevi S. Zulaikha, Laxman L. Kumarwad, and Sabbineni Poojitha. Namib beetle firefly optimization enabled DenseNet architecture for hyperspectral image segmentation and classification. International Journal of Image & Data Fusion, 15(2):190-213, 2024. https://doi.org/10.1080/19479832.2023.2284781

[12] Suresh Samudrala, and C. Krishna Mohan. Semantic segmentation of breast cancer images using DenseNet with proposed PSPNet. Multimedia Tools

and Applications, 83(15):46037-46063, 2023. https://doi.org/10.1007/S11042-023-17411-5

[13] M. Karthikeyan, and D. Raja. Deep transfer learning enabled DenseNet model for content-based image retrieval in agricultural plant disease images. Multimedia Tools and Applications, 82(23):36067-36090, 2023. https://doi.org/10.1007/S11042-023-14992-Z

[14] Babu Rajendra Prasad, and Dr. B. Sai Chandana. Human face emotions recognition from thermal images using DenseNet. International Journal of Electrical and Computer Engineering Systems, 14(2):155-167, 2023. https://doi.org/10.32985/IJECES.14.2.5

[15] Ning Wang, Yanzheng Chen, Yi Wei, Tingkai Chen, and Hamid Reza Karimi. UP-GAN: Channel-spatial attention-based progressive generative adversarial network for underwater image enhancement. Journal of Field Robotics, 41(8):2597-2614, 2024. https://doi.org/10.1002/ROB.22378

[16] Noa Barzilay, Tal Berkovitz Shalev, and Raja Giryes. MISS GAN: A Multi-IlluStrator style generative adversarial network for image to illustration translation. Pattern Recognition Letters, 151(16):140-147, 2021. https://doi.org/10.1016/J.PATREC.2021.08.006

[17] Manuel Domínguez-Rodrigo, Ander Fernández-Jaúregui, Gabriel Cifuentes-Alcobendas, and Enrique Baquedano. Use of generative adversarial networks (GAN) for taphonomic image augmentation and model protocol for the deep learning analysis of bone surface modifications. Applied Sciences, 11(11):5237-5247, 2021. https://doi.org/10.3390/APP11115237

[18] Aram You, Jin Kuk Kim, Ik Hee Ryu, and Tae Keun Yoo. Application of generative adversarial networks (GAN) for ophthalmology image domains: A survey. Eye and Vision, 9(1):6-16, 2022. https://doi.org/10.1186/S40662-022-00277-3

[19] Zhiguo Xiao, Jia Lu, Xiaokun Wang, Nianfeng Li, Yuying Wang, and Nan Zhao. WCE-DCGAN: A data augmentation method based on wireless capsule endoscopy images for gastrointestinal disease detection. IET Image Processing, 17(4):1170-1180, 2022. https://doi.org/10.1049/IPR2.12704

[20] Betelhem Zewdu Wubineh, Andrzej Rusiecki, and Krzysztof Halawa. Classification of cervical cells from the Pap smear image using the RESDCGAN data augmentation and ResNet50V2 with self-attention architecture. Neural Computing and Applications, 18(24):1-15, 2024. https://doi.org/10.1007/S00521-024-10404-X

Enhanced Prediction of Tropical Tree Biomass Using Ensemble Models

Qiucai Dang
Zhumadian Preschool Education College, Zhumadian 463000, China
E-mail: Dqc336699@163.com

Keywords: above-ground biomass, below-ground biomass, ensemble stacking, grid search optimization

Received: July 3, 2025

The present paper aims to propose a novel model to investigate its utility in evaluating the beneficial effects of tropical forest biomass. To address the multiplicity of variables, as well as the complexity and nonlinear relationships between them, five Machine Learning (ML) models, namely Gradient Boosting (GB), Extra Trees (ET), XGB, ElasticNet, and Poisson Regression, were employed to concurrently predict both the below-ground and above-ground tree biomass (BGB and AGB, respectively), as well as the total biomass (TB = BGB + AGB). Since the results of the aforementioned models were not entirely satisfactory, an additional model called the Stacking Ensemble (SE) was introduced. Each model can have its parameters optimized by Grid Search with cross-validation to make sure that there is generalization and consistent performance. The data collected were based on 175 trees from 27 ecoregional plots located in the Central Highlands ecoregion of Vietnam. The dataset was processed to investigate the proposed model's ability to predict tree biomass. The study's findings revealed that the proposed method demonstrated strong and efficient predictive capabilities for biomass estimation in forest ecoregions. The Stacking model showed the most significant improvements in the highest R 2 (0.968) and VAF (0.971), and the lowest errors, and MDAPE (23.081 percent), which means that it has a strong ability to predict and minimal bias. However, STD (105.763) was marginally higher; nevertheless, the error and strength of this variation exceeded this variance. Thus, incorporating a Stacking Ensemble (SE) model strengthens the ML approach in predicting forest tree biomass amounts.

Povzetek: Študija predlaga ansambelski model za napoved tropske drevesne biomase, ki združuje pet ML-modelov in optimizacijo z iskanjem po mreži. Stacking Ensemble doseže najboljša napovedovanja ter najnižje napake, kar občutno izboljša oceno nadzemne, podzemne in skupne biomase.

1 Introduction

1.1 The role of biomass

Given that biomass plays an unquestionable role as one of the world's vital sources of energy [1]. The disputing matter is what appropriate model would be able to recognize and prove its traits. Zhantao Song et al. (2024) in their work discussed original visions about the concept of the pyrolysis process of biomass. They argued the contribution of various factors to the challenging anticipation of physicochemical traits by applying machine learning techniques such as Random Forest, gradient boosting decision tree, extreme gradient boosting, in which R^2 was higher than 0.97 for particular surface area biochar anticipation as well as analysis, involving yield as well as N content of biochar [1].

In another study, Jia et al. (2024) exploited machine learning methods to anticipate zeolite-catalyzed biomass pyrolysis, and as a result, the Random Forest algorithm performed the highest prediction with R^2 >0.91 for their suggested models. They concluded that their selected factors and methods based on biomass characteristics can be taken into account as a plausible reference [2].

1.2 Above-ground biomass (AGB)

The term above-ground biomass (AGB) refers to the product of above-ground volume (AGV) and vegetation mass. It is also closely linked to the carbon cycle in global grassland ecosystems. Additionally, accurate estimation of AGB variations is essential for assessing carbon decomposition and its impact on climate change. It is also crucial to screen in situ-harvested AGB data before modeling [3]. Furthermore, AGB is an indispensable factor for evaluating ecosystem health and carbon storage. To estimate AGB, the above-ground volume (AGV) of vegetation is considered a high-priority parameter in research [4].

To estimate AGB variations of China's grassland ecosystems, machine learning algorithms, among which the Random Forest model with R2 = 0.83 (i.e., 83 % of the harvesting AGB variations), and RMSE = 43.84 gm−2, revealed accurate performance in estimating grassland AGB [3]. Mao et al. (2021) in their proposed model proved that structural, textural, and spectral metrics contribute to shrub AGV models. They also suggested a direct reference to specify proper vegetation metrics to screen shrub AGV. The efficiency, accuracy, and low cost

are considered to be the pros of their proposed approach for digital terrain model (DTM) output and AGV estimation; thus, it can bridge the gap between ground-based research and satellite remote sensing [4]. May et al. 2024 obtained spatially complete predictions of biomass in a tropical area. They state that this sort of spatially coherent data about AGB supplied by their model is useful to validate the eco-friendly forest handling, carbon decomposition innovations, and climate change alleviation [5].

1.3 Below-ground biomass (BGB)

Below-ground biomass (BGB) is a significant part of forest tree biomass; however, fewer studies have focused on BGB about forest biomass and carbon. This is largely because the process of measuring BGB in large trees is costly and time-consuming. As a result, researchers often use Above-Ground Biomass (AGB) to estimate BGB by applying a root-to-shoot ratio. For different forest types, researchers have also developed specific direct BGB equations [6].

In a recent study, Oliveira et al. (2024) suggested that predicting peanut BGB using their proposed alternative method—i.e., the multi-output regression (MTR) approach—would enable both researchers and farmers to quantify BGB more accurately. They proposed this method to predict multiple peanut maturity indices at the field level, helping to reduce subjectivity in determining peanut maturity [7].

1.4 Ensemble approaches

Ensemble learning is a potent machine learning technique that reduces overfitting, boosts robustness, and enhances overall performance by combining predictions from several models. Ensemble approaches combine the advantages of multiple algorithms to improve generalization rather than depending on a single model [8]. Stacking, also known as stacked generalization, is a versatile and successful ensemble technique. Stacking mixes different kinds of models, possibly with different architectures and learning strategies [9]. In contrast to bagging (e.g., Random Forest) or boosting (e.g., Gradient Boosting, XGBoost), which combine similar models (typically decision trees). Naik et al. (2022) utilized automated stacked ensemble modelling powered by machine learning for predicting aboveground biomass in forests using multitemporal Sentinel-2 data [10]. A stacking ensemble algorithm was used by Zhang et al. (2022) to reduce the biases in estimates of forest aboveground biomass derived from several remotely sensed datasets [11]. Besides, Jin et al. (2025) evaluated the impact of validation techniques and ensemble learning algorithms on estimating aboveground biomass in forests: a case study of natural secondary forests [12]. To this end, they developed models based on various outcomes, qualified to synchronously anticipate AGB, BGB, and the total amount of tree biomass, i.e., TB, in forest areas, solving the problem of carbon estimation for various forest sites.

1.5 Regression models

It is appropriate to take a brief glimpse at the regression models proposed in the present article:

The Gradient Boosting (GB) model is regarded as a strong ML algorithm for numerical optimization problems. Thanks to Leo Breiman (1998) and Jerome Friedman (2001), GB has been developed. The former used GB for decreasing variance for categorization, and the latter improved it for regression and categorization models. GB algorithms carry out numerical optimization for the models of regression and categorization, repeatedly being approximately directed towards the loss function negative gradient. Due to some complexity, it is impossible to direct precisely towards a negative gradient; normally, a weak learner is applied by a GB model to estimate the extreme decline direction [13].

Extra Trees (ET), a recently developed regression model, is considered to be an ensemble ML algorithm related to decision trees. Originally, ET is the improved form of the Random Forest algorithm for the purpose of regression or categorization performance. The reason that makes the ET regression algorithm more competitive for small-sized sample ML is that it utilizes all data to improve the branches of nodes in decision trees effectively [14]. Wang et al. (2023) in their study provided an efficient ML model utilizing an ET regression algorithm for anticipating the relevant synthesis gas traits in the process of biomass chemical looping gasification, and then compared its ability in prediction between the ET model and traditional ones. In another study, using both RF along with ET al algorithm models, researchers developed a general model to precisely predict the co-pyrolysis of coal and biomass, in which ET performed better [15]. ET is advantageous due to achieving more efficient performance than the Random Forest. Compared to RF, ET does not perform bootstrap accumulation like, i.e. it takes a random subset of data without replacement. Hence, nodes are divided randomly, but not based on the best divisions. Therefore, in the ET regression model, randomness doesn't come from bootstrap accumulation but from the random divisions of the data [16]. According to Roy (2021), RF was introduced to overcome the Decision Tree problems, giving medium variance. Accordingly, ET was proposed when accuracy was more crucial than a generalized model. It also delivers low variance.

Extreme Gradient Boosting (XGB) is another strong, multifaceted ML algorithm used for regression and taxonomy jobs. It is well-known for its exceptional capability to predict performances and deal with intricate datasets. GB involves a series of procedures, preparing models in sequence, based on which the previously produced errors are reformed by each new model. XGBoost is a type of ensemble learning technique that mixes the predictions of various ML models to yield an ultimate prediction that is more precise. Besides that, this algorithm also makes use of decision trees like basic learners during its process. To add more, XGB is intended to efficiently influence processors of high-capacity and approaches of the distribution system [17]. Ayub et al.

(2023) applied an XGB algorithm model on a multi-level factorial design outcome to predict and improve the gasification product, in which the XGB model depicts a good prediction accuracy as well as model optimization analysis. The key characteristics of the XGB are explained as an ability to handle complicated relations in data with regularizing techniques, effectively preventing overfitting; thus, it performs the calculation efficiently due to parallel processing. It considers the usage of decision trees as base learners and then makes use of regularizing techniques for model generalization at a higher dimensionality. XGB, more popularly acknowledged for its efficiency in computations, provides processing efficiency with perceptive analysis of feature significance, as well as deals with missing values smoothly [18].

ElasticNet, being a powerful linear regression technique highly beneficial in ML and statistical modeling, excels traditional linear regression models. It bears the ability to mix the penalties created by both Lasso and Ridge regressions. It is useful in particular when traditional linear regressions struggle with multicollinearity, i.e., when predictors are highly correlated [19]. That is to say, ElasticNet is advantageous due to bearing multi-dimensional datasets, selecting significant traits, and being a more consistent and reliable model where there exists collinearity. Aimed to help in solving problems of regression and developing models' performance, ElasticNet offers effective analytical means for handling multi-dimensional regression. Its common applications include characteristics selection, analysis of regression, and modeling for prediction [20]. The significance of ElasticNet regression includes multicollinearity handling, automatic feature selection, aiding in model interpretability and reducing overfitting, flexible regularization, allowing researchers to control the balance between Lasso and Ridge penalties, robustness in high-dimensional data, appropriateness for a variety of regression problems [15].

Poisson is a regression analysis where its answer is based on the distribution called Poisson. The regression suffers from a limitation of the variance equaling the mean, called Equi dispersion. As a consequence of the assumption being violated, resulting in the biased standard error, the less exact test statistics drawn from the model, and consequently, the obtained conclusions will be less valid. The Poisson regression model, therefore, cannot be used under occurrences of over-dispersion or under-dispersion. Poisson regression is one of the generalized linear models. It finds its main application because it usually happens to model occurrences of the kind that are rarely occurring [21].

The Stacking Ensemble (SE) model makes use of an ensemble generalizing approach through learning, despite the fact that it may lack instructions for appropriate non-hyperparameterized meta-learners. The necessity of applying stacking is when multiple ML methods reveal various advantages for a certain task. In this case, the stacking ensemble method employs a discrete ML technique for specifying the efficient application of various algorithms [22]. For this reason, Arif et al. (2024) developed a model of stacking ensemble, by a non-homogeneous mixture of fundamental models, for accurate yet, at the same time, interpretable prediction of lung cancer prognosis so as to recognize crucial risk factors [18].

The use of DL methods is unquestionably dominant over other traditional methods, particularly in tropical forests biomass research [6]. Although many studies have investigated tree biomass anticipation by applying various models [6], the applied models are well-established. But lack combining models as ML, ensemble, and optimization of hyperparameters approaches. This work adds value by combining them using a meticulously designed Stacking Ensemble specifically designed for predicting AGB, BGB, and TB using a small, real-world dataset from 27 eco-regions in Vietnam. The Fit Index (FI), a stability-focused evaluation metric that hasn't been used in biomass prediction before, is introduced in this study. The proposed approach provides new methodological insights that improve prediction accuracy and generalizability in tropical biomass estimation by combining rigorous preprocessing, multi-target modelling within an ecological context, and systematic hyperparameter tuning through Grid Search. Furthermore, this work differs from earlier black-box DL applications in that it incorporates Shapley Additive Explanations (SHAP) for ecological feature interpretation, which offers important ecological insight. Hence, this study was conducted to serve the purpose of bridging this gap. This subject is an expansion of an ongoing strategy to integrate remote sensing inputs acquired using a satellite or a drone and a source of biomass determinations as measured on the ground in order to develop a spatially superior, and rooted business-time dynamic biomass forecast model. Besides the otherwise plausible analytical foundation of the process, the model is capable of capturing some facets of complex nonlinear responses and enhancing the accuracy of predicting biomass over wider geographical areas and timeframes due to the use of sophisticated Stacking ensembles, enabled by Grid Search and cross-validation. Besides, the climatic variables can be included to forecast the change in biomass distribution in the case of a future climate change scenario, which can provide a significant insight both in forest management and on carbon budgeting. That is to say, designing a new model qualified to anticipate tree BGB, AGB, as well as the total of tree biomass TB (i.e., TB = BGB + AGB) concurrently, will fulfil the requirement of estimating forest carbon. On this account, making use of a community of up-to-date regression algorithms to increase the reliability for the aforementioned parameters estimation, as well as that for the newly proposed model, will assist the progressing literature in the realm of forestry science. The study proposes that integrated ensemble models will anticipate tropical tree biomass better than traditional modeling systems; as a result, the model will be dominant over conventional ones. The study objectives are twofold: firstly, designing a model to concurrently anticipate tree AGB, BGB, and TB, guaranteeing additivity of tropical forests in Vietnam by the names of Dipterocarp and Evergreen Broadleaf, and secondly, cross-validating

errors compared to a traditional model, applying the same dataset as well as anticipators in the mentioned forests.

The rest of this paper is structured as follows. That is, Section 2 discusses detailed methodology, including materials and data used in this work. Section 3 presents numerical analyses, graphical analyses, and experimental results under the heading of results and discussions. Lastly, section 4 summarizes the concluding points in the study.

2 Methodology

The present paper aimed to investigate the efficiency of a state-of-the-art model to be qualified for predicting tropical forest tree biomass effects.

This study was conducted in one of Vietnam's eight highest tropical forests, called the Central Highlands ecoregion. Two main tropical forest categories were selected for the focus of the research, i.e., Dipterocarp and Evergreen Broadleaf (See Fig. 1).

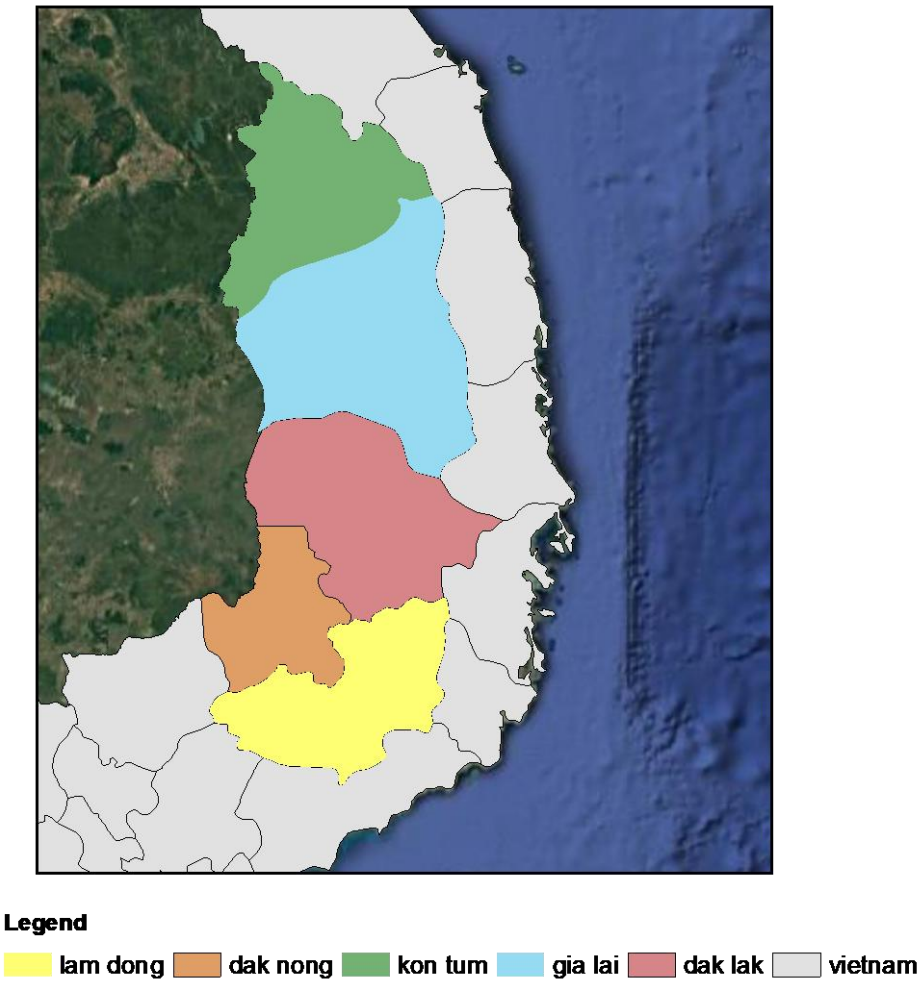


Figure 1: Sample plots, Locations for the forests Dipterocarp and Evergreen Broadleaf in the ecoregion of Central Highlands, Vietnam

In this work, the dataset was exploited in a research study conducted by Huy et al. [6]. The collected data were based on 175 trees from 27 ecoregional lots located in the Central Highlands, Vietnam. We clearly define the dataset partitioning strategy to ensure reproducibility: the entire dataset of 175 samples was randomly divided into training (80%) and testing (20%) sets. Cross-validation was used over iterations to ensure robust evaluation and minimize sampling bias. To ensure compatibility across models and better convergence during training, feature preprocessing involved removing outliers and normalizing all input variables to a [0,1] range using Min-Max scaling. The hyperparameter tuning process was carried out using Grid Search with 5-fold internal cross-validation for each machine learning model: Poisson regression, ElasticNet, XGB, Extra Trees (ET), and Gradient Boosting (GB). This allowed us to systematically explore parameter combinations and choose those that produced the best performance on training data. Based on the results of cross-validation, the Grid Search methodically investigates a predetermined set of hyperparameter values to determine which combination produces the best model performance.

A customized grid of important hyperparameters was built for every model. For instance, tree-based models such as GB, ET, and XGB had their learning rate, maximum depth, and number of estimators adjusted. We adjusted the L1 ratio and alpha (regularization strength)

for ElasticNet. Likewise, pertinent parameters for the stacking meta-learner and Poisson regression were adjusted. In order to maximize generalization and performance consistency, Grid Search was used in a cross-validation framework to guarantee that each model was trained with the best parameter settings. This method greatly enhanced both the Stacking Ensemble's overall performance and the accuracy of the individual models.

The desired targets in this dataset included the amount of above-ground tropical biomass (AGB), the amount of below-ground tropical biomass (BGB), and (TB), namely the total tropical tree biomass; equaling the summation of the below-ground and above-ground tree biomass (i.e., TB = BGB + AGB). Moreover, preprocessing and normalization operations were done on the data.

To serve the purpose of the study, five ML models, as a base learner including GB, ET, XGBoost, ElasticNet, and Poisson, were employed to synchronously anticipate both the amount of below and above-ground tree biomass (BGB and AGB, respectively) as well as the total amount of tree biomass, i.e., TB = BGB + AGB.

Owing to the individual models' mediocre performance, these five models were used as base learners to create a Stacking Ensemble (SE). Following that, a meta-learner was trained using their predictions to generate the final prediction for every biomass component.

For the purpose of assessing as well as selecting the most efficient model able to concurrently anticipate tropical tree BGB, AGB, and TB, a powerful process of cross-validation was carried out.

The Total number of the data was 175 which was randomly split ten times into two sections, involving 140 (80%) for training data, and 35 (20%) for testing data, evaluating impartially. The reason why the data was altered into data testing and training data was to conduct a data analysis satisfying accuracy and reliability in this research. A wide range of assessment metrics, such as MSE, RMSE, MAE, R2, STD, NMSE, MDAPE, and VAF, were used to evaluate performance.

the Fit Index (FI), a goodness-of-fit metric intended to assess the quality of predictions across several cross-validation realizations. A higher FI value indicates a better fit, with values approaching 1. The formula for calculating the FI is presented below.

$$FI = \frac{1}{k}\sum_{1}^{k}(1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y}_i)^2} \quad (1)$$

In the equation above k stands for the realizations number (in this study k = 10), m is the number of trees sampled in the validation dataset; and yi is the observed value. $\hat{y}$i represents the predicted value, and $\bar{y}$ shows the averaged value for BGB, AGB, and TB of the ith validated tree in the realization of kth.

The study goal was to evaluate accuracy and model consistency in light of the ecological context and the small dataset size. Metrics like R2 and VAF measure the percentage of variance explained by the models, while MSE, RMSE, and MAE quantify absolute prediction errors. Understanding normalization effects and error distribution is aided by STD and NMSE. MDAPE is a reliable percentage-based metric that works especially well with data that contains outliers or skewness, which is typical in biomass measurements. A new and comprehensible metric designed for model comparison across several validation folds, the Fit Index (FI) was introduced to reward accuracy and stability. Ultimately, when combined, these metrics make sure that the assessment covers robustness, interpretability, and predictive accuracy—all of which are critical components for ecological modelling and decision-making, demonstrating that the Stacking Ensemble model was more efficient than the other compared models. Evaluation methods for error metrics criteria are exhibited in Table 1.

Table 1: Equations for evaluation of statistical metrics criteria

Statistics	Name	Equation
MSE	Mean Squared Error	$MSE(y,\hat{y}) = \frac{\sum_{N-1}^{i=0}(y_i - \hat{y}_i)^2}{N}$
RMSE	Root Mean Square Error	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$
MAE	Mean Absolute Error	$MAE = \frac{\sum_{n}^{i=1}\lvert y_i - \hat{y}_i\rvert}{n}$
R^2	Determination Coefficient	$R^2(y,\hat{y}) = 1 - \frac{\sum_{N}^{i=1}(y_i - \hat{y}_i)^2}{\sum_{N}^{i=1}(y_i - \bar{y})^2}$
STD	Standard Deviation	$STD = \sqrt{\frac{\sum_{n}^{i=1}(x_i - \bar{x})^2}{n-1}}$
NMSE	Normalized Mean Square Error	$1 - \frac{\lVert x - y\rVert_2}{\lVert x - \bar{x}\rVert}$
MDAPE	Median Absolute Percentage Error	$median\left(\left\lVert\frac{e_1^{abs} - e_1^{\widehat{pre}}}{e_1^{abs}}\right\rVert\right) * 100\%$
VAF	Variance Account Factor	$\left(1 - \frac{var(t_n - y_n)}{var(t_n)}\right) * 100$

n represents observations number, y_i is ith observed value, $\hat{y}_i$ shows ith predicted value, and $\bar{y}$ is the observations average.

Graphical analyses were also carried out to assess the accuracy of the recommended model performance. Illustrated in different plots, they give the reader illuminating perceptions of the suitability and accuracy of the models, all which have been discussed in section 3 of this paper.

As an overview of the research, the general flowcharts of the whole study have been demonstrated below (See Fig. 2 and Fig. 3). Fig. 2 also illustrates a brief comprehension of the step-by-step research methodology. That is to say, the research process begins with the dataset, going through analyzing and normalizing them, next, dividing the normalized data into train and test. More important part is here where the proposed ML models are evaluated based on an array of specific metrics to opt an appropriate model which is appeared to be Stacking Ensemble. Finally, ensemble models are also assessed on the basis of evaluation metrics to choose the best one. Hence, the results are saved for future usage.

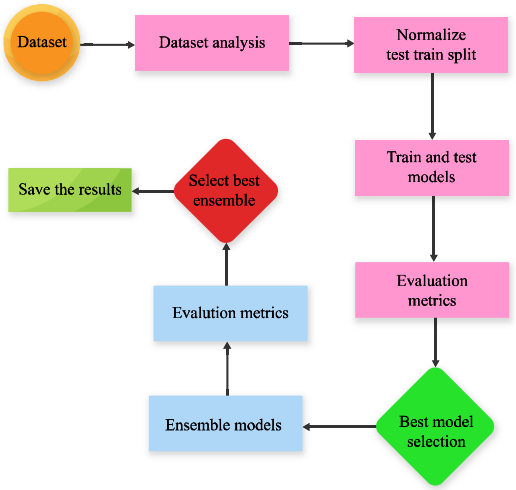


Figure 2: General flowchart of the whole research process for applying the proposed model

Figure 3 shows the modeling procedure involving data collection process for the purpose of theory, and then applying six ML models to concurrently predict tropical forest tree biomass and specifying the best reliable model for such prediction by comparing the selected ML models with the aid of evaluation metrics.

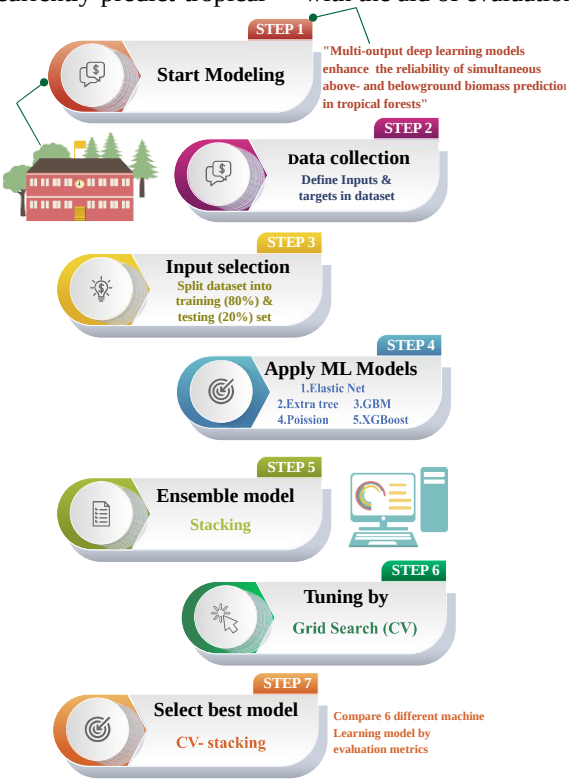


Figure 3: Flowchart of modeling procedure showing the process of employing ML models concurrently

The flowcharts of the six regression models, including ElasticNet, GB, ET, XGB, Poisson, and SE have been illustrated respectively in the following figure. The optimization of hyperparameters is also utilized through Grid Search tuning. These models were utilized with the goal of synchronously predicting ABG, BGB, and TB = BGB + AGB. Additionally, each proposed regression will also be discussed briefly below.

2.1 Base machine learning models

ElasticNet regression is an extension of linear regression that integrates both regularizing penalties of Lasso (abbreviated as L1) and Ridge (abbreviated as L2) into the loss function. This combination allows ElasticNet to deal with circumstances where there are a large number of characteristics, and they are also highly correlated. Its mathematical formulation is shown below.

$$ElasticNet = \sum_{i=1}^{n}(y_i - y(x_i))^2 + \alpha \sum_{j=1}^{p}|w_j| + \alpha \sum_{j=1}^{p}(w_i)^2 \quad (2)$$

In Elastic Net regression, the parameters alpha and l1_ratio bear significant roles in specifying the regularization technique used in the model. These parameters control the trade-off between the L1 and L2 penalties. In the presented formula α is the regularization strength parameter in ElasticNet. It supervises the whole strength of regularization applied to the model. For $\alpha = 0$, no regularization is applied, and Elastic Net equals Ordinary Least Squares (OLS) regression. For $\alpha = 1$, regularizations of both L1 and L2 are applied, blending their penalties. For 0<α<1, this model employs a mixture of L1 and L2 regularization, permitting a flexible mixture of penalties. L1 Ratio (l1_ratio) is the blending parameter that identifies the balance between L1 and L2 penalties. It controls the proportion of the penalty determined to the L1 norm relative to the L2 norm. For l1_ratio=0, the model applies only regularization of L2 (which equals Ridge regression). For l1_ratio=1, it uses merely regularization of L1 (which equals Lasso regression). For 0<l1_ratio<1, Elastic Net applies a mixture of both L1 and L2 regularization, allowing for a combination of penalties [23].

As shown in Fig. 4, applying the Elastic Net model in this study involves several linear steps. Because of the multicollinearity between predictors that are specific to trees and sites, ElasticNet regression was used. It made feature selection and coefficient shrinkage possible at the same time by combining L1 and L2 penalties. Grid Search with 5-fold cross-validation was used to optimize the regularization parameters (l1_ratio and alpha), which enhanced the generalization and stability of the model.

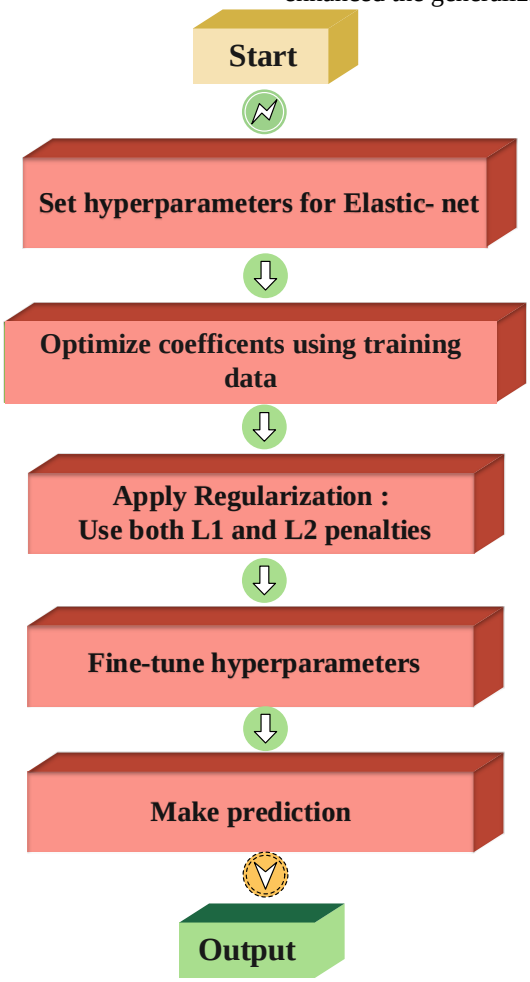


Figure 4: The steps of the ElasticNet model applied for predicting tropical tree biomass

Extra Trees (ET) regression is a type of ensemble ML strategy that accumulates the outcomes of various decision trees decorrelated, similar to Random Forest regression [24]. ET regression employs a conventional top-down technique to create a collection of regression trees. RF model applies two stages, respectively, including bootstrapping and bagging [25].

Hameed et al. (2021) have discussed the Random Forest model as an array of decision trees in their article, and used its equation as follows:

$$T(x, \theta_1, \ldots, \theta_r) = \frac{1}{R}\sum_{r=1}^{R} T(x, \theta_r) \quad (3)$$

In which T (x, θr) demonstrates the Tth tree prediction, in which θ presents a uniform independent distribution vector appointed before the tree growth. All these trees are blended and averaged in an ensemble of them (i.e., shaping forest), named T(x).

According to Hameed et al. (2021), two main differences between the ET and the RF systems are cited as follows. Firstly, two main differences between the ET and the RF systems are cited as follows. Firstly, ET exploits all the divided nodes as well as cutting points, selecting randomly from the cut points. Secondly, the algorithm applies all the samples to help the tree grow so as to limit bias.

Two parameters involved in the ET model for controlling the splitting process are k and nmin; k represents the characteristic number, chosen randomly in the node, while nmin refers to the minimum size of the sample anticipated nodes division. Moreover, respectively via k and nmin, the feature selection strength and the average strength of output noise are specified. The abovementioned two factors enhance the accuracy and decrease the ET model overfitting [19].

Figure 5 demonstrates the design of the ET model proposed in the present study. According to Fig. 6, training data are processed through the ET model as follows. N predicted outputs result from N tree decisions. Consequently, the obtained output predictions are averaged to result in the optimal output.

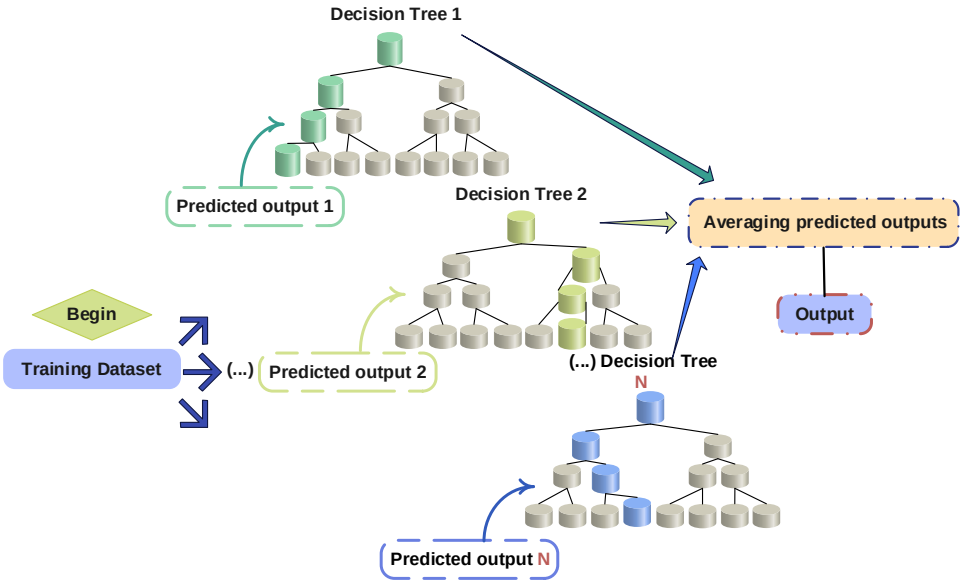


Figure 5: The steps of Extra Trees (ET) model used for predicting tropical tree biomass

The Gradient Boosting (GB) algorithm's job is to find a function $T(x_i)$, minimizing some loss function $\mathscr{L}$ [$T(x_1),\ldots, T(x_n)$], in which x_i is a vector with k dimensions for i=1,…, n. This algorithm begins with a primary prediction $T_0(x_i)$ and carries on repeatedly so that $T_m(x_i) = T_{m-1}(x_i) + h_m(x_i)$. Supposedly, $h_m(x_i)$ would experience extreme reduction direction in $\mathscr{L}$ [$T_{m-1}(x_1),\ldots,T_{m-1}(x_n)$]. Such direction is delivered via $\mathscr{L}$ with a negative gradient assessed in [$T_{m-1}(x_1)\ldots, T_{m-1}(x_n)$]. Despite being very demanding or sometimes impossible for a function detection of $h(x_i)$ to approximate $\mathscr{L}$ assessed in [$T_{m-1}(x_1),\ldots, T_{m-1}(x_n)$], when $h(x_i)$ approximation is estimated, GB algorithm progresses through $T_m(x_i) = T_{m-1}(x_n) + \alpha h_m(x_i)$ for $\alpha>0$ which is a supposed learning rate [9].

As the GB model is represented in Fig. 6, the data first goes through a bootstrap sampling to be split into T data subsets for which there would be hT tree decisions. Thereafter, there is a one-to-one result for each tree decision; altogether, hT(x) results. Ultimately, the attained results' average is calculated to produce the final result, namely H(x). The ability of Extra Trees (ET) and Gradient Boosting (GB) to model intricate non-linear relationships—which are common in ecological systems—led to their selection. With a small dataset size (n=175), ET's randomized split selection was especially helpful in reducing overfitting. To attain the best bias-variance tradeoffs, we employed Grid Search to adjust variables like the number of estimators, tree depth, and leaf size.

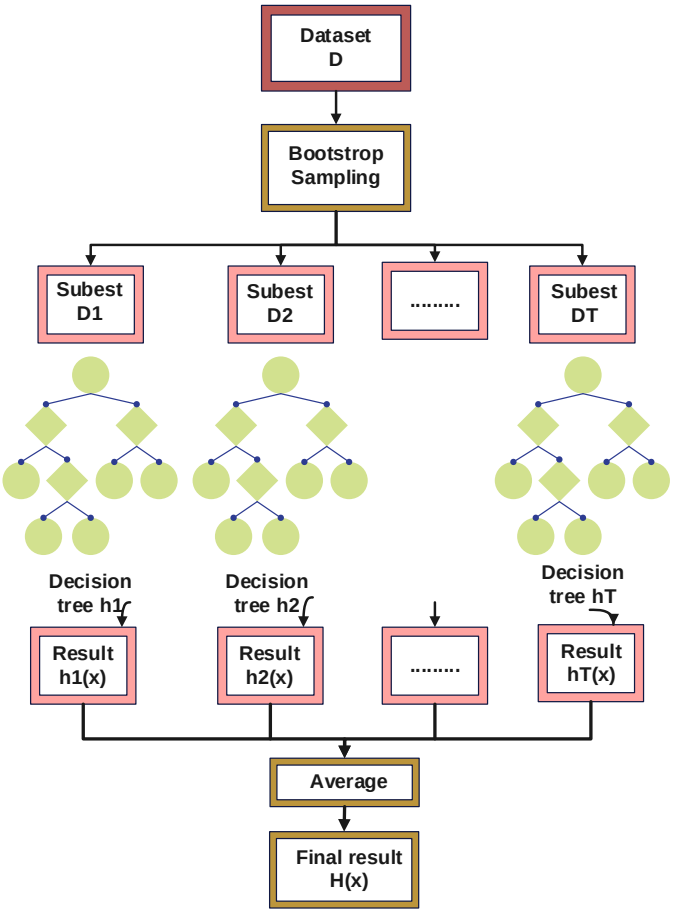


Figure 6: The stages of Gradient Boosting (GB) model employed for tropical tree biomass predictions

Given that the Poisson distribution is the basis for Poisson regression, it identifies the probability of the occurrence of any number of events in a steady interval of time or space, with the assumption that the events occur at a constant rate and are independent of each other. The formula for the Poisson distribution is given by:

The Poisson distribution could be calculated from the formula below:

$$P(X = k) = \left(\frac{\lambda^k e^{-\lambda}}{k!}\right) \quad (4)$$

In the above equation, X is the number of random occurrences, and λ represents the average or mean of the events. Poisson regression exploits its distribution to provide a comprehension of the predictor variables' relationships along with that of the count data in the dataset. In this regression, the expected value (mean) of the count variable (namely Y) is designed as a model of a linear mixture of predictor variables (namely X):

$$\lambda = \exp(\beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta nXn) \quad (5)$$

in which: λ is the expected count, which represents the occurrence proportion, β0 is the intercept term, β1, β2, …, βn represent coefficients related to each predictor variable. The link function in Poisson regression is the natural logarithm (log-link), ensuring the predicted values are not negative. This model is evaluated via maximum likelihood estimation, and the coefficients (β) are specified to maximize the probability of observing the actual count data in the model [26].

The approach for the application of the Poisson model is well-illustrated in Fig. 7, which experiences various stages in a linear pattern. To begin with, a point cloud is taken as an input; second, the surface normal of all the points is detected by computing the eigenvector over the k-nearest neighbors of each point. Third, an octree with a predefined depth d is selected for categorizing the reconstructed surface. Then, the Gradient of the indicator function (Vx) equated to the vector V is defined by the point cloud. The next stage involves defining an indicator function X with the value of 1 inside and 0 outside the surface. Thus, Vx=V and the divergence operator is applied to either side; i.e. $\Delta x \equiv \nabla.\nabla x = \nabla.V$. On the next stage, the indicator function x is solved as a standard Poisson problem. The marching cube algorithm is used to extract the surface from the solved indicator function x. Eventually, the reconstructed surface is stored in the octree of depth d. Since AGB, BGB, and TB are skewed and non-negative, Poisson regression was employed. Although it was initially created for count data, its formulation fits biomass distributions quite nicely. To make sure the Poisson model's assumptions held true in this situation, diagnostic tests were conducted.

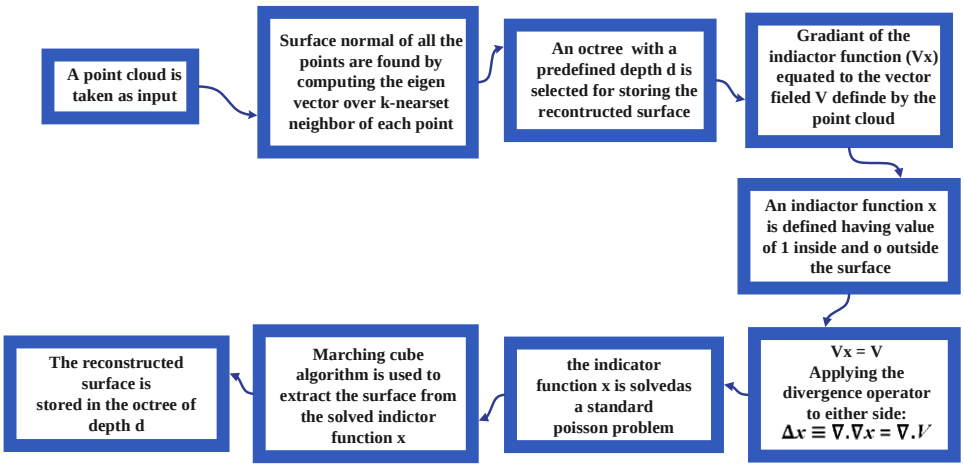


Figure 7: The stages of the Poisson model employed for tropical tree biomass predictions

2.2. Hyperparameter tuning

Grid Search was applied in a 5-fold cross-validation framework to find optimal hyperparameters for all the models. For tree-based models like Gradient Boosting, Extra Trees, and XGB, the number of estimators, learning rate, and max tree depth were systematically changed to optimize model complexity and predictability. In the ElasticNet scenario, the regularization parameter (alpha) and L1 fraction were tuned to prevent overfitting and cause sparsity. In Poisson regression, tuning was for the regularization parameters and the number of iterations to achieve better convergence. After separately tuning each of the base models, their outputs were fed into a meta-learner in the Stacking Ensemble, whose parameters also were tuned via Grid Search. This broad tuning process ensured that all models, including the ensemble, reached optimal generalization and performance [27].

The Stacking Ensemble was selected due to the meta-learner included, as it blends heterogeneous base learners with varying predictive ability and error behaviors, as well as generalizes well. Due to its nature of broad application in addressing multicollinearity on regression prediction of the base models and capturing of non-linear relationships, a tree-based learner was applied as the meta-model in this research. Stacking model showing an RMSE of 18.298, MAE of 12.422, and R2 of 0.968 performed significantly better compared to any of the base models on the test data, meaning that the ensemble was able to selectively leverage the strengths of each of the related models to generate more stable and accurate predictions of biomass.

In the Stacking model, presented in Fig. 8**,** training data are processed on the basis of three level 0 models separately. Each model's prediction results are gathered as other processed training data in the study. All of the base learners' predictions (GB, ET, XGB, ElasticNet, and Poisson) were aggregated by the Stacking Ensemble. To avoid overfitting and information leakage, the meta-learner, a Ridge regression model, was trained on out-of-fold predictions. We were able to improve overall predictive performance by combining the complementary strengths of all models—capturing distribution-specific, linear, and non-linear trends—into this ensemble. Table 2 provides the hyperparameters chosen by the stacking meta learner for the models.

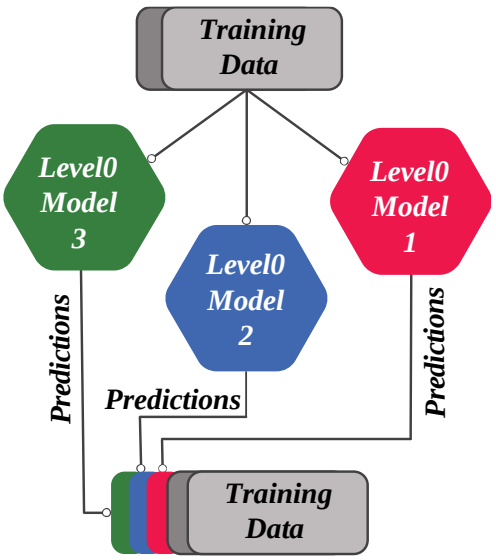


Figure 8: Stacking model procedures used for tropical tree biomass

Last but not least, XGB is one of the common algorithms in ML. It is based on the ensemble learning framework, following the gradient boosting algorithm. Thus, it is applicable for the tasks of supervised learning, i.e., regression, ranking, and categorization. XGB is a predictive model that combines multiple individual models' predictions iteratively. It works by adding weak learners into the ensemble one after another, such that at every step, a new learner tries to correct the errors of the prior ones. It also minimizes a prespecified loss function during training data using some sort of gradient descent optimization [13].

In summary, the XGB is developed in three stages straightforwardly: First, a primary model, namely F0, was used to predict, i.e. the aimed variable. The XGB model is related to a residual (y–F0). Second, the residuals obtained in the prior stage are adapted to a new model called h1. Third, the combination of F_0 and h_1 delivers F_1, which is the promoted form of F_0. Consequently, the MSE metric system from F1 will be lower than that from F_0.

$$F_1(X) < -F_0(x) + h_1(x) \tag{6}$$

For improving F1's performance, a residuals model of F1 can be designed, and an original model called F2 is presented.

$$F_2(X) < -F_1(x) + h_2(x) \tag{7}$$

This process would be iterated for a number of 'n' stages up until potentially minimizing residuals as much as probable, i.e.

$$F_n(X) < -F_{n-1}(x) + h_n(x) \tag{8}$$

It is worth mentioning that additive learners would not mess with the functions developed in prior iterations but add information of their own in order to bring down the error values. First, the model begins with some function F0(x). This F0(x) needs to minimize the loss function or MSE, hence:

$$\begin{aligned} &F_0(x) = argmin_\Upsilon \textstyle\sum_{i=1}^{n} L(y_i, \Upsilon) \\ &argmin_\Upsilon \textstyle\sum_{i=1}^{n} L(y_i, \Upsilon) = \\ &argmin_\Upsilon \textstyle\sum_{i=1}^{n} (y_i - \Upsilon)^2 \end{aligned} \tag{9}$$

Regarding the prime differential of this equation with γ, it is observed the function is minimized at the mean i=1, …, n. Thus, the promoting model can proceed with:

$$F_0(x) = \frac{\sum_{i=1}^{n} y_i}{n} \tag{10}$$

F0(x) presents the first step of predictions in this model. Next, for each instance, the residual error is expressed as: (yi – F0(x) [28].

In Fig. 9, the XGB model employs a multifaceted approach to make predictions about input data. Afterwards, the average of predictions is calculated and an ultimate XGB prediction is thus generated .Because of its exceptional performance with structured tabular data and its integrated regularization, which helps avoid overfitting, XGB was included. It was well-suited for this task because of its efficient handling of non-linearities, support for missing values, and robustness to noise, even with the small sample size. Grid Search was used to optimize important hyperparameters, such as learning rate, maximum depth, and gamma.

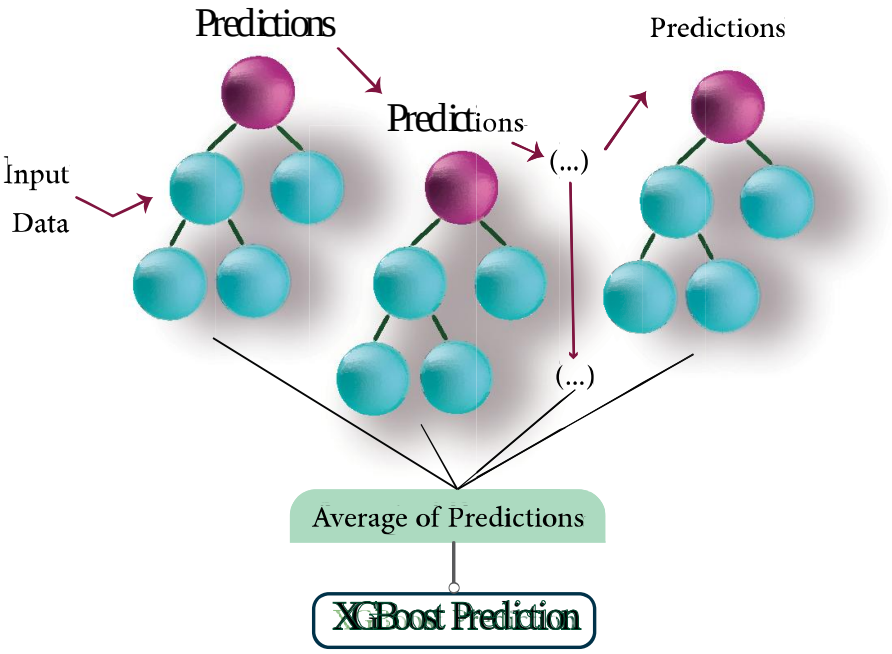


Figure 9: The procedure used in the XGB model for predicting tropical tree biomass

3 Results and discussion

3.1 Exploratory data analysis

To display how closely related the multiple variables of the study data are, a Pearson correlation heatmap is exploited as an effective color-coded visual matrix (See Fig. 10). Variables are demonstrated with rows and columns, and the cells define the relationship between variables two by two. The color shading for each cell indicates the correlations' direction and their strength: the darker the color of a cell, the stronger the correlation of the related variables. As it is obvious from this tabulated heatmap, the colors are darker for stronger correlations and lighter for weaker ones. Additionally, the green colors represent the positive correlations; that is, when one variable increases, the other variable tends to go up, whereas in the case of negative correlations, when one variable increases, the other variable tends to drop. Purple colors have been used.

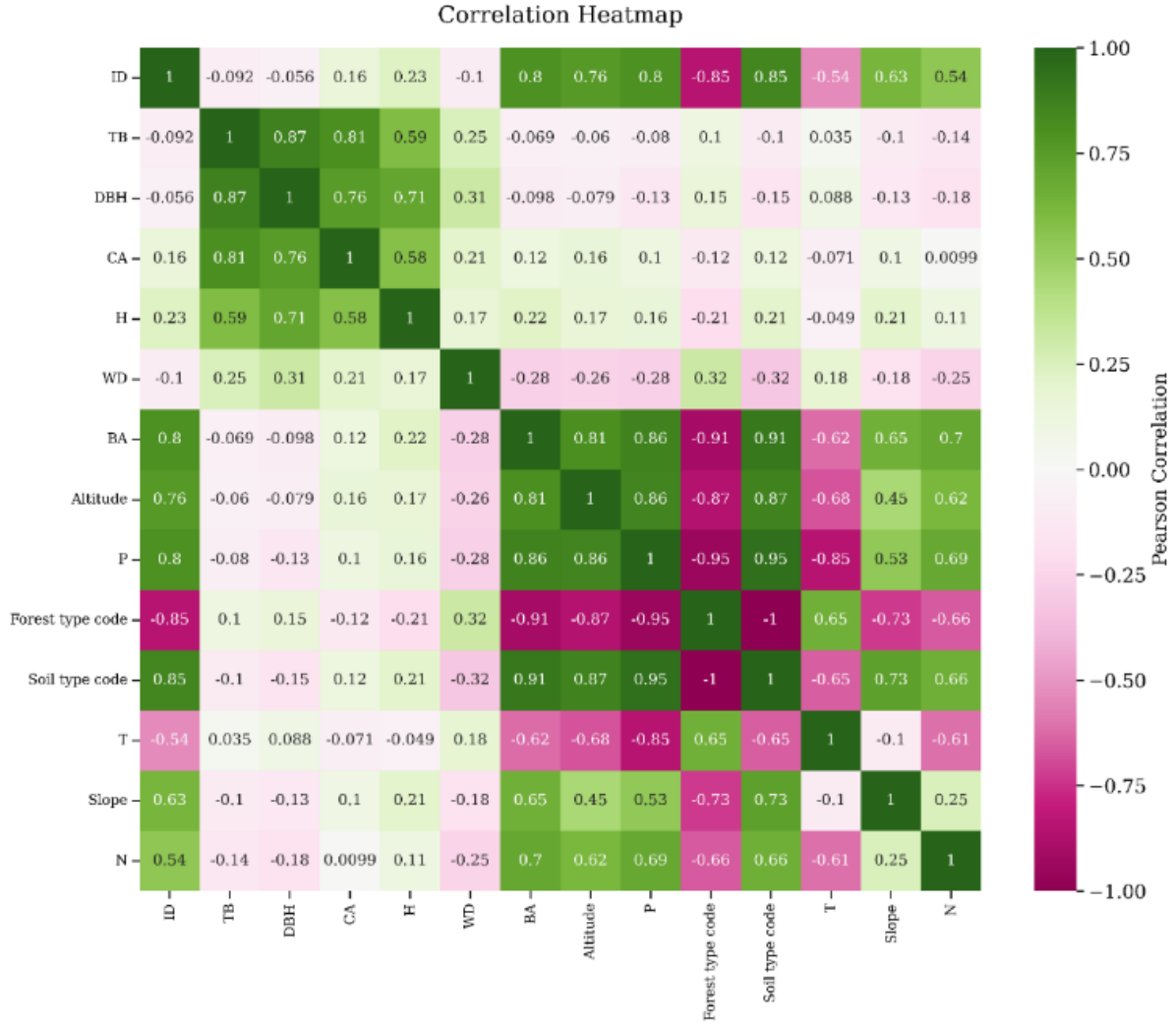


Figure 10: Pearson Correlation Heatmap for detecting the relationship between studied variables.

In Fig. 11, a pair plot visualization for the distribution of dataset parameters is shown for exploring the analysis of the data. In a pair plot, the data is visualized to find the relation between them, where variables are continuous as well as categorical, or form the most divided clusters. Dispersions of the parameters indicate the fact that most features are not evenly distributed. CA, WT, and P are skewed or clustered, and the values of these variables are focused on particular ranges. Scatter plot graphs such as CA versus WT or HA versus CA show positive relationships, which hold good, indicating potential multicollinearity that could be important to model. As a contrast, the variable types like forest type code and soil type code arrive in horizontal bands or discrete groups, as they are categorical. These trends suggest that the explanatory power of the data set is in part due to a mixture of continuous gradations in combination with categorical differences. The distributions in classes are depicted by the colors, and it can be observed that there is clearly a grouping in the plots, either of altitude, or of CA, or of WT. Following is a pair plot providing a high interface level to derive enlightening statistical information about the dataset; i.e., the variations in each plot can be observed, and the crucial diagonal secondary plots show each variable distribution. This pair plot for the relationship between variables of total amount of biomass, namely TB, is also demonstrated in Fig. 12, which more explicitly explores how the CTB classes are distinguished in terms of predictors. In this instance, the scatter plots show that for most variable pairs, the CTB categories are

predominantly overlapping, indicating that no set of variables can completely separate the CTB classes. However, there are regions, particularly in pairings like CA vs. WT or CA vs. P, where some CTB groups are grouped more closely together or are bunched into more constricted value ranges. The histograms on the diagonal also emphasize the bunched character of observations within given intervals, further underscoring that the dataset is skewed in variable distribution. This class grouping within specific regions suggests that individual variables may not always be able to differentiate CTB, but groups of predictors likely have predictive value. Further, the mixture of continuous and discrete variables introduces difficulty, as seen in the scatter plots, where some categories of CTB extend across different bands, with others overlapping.

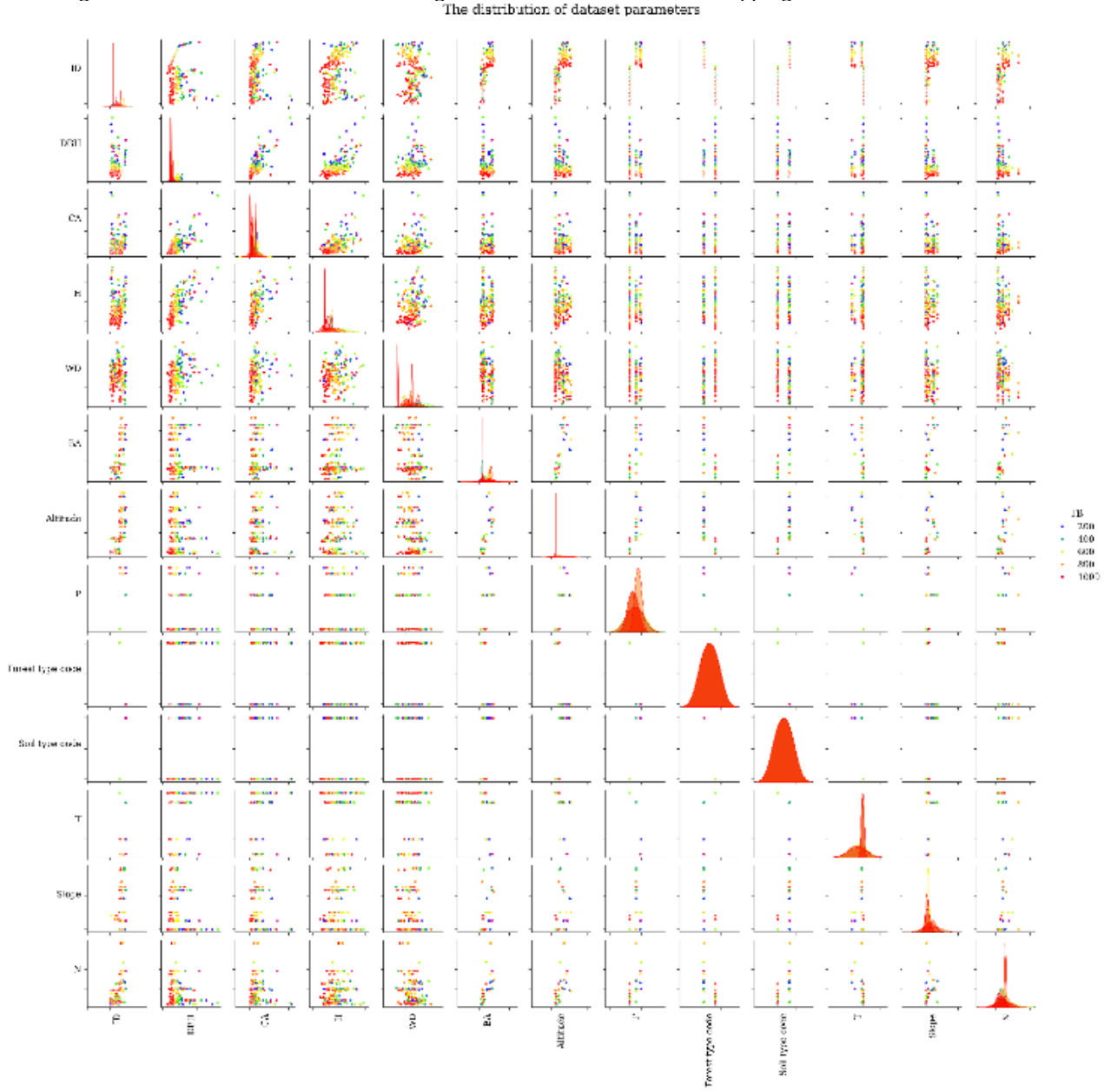


Figure 11: Pair plot for specifying the distribution of dataset parameters as well as their relationship

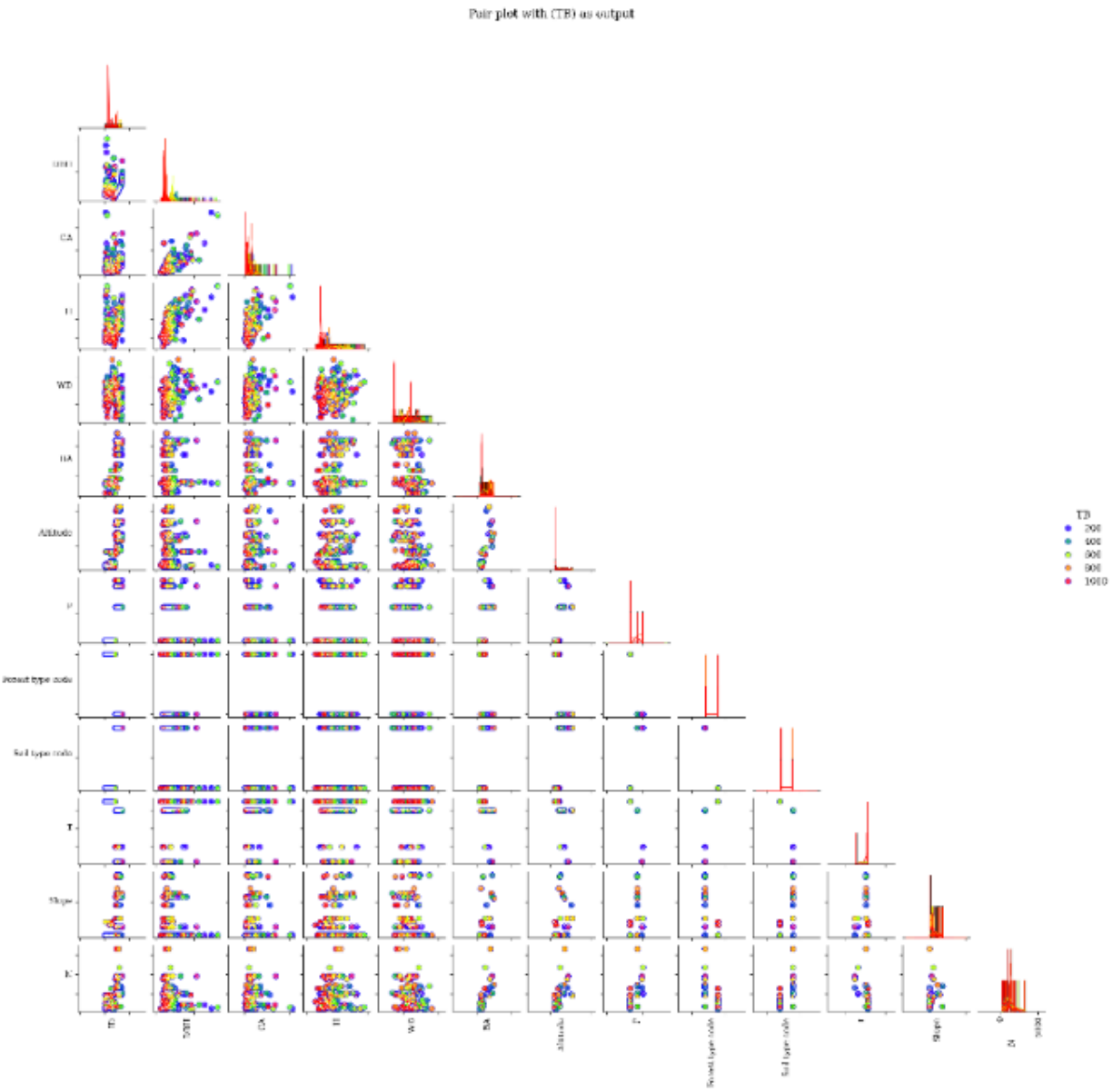


Figure 12: Pair plot for showing the relationship between the variables of total tropical tree biomass (TB) and their distributions

3.2. Machine learning results

In the present investigation, six methods were employed in two forest locations, i.e., Dipterocarp and Evergreen Broadleaf, to concurrently anticipate BGB, AGB, and TB = BGB + AGB. To assess the base models' performance, Table 3 presents the findings of the error metrics criteria for each recommended models considering the train and test data. By comparing these error metrics along with FI, it was detected that the Stacking Ensemble model was optimal than other models. The very large R 2 values close to 0.999 on the training dataset is an indicator of overfitting or data leaks. To prevent this we made sure to have rigorous separation of the training and test data and we optimized our hyperparameters using the Grid Search with cross-validation to prevent overfitting. The rock-bottom R2 values on the testing data (such as 0.962) are indicative of a lack of overfitting, so the overfitting appears to be contained. Further improvements with regard to regularization and data augmentation will be necessary in future computations to minimize the chances of overfitting. The test results indicate that the Stacking model outperforms the others in nearly all metrics, demonstrating higher predictive accuracy and reliability. Its mean squared error (MSE) is considerably low at 334.82, indicating lower average squared discrepancies between the predicted value and actual value compared to other models like ElasticNet (2378.17) and Extra Trees (1216.54). In the same vein, root mean squared error (RMSE) for Stacking is 18.30, a far cry from those of ElasticNet (48.77) and Gradient Boosting (41.75), meaning they were more precise in their predictions. Mean absolute error (MAE) performs the same, at 12.42 for Stacking, a far better performance than for models such as Poisson regression (19.32) and XGB (21.18).

In regard to explained variance and fit, Stacking had the best R^2 value of 0.968 across all models, which means it accounts for nearly 97% of the test data variance. This is

significantly better than ElasticNet's R^2 score of 0.77 and Extra Trees' R^2 score of 0.88. The Stacking model's normalized mean squared error (NMSE) is 0.032, the minimum, with no significant normalized error against data variance. Likewise, its variance accounted for (VAF) is 0.971, indicating impeccable consistency of predicted and actual values. The median absolute percentage error (MDAPE) of 23.08 and the standard deviation (STD_dev) of residuals of 105.76 also illustrate the consistency of the model in its performance.

On the other hand, the other models show higher error measures and lower variance explanation, with the Stacking model remaining the most accurate and consistent for the test set in this comparison. This claim was on the basis of the higher R^2 value of the Stacking model for both train and test data. According to this table, the higher R2 and VAF for each model make them worthy of a better model; in this case SE model. On the other hand, the lower the other metrics such as MSE, RMSE, MAE, NMSE, MDAPE, and STD, the more the model would have the merit of being an efficient predictor. Therefore, the Stacking model is deemed the most efficient and performs better than the other models for both training and testing data. In contrast, ElasticNet shows weaker performance in predicting the variables. Furthermore, the results of the employed evaluation metrics are presented and thoroughly discussed using relevant figures at the end of this section. Improved accuracy and stability on both forest types were indicated by the lower MSE, RMSE, and MDAPE, but higher R2 and VAF of the Stacking Ensemble, which consistently outperformed the other algorithms. ElasticNet performed poorly because of its linear framework, which failed to properly capture the intricate, nonlinear patterns in biomass data. Because Stacking possessed the ability to combine the powers of tree-based models like GB, ET, and XGB, it outperformed them despite the fact that they were moderate.

Figure 13 below is an illustration of the data values obtained via the ML parameters; i.e., ElasticNet, Extra Trees, GB, Poisson, Stacking, XGB; and accordingly, a comparison of these parameters in detail, along with their distance from the target value data, is presented.

Table 3: Error metrics criteria result for the proposed ML models considering the train and test datasets.

Models / Metrics	ElasticNet	Extra Trees	GB	Poisson	Stacking	XGB
Train						
MSE	4026.476	4.933E-26	5.800	1725.207	1153.269	1.72544E-05
RMSE	63.455	2.221E-13	2.408	41.536	33.960	0.004
MAE	32.550	7.82E-14	1.821	16.042	11.562	0.003
R^2	0.788	0.999	0.999	0.909	0.939	0.999
NMSE	0.212	2.601E-30	0.000	0.091	0.061	9.09812E-10
MDAPE	67.280	1.651E-13	5.016	29.535	8.722	0.008
STD_dev	100.279	137.713	137.491	150.171	134.774	137.713
VAF	0.788	0.999	0.999	0.909	0.939	0.999
Test						
MSE	2378.167	1216.538	1743.162	1051.301	334.820	2090.796
RMSE	48.766	34.879	41.751	32.424	18.298	45.725
MAE	36.818	18.068	21.948	19.320	12.422	21.178
R^2	0.770	0.882	0.831	0.898	0.968	0.797
NMSE	0.230	0.118	0.169	0.102	0.032	0.203
MDAPE	68.390	32.640	34.406	31.075	23.081	30.099
STD_dev	100.272	85.241	76.508	78.865	105.763	75.760
VAF	0.777	0.894	0.853	0.924	0.971	0.817

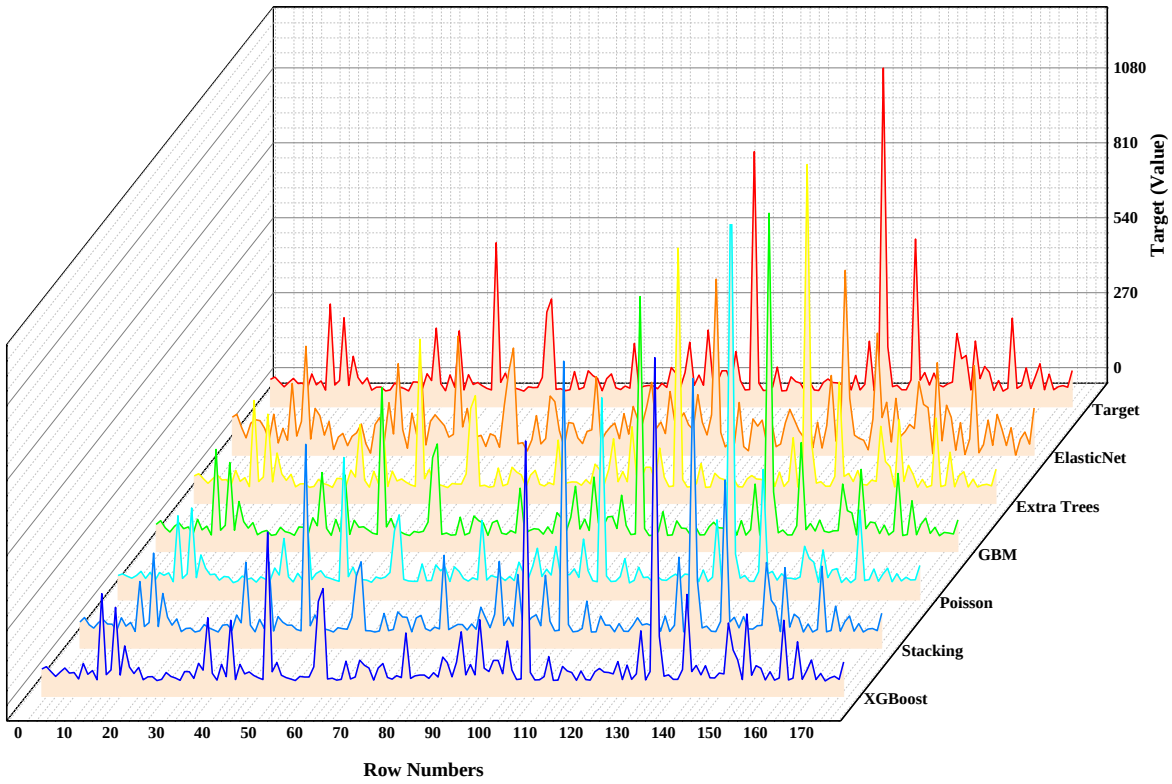


Figure 13: Value plot of comparing ML parameter values with the target value.

Figure 14 shows the results for R^2 as a significant error metric criterion, suggesting how well the employed ML models' predictions fit the real data. Shown in the figure, the model's prediction values align closely to the norm line (when $R^2 = 1$) is considered to be a superior as well as more accurate model. This result is in line with a higher R^2 value (approximately 0.939) for the test data and 0.968 for the training data in the proposed Stacking model.

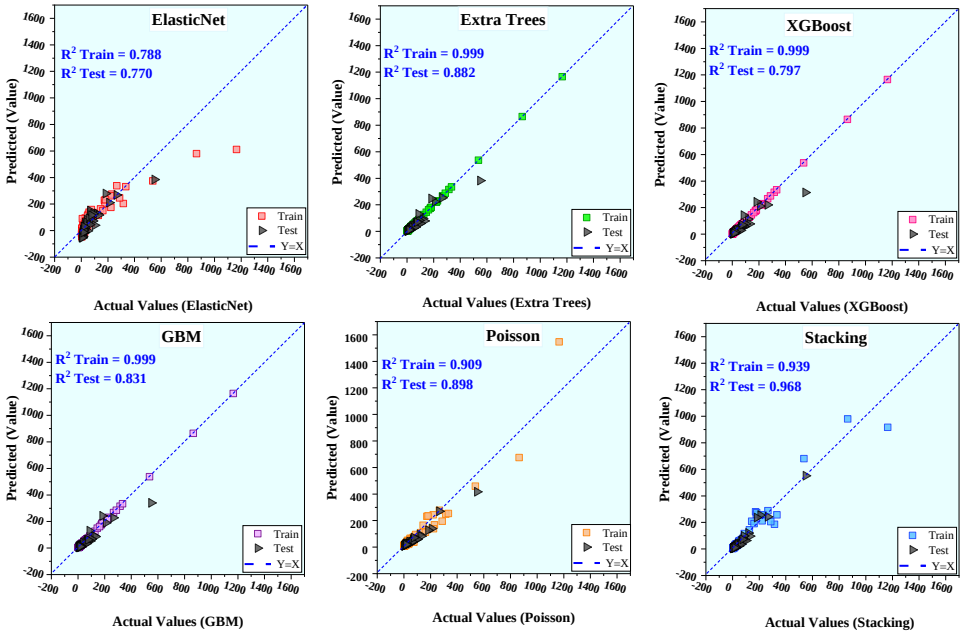


Figure 14: Comparing the coefficient of determination (R2) for each ML model.

The frequency of each error value for each ML method's predictions is represented in Fig. 15. The error analysis was conducted for both train and test parameters, and the ML models were assessed to examine their accuracy. As a result, the error in the ML models' prediction performance ought to be almost zero to be an adequate model for the aim of the study.

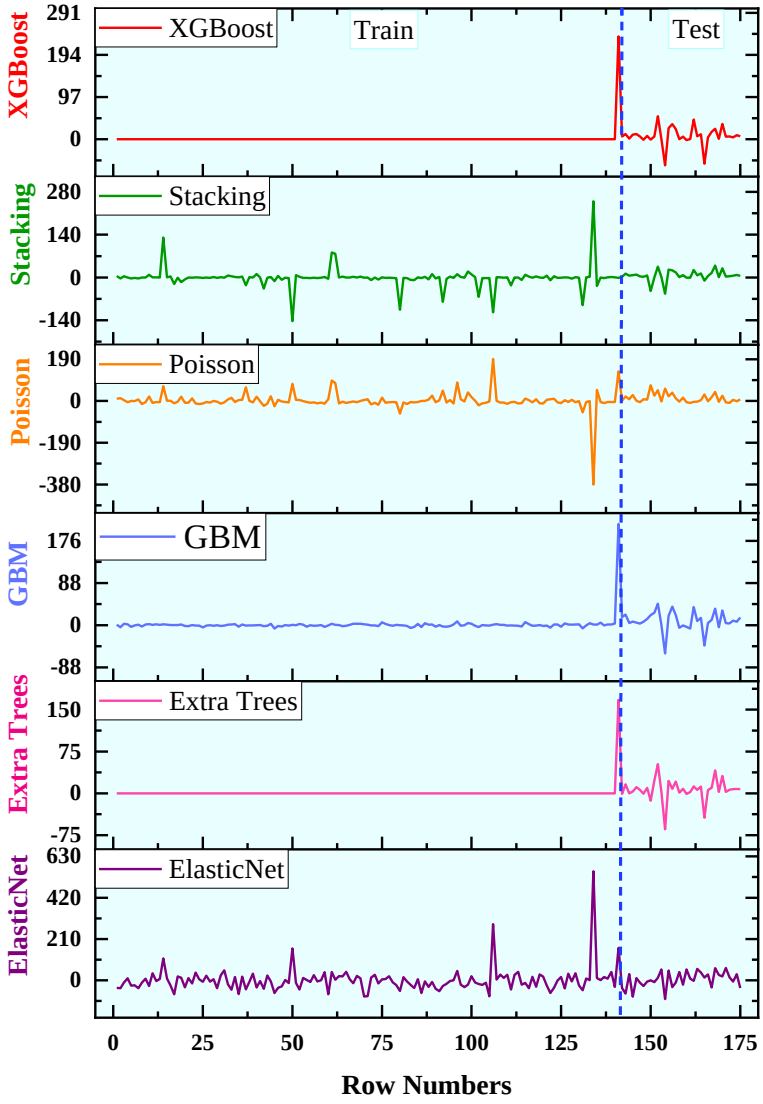


Figure 15: Comparing error values for ML models

Moreover, according to Fig. 16, the error values for the proposed ML models have been illustrated from the least error value to the most errors for both train and test data of each model, moving from left to right. A model with the least error values (i.e., approximately zero) would be the best predictor among the employed ML models. This visualization highlights that the data has intense recurrent peaks—suggesting non-uniform distributions with dominating clusters—and that these patterns persist but evolve subtly across different sections of the dataset. When one of the groups describes a stacking model, its activity can be visually compared with the other groups by looking at how close the mean is to zero and how much and steady the standard deviation is.

Based on the plot, we see that stacking seems to be more accurate than individual models, but by a very small margin. Compared to its predictions, it has fewer errors and reduced variance, indicating that it has a stronger generalization and stability. On the contrary, although other models have also performed adequately, they exhibit some spread or deviations that are a bit higher than the mean.

Overall, it appears that stacking should produce a more consistent and less erratic result than single models, thus making a superior comparison to the single models in terms of performance.

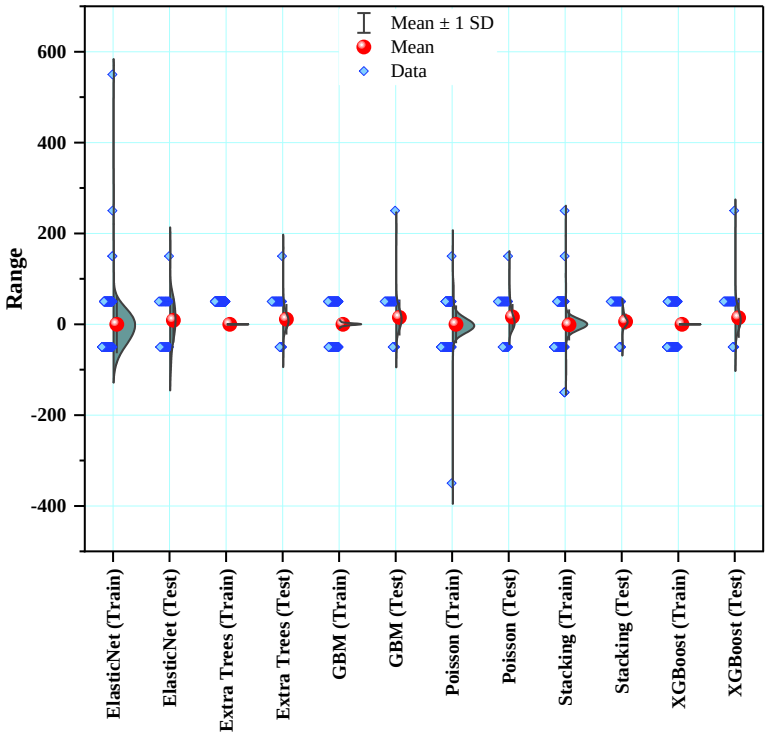


Figure 16: Boxplot of ML models' error values for both train and test data

The following figure (Fig. 17) illustrates a comparison of proposed models in terms of two important statistical evaluation metrics, namely R^2 and VAF, estimated for both the test and train datasets. As the values of these metrics show, all the models perform efficiently in the prediction except the ElasticNet model, which performs weaker than others, having lower VAF and R^2. Stacking and XGB model performances are stronger than the rest of the models, bearing higher VAF and R^2. Based on the plot, we see that stacking seems to be more accurate than individual models, but by a very small margin. Compared to its predictions, it has fewer errors and reduced variance, indicating that it has a stronger generalization and stability. On the contrary, although other models have also performed adequately, they exhibit some spread or deviations that are a bit higher than the mean.

Overall, it appears that stacking should produce a more consistent and less erratic result than single models, thus making a superior comparison to the single models in terms of performance.

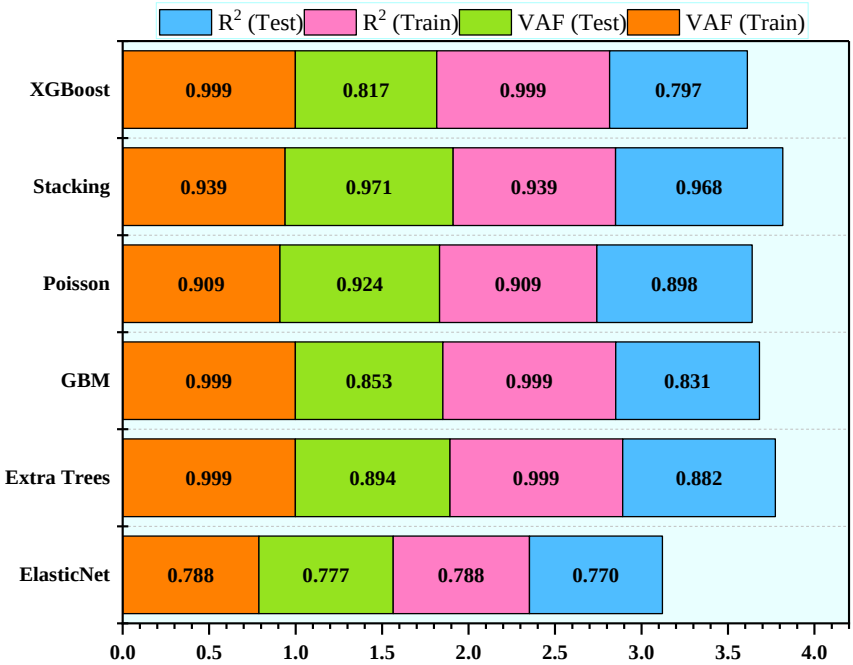


Figure 17: Comparison of Models based on VAF and R2 metrics.

The other evaluation metrics, including MSE, NMSE, MAE, RMSE, STD, and MDAPE, are applied for comparison among the models, supposing that the lowest value of these metrics for each model allows that model to be the best predictor. In this case, the stacking model for both the train and test datasets is the lowest compared to the other models (See Fig. 18).

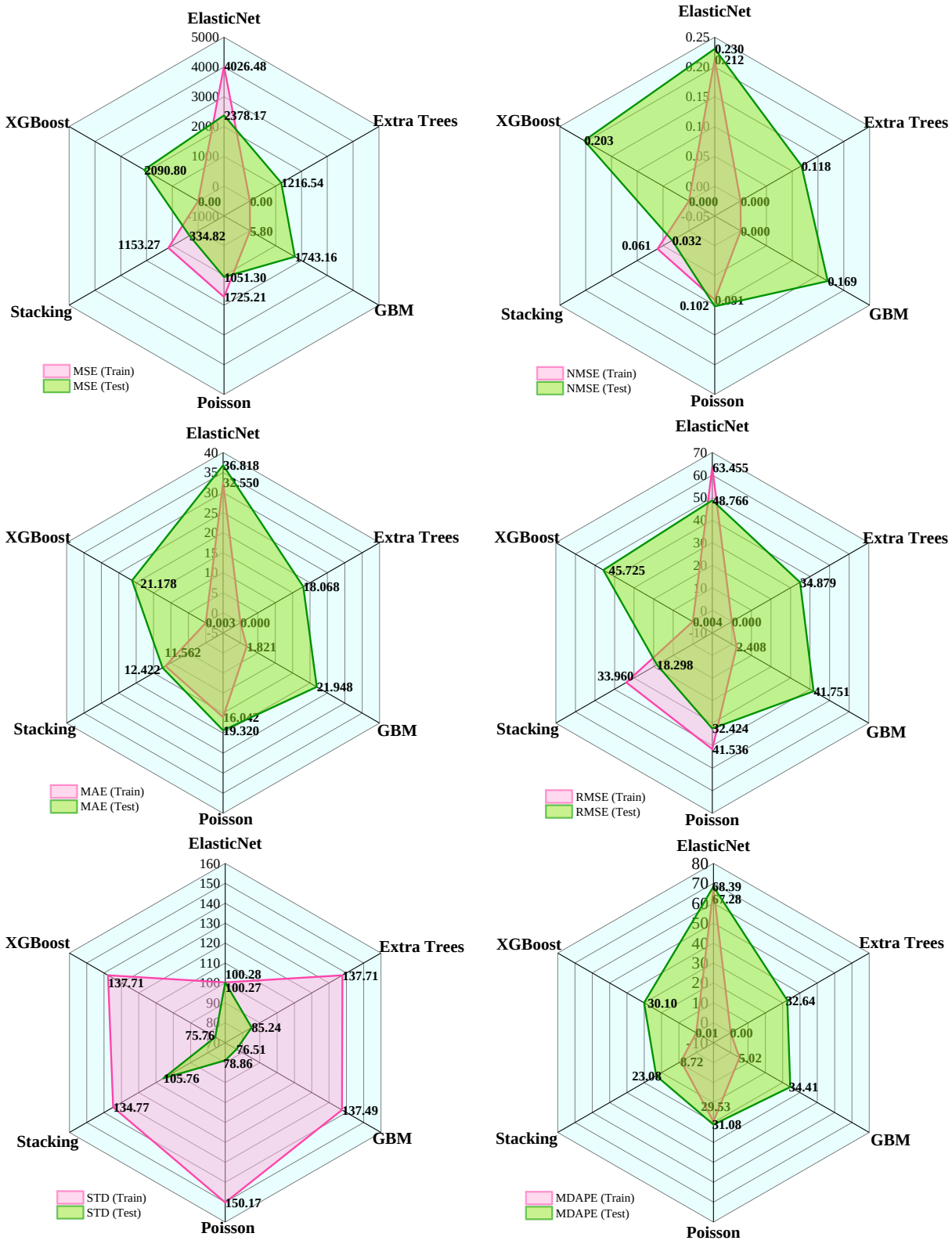


Figure 18: Comparison of Models based on MSE, NMSE, MAE, RMSE, STD, and MDAPE metrics.

Another graphical tool used to compare the models' performance is the Taylor diagram. This diagram evaluates models based on their accuracy, using metrics such as the correlation coefficient, standard deviation (STD), and RMSE. In the diagram, the models' performance is represented by circles, where better

performance is indicated by points closer to the reference point [29]. Taylor diagrams for predicting tropical tree biomass are shown in Fig. 19. As seen in these diagrams, the RMSE of the Stacking model is lower than that of the other models, and its correlation coefficient exceeds 0.9, outperforming the other models in this regard. In contrast, the RMSE of the ElasticNet model is higher than that of the other machine learning (ML) models, and its correlation coefficient is lower. These findings, based on the correlation coefficient, STD, and RMSE, confirm that the Stacking model outperforms the other models.

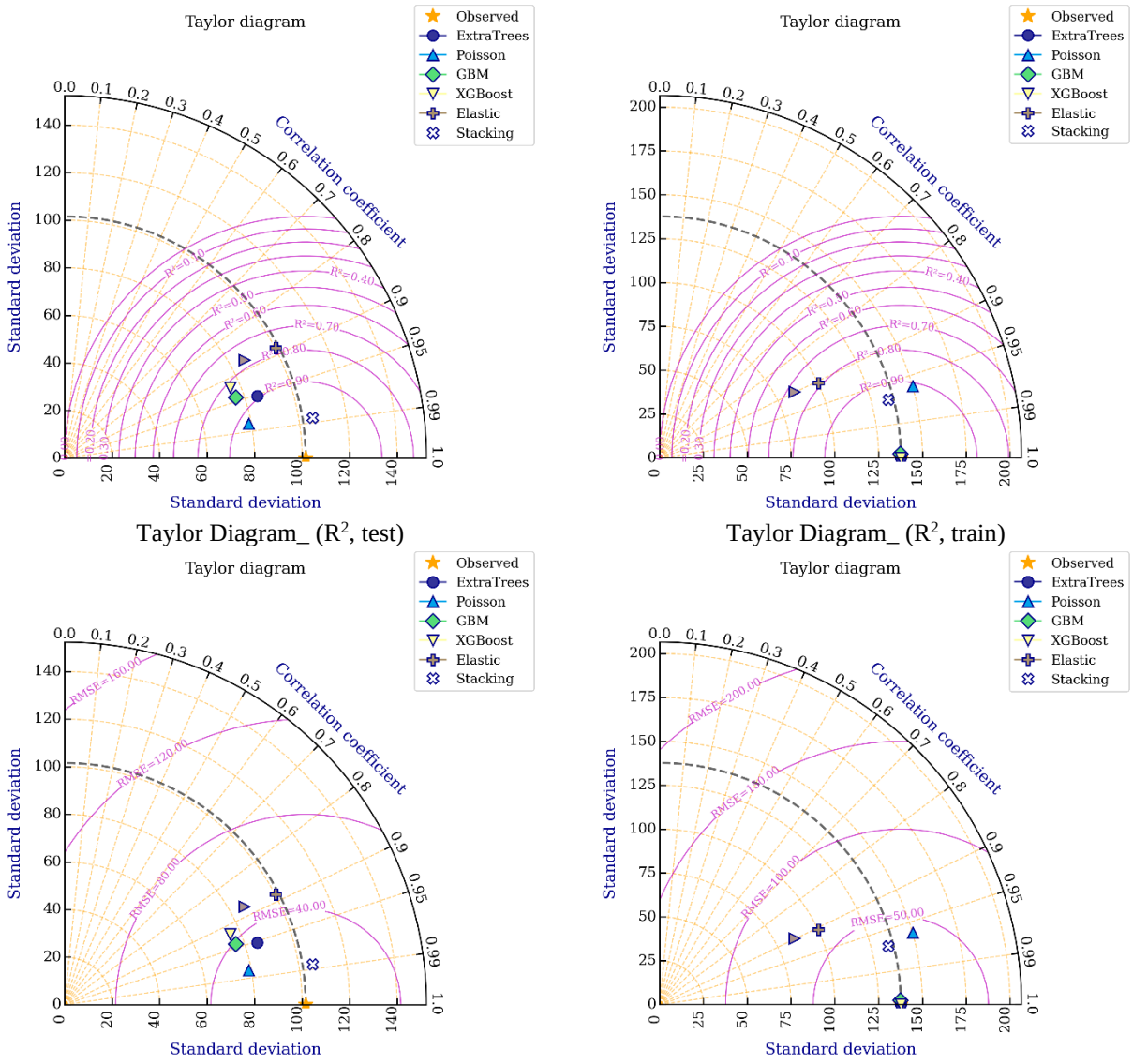
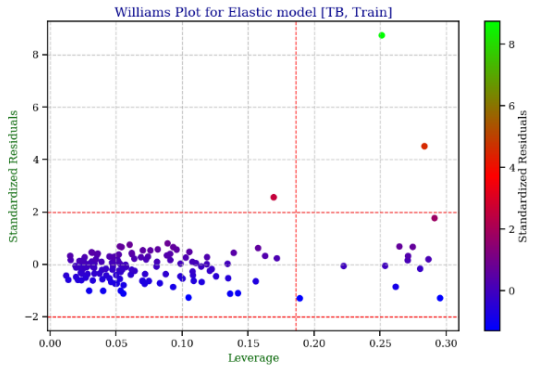


Figure 19: Taylor diagrams for models' comparison based on RMSE, STD, and R metrics

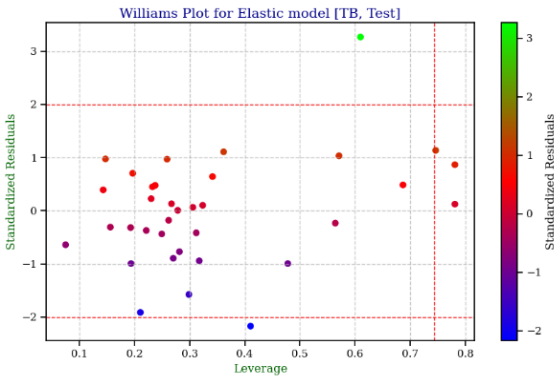
The last plot to be discussed for model comparison is the Williams plot. This plot is used to compare a specific group of compounds in terms of leverage values and standardized residuals [30]. William's plot shows the standardized residuals on the y-axis and leverages on the x-axis of the training and testing datasets. From this plot, the applicability domain is implemented within a squared area inside ±2 standard deviations and a threshold h* in leverage (h* = 3p´/n, being p´ model parameters and n compounds number). The majority of data ought to be located within this area, conceptualizing that they have to be inliers and influential in the model. The Williams plots of the Stacking model on both training and test sets are indicators of good model performance and generalization. Most of the observations in the test plot lie comfortably within the satisfactory limits for leverage and standardized residuals (±2), which is an indicator that the model predictions are not biased and stable and possess minimal outliers. Only a minimal number of observations being out of the ±2 boundary and leverage constraint signifies that there are very few influential or problematic points.

Similarly, the training plot shows tightly clumped residuals around zero with the majority of data points having little leverage, which would mean that the model has not over-fit the training data. Even some of the residuals fall outside ±3 or do possess relatively higher leverage, those are scattered and do not invalidate the model. The similar trend in both plots confirms that the

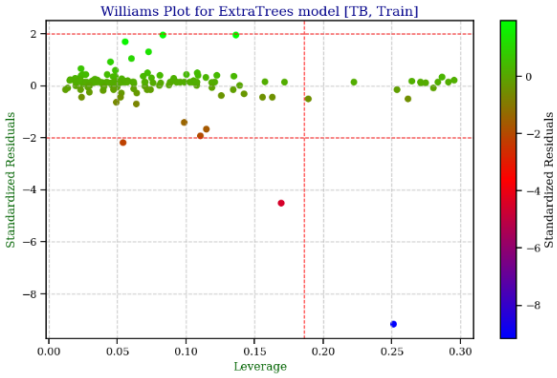
Stacking model works well on unseen data, learning the inherent pattern significantly without being overfit and robust. To be an efficient model predictor, the data must lie within this domain (See Fig. 20) [23].



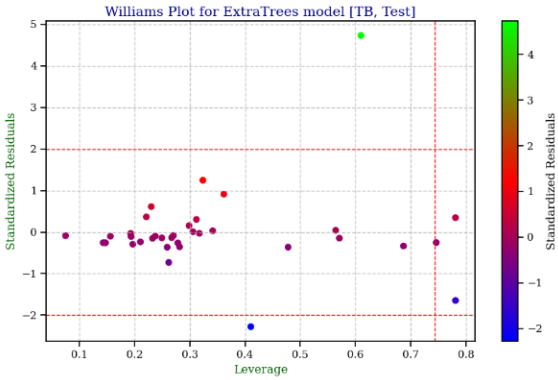
Elastic Net Williams Plot [TB, Train]



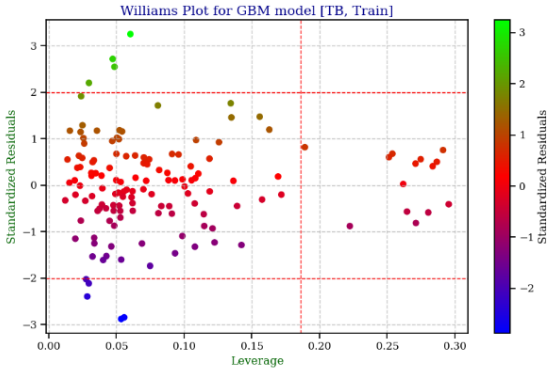
Elastic Net Williams Plot [TB, Test]



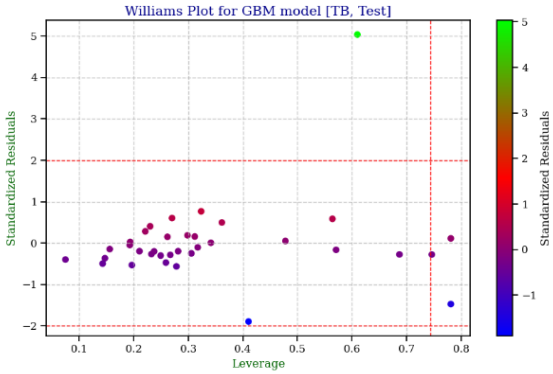
Extra Trees Williams Plot [TB, Train]



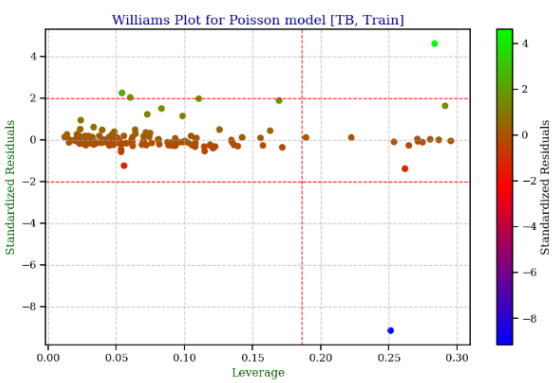
Extra Trees Williams Plot [TB, Test]



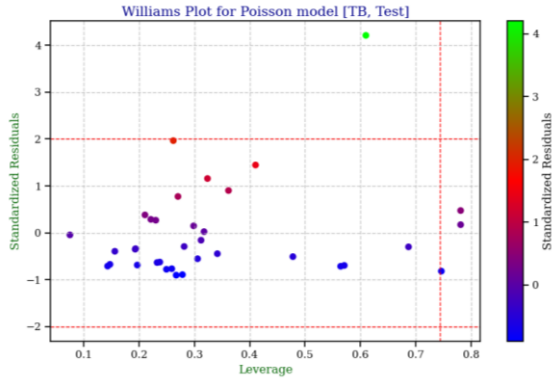
GBM Williams Plot [TB, Train]



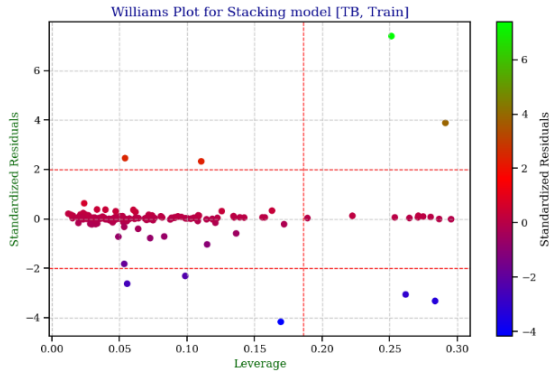
GBM Williams Plot [TB, Test]



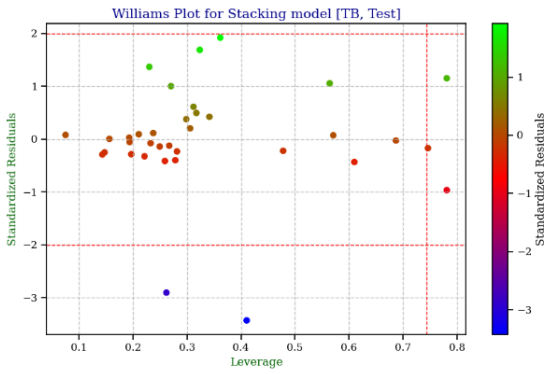
Poisson Williams Plot [TB, Train]



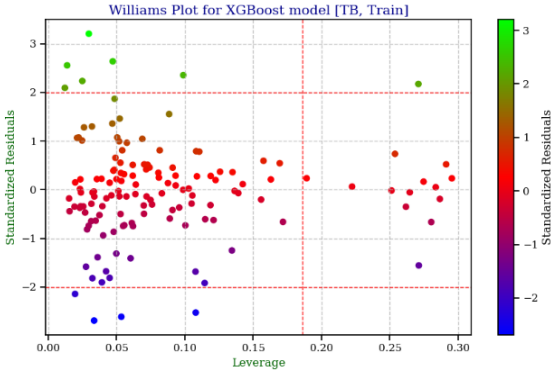
Poisson Williams Plot [TB, Test]



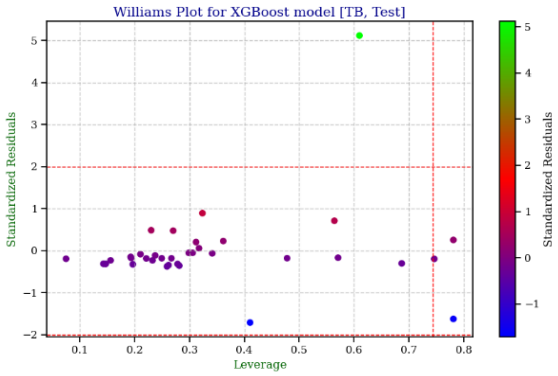
Stacking Williams Plot [TB, Train]



Stacking Williams Plot [TB, Test]



XGBoost Williams Plot [TB, Train]



XGBoost Williams Plot [TB, Test]

Figure 20: Williams plots for models' comparison based on standard residuals and leverage.

3.3. Comparison with foundation models and LLMs

Large Language Models, or LLMs, are sophisticated AI systems that have been trained on enormous text datasets to comprehend and produce human language. Google created BERT, which is perfect for tasks like classification and question answering because it can analyze words in both directions and understand context. With its emphasis on producing relevant and coherent text, OpenAI's GPT is effective for tasks like content creation, dialogue, and summarization. Both use Transformer architecture, but GPT is more focused on generation and BERT on comprehension. The proposed SE model adds domain-specific efficiency, whereas models such as BERT and GPT are effective for general-purpose NLP tasks. We specifically highlighted how, in contrast to the extensive, data-intensive training of LLMs, SE makes use of structured, domain-relevant features. This study also indicates SE's improved interpretability and reduced computational cost, both of which are important for ecological modelling.

4 conclusions

The study was implemented on eight tropical forests in Vietnam, using the forestry variables, i.e., AGB, BGB, and TB. In an attempt to solve the problem of predicting the mentioned variables, the study used an MGDL regression strategy, which later proved to be an efficient model to bear a strong ability to predict tropical forest biomass. To this, five models were selected as major algorithms to unravel the issue of biomass prediction. These models included Gradient Boosting (GB), Extra Trees (ET), XGB, ElasticNet, and Poisson, all of which were employed to synchronously anticipate both the amount of AGB, BGB, as well as TB = BGB + AGB. Then optimized by Grid Search. Additionally, the SE model was joined to the aforementioned models so as to allow the results to become satisfactory, i.e., mainly for the cross-validation purpose. Therefore, the recommended method's performance was investigated in terms of two sets of actual data, namely training and testing data.

The outcome of this study presented that the recommended method had a vigorous efficacy to estimate the amount of forest biomass. That is to say, employing a simultaneous group of ML models resulted in a significant impact on predicting forestry above- and below ground, as well as the sum of the biomass. The very high R^2 values of near 0.999 in the training set are definitely cause for alarm for overfitting or data leakage. We dealt with this by ensuring strict separation of training and test datasets such that there was no information leakage. We also employed Grid Search with cross-validation during hyperparameter tuning to allow maximum model complexity without over-fitting. The test set results, with R^2 scores considerably lower (e.g., 0.968 for the best model), are a sign of good generalization and suggest that while there may be some overfitting, it is controlled. More regularization and more

data will be tried in future research to reduce the possibility of overfitting even more. Based on the provided metrics, the Stacking ensemble model performed clearly superior to each of the standalone models on the test set. That is because it is capable of leveraging the prediction power of various base learners—ElasticNet, Extra Trees, Gradient Boosting, Poisson Regression, and XGB—and minimizing their respective errors through a meta-learner. Stacking takes into consideration the stand-alone strengths of linear as well as nonlinear models and results in improved generalization and less overfitting.

Quantitatively, the Stacking model achieved the highest coefficient of determination (R^2 = 0.968) and variance accounted for (VAF = 0.971) on the test set, indicating that its predictions were most highly correlated with actual biomass values. It generated the lowest mean squared error (MSE = 334.820), root mean square error (RMSE = 18.298), and mean absolute error (MAE = 12.422), indicating high accuracy and low prediction bias. In terms of normalized error, it also had an NMSE of just 0.032, and the mean directional accuracy percentage error (MDAPE) decreased to 23.081%, significantly better than other models. Although its test standard deviation (STD = 105.763) was slightly greater, this is the natural result with better prediction accuracy and range coverage for both train and test data, where the results showed R2 equals 0.939 for the testing data and 0.968 for the training data in this study data analysis. Therefore, adding the SE model to the proposed models is recommended for predicting forest biomass effects. As a result, this is evidence of the poor performance of this model. The William plots residuals display that its majority corresponds to the tolerant parameters, and very limited outliers or high-leverage points are there in the test and train subsets. This implies that the Stacking model produces reliable, unbiased, and non-overfit predictions, and there is indeed powerful generalization and performance.

Declarations

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Authors' contributions QD

Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Acknowledgements

I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Ethical approval

The research paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] H. C. Zhantao Song, Xiong Zhang, Xiaoqiang Li, Junjie Zhang, Jingai Shao, Shihong Zhang, Haiping Yang, "Machine learning assisted prediction of specific surface area and nitrogen content of biochar based on biomass type and pyrolysis conditions," *J Anal Appl Pyrolysis*, vol. 183, 2024.

[2] L. Jia, W. Shao, J. Wang, Y. Qian, Y. Chen, and Q. Yang, "Machine learning-aided prediction of bio-BTX and olefins production from zeolite-catalyzed biomass pyrolysis," *Energy*, vol. 306, p. 132478, 2024.

[3] H. Wu, S. An, B. Meng, X. Chen, F. Li, and S. Ren, "Retrieval of grassland aboveground biomass across three ecoregions in China during the past two decades using satellite remote sensing technology and machine learning algorithms," *International Journal of Applied Earth Observation and Geoinformation*, vol. 130, no. November 2023, p. 103925, 2024, doi: 10.1016/j.jag.2024.103925.

[4] P. Mao *et al.*, "An improved approach to estimate above-ground volume and biomass of desert shrub communities based on UAV RGB images," *Ecol Indic*, vol. 125, p. 107494, 2021, doi: 10.1016/j.ecolind.2021.107494.

[5] P. B. May *et al.*, "Mapping aboveground biomass in Indonesian lowland forests using GEDI and hierarchical models," *Remote Sens Environ*, vol. 313, p. 114384, 2024.

[6] B. Huy, N. Quy Truong, K. P. Poudel, H. Temesgen, and N. Quy Khiem, "Multi-output deep learning models for enhanced reliability of simultaneous tree above- and below-ground biomass predictions in tropical forests of Vietnam," *Comput Electron Agric*, vol. 222, no. December 2023, p. 109080, 2024, doi: 10.1016/j.compag.2024.109080.

[7] M. F. Oliveira *et al.*, "Predicting below and above-ground peanut biomass and maturity using multi-target regression," *Comput Electron Agric*, vol. 218, p. 108647, 2024.

[8] G. Kunapuli, *Ensemble methods for machine learning*. Simon and Schuster, 2023.

[9] R. Dey and R. Mathur, "Ensemble learning method using stacking with base learner, a comparison," in *International Conference on Data Analytics and Insights*, Springer, 2023, pp. 159–169.

[10] P. Naik, M. Dalponte, and L. Bruzzone, "Automated machine learning driven stacked ensemble modeling for forest aboveground biomass prediction using multitemporal Sentinel-2 data," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 16, pp. 3442–3454, 2022.

[11] Y. Zhang, J. Ma, S. Liang, X. Li, and J. Liu, "A stacking ensemble algorithm for improving the biases of forest aboveground biomass estimations from multiple remotely sensed datasets," *GIsci Remote Sens*, vol. 59, no. 1, pp. 234–249, 2022.

[12] J. Liu, Y. Niu, Z. Jia, and R. Wang, "Assessing the ethical implications of artificial intelligence integration in media production and its impact on the creative industry," *MEDAAD*, vol. 2023, pp. 32–38, 2023.

[13] R. Huang, C. McMahan, B. Herrin, A. McLain, B. Cai, and S. Self, "Gradient boosting: A computationally efficient alternative to Markov chain Monte Carlo sampling for fitting large Bayesian spatio-temporal binomial regression models," *Infect Dis Model*, vol. 10, no. 1, pp. 189–200, 2025, doi: 10.1016/j.idm.2024.09.008.

[14] Z. Wang, L. Mu, H. Miao, Y. Shang, H. Yin, and M. Dong, "An innovative application of machine learning in prediction of the syngas properties of biomass chemical looping gasification based on extra trees regression algorithm," *Energy*, vol. 275, p. 127438, 2023.

[15] H. Wei, K. Luo, J. Xing, and J. Fan, "Predicting co-pyrolysis of coal and biomass using machine learning approaches," *Fuel*, vol. 310, p. 122248, 2022.

[16] R. (Bob) Roy, "No Title." Accessed: Jun. 27, 2021. [Online]. Available: https://bobrupakroy.medium.com/extra-trees-classifier-regressor-5b5f6abe8228

[17] B. Kıyak, H. F. Öztop, F. Ertam, and İ. G. Aksoy, "An intelligent approach to investigate the effects of container orientation for PCM melting based on an XGBoost regression model," *Eng Anal Bound Elem*, vol. 161, pp. 202–213, 2024.

[18] Y. Ayub, J. Ren, T. Shi, W. Shen, and C. He, "Poultry litter valorization: Development and optimization of an electro-chemical and thermal tri-generation process using an extreme gradient boosting algorithm," *Energy*, vol. 263, p. 125839, 2023.

[19] A. Jain, "No Title." Accessed: Feb. 05, 2024. [Online]. Available: https://medium.com/@abhishekjainindore24/elastic-net-regression-combined-features-of-l1-and-l2-regularization-6181a660c3a5

[20] J. Liu *et al.*, "A new application of Elasticnet regression based near-infrared spectroscopy model: Prediction and analysis of 2, 3, 5, 4′-tetrahydroxy stilbene-2-O-β-D-glucoside and moisture in Polygonum multiflorum," *Microchemical Journal*, vol. 199, p. 110095, 2024.

[21] Purhadi, D. N. Sari, Q. Aini, and Irhamah, "Geographically weighted bivariate zero inflated generalized Poisson regression model and its application," *Heliyon*, vol. 7, no. 7, p. e07491, 2021, doi: 10.1016/j.heliyon.2021.e07491.

[22] U. Arif, C. Zhang, S. Hussain, and A. R. Abbasi, "An Efficient Interpretable Stacking Ensemble Model for Lung Cancer Prognosis," *Comput Biol Chem*, p. 108248, 2024.

[23] H. Yıldırım and M. R. Özkale, "A Novel Regularized Extreme Learning Machine Based on L 1-Norm and L 2-Norm: a Sparsity Solution Alternative to Lasso and Elastic Net," *Cognit Comput*, vol. 16, no. 2, pp. 641–653, 2024.

[24] S. M. Mastelini, F. K. Nakano, C. Vens, and A. C. P. de Leon Ferreira, "Online extra trees regressor," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 10, pp. 6755–6767, 2022.

[25] M. M. Hameed, M. K. Alomar, F. Khaleel, and N. Al-Ansari, "An Extra Tree Regression Model for Discharge Coefficient Prediction: Novel, Practical Applications in the Hydraulic Sector and Future Research Directions," *Math Probl Eng*, vol. 2021, 2021, doi: 10.1155/2021/7001710.

[26] "Understanding Poisson Regression." Accessed: Nov. 05, 2023. [Online]. Available: https://medium.com/@data-overload/understanding-poisson-regression-a-powerful-tool-for-count-data-analysis-b7184c61bfde

[27] M. A. Alemayehu, S. D. Kebede, A. D. Walle, D. N. Mamo, E. B. Enyew, and J. B. Adem, "A stacked ensemble machine learning model for the prediction of pentavalent 3 vaccination dropout in East Africa," *Front Big Data*, vol. 8, p. 1522578, 2025.

[28] "XGBoost Algorithm." Accessed: Sep. 04, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/#:~:text=XGBoost builds a predictive model,made by the existing ones.

[29] M. Ehteram, A. N. Ahmed, P. Kumar, M. Sherif, and A. El-Shafie, "Predicting freshwater production and energy consumption in a seawater greenhouse based on ensemble frameworks using optimized multi-layer perceptron," *Energy Reports*, vol. 7, pp. 6308–6326, 2021, doi: 10.1016/j.egyr.2021.09.079.

[30] A. Beheshti, E. Pourbasheer, M. Nekoei, and S. Vahdani, "QSAR modeling of antimalarial activity of urea derivatives using genetic algorithm-multiple linear regressions," *Journal of Saudi Chemical Society*, vol. 20, no. 3, pp. 282–290, 2016, doi: 10.1016/j.jscs.2012.07.019.

HematoFusion: A Weighted Residual-Vision Transformer Ensemble for Automated Classification of Haematologic Disorders in Microscopic Blood Images

Mouna Saadallah, Latefa Oulladji, Farah Ben-Naoum
Evolutionary Engineering and Distributed Information Systems Laboratory, Department of Computer Science, Djillali Liabes University, Sidi Bel Abbes, Algeria
E-mail: mouna.saadallah@univ-sba.dz, latifa.oulladji@univ-sba.dz, farah.bennaoum@univ-sba.dz

Keywords: Medical imaging, neural networks, red blood cell, leukemia, lymphoma

Received: May 27, 2025

Haematologic malignancies pose a significant global challenge, with 1.34 million new cases reported in 2019 and leukemia claiming 311,594 lives in 2020. Early diagnosis of these blood disorders increases survival chances by enabling prompt treatment, yet their complexity and variable cellular morphology hinder accurate detection. Advances in Medical Imaging and AI, particularly Image Classification, offer solutions by analyzing blood samples for subtle morphological patterns. This study advances the field by introducing a novel data set for the classification of red blood cells and using open-source data for the classification of leukemia and lymphoma (each covering 29,363; 16,811; and 1,436 images, respectively). We fine-tuned multiple AI models, including EfficientNetB3, ResNet50V2, and a pretrained Vision Transformer (ViT), and combined their strengths into a weighted ensemble framework. Evaluated across various metrics (including accuracy, precision, recall, etc.), the proposed HematoFusion model excelled, achieving 96% accuracy in the morphology of red blood cells, 99% in Leukemia, and 96% in Lymphoma, surpassing most existing models in terms of accuracy while covering a wider range of haematologic disorders. These findings demonstrate the potential of integrated AI frameworks to improve haematologic diagnostics with precision and reliability.

Povzetek: HematoFusion je uteženi ansambel ResNet50V2 in Vision Transformer, namenjen avtomatski klasifikaciji hematoloških motenj iz mikroskopskih slik. Sistem uporablja nov RBC-nabor podatkov ter odprtokodne nabore levkemije in limfoma ter izboljša zanesljivost diagnostičnega razpoznavanja krvnih celic.

1 Introduction

The collection of blood samples is crucial to understanding diseases, preventing them, and thoroughly providing treatment.

The diagnosis of blood cell diseases hinges significantly on determining the patient's Blood Cell Count (BCC) and observing the appearance of cells under a microscope. It serves as a guide for the pathologist or biologist, providing vital information on diseases that are indicative of quantitative (variations in the number of cells) or qualitative (structural or functional) abnormalities in blood cells [11].

Patients admitted to consultation often suffer haematologic dysfunction (either qualitative or quantitative), some examples of the most common cases requiring medical evaluation are caused either by a decrease in complete blood count, anemia, for instance, sees a decrease in the number of Red Blood Cells (RBCs) or the level of hemoglobin, or an increase/ concentration in the amount of RBCs, as marked in the condition of Erythrocytosis. Other conditions mark a change in the cell's shape and/or size, including: microcyte, macrocyte, echinocytes, codocyte, acanthocyte, spherocyte, and more. White Blood Cells (WBC) and platelet disorders can mainly be described as quantitative, for example: Leukopenia, leukocytosis, neutropenia, lymphocytopenia (WBC), thrombocytosis, or thrombocytopenia (platelets). Most qualitative disorders are of cancer or proliferative disorders, including Leukemia, lymphoma, myeloma (WBC), and Hemophilia (platelets) [9].

The pathologist, along with other medical professionals, depends on studying and examining body tissues to perform diagnostics. The microscope is the main tool used to observe blood cells, which provides a detailed description of them in terms of shape and count. Blood cell observation can be extremely challenging with the naked eye and requires enormous concentration and focus; modern technologies, however, recommend new techniques involving the use of a camera to capture microscopic images that can be exploited for further studies and examination. Some existing solutions like EasyCell ®Assistant, Vision Hema®Assist, include a stand-alone tool using highly costly robots and integrated microscopes that can assist the pathologist to make decisions and save time; nonetheless, due to their high costs and unavailability in public hospi-

tals and laboratories, these solutions cannot be relied upon entirely. This leads us to consider cheaper and more effective innovations emerging in recent years, including Deep Learning (DL) and its various contributions. DL has been widely implemented by researchers in the medical field, and it has given promising results regarding medical imaging (MRI, X-Rays, CT...) [30, 52] and enabled medical professionals to rapidly diagnose and detect abnormalities in the human body without exhausting analysis and observations.

This paper aims to improve classification accuracy for haematologic diseases by leveraging ensemble learning techniques applied to multi-source microscopic datasets, preserving the full spectrum of morphologic variability. The latest DL techniques were exploited, including transfer learning and fine-tuning Convolutional Neural Network (CNN) models and the recently emerging visual transformer (ViT) [28]. The ResNet50V2 and EfficientNetB3 networks were chosen as they were preferable for microscopic image classification, and the latter was suitable for scenarios with limited computing resources. We acquired different sources for our data set that study not only Red Blood Cell disorders but also White Blood Cells (WBC). The CNN and ViT models were separately trained using the completed data set, and the results were later combined to enhance the performance.

A description of our contribution is provided on the following lines:

- A meticulously curated data set for Red Blood Cell Morphology using samples collected in the Anti-Cancer Center in El-Oued, Algeria.
- The base architecture of EfficientNetB3 was used with transfer learning, leveraging pretrained weights from ImageNet. It was additionally fine-tuned for the task of blood cell classification.
- The ResNet50V2 was also integrated and transfer-learned as a base architecture and eventually fine-tuned by adding dense layers and regularization techniques that serve to enhance the model's performance.
- A pretrained ViT model was applied to our data set to classify blood cell images through self-attention mechanisms. The model was fine-tuned by optimizing hyperparameters to improve accuracy.
- A hybrid CNN/ViT model was developed by combining the strengths of CNNs for local feature extraction with those of ViT which captures global features more efficiently.

2 Related work

Pathology and detecting blood disorders require a mass of work and time by a biologist to prepare the blood, test it, and analyze it.

Nevertheless, the emergence of developed technologies, such as deep learning, made things much easier for biologists/ pathologists, as it assists them with the process of analyzing the blood smear and detecting abnormalities in cell type, shape, and aggregation. If done entirely by the pathologist, this step may take hours or even days when necessary, which causes a decline in the health worker's focus and even eyesight.

This urged the need to automate the task to alleviate the pressure on them. Many studies have been conducted to address this problem by exploiting the use of Artificial Intelligence and its diverse techniques.

In its earliest phase, peripheral blood image analysis was inspired by the emerging use of Artificial Intelligence in the medical field and its automation. Kim KS, et al. [2] designed a system that uses a CCD camera attached to the microscope to capture the peripheral images, preprocessing techniques such as edge enhancement and noise removal were applied, and the images were later classified into 15 types of Red Blood Cell abnormalities and 5 normal shapes of White Blood Cells using neural networks. Following that, neural networks, mainly Convolutional Neural Networks, were explored for blood cell image analysis and classification. WBC and its 5 different normal cell shapes (Neutrophils, Lymphocytes, Monocytes, Basophils, Eosinophils) were the easiest to classify and readily available [14] [56] [37]. Classification accuracy reached 96% using a simple neural network that consists of 16 neurons input layers and 10 nodes in a single hidden layer to achieve a minimum error less than 10^{-4} and the output layer with 5 neurons to classify each type [14]. Ali et al. [54] proposed the VGG16-ViT network that uses two online datasets to classify WBC subtypes, achieving excellent precisions of 98.99% and 99.95% on each dataset.

The DenseNet121 model [12] was used by Bozkurt F. [27] on the open-access data set provided by Paul Mooney, available on Kaggle.com [18], reaching an accuracy of 98%. Another Two-Module Deformable CNN and Transfer learning was proposed by Yao Xufeng, et al. [37], whilst the first module initializes the ImageNet [3] characteristic weights, the second module was designated for classification. The authors achieved precisions of 95.7%, 94.5%, and 91.6% for low-resolution and noisy undisclosed data sets, BCCD data set [20], respectively.

Some of the studies, however, focused solely on the classification of one disease. Leukemia is one of the most common blood cancers, leading to growing interest in developing new diagnostic systems for early detection and prevention. In this context, CNNs have gained significant attention due to their efficiency and high accuracy in image-based classification tasks. Areen K. et al. [47] compared in their study multiple CNN-based algorithms (AlexNet, DenseNet, ResNet, and VGG16), employing three datasets (ALL-IDB, ASH ImageBand, and images captured at JUST); reaching an accuracy of 94%. DeepLeukNet proposed by Saeed et al. [53] was conceived to classify Acute Lymphoblastic Leukemia (ALL) subtypes

employing a CNN-based classifier on the ALL-IDB1 and ALL-IDB2 datasets, attaining 99.61% accuracy. Kasim et al. [55] leverages the online datasets ALL-IDB and Munich AML Morphology datasets for multi-class classification of Leukemia subtypes using pretrained CNN architectures and other classification models, including RandomForest, SVM, and Extreme Gradient Boosting. The highest accuracy achieved by this method was of 88%. In recent studies, Vision Transformers (ViTs) have been employed for the classification of Leukemia subtypes. Swain et al. [59] proposed in their research a model based solely on ViTs and classified ALL subtypes. The accuracy of the test set reached 99.67%. A similar approach was implemented by Prasad et al. [51] who attained an overall accuracy of 98.01% for the automatic detection of ALL. Others opted for architectures combining both CNNs and ViTs to further enhance feature extraction. For instance, Tanwar et al. [60] combined in their study the ResNet50 model with the ViT, establishing a dual-stream architecture, reaching an accuracy of 99%.

DL also proved efficient in the classification of other types of cancer such as Lymphoma. Its potential was thoroughly explained by several researchers [58] [35], stressing the application of CNNs and ensemble techniques. Ozgur et al. [49] developed a triple classification system of various Lymphomas: CLL, FL, and MCL, and employed a combination of ML and DL algorithms, reaching precisions of 94%, 92%, and 82%, respectively.

Sickle Cell Anemia and Malaria can be diagnosed by examining the patient's RBCs. Harahap Mawaddah, et al. [29] used a data set that regroups 27.588 images of infected and healthy individuals' RBCs provided by Yasmin M. Kassim et al. [23]. 2 CNN architectures were compared during the classification. LeNet-5 [1] was deemed more precise than DRNet [46] in classifying RBCs affected by Malaria, with accuracies of 95.7% and 95%, respectively. Alzubaidi Laith, et al. [22] introduced a CNN classifying RBC into 3 classes, namely normal, abnormal, and miscellaneous. They used the same network as a feature-extractor, then applied the Error Correcting Output Codes (ECOC) classifier for the classification task, achieving an accuracy of 92.06% In addition to neural networks, Machine Learning techniques were also employed to address the problem of Blood Cell Image Analysis. Aliyu Hajara Abdulkarim, et al. [17] compared Support Vector Machine (SVM) against Deep Learning methods using AlexNet architecture [5]. The dataset used was open-sourced and distinguished 4 types of RBC abnormalities along with their normal shape. The accuracy for the CNN model was relatively weak and couldn't exceed 33%, while the SVM model achieved a perfect 100% on the RBC data set. The latter was deployed with the Radial Basis Function (RBF) default setting; this same network was employed by Syahputra Mohammad Fadly, et al. [15], achieving an accuracy of 83.3% using Canny Edge Detection for preprocessing and feature extraction to classify 3 types of RBC abnormalities.

Label-free identification was also explored by various researchers using an imaging flow cytometer to classify unstained WBCs [19] and optofluidic time-stretch microscopy along with Machine Learning for aggregated platelets detection as well as single platelets and WBC [13].

Visual or Vision Transformers were introduced by Dosovitskiy, et al. [28] in 2020, to exploit transformers in visual applications. Given that image classification is rather a novel concept for transformers, it may take a while to fully develop and exploit ViT in this regard. Compared to ViT, CNN can handle large-scale data sets better and offer excellent results. ViT, however, is known for its understanding of global context and dependencies, although it requires pretraining large amounts of data to achieve comparable results to CNN. [34] Therefore, an ensemble ViT/CNN model can be an excellent approach to incorporate ViT's efficiencies with CNN, this was previously done by Y.Barhoumia, et al. [26] to address another consistent problem of Intra Cranial Hemorrhage Classification. It was also employed by Jiang Zhencun, et al. [32] to diagnose ALL. The ensemble model method used is the weighted-sum model; the output results of the ViT models are multiplied by a coefficient of 0.7, and the output results of the EfficientNet model [21] are multiplied by a coefficient of 0.3. The authors later combined the results to get the final prediction result. The ViT-CNN ensemble model achieved outstanding results with an accuracy of 99.03%, exceeding the models in the literature.

A comparative summary of recent studies on cancer classification using deep learning methods is presented in Supplementary Material: Section 1 (Table S1), which provides the datasets used, classification techniques, number of classes, and accuracy values reported.

3 Methods

3.1 Data acquisition

The data set used for the classification was acquired by combining different sources.

1. The **Chula RBC-12-data set** [33] of RBC blood smear images, which contains a total of 706 smear images describing 13 classes of RBC, and comprising over 20K images of normal and pathological RBC. The images provided were collected at the Oxidation in Red Cell Disorders Research Unit, Chulalongkorn University in 2019, with a DS-Fi2-L3 Nikon microscope used at 1000x magnification. The 13 classes are specified as follows: Normal cell, Macrocyte, Microcyte, Spherocyte, Target cell, Stomatocyte, Ovalocyte, Teardrop, Burr cell, Schistocyte, uncategorized, Hypochromia, Elliptocyte. 2 classes were neglected for the lack of blood smear images.

2. The **ThalassemiaPBS-data set** [40] contains 7108 images of peripheral blood smear images of four thalassemia patients for nine cell types (Elliptocyte, Teardrop, Normal cell, Cigar cell, Stomatocyte, Target

cell, Hypochromia, Spherocyte, Acanthocyte). The images were collected by a clinical pathologist from the Clinical Pathology Laboratory of the Faculty of Medicine, Public Health and Nursing, Universitas Gadjah Mada, Indonesia, using the Olympus CX21 microscope attached with an Optilab advance plus camera with 1000x total magnification.

3. The **RBC-mini data set, Anti-Cancer Center El-Oued, Algeria** [57]: A small data set fragment (mini-batch) provided by the specialized healthcare facility: Anti-Cancer Center in El-Oued, Algeria, that contains a total of 13 blood smear images, regrouping 5 different types of RBC disorders: Burr-cells, ovalocyte, schistocyte, stomatocyte, and teardrop. The blood smear images were captured in May 2024 using an optical microscope with a x1000 magnification. These images were integrated to augment the diversity of the RBC class and mitigate overfitting risks, not to serve as a core data source.

   Table 1 regroups all 3 sources of RBC data sets and lists the size of each data set per type of cell disorder, before and after the application of data augmentation techniques described in Section 3.3.1.
   The total size of the RBC data set is 29,363.

4. The **Raabin-Leukemia data set** [39] is a free-access data set of microscopic images of blood cells, focusing on cases related to Leukemia. 2 experts labeled the cells, and the samples were captured from patients at the Takht-e Tavous laboratory in Tehran, Iran. The Zeiss microscope and LG J3 smartphone camera were used for the imaging.

5. The **Malignant Lymphoma Classification data set** [4] contains a significant number of labeled histopathological images of lymphoma, 3 types of this cancer are covered in this data set: Chronic lymphocytic leukemia (CLL), follicular lymphoma (FL), and mantle cell lymphoma (MCL), through biopsies sectioned and stained with Hematoxylin/Eosin (H+E).

   Tables 2 and 3 present the Lymphoma and Leukemia datasets, respectively, compiled from the Malignant Lymphoma Classification dataset and the Raabin-Leukemia dataset. The tables show the class distribution before and after applying the data augmentation techniques described in Section 3.3.1. The Lymphoma dataset consists of 1,436 images, while the Leukemia dataset contains 16,811 images.

3.2 Image cropping

Figure 1 presents a representative blood smear image from the Chula RBC-12 data set.[33] Each image was manually cropped to focus on individual RBCs and relevant regions of interest. They were subsequently categorized based on specific morphological characteristics. The organization of these classes was performed meticulously to ensure consistency with the referenced files provided by the authors.
The accompanying ”Label” folder within the data set houses a series of files providing detailed annotations for each image, structured in a specific format: the x-coordinate, y-coordinate, and the corresponding RBC type encoded as a numerical value (each class is given a unique value from 1 to 11).
This labeling system facilitates the task of accurate identification and classification of RBCs, thereby serving as a foundation for various haematological studies and the development of automated diagnostic tools.
The same process was replicated on the RBC-mini data set which we collected in collaboration with the Anti-Cancer Center in El-Oued, Algeria. The resulting blood smears were preprocessed and cropped using the OpenCV library as described below. The extracted images were manually labeled under the supervision of specialists at the Center.

1. Load the Image using OpenCV.
2. Preprocess the Image by converting it to grayscale and applying thresholding or edge detection to highlight the cells.
3. Find the cells using contour detection.
4. Extract each cell based on the detected contours and save them as separate image files.

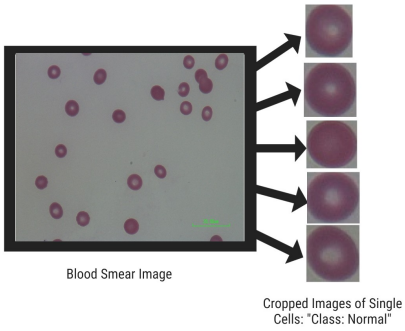


Figure 1: Images of single-cells cropped from one blood smear image

The images provided by the **ThalassemiaPBS-data set** [40] already consisted of single cells; therefore, no further preprocessing was needed. This process was necessary to isolate and classify specific morphological abnormalities.
In contrast, the leukemia [39] and lymphoma [4] data were not cropped as single cells. Instead, the whole blood smear images were retained as input, since the spatial context and global information contained in the whole smear image are all positively contributing towards the classification of leukemia subtypes and malignant lymphomas. These differences in preprocessing reflect the varying nature of the diagnostic tasks and were

Table 1: The complete Red Blood Cells data set description by type of cell and data size, including before and after augmentation

Index	Type of RBC	No. of images [33]	No. of images [40]	No. of images [57]	Total (with augm)
1	Acanthocyte	0	354	0	1432
2	Burr Cell	90	0	10	982
3	Cigar Cell	455	24	0	1893
4	Hypochromia	90	222	0	1284
5	Normal	1812	1426	0	3292
6	Ovalocyte	114	1211	4	3735
7	Schistocyte	108	0	8	453
8	Spherocyte	92	562	0	2640
9	Stomatocyte	49	382	3	1792
10	Target Cell	651	851	0	3912
11	Teardrop	26	2085	6	7948

Table 2: The Lymphoma data set description by type of cell and data size, including before and after augmentation

Category	Subtype	Before	After
Lymphoma	CLL	113	443
	FL	139	526
	MCL	122	467

Table 3: The Leukemia data set description by type of cell and data size, including before and after augmentation

Category	Subtype	Before	After
Leukemia	ALL (L1)	377	1131
	ALL (L2)	3595	3595
	AML (m0)	672	997
	AML (m1)	425	1700
	CLL	1071	3741
	CML	1624	5647

taken into account during the design of the model pipelines.

3.3 Data processing

3.3.1 Data augmentation

Data augmentation is a technique that is essential in image processing. It consists of artificially enhancing the size of a given data set by making changes to the original images. Furthermore, this method presents a solution to improving the model's performance by mitigating common issues like overfitting.

These variations of the existing images generated by the data augmentation techniques provide a more robust data set. These alterations can consist of simple geometric transformations, and color or noise introductions, all designed to make the model's predictions more generalizable and accurate.

In the present study, three primary data augmentation techniques were employed, namely: flipping, which involves mirroring the image horizontally or vertically, rotation, which involves altering the image by turning it by a specified degree, and Gaussian blurring, which can help reduce noise and minor details by adding a Gaussian filter to the image.

When combined, these augmentation techniques allowed us to enrich our data set, all the while relying on additional data preprocessing techniques that will be introduced in the following sections.

3.3.2 Data resizing

Another vital preprocessing technique before training the model is "Resizing". Since our data set is acquired from various sources, it is rather imbalanced, and images come in different sizes and shapes. Therefore, the sizes must be standardized into a uniform squared dimension before feeding the images into the model. This will allow the model to learn efficiently and improve its accuracy. Each model expects a certain target size for the images. ResNet50V2 model, for instance, requires a target size of (224, 224, 3); we were able to apply it using the flow_from_directory() method in Keras. EfficientNetB3, however, expects input images of shape (300, 300, 3) by default, but the model can accept other input shapes as long as the shape is at least 224 × 224 and the number of channels is 3 (RGB); thus, the input size was resized to (224, 224, 3) to reduce computation time and memory usage.

When provided with the target size, Keras uses bilinear interpolation by default for the image resizing operation. The formula specified below is a representation of the process in which the original coordinates are mapped to new ones using interpolation.

$$\text{new}(i', j') = \text{interpolate}\left(\text{orig}\left(i \cdot \frac{H_{\text{orig}}}{H_{\text{tgt}}}, j \cdot \frac{W_{\text{orig}}}{W_{\text{tgt}}}\right)\right) \quad (1)$$

3.3.3 Data rescaling

To ensure uniformity across input data and improve model training, all images were rescaled using appropriate preprocessing techniques depending on the model architecture. To further enhance the CNN-based models' efficiency, we've used "Rescaling", a technique in which the image's range of pixel values is changed to a standard or normalized range.

There are two common rescaling techniques: Standardization and normalization. In our paper, we've opted for the latter, which ensures that various pixel values are used during the model's learning process. The pixels of a given image can be represented as integers in the range of 0 to 255 in the case of an 8-bit image. Rescaling modifies these values into a different range of -1 to 1 or 0 to 1 when using normalization.

Likewise, we've used the flow_from_directory() method to rescale the images by a factor of 1/255 for our training, validation, and test batches. This method uses a form of Min-Max Scaling, where each pixel value is divided by 255. The minimum value (0) in this case maps to 0, and the maximum value (255) in turn maps to 1. Its formula can be defined as follows:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{2}$$

Meanwhile, for the ViT model, a different preprocessing strategy was implemented to fit its expected input distribution. Mean–standard deviation normalization was applied to standardize the image data and improve the model's convergence. Each image's pixel values were normalized using the following channel-wise means and standard deviations:

- Mean: [0.485, 0.456, 0.406]
- Standard deviation: [0.229, 0.224, 0.225]

This normalization follows the formula:

$$\text{normalized_pixel}(i, j) = \frac{\text{pixel}(i, j) - \text{mean}}{\text{std_dev}} \tag{3}$$

Additionally, Supplementary Material: Section 2.2 includes the parameter-level details of the aforementioned data augmentation techniques.

3.4 Proposed solution

In this section, we present the architectures we employed for our blood-cell classification system based on the latest deep-learning techniques. Three state-of-the-art models were explored for this task: EfficientNetB3, ResNet50V2, and Vision Transformer (ViT). To further enhance the classification accuracy, we developed ensemble models combining the strengths of ViTs and CNNs. In training, Transfer Learning was used to fine-tune each of the cited architectures, and the hyperparameters were optimized to suit the specifics of the corresponding data set.

The choice of models and more specific details are explained later in the section.

3.4.1 EfficientNetB3

A member of the EfficientNet family that was first introduced in May 2019 by [21]. This architecture was chosen due to its superior performance in feature extraction and its ability to balance computational efficiency with high accuracy, making it well-suited for tasks like blood cell classification.

EfficientNets are developed based on AutoML and compound scaling. The authors first used the AutoML MNAS Mobile framework to develop a baseline network, which they named EfficientNetB0, the first of the EfficientNet family. They then used the compound scaling method to scale up and obtain the series from B1 to B7.

The architectures achieved higher accuracy and efficiency despite being smaller and, thus faster than other models.

In our paper, we have opted for the B3 version which gave promising initial results, additional layers were added to adapt the model for blood cell classification.

We additionally adjusted key hyperparameters meticulously during training, such as learning rate, batch size, and dropout rate.

Figure 2 shows the architecture of the EfficientNetB3 base model that we have adopted for our specific classification task. The architecture was created using diagrams.net (formerly known as draw.io). [31]

The model is first fed microscopic images resized to (300, 300, 3) and processed through its pretrained backbone. The default fully connected classification head of EfficientNetB3 had been removed since it is only specific to the ImageNet data set it was trained on (containing 1000 classes), which allowed us to add the custom classification head tailored to our data set.

Three versions of the same architecture were used, each with a modified softmax layer to adapt to our three different data sets: *(1)* for RBC classification, 11 classes, *(2)* for Leukemia classification, 6 classes, *(3)* for Lymphoma classification, 3 classes.

The EfficientNetB3 backbone acts as a feature extractor, extracting spatial and hierarchical features that are later fed to the added layers for learning. The first five layers are frozen to prevent the weights from being updated during training. This helps to adapt the deeper layers to our data set; whereas freezing more layers could have resulted in under-fitting since the data set has unique characteristics that are significantly different from the original ImageNet data set. Deeper layers of the EfficientNetB3 backbone are unfrozen to enable the model to capture more patterns. (eg: cell morphology, staining patterns, etc).

This version of the model expects a (300, 300, 3) input shape by default, we have resized the input size to (224, 224, 3) to speed up training and reduce memory usage due to limited resources and the size of our data set which is

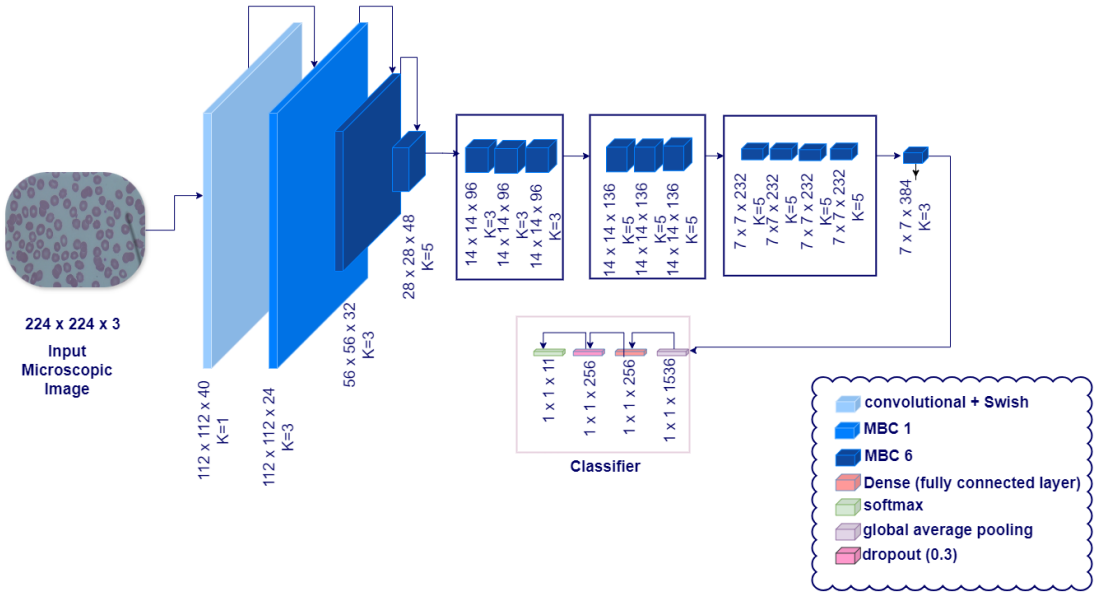


Figure 2: EfficientNetB3 model for blood cell classification: The model processes 300x300x3 microscopic images through convolutional layers with Swish activation, followed by mobile inverted bottleneck blocks (MBConv 1 and 6). The first 5 layers are frozen, with fine-tuned deeper layers. A custom classification head is added for task-specific classification

rather small.

A dropout layer is added after the global average pooling (GAP) to reduce the overfitting that we have been subjected to due to the depth of the network against the small size of the data set. A dropout rate of 0.3 was employed, thus deactivating 30% of neurons during training. This prevents the model from relying on these specific neurons.

A fully connected layer with 256 units using the ReLU activation function is added, serving to learn complex representations.

The classification head is completed with the final output dense layer. The number of units corresponds to the number of classes in each of our 3 data sets. The softmax activation function is used to specify the multi-class classification.

3.4.2 ResNet50V2

Deep convolutional neural networks have contributed to the Image-Classification field significantly, providing a robust platform to researchers ever since the emergence of the first-ever deep neural network; LeNet in 1998. Later on, in 2012, the idea of Dropout was presented, allowing the model to avoid overfitting.

Researchers next focused solely on adding more convolutional layers to increase the depth of the model, and thus, its efficiency. However, the idea of simply stacking up layers didn't benefit researchers as it introduced a whole new issue, the accuracy degradation, which unexpectedly wasn't due to overfitting, rather, it was caused by the Vanishing Gradient Effect. [6]

Residual Neural Networks to the rescue! In 2015, ResNet152, the first of the ResNet family was introduced. It consists essentially of modularized architectures that stack building blocks of the same connecting shape, called short-cut connections, that skip one or more layers. [10] These connections in ResNet work by performing identity mapping, the outputs of this mapping are added to those of the stacked layers as illustrated in Figure 3.

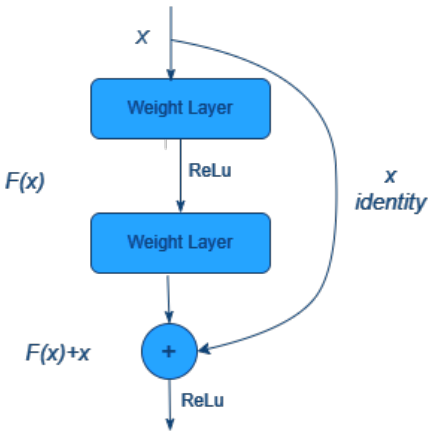


Figure 3: Residual learning - a building block

ResNet50V2 is a residual neural network variant that employs skip connections to prevent vanishing gradients during back-propagation; This ensures efficiency in learning complex features present in microscopic blood cell images.

Figure 4 presents the architecture of the ResNet50V2

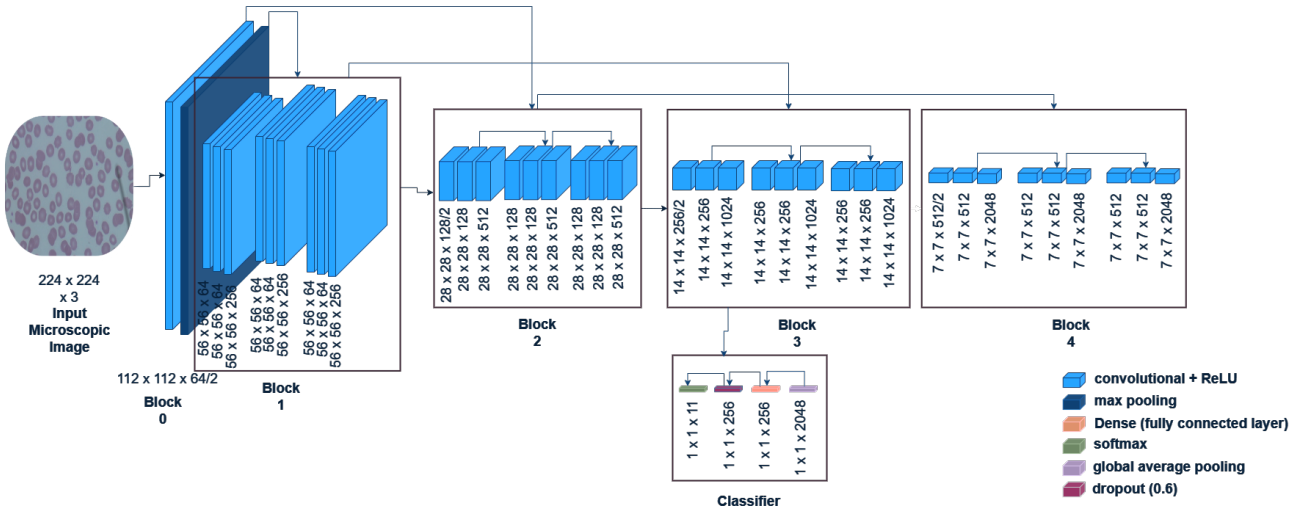


Figure 4: ResNet-50v2 model for blood cell classification: The model processes 224 × 224 × 3 microscopic images through a series of convolutional layers with ReLU activation. It consists of four main blocks with residual connections and employs bottleneck blocks (1x1, 3x3, 1x1 convolutions). A global average pooling layer is added, followed by a fully connected classification head, and a softmax activation for predicting blood cell classes. Key components include skip connections, dropout (0.6), and task-specific fine-tuning

base model that we have employed for our classification. Similarly, the model was also designed using the diagrams.net tool.
The model is fed a microscopic image of size (224,224,3) that is previously preprocessed and normalized.
It consists of 50 layers, focusing more on improved gradient flow and training stability by introducing pre-activation residual blocks and applying batch-normalization and activation (ReLU) before convolutions.
The network's initial block captures low-level features such as edges, textures, and patterns by implementing convolutional and pooling layers, followed by 4 residual blocks, with skip connections to prevent vanishing gradient problems. Higher-level features are extracted using down-sampling (strides).
The final output of these blocks is passed through a global average pooling layer (GAP) to reduce the feature map to a 1D vector; A fully connected layer and a softmax classifier are added.

The base model acts as a feature extractor, and the custom layers act as a task-specific classifier, tailored to blood-cell classification. Similar to the EfficientNet, the model was transfer-learned by freezing the first 5 layers. This prevents overfitting as the learning focuses on the deep layers, and a dropout layer is also added to ensure the model does not memorize the training data. Preceded by a 256 units dense layer that allows the model to combine the learned features to improve classification and the classifier ends with a dense layer that has the same number of neurons as the classification classes: (1) for RBC classification, 11 classes, (2) for Leukemia classification, 6 classes, (3) for Lymphoma classification, 3 classes.

3.4.3 Experimental hyperparameters

Table 4 presents a breakdown of the hyperparameters and setup used in the experiments based on the training pipeline using Keras/Tensorflow, along with their purposes, as well as the strategies employed to transfer-learn and fine-tune the models and achieve the best accuracies possible.

The EfficientNetB3 and ResNet50V2 models were both trained using the same hyperparameters detailed in Table 4.

3.4.4 Experimental environment

Hardware: The experiments for all 3 models were conducted by Google Colab; It typically consists of NVIDIA Tesla GPUs.

Software: *Platform:* Google Colab, a hosted Jupyter Notebook environment.
Framework(s): Tensorflow v2.18.0 and Keras v3.6.0 were used to develop, train, and evaluate the 3 models.
Python: Version (3.10).
Libraries: matplotLib, numpy, PIL, joblib, and more were used to preprocess, analyze, and visualize data.

Storage: The 3 data sets were preprocessed and split into Training, Validation, and Test data sets, each stored in Google Drive, which is mounted to the Colab environment for access. The detailed data split strategy, including training, validation, and testing partitions, is provided in Supplementary Material: Section 2.1.

Table 4: Experimental hyperparameters for training the CNN models and their purposes

Hyperparameter	Value	Purpose
Optimizer	Adam (LR=10^{-4})	A low learning rate is employed to fine-tune pre-trained layers.
Loss function	Categorical crossentropy	Used for multi-class classification.
Metrics	Accuracy	To monitor the number of correctly classified instances during training.
Steps-per-epoch	ResNet50V2: 20, EfficientNetB3: 200	The number of training batches processed per epoch.
Validation steps	ResNet50V2: 10, EfficientNetB3: 316	The number of validation batches processed per validation step.
Epochs	ResNet50V2: 300, EfficientNetB3: 10	Specifies the training schedule which allows gradual convergence.

3.5 ViT model

Visual or Vision Transformers (ViT) is a novel approach introduced by Dosovitskiy et al. [28]. It uses the concept of transformers designed specifically for visual applications and image classification tasks in particular.
When using the transformer blocks in ViT, the multi-head attention mechanism is applied to integrate global context efficiently and learn high-level features. [42]
Following the success of NLP transformers [16], Dosovitskiy et al. were inspired to develop a new attention-based class of models that can be exploited in Computer Vision.
Compared to NLP transformers, ViT only uses the encoder attention branch, neglecting the decoder attention branch, whilst word tokens are replaced by image patches.
In a normal CNN, the entire image is taken as input, whereas in ViT, the image is first divided into equal-sized patches, which are passed through some linear layers; the outputs of this layer are known as patch embeddings. Between these embeddings, we have the position embeddings, which serve to provide the model with some positional information regarding the sequence of these patches. Afterward, another learnable token is added to the position embedding for image classification purposes.

Figure 5 presents the architecture of the ViT model we have employed for our blood cell classification task. Prior to the training phase, the data was first prepared and processed to fit the model's requirements and expected input. The data set was initially split into training, validation, and test sets and stored in specific folders. The ImageFolder utility was used to load the images and associate them with their corresponding classes based on the folder names provided. The images were later resized to fit the shape expected by the ViT model: 224×224. Further normalization was implemented to standardize the image data, and make it more suitable for the model (See Section 3.3.3).

Similarly to the CNN architecture, three versions were implemented for each data set: *(1)* for RBC classification, 11 classes, *(2)* for Leukemia classification, 6 classes, *(3)* for Lymphoma classification, 3 classes.

The pretrained backbone uses the google/vit-base-patch16-224-in21K model from the Hugging Face library [25] as a feature extractor. The model was trained on the ImageNet-21K data set [8]. It was fine-tuned to adapt to the blood cell classification task, where the number of labels was defined as the number of classes in the data set as mentioned previously in this section.

The transformer encoder depicted in Figure 5 is first provided with the embedded patches (Patch-embedding/ Position-embedding). The input image is divided into fixed-size patches of 16x16. We next apply a linear projection on the flattened patches to form a fixed-dimensional vector. Unlike CNNs, Transformers require position embeddings to learn and capture the input's order of sequence [38]. It serves to improve accuracy and encode the spatial information of the patches.

The combined embedded patches are fed into the Transformer Encoder to go through a series of L layers, each including a list of components, as follows:

1. *Multi-head self-attention* is a mechanism that enables the model to learn global patterns by splitting the process of self-attention into multiple heads, where each head focuses on the interaction between patch-embeddings differently.[16] The attention calculations are eventually merged to give a more global score.

2. The *output* of the Multi-head attention is added to the input of the next component by a *skip connection* (residual connection) after normalization. As explained earlier, residual connections are added to prevent the vanishing gradient during training.

3. To further enhance the model's learning through patch-embeddings, a *feed-forward network (FFN)*

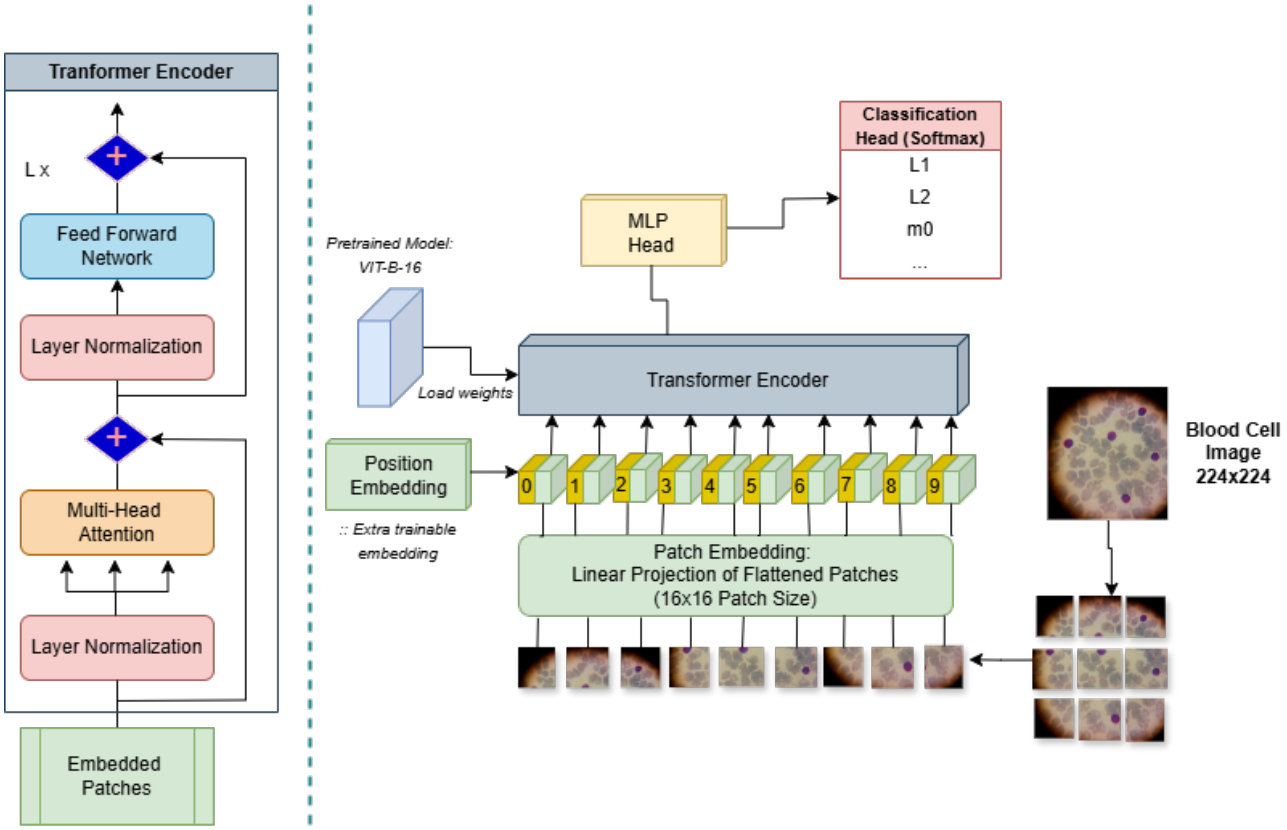


Figure 5: Vision Transformer (ViT) Architecture for blood cell classification: The model processes 224×224 microscopic images through patch embeddings and position encoding, which are later fed to the transformer encoder with loaded weights from the pretrained ViT-B-16 in 21K model. After passing through the transformer encoder, the embeddings are used as the input to the classification head (MLP + Softmax)

[48] is fed to the normalized output of the Multi-head attention, which consists of fully connected layers with a GeLU activation in between. This allows the model to capture local transformations.

4. Similarly to the Multi-head self-attention block, the *FFN* is normalized and added to the *residual connection*.

The output of the Transformer Encoder is a sequence of embedding, enriched with local and global contextual information, independently for each patch.
After passing through the Transformer Encoder, the embedding corresponding to the special classification token (cls) is used as the input to the classification head that consists of a Multi-Perceptron head (MLP), and a softmax classification head.
The MLP takes the output of the Transformer Encoder and feeds it into a series of fully connected layers to prepare the data for the softmax classification head that processes it to the desired classes.

3.5.1 Experimental hyperparameters

Table 5 outlines the hyperparameters used for training a ViT model using the backbone google/vit-base-patch16-224-in21k, along with their respective values; detailing the batch size, the learning rate, and the optimizer employed. The OneCycleLr Scheduler was used as a strategy to vary the learning rate during training; each cycle uses a maximum of 10^{-3} as a learning rate. Other parameters include the CrossEntropyLoss function and a total of 10 epochs (624 batches per epoch) to train the model.

Table 5: Experimental hyperparameters for training the ViT model

Hyperparameter	**Value**
Batch size	32
Learning rate	10^{-4} (initial)
Optimizer	AdamW
Scheduler	OneCycleLR
Scheduler Max lr	10^{-3}
Number of epochs	10
Loss Function	CrossEntropyLoss
Model backbone	google/vit-base-patch16-224-in21k

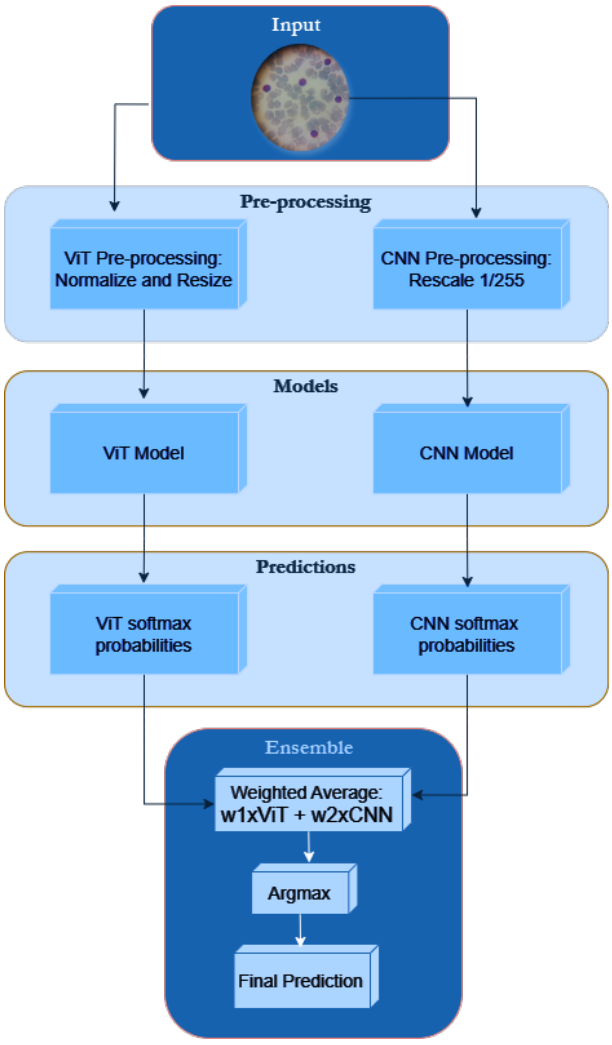


Figure 6: Workflow of the weighted average ensemble method: The input is preprocessed to fit the ViT and the CNN models, and the predictions of the models are later combined using the weighted average approach to generate the final prediction

3.5.2 ViT-CNN ensemble model

To further enhance the performance of our models, an ensemble method was introduced to seek the opinions of several models and combine them to achieve highly accurate classifications than those of the raw models when trained separately [44] [50]. Through our experiments, we have observed the superiority of residual networks in training and efficient learning, while the ViT model performed better in certain instances, focusing more on learning complex features. Thus, we incorporated in our methodology a dual-architecture ensemble, combining the residual network's efficiency with the high precision obtained by the ViT.

Figure 6 presents the flowchart of the ResNet-ViT ensemble model that we have implemented.
The weighted-average ensemble method was selected after experimenting with the most prevalent methods in image classification tasks, namely, maximum voting, the averaging method, and the weighted sum. In the weighted-average method, the models are assigned different weights after training, defining the importance of each model for prediction.

The weighted-average ensemble combines predictions from a CNN (M_1) and a Vision Transformer (M_2), with output probabilities for class c denoted as $P_1(c)$ and $P_2(c)$, respectively, obtained via the softmax function to ensure $\sum_c P_i(c) = 1$ for $i = 1, 2$. The weights w_1 and w_2 are assigned to M_1 and M_2 based on validation performance. The ensemble probability for class c is computed as:

$$P_{\text{ensemble}}(c) = \frac{w_1 P_1(c) + w_2 P_2(c)}{w_1 + w_2} \tag{4}$$

The final class prediction is determined by selecting the class with the highest ensemble probability:

$$\hat{c} = \arg\max_c P_{\text{ensemble}}(c) \tag{5}$$

Preprocessing: The input fed to the already trained models is first preprocessed; each model is preprocessed differently. The ViT uses normalization with mean and std, while the CNN uses simple rescaling (1/255). The models are later loaded to make predictions, and both models output probabilities for the different classes; we used the softmax function to ensure that they sum up to 1.

Weight Selection: Weights w_1 and w_2 were determined through a grid search over predefined pairs, specifically $[(0.3, 0.7), (0.4, 0.6)]$, where each pair sums to 1 to maintain normalized probabilities. The grid search evaluated each weight combination on a validation subset using classification accuracy as the performance metric. The pair that achieved the highest accuracy was selected. Further details about the ensemble weight selection and performance across datasets are provided in the Supplementary Material: Section 3.

4 Results

This section provides an in-depth analysis of the results obtained from our experiments. First, we explore the performance of our individual models, using the insights present in the confusion matrix and focusing on metrics such as: (1) accuracy, (2) precision, (3) recall, (4) F1-score, (5) Cohen kappa, and (6) AUC scores, followed by an evaluation of the ensemble model, HematoFusion. The evaluation is conducted across the three data sets we have introduced in earlier sections, and the results are eventually interpreted in the context of existing literature.

The *Accuracy* is calculated by measuring the number of predicted cases. When achieved, a high accuracy means the overall performance of the model is good. However, in the case of imbalanced data sets, high accuracy can be misleading, and other metrics are necessary to further evaluate the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

The *Precision* is calculated by measuring the number of correctly predicted positive cases. A high precision is achieved only if most of the positive cases are correctly predicted. [45]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

The *Recall*, also known as *Sensitivity* or *True Positive Rate* measures whether all relevant cases of the data set were correctly predicted.[45]

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

To address the accuracy's shortcomings in handling imbalanced data sets, which is the case in our paper, the *F1-score* was introduced for balanced evaluations, combining precision and recall in one metric. The F1-score eventually only achieves a high accuracy when both precision and recall are high. [43]

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

The *Cohen Kappa* was introduced as a statistical measure of the agreement between the predicted labels and their actual values. If κ=1, the perfect agreement is achieved, if κ=0, the agreement is random, and if κ<0, it means the model achieved more than a random agreement. [36]

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{10}$$

where:

P_o = Observed agreement (accuracy)

P_e = Expected agreement based on chance

$$P_e = \sum_{i=1}^{k} \frac{(A_i \cdot B_i)}{N^2}$$

The *Area Under the Curve (AUC)* score, specifically the Area Under the Receiver Operating Characteristic (ROC) curve [24], evaluates a model's ability to discriminate between classes at various thresholds of classification. An AUC score approaching 1 indicates high discriminative capability, particularly useful for unbalanced datasets that are common in blood cell classification, where accuracy becomes misleading based on class difference.

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR} \tag{11}$$

where:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \text{ (True Positive Rate or Sensitivity)}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \text{ (False Positive Rate).}$$

4.1 Classification results

To analyze the model's performance during training, the accuracy and loss were both monitored and visualized through the training curves over successive epochs.
Figures S5, S6, and S7 (Supplementary Material Section 7) depict the training curves for the RBC Morphology, Leukemia, and Lymphoma data sets, respectively.

Additionally, for further visual evaluation of the classification performance, confusion matrices were computed on the test set, showing the number of accurate and inaccurate predictions of instances, namely: True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP).
The confusion matrices generated for the RBC Morphology, Leukemia, and Lymphoma datasets are displayed, respectively, in Figures 7, 8, and 9.

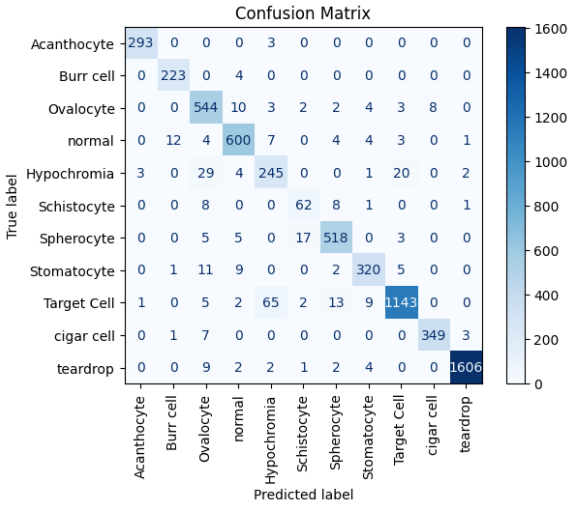
These confusion matrices were used to compute the quantitative metrics for a more specific evaluation.
Detailed tables presenting per-class performance metrics (precision, recall, and F1-score, kappa) for each model and dataset combination are provided in the Supplementary Material: Section 4, Tables S3–S5.

The results for the classification of RBC Morphology, Leukemia, Lymphoma, the Ensemble model, HematoFusion are summarized in Tables 6, 7, and 8, respectively. To test model stability and robustness over sets, we conducted a bootstrapping analysis. This technique provides us with an estimate of results' variability and increases the validity of our performance claims over single-run statistics. The detailed bootstrapping results with distribution plots and summary statistics are presented in Supplementary Material: Section 6 (Table S7 and Figures S2–S4).
Furthermore, we have calculated and included the AUC scores (Table S6) and ROC curves (Figure S1) for our individual models across each dataset, as shown in Supplementary Material: Section 5.

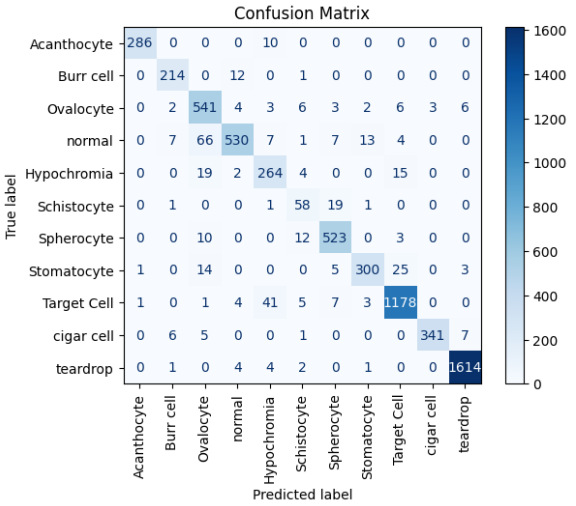
5 Discussion

5.1 Interpretation of results

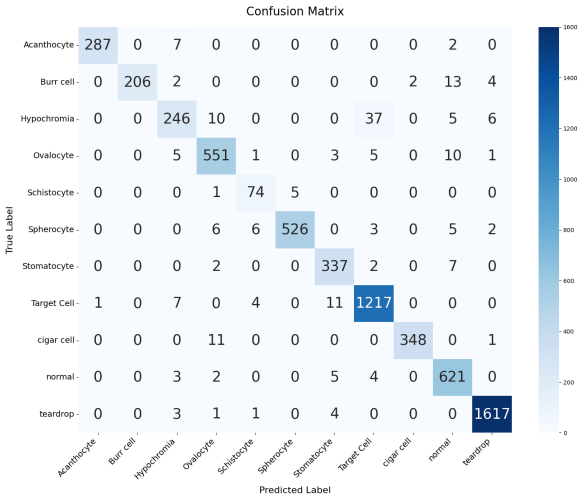
The convergence of the ResNet50V2 model illustrates a steady reduction in training loss, and the accuracy becomes stable after reaching a certain number of epochs. The ViT model has demonstrated higher fluctuation in both accuracy and loss during training.



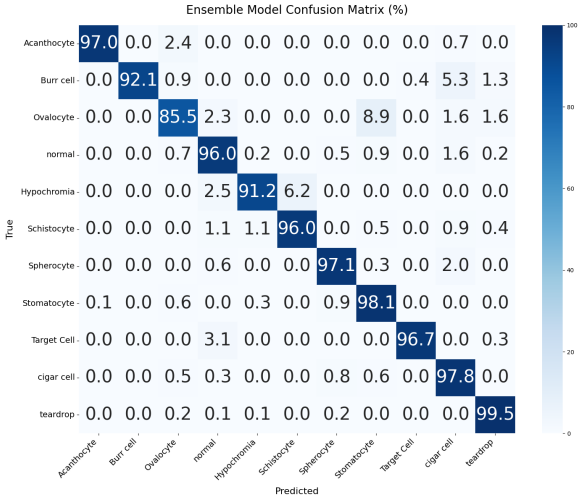
(a) EfficientNetB3 model



(b) ResNet50V2 model



(c) Pretrained ViT model



(d) HematoFusion model

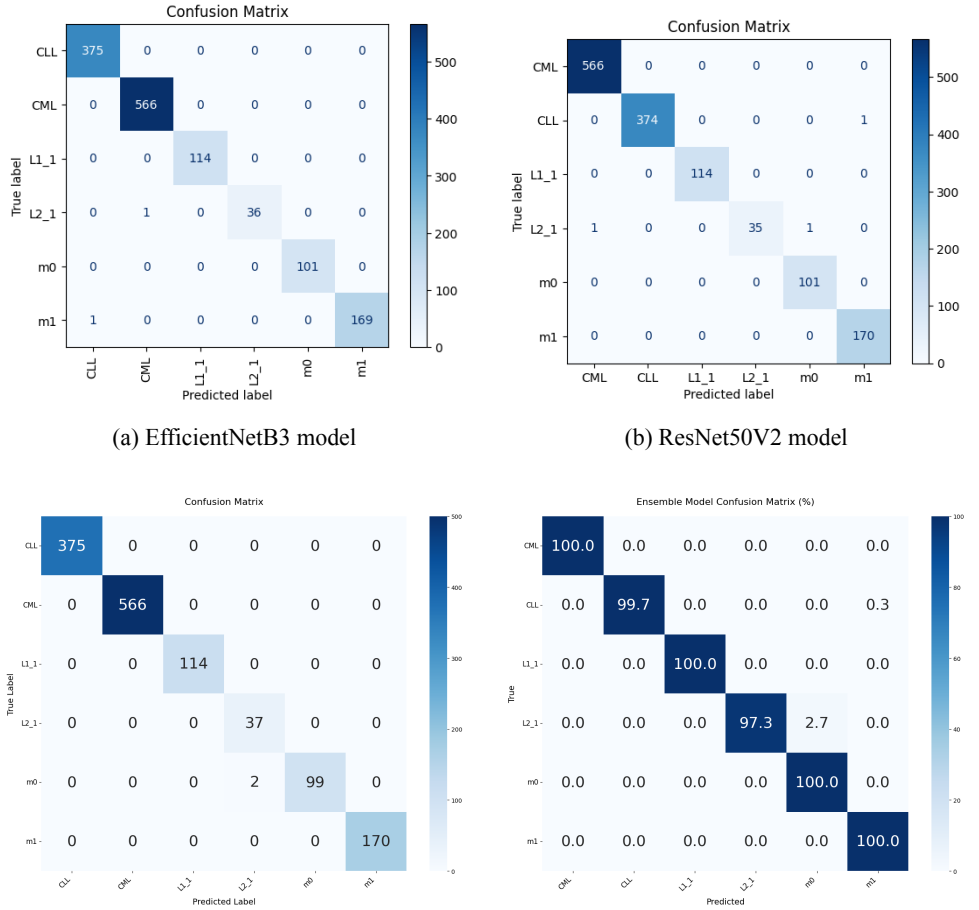
Figure 7: Confusion matrices for the classification performance of the four models on the RBC data set: (a) EfficientNetB3 model, (b) ResNet50V2 model, (c) pretrained ViT model, (d) HematoFusion model combining the ViT and ResNet50V2 models.

Table 6: RBC Morphology classification results across the three individual models with detailed metrics for evaluation

Model	Model Performance						
	Train Acc	Val Acc	Test Acc	Kappa	Recall	F1-score	Precision
EfficientNetB3	0.99	0.91	0.97	0.93	0.92	0.92	0.92
ResNet50V2	0.98	0.98	0.92	0.92	0.93	0.93	0.93
ViT	0.98	0.94	0.96	0.96	0.94	0.94	0.94

Table 7: Leukemia classification results across the three individual models with detailed metrics for evaluation

Model	Model Performance						
	Train Acc	Val Acc	Test Acc	Kappa	Recall	F1-score	Precision
EfficientNetB3	1.0	1.0	1.0	0.99	0.99	0.99	0.99
ResNet50V2	1.0	1.0	1.0	0.99	1.0	1.0	1.0
ViT	0.99	0.99	0.99	0.99	1.0	1.0	1.0



(a) EfficientNetB3 model (b) ResNet50V2 model

(c) Pretrained ViT model (d) HematoFusion model

Figure 8: Confusion matrices for the classification performance of the four models on the Leukemia data set: (a) EfficientNetB3 model, (b) ResNet50V2 model, (c) pretrained ViT model, (d) HematoFusion model combining the ViT and ResNet50V2 models.

Table 8: Lymphoma classification results across the three individual models with detailed metrics for evaluation

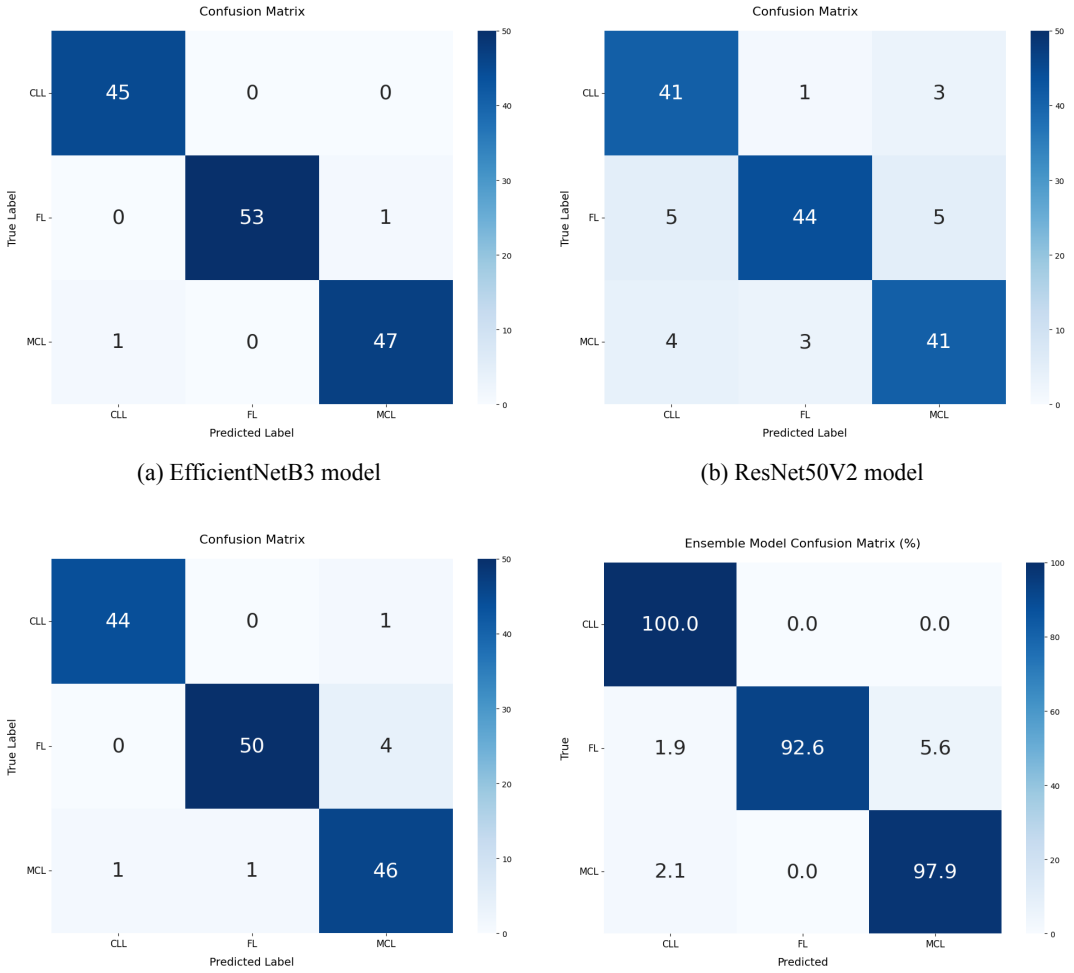
Model	Model Performance						
	Train Acc	Val Acc	Test Acc	Kappa	Recall	F1-score	Precision
EfficientNetB3	1.0	0.99	0.99	0.97	0.99	0.99	0.99
ResNet50V2	1.0	0.91	0.96	0.91	0.96	0.96	0.96
ViT	0.98	0.98	0.95	0.92	0.98	0.98	0.98

Table 9: HematoFusion ensemble model classification results across the three datasets, showing detailed evaluation metrics

Dataset	Ensemble Model Performance				
	Best Acc	Kappa	Recall	F1-score	Precision
RBC	0.96	0.94	0.97	0.97	0.97
Leukemia	0.99	0.99	1.00	1.00	1.00
Lymphoma	0.96	0.95	0.97	0.97	0.97

When comparing the True Positives of the proposed HematoFusion model with the individual models, we can clearly observe an increase in the rates of correctly classified cases and a decrease in the misclassification rates.

The individual models struggled with predicting the Hypochromia class. The ensemble model, on the other hand, exhibited a stronger ability to recognize this class. Whereas, Acanthocyte and Teardrop were easier to identify, owing to their distinguishable shapes, which was reflected in the high number of TP.



(a) EfficientNetB3 model

(b) ResNet50V2 model

(c) Pretrained ViT model

(d) HematoFusion model

Figure 9: Confusion matrices for the classification performance of the four models on the Lymphoma data set: (a) EfficientNetB3 model, (b) ResNet50V2 model, (c) pretrained ViT model, (d) HematoFusion model combining the ViT and ResNet50V2 models.

In Table 6, a slight overfitting is observed due to the class imbalance; thus, the accuracy can be misleading for accurate measurement of the performance. This, however, was addressed with the use of precision, recall, and F1-score. Although EfficientNetB3 was slightly better on test accuracy (0.97), the ViT model outperformed in stronger and more descriptive metrics, like Kappa score, precision, recall, and F1-score, which indicate more balanced performance with class imbalance. Therefore, ViT is graded as the overall best performer on the RBC morphology dataset. Table 7 shows more consistent results on the Leukemia data set across all models, achieving perfect classification, which indicates better generalizations.

Both ResNet50V2 and EfficientNetB3 achieved comparable top-tier performances on the Leukemia dataset, with identical test accuracy, precision, recall, and F1-score, and minor variations in other evaluation metrics. EfficientNetB3, alternatively, outperforms the other models on the Lymphoma classification (Table 8), reaching almost perfect accuracies.

The Ensemble model, HematoFusion, demonstrates more uniform results across all data sets in terms of all evaluating metrics, mitigating the issues with the class imbalance, as evidenced by its performances, leveraging the strengths of both the ViT and ResNet50V2 models that struggled with some classes.

The precision improved by 4% in the RBC data set and reached a perfect 100% for Leukemia classification, averaging the performances of the individual models on the Lymphoma data set with a precision of 97% on the test set. Despite the strong performance of our proposed solution, further improvement could be implemented to help the model generalize better and address the issue of class imbalance efficiently.

5.2 Comparative study

Table 9 presents a breakdown of the performance of the proposed solution across all three datasets, outlining the accuracy and precision of the model when compared to the lit-

Table 10: Comparative results of the proposed solution and the literature across different metrics for each data set

Dataset	HematoFusion		Literature	
	Accuracy	**Precision**	**Accuracy**	**Precision**
RBC	**0.96**	**0.97**	0.98	0.97
Leukemia	**0.99**	**1.0**	0.99	0.99
Lymphoma	**0.96**	**0.97**	0.96	0.96

erature.
In a bid to substantiate the efficiency of our proposed solution, we evaluated it against the following models:

1. *Literature [7] RBC Classification:* The authors presented a Maximum Voting based Ensemble Model to classify Dacrocyte (Teardrop), Schistocyte, and Elliptocyte cells (Cigar cells) in Iron Deficiency Anemia.
   The average classification precision and accuracy of the latter reached a maximum of 97% and 98%, respectively; While both models achieved the same precision of 97%, the model in the Literature reported a slightly higher accuracy (98%) compared to HematoFusion's 96%. Nonetheless, it's worth noting that our data set comprises 11 classes against the 3 classes studied in this article.

2. *Literature [32] Leukemia Classification:* The authors proposed a ViT-CNN Ensemble Model for the diagnosis of Acute Lymphoblastic Leukemia (ALL), which is one of the 6 classes that we analyzed in our paper. Compared to the model in the Literature, which achieved 99% accuracy and 99% precision on the Leukemia dataset, HematoFusion matched the accuracy (99%) but outperformed in precision, achieving a perfect 100%.

3. *Literature [41] Lymphoma Classification:* Malignant Lymphoma (ML) was addressed in this paper, which is among the 3 classes that appear in our Lymphoma data set.
   The proposed hybrid model used the combined features of 3 deep learning networks, namely, MobileNet-VGG16, VGG16-AlexNet, and MobileNet-AlexNet, to classify the models by the XGBoost and DT algorithms, reaching an average accuracy and precision of 96%.

An extended version of this comparison, covering a broader range of SOTA models and datasets, is provided in Supplementary Material: Section 8 (Table S8).
Overall, our proposed HematoFusion Ensemble model achieved a reliable performance across the 3 data sets, despite the imbalanced data set and the high number of classes in the case of RBC Morphology Classification.

5.3 Limitations

Although the reported results show high precision, reaching up to 99%, this should be interpreted with caution due to known issues like *dataset imbalance*. As identified previously, some classes were underrepresented, and this could result in biased learning as well as overfitting. To mitigate this, data augmentation techniques were employed (as outlined in Supplementary Section 2.2), and performance was monitored across a variety of metrics (precision, recall, F1-score, Cohen's Kappa, and AUC scores) rather than simply accuracy. However, we are aware that the lack of external validation data limits generalizability. Although high-performance metrics are presented, the models have not been prospectively validated within a real clinical workflow. Their incorporation into clinical decision-making would require extensive regulatory testing and interpretability evaluation. Additionally, while conventional regularization techniques such as dropout and data augmentation were applied to address overfitting, we recognize the need for more advanced strategies. Future work will explore class imbalance mitigation techniques such as SMOTE, GAN-based synthetic image generation, and uncertainty-aware training beyond testing on independent cohorts, to further assess the robustness of the model in actual clinical settings. Furthermore, we intend to conduct ablation studies on ensemble weight parameters and data augmentation strategies to evaluate their individual contributions.

6 Conclusion

In this study, the problem of pathological blood cell classification was addressed through the use of novel deep-learning strategies. We curated a data set for RBC Morphology classification, consisting of samples from three different sources. The process involved preprocessing techniques to establish a data set aligned with our research objectives; 2 other data sets were acquired, targeted for Lymphoma and Leukemia classifications separately.
Three distinct individual models were applied for each of the data sets: the EfficientNetB3, ResNet50V2, and a pretrained ViT model. To leverage the strengths of both the CNN and ViT architectures, an Ensemble model using the weighted average method was developed.
The present findings confirm that the proposed HematoFusion model mitigates the shortcomings of the individual models by enhancing the accuracy, precision, and sensitivity, achieving more consistent results across the three data sets. While HematoFusion demonstrates competitive or superior performance on Leukemia and Lymphoma classification, particularly in precision and F1-score, it performs

comparably on RBC classification, despite its higher number of classes and the issue of data imbalance that resulted in a few cases of overfitting. We additionally acknowledge certain limitations in predicting a couple of classes. These are the key components to overcome in future research. Future studies should also be devoted to covering more pathological blood disorders and implementing further processing and data augmentations to alleviate the issue of class imbalance and overfitting.

Overall, this paper provides a foundation for future developments by establishing baseline data that future researchers can expand upon to address the limited data available for RBC Morphology and combining the strengths of the residual networks and vision transformers for a more robust framework.

References

[1] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: https://doi.org/10.1109/5.726791.

[2] KS Kim et al. "Analyzing blood cell image to distinguish its abnormalities". In: *Proceedings of the eighth ACM international conference on multimedia*. New York: Association for Computing Machinery, 2000, pp. 395–397. DOI: https://doi.org/10.1145/354384.354543.

[3] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Miami: Ieee, 2009, pp. 248–255. DOI: https://doi.org/10.1109/CVPR.2009.5206848.

[4] Nikita Orlov et al. "Automatic Classification of Lymphoma Images With Transform-Based Global Features". In: *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society* 14 (2010), pp. 1003–13. DOI: https://doi.org/10.1109/TITB.2010.2050695.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Lake Tahoe, Nevada: Curran Associates, Inc., 2012, pp. 1097–1105.

[6] K. He et al. *Deep Residual Learning for Image Recognition*. Preprint at https://arxiv.org/abs/1512.03385. 2015.

[7] Mahsa Lotfi et al. "The detection of dacrocyte, schistocyte and elliptocyte cells in iron deficiency anemia". In: *2015 2nd International conference on pattern recognition and image analysis (IPRIA)*. Rasht, Iran: IEEE, 2015, pp. 1–5. DOI: https://doi.org/10.1109/PRIA.2015.7161628.

[8] O. Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. Preprint at https://arxiv.org/abs/1409.0575. 2015.

[9] J. C. Chapin and M. T. Desancho. "Hematologic Dysfunction in the ICU". In: *Critical Care*. Ed. by J. M. Oropello, S. M. Pastores, and V. Kvetan. New York: McGraw-Hill Education, 2016.

[10] Kaiming He et al. *Identity Mappings in Deep Residual Networks*. Preprint at http://arxiv.org/abs/1603.05027. 2016.

[11] Kenneth Kaushansky et al. *Williams Hematology*. New York: McGraw-Hill Education, 2016.

[12] Gao Huang et al. "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu: IEEE, 2017, pp. 4700–4708. DOI: https://doi.org/10.48550/arXiv.1608.06993.

[13] Yiyue Jiang et al. "Label-free detection of aggregated platelets in blood by machine-learning-aided optofluidic time-stretch microscopy". In: *Lab on a Chip* 17.14 (2017), pp. 2426–2434. DOI: https://doi.org/10.1039/C7LC00396J.

[14] Mazin Z Othman, Thabit S Mohammed, and Alaa B Ali. "Neural network classification of white blood cell using microscopic images". In: *International Journal of Advanced Computer Science and Applications* 8.5 (2017), pp. 99–103. DOI: https://doi.org/10.14569/IJACSA.2017.080513.

[15] Mohammad Fadly Syahputra, Anita Ratna Sari, and Romi Fadillah Rahmat. "Abnormality classification on the shape of red blood cells using radial basis function network". In: *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*. Kuta Bali, Indonesia: IEEE, 2017, pp. 1–5. DOI: https://doi.org/10.1109/CAIPT.2017.8320739.

[16] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017). DOI: https://doi.org/10.48550/arXiv.1706.03762.

[17] Hajara Abdulkarim Aliyu et al. "Red blood cell classification: deep learning architecture versus support vector machine". In: *2018 2nd international conference on biosignal analysis, processing and systems (ICBAPS)*. Kuching, Malaysia: IEEE, 2018, pp. 142–147. DOI: https://doi.org/10.1109/ICBAPS.2018.8527398.

[18] Paul Mooney. *Blood Cell Images*. 2018. URL: https://www.kaggle.com/datasets/paultimothymooney/blood-cells.

[19] Mariam Nassar et al. "Label-free identification of white blood cells using machine learning". In: *Cytometry Part A* 95.8 (2019), pp. 836–842. DOI: https://doi.org/10.1002/cyto.a.23794.

[20] N. C. Shenggan. *BCCD Dataset*. https://github.com/Shenggan/BCCD_Dataset. 2019.

[21] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Long Beach, California: PMLR, 2019, pp. 6105–6114. DOI: https://doi.org/10.48550/arXiv.1905.11946.

[22] Laith Alzubaidi et al. "Classification of red blood cells in sickle cell anemia using deep convolutional neural network". In: *Intelligent Systems Design and Applications*. Ed. by Ajith Abraham et al. Vol. 1. Cham: Springer International Publishing, 2020, pp. 6–8. DOI: https://doi.org/10.1007/978-3-030-16657-1_51.

[23] Yasmin M Kassim et al. "Clustering-Based Dual Deep Learning Architecture for Detecting Red Blood Cells in Malaria Diagnostic Smears". In: *IEEE Journal of Biomedical and Health Informatics* 25.5 (2020), pp. 1735–1746. DOI: https://doi.org/10.1109/JBHI.2020.3034863.

[24] Tatiana Cristina Figueira Polo and Hélio Amante Miot. *Use of ROC curves in clinical and experimental studies*. 2020. DOI: https://doi.org/10.1590/1677-5449.200186.

[25] Thomas Wolf et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771 [cs.CL]. URL: https://arxiv.org/abs/1910.03771.

[26] Yassine Barhoumi and Ghulam Rasool. *Scopeformer: n-CNN-ViT hybrid model for intracranial hemorrhage classification*. 2021. DOI: https://doi.org/10.48550/arXiv.2107.04575.

[27] Ferhat Bozkurt. "Classification of blood cells from blood cell images using dense convolutional network". In: *Journal of Science, Technology and Engineering Research* 2.2 (2021), pp. 81–88. DOI: https://doi.org/10.53525/jster.1014186.

[28] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. Preprint at https://arxiv.org/abs/2010.11929. 2021.

[29] Mawaddah Harahap et al. "Implementation of Convolutional Neural Network in the classification of red blood cells have affected of malaria". In: *Sinkron: jurnal dan penelitian teknik informatika* 5.2 (2021), pp. 199–207. DOI: https://doi.org/10.33395/sinkron.v5i2.10713.

[30] Danish Jamil et al. "Diagnosis of gastric cancer using machine learning techniques in healthcare sector: a survey". In: *Informatica* 45.7 (2021). DOI: https://doi.org/10.31449/inf.v45i7.3633.

[31] JGraph. *diagrams.net, draw.io*. Oct. 2021. URL: https://www.diagrams.net/.

[32] Zhencun Jiang et al. "Method for diagnosis of acute lymphoblastic leukemia based on ViT-CNN ensemble model". In: *Computational Intelligence and Neuroscience* 2021.1 (2021), p. 7529893. DOI: https://doi.org/10.1155/2021/7529893.

[33] Korranat Naruenatthanaset et al. *Red Blood Cell Segmentation with Overlapping Cell Separation and Classification on Imbalanced Dataset*. Preprint at https://arxiv.org/abs/2012.01321. 2021.

[34] Maithra Raghu et al. "Do vision transformers see like convolutional neural networks?" In: *Advances in neural information processing systems* 34 (2021), pp. 12116–12128. DOI: https://doi.org/10.48550/arXiv.2108.08810.

[35] Georg Steinbuss et al. "Deep learning for the classification of non-Hodgkin lymphoma on histopathological images". In: *Cancers* 13.10 (2021), p. 2419. DOI: https://doi.org/10.3390/cancers13102419.

[36] Željko Vujović et al. "Classification model evaluation metrics". In: *International Journal of Advanced Computer Science and Applications* 12.6 (2021), pp. 599–606. DOI: https://doi.org/10.14569/IJACSA.2021.0120670.

[37] Xufeng Yao et al. "Classification of white blood cells using weighted optimized deformable convolutional neural networks". In: *Artificial Cells, Nanomedicine, and Biotechnology* 49.1 (2021), pp. 147–155. DOI: https://doi.org/10.1080/21691401.2021.1879823.

[38] Kai Jiang et al. "The encoding method of position embeddings in vision transformer". In: *Journal of Visual Communication and Image Representation* 89 (2022), p. 103664. DOI: https://doi.org/10.1016/j.jvcir.2022.103664.

[39] Zahra Mousavi Kouzehkanan et al. "A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm". In: *Scientific Reports* 12.1 (2022), p. 1123. DOI: https://doi.org/10.1038/s41598-021-04426-x.

[40] Dyah Aruming Tyas et al. "Erythrocyte (red blood cell) dataset in thalassemia case". In: *Data in Brief* 41 (2022), p. 107886. DOI: https://doi.org/10.1016/j.dib.2022.107886.

[41] Mohammed Hamdi et al. "Hybrid Models Based on Fusion Features of a CNN and Handcrafted Features for Accurate Histopathological Image Analysis for Diagnosing Malignant Lymphomas". In: *Diagnostics* 13.13 (2023), p. 2258. DOI: https://doi.org/10.3390/diagnostics13132258.

[42] Rojina Kashefi et al. *Explainability of Vision Transformers: A Comprehensive Review and New Perspectives*. Preprint at https://arxiv.org/abs/2311.06786. 2023.

[43] Gireen Naidu, Tranos Zuva, and Elias Mmbongeni Sibanda. "A review of evaluation metrics in machine learning algorithms". In: *Computer science on-line conference*. Springer. 2023, pp. 15–25. DOI: https://doi.org/10.1007/978-3-031-35314-7_2.

[44] Austin H Routt et al. "Deep ensemble learning enables highly accurate classification of stored red blood cell morphology". In: *Scientific Reports* 13.1 (2023), p. 3152. DOI: https://doi.org/10.1038/s41598-023-30214-w.

[45] Hongwei Shang et al. "Precision/recall on imbalanced test data". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 9879–9891. URL: https://proceedings.mlr.press/v206/shang23a.html.

[46] Enquan Yang et al. "DRNet: Dual-stage refinement network with boundary inference for RGB-D semantic segmentation of indoor scenes". In: *Engineering Applications of Artificial Intelligence* 125 (2023), p. 106729. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2023.106729.

[47] Areen K Al-Bashir, Ruba E Khnouf, and Lamis R Bany Issa. "Leukemia classification using different CNN-based algorithms-comparative study". In: *Neural Computing and Applications* 36.16 (2024), pp. 9313–9328. DOI: https://doi.org/10.1007/s00521-024-09554-9.

[48] Martin Moller. "Efficient training of feed-forward neural networks". In: *Neural Network Analysis, Architectures and Applications*. CRC Press, 2024, pp. 136–173. DOI: https://doi.org/10.1201/9781003572886-8.

[49] Emine Özgür and Ahmet Saygılı. "A new approach for automatic classification of non-hodgkin lymphoma using deep learning and classical learning methods on histopathological images". In: *Neural Computing and Applications* 36.32 (2024), pp. 20537–20560. DOI: https://doi.org/10.1007/s00521-024-10229-8.

[50] Sajida Perveen et al. "A framework for early detection of acute lymphoblastic leukemia and its subtypes from peripheral blood smear images using deep ensemble learning technique". In: *IEEE Access* 12 (2024), pp. 29252–29268. DOI: https://doi.org/10.1109/ACCESS.2024.3368031.

[51] Prakeerth Prasad and Jani Anbarasi L. "Acute Lymphoblastic Leukemia Subtypes Detection using Vision Transformer Model". In: *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*. 2024, pp. 1413–1418. DOI: https://doi.org/10.1109/ICDICI62993.2024.10810888.

[52] Ruaa Sadoon and Adala Chaid. "Classification of pulmonary diseases using a deep learning stacking ensemble model". In: *Informatica* 48.14 (2024). DOI: https://doi.org/10.31449/inf.v48i14.6145.

[53] Umair Saeed et al. "DeepLeukNet—A CNN based microscopy adaptation model for acute lymphoblastic leukemia classification". In: *Multimedia Tools and Applications* 83.7 (2024), pp. 21019–21043. DOI: https://doi.org/10.1007/s11042-023-16191-2.

[54] Md Shahin Ali et al. "A Hybrid VGG16-ViT Approach With Image Processing Techniques for Improved White Blood Cell Classification and Disease Diagnosis: A Retrospective Study". In: *Health Science Reports* 8.6 (2025), e70859. DOI: https://doi.org/10.1002/hsr2.70859.

[55] Sazzli Kasim et al. "Multiclass leukemia cell classification using hybrid deep learning and machine learning with CNN-based feature extraction". In: *Scientific Reports* 15.1 (2025), p. 23782. DOI: https://doi.org/10.1038/s41598-025-05585-x.

[56] Aniel Mahendren et al. "White Blood Cells Classification: A Feature-Based Transfer Learning Approach". In: *Selected Proceedings from the 2nd International Conference on Intelligent Manufacturing and Robotics, ICIMR 2024, 22-23 August, Suzhou, China*. Ed. by Wei Chen et al. Singapore: Springer Nature Singapore, 2025, pp. 757–763. ISBN: 978-981-96-3949-6. DOI: https://doi.org/10.1007/978-981-96-3949-6_63.

[57] Mouna Saadallah. *Red Blood Cell Morphology Dataset for Image Classification*. Zenodo, Feb. 2025. DOI: https://doi.org/10.5281/14936017. URL: https://zenodo.org/records/14936017.

[58] Vera Sorin et al. "Deep Learning Applications in Lymphoma Imaging". In: *Acta Haematologica* (2025). DOI: https://doi.org/10.1159/000547427.

[59] KP Swain, SK Swain, and SR Nayak. "Vision Transformer-Based Automated Classification of Acute Lymphoblastic Leukemia". In: *2025 International Conference on Emerging Systems and Intelligent Computing (ESIC)*. IEEE. 2025, pp. 584–588. DOI: https://doi.org/10.1109/ESIC64052.2025.10962707.

[60] Vishesh Tanwar et al. "Enhancing blood cell diagnosis using hybrid residual and dual block transformer network". In: *Bioengineering* 12.2 (2025), p. 98. DOI: https://doi.org/10.3390/bioengineering12020098.

Deep Learning and Rule-Based Hybrid Model for Enhanced English Composition Scoring Using Attention Mechanisms and Graph Convolutional Networks

Ruimin Li
Zhoukou Vocational and Technical College, Zhoukou 466000, China
E-mail: laogui9029@126.com

Keywords: English essay grading, deep learning, artificial rules, graph convolutional network, wide&deep architecture

Technical paper

Received: July 8, 2025

Through the profound exploration conducted on AI technology in the field of education, early automatic scoring systems for English compositions have problems such as high misjudgment rate and low efficiency. To improve the efficiency, accuracy, and stability of the English composition grading model, a deep learning and manual rule-based English composition grading model was designed. The research extracted sequence features by introducing attention mechanisms, enhancing contextual correlation analysis, and aggregating global features through graph convolutional networks to extract high-order semantic relationships. Finally, a visual manual scoring rule was designed, which integrated deep semantic features and manual rule features through the Wide&Deep architecture to jointly optimize the scoring results. The experiment outcomes indicated that the accuracy recall curve area of the research method was 92.3%. In practical application testing, the highest group stability index of the research method was 0.07 in June. When faced with 600 concurrent requests, the average response time of the research method reached a stable value of 3.4 seconds. The outcomes above demonstrated that the English essay scoring model, which combines deep learning with manual rules as proposed by the research, exhibited excellent accuracy, speed, and stability. It effectively addressed the issues of a high misjudgment rate and low efficiency found in traditional scoring systems, thereby enhancing the model's reliability.

Povzetek: Razvit je hibridni model za ocenjevanje angleških esejev, ki združuje globoko učenje z ročnimi pravili. Z Word2Vec, mehanizmom pozornosti in GCN zajame lokalne ter globalne semantike, Wide&Deep pa združi pravila in značilke.

1 Introduction

English writing ability is one of the core indicators of language learning, and traditional manual scoring methods face bottlenecks such as low efficiency and strong subjectivity [1, 2]. Early automatic scoring systems mainly relied on rule-based methods to detect surface errors through pre-defined grammar and spelling rules, but it was difficult to evaluate the quality of content and logic, resulting in a high rate of misjudgment [3]. With the advancement of technology, machine learning (ML) algorithms have been introduced to comprehensively consider vocabulary, syntax, and other elements through feature engineering. However, a substantial quantity of annotated data support is still needed, and the generalization ability is insufficient [4]. The existing scoring system cannot meet the automatic scoring requirements for English compositions, and there is an urgent need for a stable, efficient, and accurate scoring model. Deep learning (DL) models can improve semantic understanding through end-to-end learning, but they lack transparency and are difficult to capture grammatical details. Artificial rules have a high degree of interpretability, but cannot adapt to open content evaluation. The two methods complement each other in advantages [5]. In light of the preceding circumstances, to ensure the stability, accuracy, and efficiency of the scoring model, an innovative English composition scoring model based on DL and artificial rules has been designed. The research uses Word2Vec model to convert essay text into a matrix of word vectors, capturing semantic information of vocabulary. It introduces attention mechanism and graph convolutional network to extract local sequence features and semantic graph features, and concatenating the two features to generate deep semantic features, constructing graph adjacency matrix to dynamically capture the relationships between sentences. Then, artificial rule features are generated through feature concatenation, and the Wide&Deep architecture is used to fuse deep semantic features with artificial rule features. Finally, combining multi-dimensional manual rule evaluation, the research achieves dynamic comprehensive scoring of the entire English composition. It is anticipated that research methodologies will offer a theoretical foundation for grading essays in different languages.

2 Related works

English composition grading is an important part of the educational evaluation system, playing a crucial role in achieving teaching objectives and optimizing teaching strategies. Ramesh et al. proposed AI and ML techniques for evaluating automatic paper grading in response to issues such as time-consuming manual assessments and lack of reliability in the education system. During the research process, the limitations and research trends of the current study were analyzed. The outcomes revealed that the research method had a good effect [6]. Fokides et al. compared the accuracy and qualitative aspects of the corrections and feedback generated by ChatGPT with educators regarding the effectiveness of ChatGPT on elementary school students' essays written in English. The outcomes revealed that ChatGPT surpassed educators in regard to both the volume and the caliber of output [7]. Shahzad et al. proposed using random forests as classifiers for off topic paper detection to address the prediction problem of whether an article deviates from the topic. The outcomes revealed that the research method had high accuracy [8]. Erturk et al. pointed out the low reliability and effectiveness of essay style evaluation tools, and believed that the system's decrease in paper scores was related to boredom in the labeling. The outcomes revealed that higher levels of boredom were correlated with lower scores [9]. Sharma et al. proposed a system that combines handwriting recognition models and automatic paper grading to address the time-consuming issue of grading handwritten papers in educational environments. During the research process, the performance of downstream tasks in paper scoring was analyzed based on Transformer context embedding. The outcomes revealed that the research method had good performance [10].

Many scholars both within the country and abroad have carried out profound investigations and application of Word2Vec and artificial rules. Mohammed et al. conducted an exhaustive examination of diverse approaches within the realm of ensemble learning to address the issue of time-consuming hyperparameter tuning in DL. Various features or factors that affect the success of integration methods were explained during the research process. The outcomes revealed that the research method could provide accurate theoretical support [11]. Tropsha et al. proposed a "deep quantitative structure-activity relationship" model for virtual screening of molecular databases. The outcomes revealed that the research method had a good effect [12]. Whang et al. proposed a fairness measure and unfairness mitigation technique to address the issues of bias and unfairness in traditional data management. The outcomes revealed that the research method had good data management performance [13]. Pereira et al. proposed an ML system for multi animal pose tracking to address the challenge of using DL and computer vision techniques to study the social behavior of multiple animals in natural environments. The outcomes revealed that the research method had good efficiency and accuracy [14]. Olan et al. designed an explanatory algorithm to address the impact of AI on the decision-making process in the supply chain field. The composition of interpretable AI and decision support systems was determined during the research process. The outcomes revealed that the research method could effectively enhance decision-making ability in the context of supply chain [15].

In summary, existing research has played a good role in the technological advancement of English composition grading models, but it still has limitations such as low grading efficiency and significant subjective differences. The automatic scoring model based on DL can extract multi-level information such as linguistic features and semantic information, which can simulate the process of manual scoring to a certain extent, while manual rules can handle complex grammar rules and subtle semantic differences. Therefore, based on this, a DL and artificial rule-based English composition grading model was designed. The goal is to align with the design standards for automated English composition grading and to significantly boost both the accuracy and efficiency of the grading workflow.

3 Design of english composition scoring model

3.1 Intelligent english composition scoring model based on deep semantic text features

As an important part of the education evaluation system, English composition grading has undergone an evolution from traditional manual grading to automated grading. However, existing automated grading systems are mostly based on shallow text features, resulting in significant errors in their grading results [16, 17]. DL models can effectively improve the accuracy and reliability of English composition grading from three aspects: feature extraction, semantic understanding, and grading prediction [18]. The study converts the original English composition text into a numerical word embedding matrix, and the text to word embedding conversion formula is shown in Equation (1).

$$\mathbf{E} = \text{Embedding}(\mathbf{X}) \quad (1)$$

In Equation (1), $\mathbf{X}$ represents the input English composition text sequence, represents the word embedding matrix, and $\text{Embedding}()$ represents a DL embedding function. Next, the research will investigate the use of the Woed2Vec learning model to map each word to a high-dimensional space, capturing the semantic and positional information of the word. The Word2Vec learning model has two training models: continuous bag of words and skip word. The frameworks of the two models are presented in Figure 1.

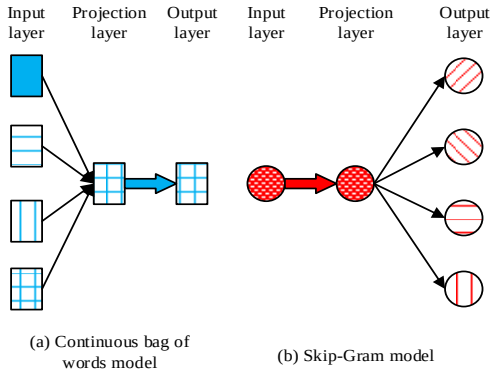


Figure 1: Framework diagram of continuous bag of words model and skip-gram model

As shown in Figure 1, the training process of both the continuous bag of words training model and the skip word training model goes through the input layer, the mapping layer, and finally outputs the results from the output layer. However, the continuous bag of words model aggregates and maps multiple features, and then outputs the results, while the skip word model maps the features and performs classification output. The study combines continuous bag of words training model and skip word training model to train and detect the sequence features and semantic map features of English compositions, and then scores the English compositions based on the detection results. In the process of extracting sequence features from English compositions, in order to break through the sequence limitations of DL models, self-attention mechanism is introduced, and its function expression is shown in Equation (2).

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2}$$

In Equation (2), $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the query matrix, key matrix, and value matrix, respectively. d_k is the dimension of the key or query vector and $\text{softmax}()$ represents the normalization function. To enhance the model's ability to express complex sequence patterns, a multi-head attention mechanism is introduced, and its calculation formula is shown in Equation (3).

$$\begin{cases} \text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)\mathbf{W}^O \\ \text{wherehead}_i = \text{Attention}(\mathbf{X}\mathbf{W}_i^Q,\mathbf{X}\mathbf{W}_i^K,\mathbf{X}\mathbf{W}_i^V) \end{cases} \tag{3}$$

In Equation (3), head represents the number of attention heads. $\mathbf{X}\mathbf{W}_i^Q$, $\mathbf{X}\mathbf{W}_i^K$, and $\mathbf{X}\mathbf{W}_i^V$ represent the projection matrices of the i th head query vector, head key vector, and head value vector, $\mathbf{W}^O$ represents the output fusion matrix, and $\text{Concat}()$ represents the concatenation operation, The number of attention heads is 8, which is determined by GPU video memory optimization test. In the process of extracting semantic graph features from English compositions, in order to dynamically capture the relationships within sentences, a semantic graph adjacency matrix is constructed, and its construction formula is shown in Equation (4).

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{E}\mathbf{E}^T}{\sqrt{D}}\right) \tag{4}$$

In Equation (4), $\mathbf{A}$ represents the adjacency matrix, $\mathbf{E}^T$ represents the transpose matrix of the word embedding matrix $\mathbf{E}$, and D is the embedding dimension. Continuing with the study of iteratively updating node features to capture higher-order relationships in semantic graphs, the graph convolution feature propagation formula is shown in Equation (5).

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{\Theta}^{(l)}\right) \tag{5}$$

In Equation (5), $\mathbf{H}^{(l)}$ represents the node feature matrix of the l th layer, $\mathbf{D}^{-\frac{1}{2}}$ is used for normalization, $\mathbf{\Theta}^{(l)}$ represents the learnable weight matrix, σ is the activation function, and introduces nonlinearity to enhance the model's expressive power. Finally, the study integrates all node features and aggregates them into a graph level feature vector to represent the global semantics of the entire English composition. The graph level feature aggregation formula is shown in Equation (6).

$$\begin{cases} \mathbf{z} = \sum_{i=1}^{N} \alpha_i \mathbf{h}_i^{(L)} \\ \alpha_i = \dfrac{\exp(\mathbf{w}^{\bullet}\mathbf{h}_i^{(L)})}{\sum_j \exp(\mathbf{w}^{\bullet}\mathbf{h}_j^{(L)})} \end{cases} \tag{6}$$

In Equation (6), $\mathbf{h}_i^{(L)}$ is the feature vector of the i th node, L is the total number of layers, $\mathbf{z}$ is the graph level feature vector representing the semantic summary of the entire text, α_i is the attention weight representing the importance of node i to global features, $\mathbf{w}$ is the learnable weight vector used to calculate attention scores, and N is the number of nodes or words. In summary, the detection model structure that integrates the sequence features of English compositions with the semantic map features of English compositions is shown in Figure 2.

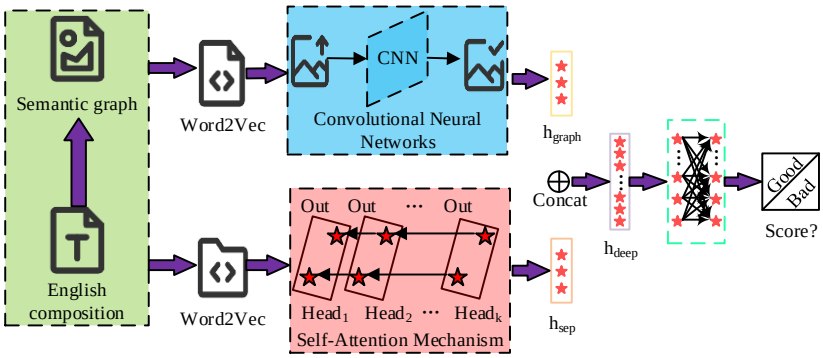


Figure 2: Detection model integrating sequence and graph features of english compositions

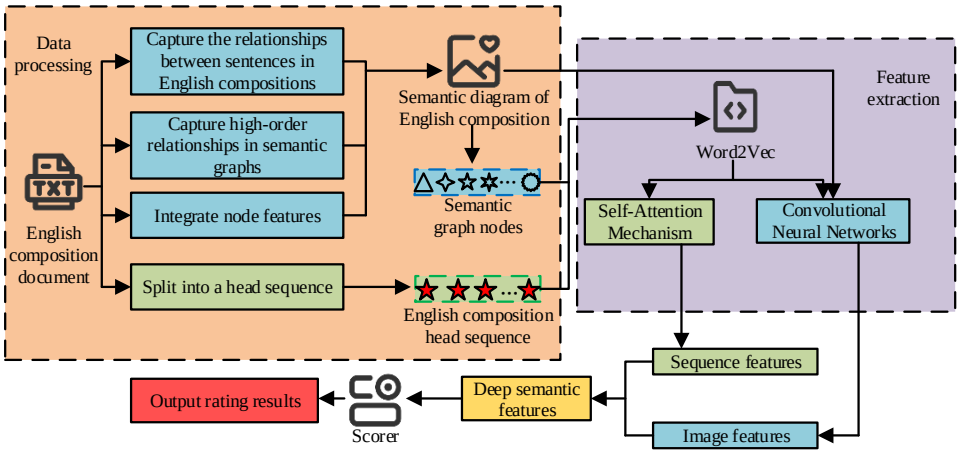


Figure 3: Intelligent scoring model for english compositions based on deep semantic text features

As shown in Figure 2, the detection model that integrates English composition sequence features and semantic graph features receives two types of input data, and the semantic graph captures the semantic relationships between phrases and concepts. Then the semantic graph and English composition are processed by Word2Vec, converting discrete words into dense, low dimensional real valued vectors. Next, semantic graph features are extracted through graph convolutional networks, while sequence features are extracted by introducing attention mechanisms. Subsequently, feature fusion is performed, and the sequence feature vectors and graph feature vectors extracted from two parallel paths are concatenated to form deep semantic features. Finally, the model evaluates the output rating results. The fusion detection model solves the limitation of single feature representation ability by fusing two complementary feature representations. The deep semantic text feature expression of its fusion model is shown in Equation (7).

$$h_{\text{deep}} = \sigma\left(h_{seq} \| h_{graph}\right) \tag{7}$$

In Equation (7), h_{deep} represents the deep semantic features of English composition, $\|$ represented by vector concatenation symbols, h_{seq} and h_{graph} respectively represent the sequence features and graph features of English composition. In summary, the intelligent scoring model for English composition based on deep semantic text features is shown in Figure 3.

As shown in Figure 3, the English composition scoring model based on deep semantic text features achieves accurate evaluation by integrating multi-level semantic information. The model first performs structured parsing on the input English essay document, breaks down the title sequence to highlight the article structure, and preserves contextual information through node feature integration. In the feature extraction stage, multimodal technology is used to deeply fuse semantic information. On the one hand, the title sequence is embedded in Word2Vec and local sequence features are extracted through self-attention mechanism. On the other hand, semantic graph nodes model global semantic relationships through graph convolutional networks. The two types of features are further combined with image features to form a unified deep semantic feature vector. Finally, the rater performs regression analysis based on deep semantic features and outputs objective scoring results.

In summary, the implementation details of the entire research framework are as follows: (1) Word2Vec is used to convert English essay texts into dense word vector matrices. The continuous bag-of-words model predicts core words through contextual word prediction. The input layer aggregates multiple contextual word vectors, while the mapping layer summarizes them to output core word probabilities. The skip-word model predicts contextual words based on core words. Both models undergo

negative sampling optimization, with 300-dimensional embeddings and a 5-window contextual size. (2) During attention mechanism feature extraction, the input word vector matrix is linearly transformed to generate query matrices, key matrices, and value matrices, each with 64 dimensions. The multi-head architecture employs 8 heads, where each head independently computes attention and outputs concatenated linear fusion results. The final sequence features are generated. (3) In graph convolutional network semantic feature extraction, the adjacency matrix embedding dimension is 300. The feature propagation and aggregation process learn a weight matrix with 128 dimensions across 2 layers.

3.2 Intelligent english composition scoring model combined with artificial rules

Although the English composition grading model based on deep semantic text features can effectively grade English compositions, it generally relies on manually defined grading templates, and candidates can avoid deduction types through simple writing techniques, lacking interpretability [19]. In the field of composition checking, artificial rules are usually expressed in formal language and automatically detected through natural language processing tools. The English scoring model combined with artificial rules can effectively solve the problem of lack of interpretability in DL models, so further research is needed to introduce artificial rules [20]. Based on manual rules to quantify the basic language quality of sentences, the basic formula for scoring errors in English compositions is shown in Equation (8).

$$E_s = \frac{1}{1+\lambda\cdot(C_{spell}+C_{gram})}\times F \tag{8}$$

In Equation (8), E_s is the score for incorrect sentences, with a maximum score of F, C_{spell} is the number of spelling errors, C_{gram} is the number of grammar errors, and λ is the error penalty coefficient, its value is 0.1, and the error rate is lowest when its value is 1 through grid search verification. Continuing with the study of balancing the importance of each dimension through artificial rules, the formula for weighting the multidimensional excellence of sentences is shown in Equation (9).

$$Q_s = \beta_1\cdot V_s + \beta_2\cdot G_s + \beta_3\cdot T_s + \beta_4\cdot P_s \tag{9}$$

In Equation (9), Q_s represents the overall excellence score of the sentence, V_s represents the vocabulary score, G_s represents the syntactic complexity score, T_s represents the part of speech diversity score, P_s represents the rectangle score, and β_i represents the artificial rule weight. Then, to evaluate the logical rigor of English essay paragraphs, a scoring formula for paragraph cohesion strength is introduced, and its specific expression is shown in Equation (10).

$$C_p = \frac{\sum_{k=1}^{n}\gamma_k\cdot I(\text{conn}_k)}{\text{N}}\times\frac{R_{\text{cohere}}}{1+\log(\text{L})} \tag{10}$$

In Equation (10), C_p represents the coherence score of the paragraph, $I(\text{conn}_k)$ represents the validity indicator function of the k th connector, γ_k represents the weight of the connector, R_{cohere} represents the semantic coherence ratio, N represents the number of sentences, and L represents the length of the paragraph. Finally, the study aims to achieve dynamic comprehensive scoring of the entire English composition through multi-dimensional manual rule evaluation. The scoring formula is shown in Equation (11)

$$Score = \lambda\cdot\left(\frac{\sum_{j=1}^{m}Q_{s_j}}{m}\right)+\mu\cdot\left(\frac{\sum_{p=1}^{l}C_{p_p}}{l}\right)+\nu\cdot\text{Sim}_{\text{content}}\cdot\text{Sim}_{\text{str}} \tag{11}$$

In Equation (11), $Score$ represents the final score of the composition, Q_{s_j} represents the excellence score of the j th sentence, C_{p_p} represents the coherence score of the p th paragraph, $\text{Sim}_{\text{content}}$ and Sim_{str} represents the similarity of content and structure, λ,μ,ν all meet the requirement of $\lambda+\mu+\nu=1$. The artificial rules are constructed based on expert knowledge, employing a method that quantifies sentence-level errors and sentence excellence through predefined weights to achieve digital transformation. The primary linguistic features targeted include surface errors, sentence-level errors, and paragraph-level errors. By integrating deep semantic features through a Wide&Deep architecture, the rules enhance interpretability while capturing subtle errors and reducing subjective variations. Experimental validation demonstrates their effectiveness in lowering bias values and misjudgment rates, as well as improving scoring stability. In summary, the feature extraction framework for the manual scoring rules of English compositions is shown in Figure 4.

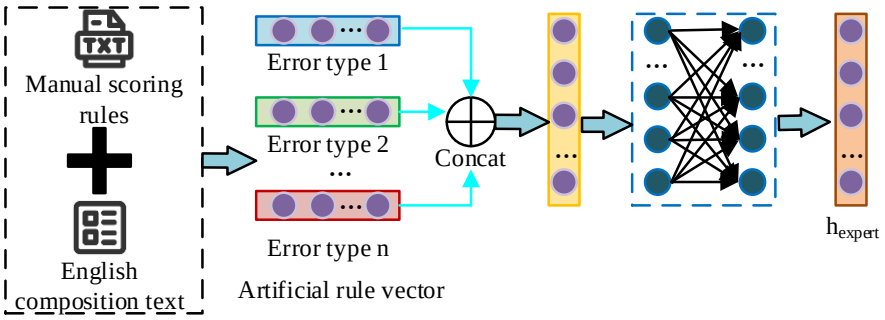


Figure 4: Feature extraction of artificial rules for english composition

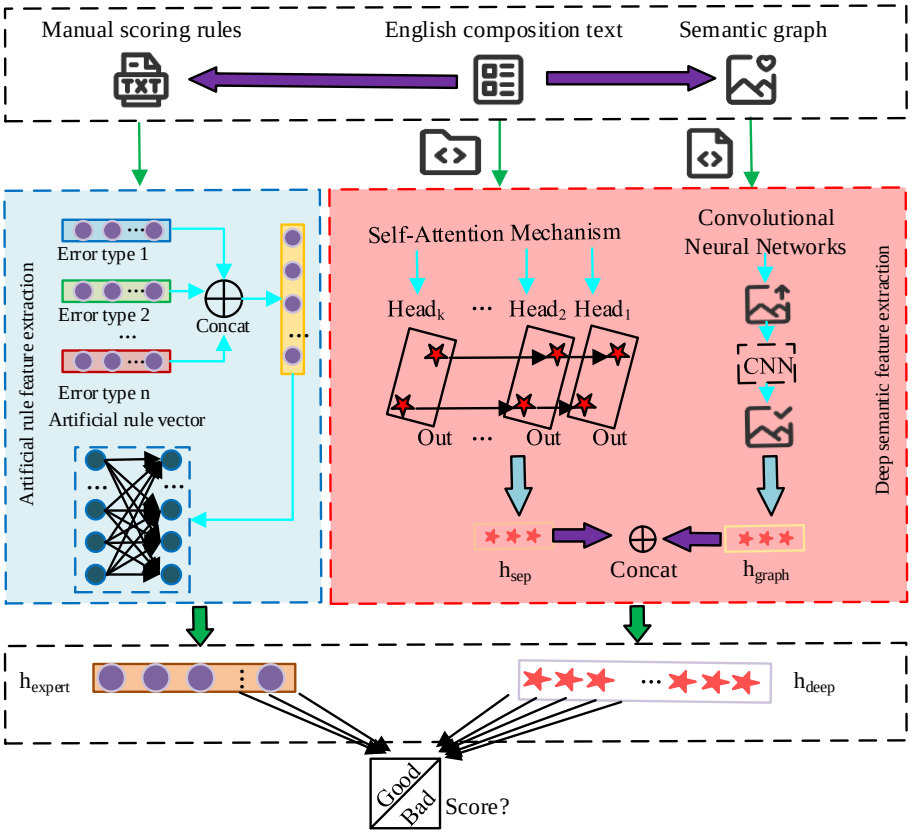


Figure 5: Network structure of english composition error detection combined with artificial rules

As shown in Figure 4, in the feature extraction framework of manual scoring rules for English compositions, structured manual scoring rules are input together with the original English composition text as initial data. Then, the manual rules are decomposed into different types of errors, and each type of rule is quantified as a numerical vector to achieve the digital transformation of expert knowledge. Then, the artificial rule vector is concatenated with the semantic vector of the composition text to form a mixed feature that combines both artificial rules and text semantics. Finally, after processing, output the characteristics of the manual scoring rules for English compositions. The study aims to achieve the organic integration of artificial rules and DL models by converting discrete artificial rules into continuous features. The specific expression is shown in Equation (12).

$$h_{expert} = \sigma(\mathbf{W}(\|_{v \in rud} \mathbf{x}_i) + b) \tag{12}$$

In Equation (12), h_{expert} represents the artificial rule feature, $v \in rud$ represents the set of error types, $\mathbf{x}_i$ represents the artificial rule vector for the error type, and b represents the bias term. Next, the study uses the Wide&Deep structure to fuse shallow features of artificial rules with deep semantic text features, achieving the final error classification prediction. The fusion formula is shown in Equation (13).

$$y = Softmax(\mathbf{W}_{wide} h_{expert} + \mathbf{W}_{deep} h_{deep} + b) \tag{13}$$

In Equation (13), y represents the rating result, and $\mathbf{W}_{wide}$ and $\mathbf{W}_{deep}$ represent the weight matrices of

parts *wide* and *deep*. In summary, the network structure of English composition error detection combined with manual rules is shown in Figure 5.

As shown in Figure 5, the English composition error detection network structure combined with manual rules improves detection accuracy and interpretability through dual channel feature fusion. The model receives dual source inputs: the manual scoring rules are decomposed and vectorized into quantifiable rule vectors, covering error types such as grammar, logic, rhetoric, etc. The model synchronizes the construction of semantic maps for original English compositions, extracts logical relationships between sentences, and performs sequence analysis to capture word order features. Then, the deep semantic features obtained from the dual source input are evaluated together with the artificial rule features, and the results are judged. The study introduces binary cross entropy loss to measure the difference between misclassified predictions and true labels. The specific expression of the loss function is shown in Equation (14).

$$Loss = -\hat{y}\log(y) - (1-\hat{y})\log(1-y) \tag{14}$$

In Equation (14), $Loss$ represents the loss function value, $\hat{y}$ represents the true label of the sample, and y represents the predicted probability output of the model. Finally, the study evaluates the performance of the model by calculating its accuracy, as shown in Equation (15).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{15}$$

In Equation (15), $Accuracy$ represents the accuracy of the model, TP and TN are the number of essays correctly rated as low or high by the model, FP and FN are the number of essays incorrectly rated as low or high by the model. In summary, the scoring process of the English composition scoring model based on DL and artificial rules is shown in Figure 6.

As shown in Figure 6, the English essay scoring model based on DL and artificial rules improves scoring accuracy and interpretability through dual channel feature collaboration. The model takes English composition text and manual scoring rules as dual source inputs: on the one hand, it generates a semantic map through multi-level parsing of the original text, and on the other hand, it breaks down the document into sequential features according to its structure, preserving the framework information of the article. Next, in the feature extraction stage, a bimodal DL architecture is adopted. After Word2Vec vectorization of semantic graph nodes, a graph convolutional network models global semantic relationships and outputs deep features. Sequence nodes extract local language patterns through self-attention mechanisms, generate sequence features, and concatenate the two to form deep semantic features. On the other hand, breaking down manual rules in textual form into quantifiable dimensional vectors enables the digital transformation of expert knowledge. Finally, the artificial rule features are optimized using binary cross entropy and combined with deep semantic features to generate rule enhanced deep features. The rater then performs regression analysis based on the rule enhanced deep features to output the final English essay grading results.

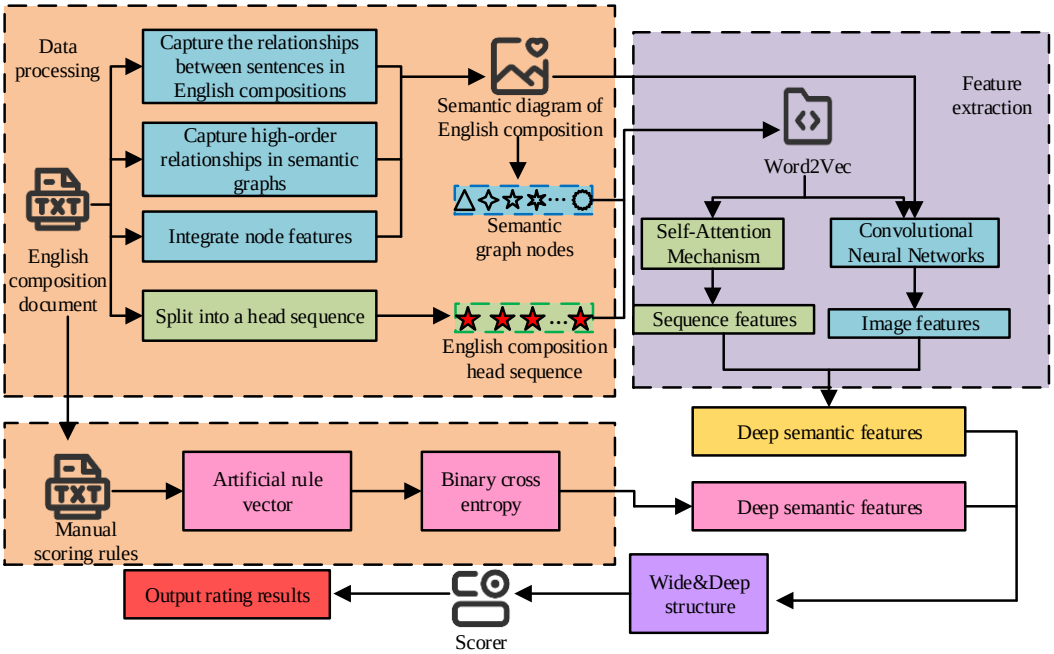


Figure 6: Scoring process of english composition scoring model based on dl and artificial rules

4 Validation of English composition grading model based on DL and artificial rules

4.1 Performance testing of English composition scoring model based on DL and artificial rules

To confirm the capability of the English essay grading model based on DL and artificial rules, a simulation model was constructed for testing. The testing environment and specific configuration are presented in Table 1.

Table 1: Test environment and specific configuration

Testing environment	Specific configuration
GPU	NVIDIA Tesla V100/A100
CPU	Intel Xeon Gold 6248R
Memory	256GB DDR4
Storage	2TB NVMe SSD + 10TB HDD
DL framework	PyTorch 1.12 / TensorFlow 2.10
Feature engineering tools	Scikit-learn 1.2 + Gensim 4.3
Support for large models	Transformers 4.28

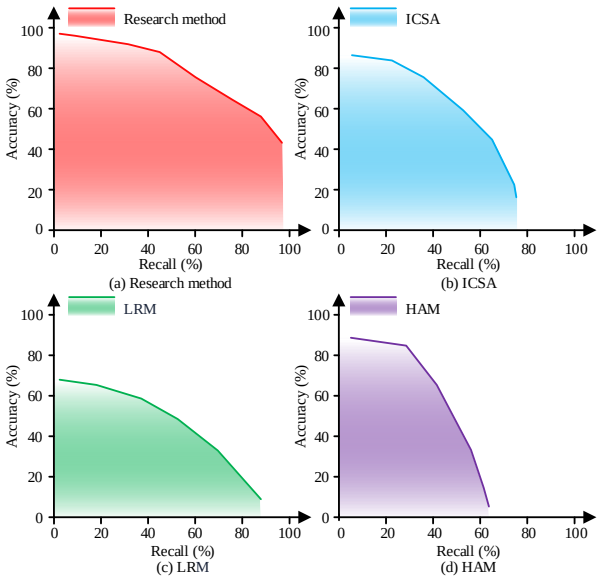


Figure 7: Accuracy recall curve of different methods

As shown in Table 1, the specific configurations in the table were used for performance testing, using the Kaggle ASAP dataset. The research methods were compared with the Integrated Classification Scoring Algorithm (ICSA), Linear Regression Model (LRM), and Hierarchical Attention Model (HAM). The accuracy recall curves and curve areas of the four methods were compared, and the results are presented in Figure 7.

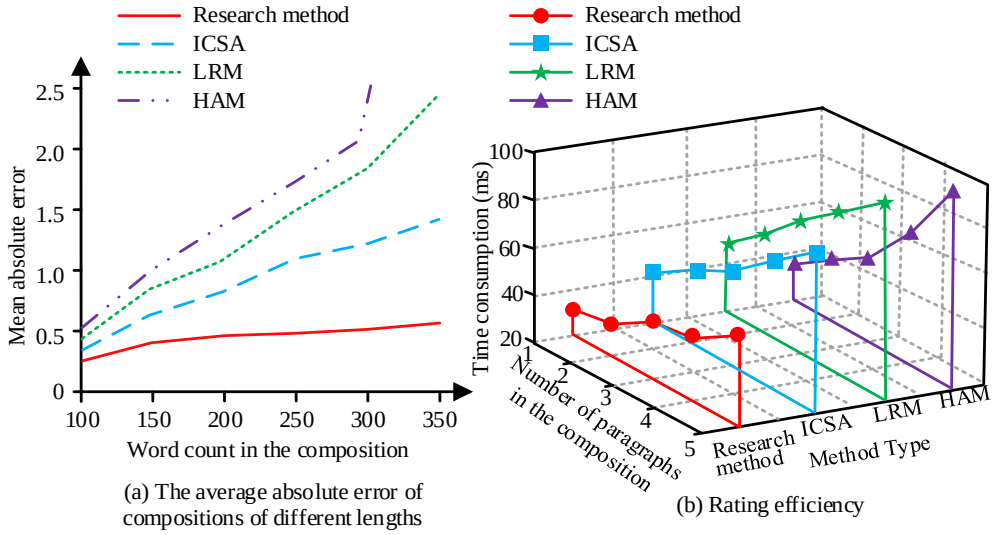
As shown in Figure 7, the shape and area of the accuracy recall curve of different methods are different. In Figure 7 (a), the accuracy recall curve of the research method was close to a rectangle, with a curve area of 92.3%. In Figure 7 (b), the accuracy recall curve of the ICSA algorithm was 71.6%. In Figure 7 (c), the curve of the LRM model belonged to low accuracy high recall, which was prone to false positives. As shown in Figure 7 (d), the curve of the HAM model belonged to high accuracy low recall, which was prone to missed detections. Overall, compared to comparative methods, research methods had higher accuracy and inspection coverage. The mean absolute error (MAE) of the scoring results of the four methods under different numbers of writing words, as well as the scoring time under different numbers of writing paragraphs, were compared, and the outcomes are presented in Figure 8.

In Figure 8 (a), the MAEs of the scoring results of the four methods all increased with the increase in the number of English composition words. The MAE of the research method's scoring results had the smallest increase. When the word count in the composition was 100, the MAE of the research method's scoring results was 0.25. When the word count in the composition was 350, the MAE of the research method's scoring results was 0.52. The MAE for the two types of composition word counts only increased by 0.27. The MAE of the scoring results for the other three methods was significantly greater than that of the research method at different numbers of words in the composition. In Figure 8 (b), the scoring time of all four methods increased with the number of paragraphs in the essay. When the English essay had only one paragraph, the scoring time of the research method was 32 ms, and when the essay had five paragraphs, the scoring time was 42 ms. However, the scoring time of the other three methods at different paragraph counts was significantly greater than that of the research method. Overall, compared to the comparative methods, the research methods had better robustness. In conclusion, the English essay grading model proposed by the research based on DL and artificial rules had high reliability, accuracy, and good robustness. After validating the performance of the research methodology, the study further investigated the synergistic effects of the fusion architecture through ablation experiments. First, independent testing of deep models revealed that removing manual rules reduced grammatical error detection accuracy, demonstrating their constraint effect on surface errors. Next, independent testing of rule models showed increased semantic coherence score deviations in long texts when graph convolutional networks were removed, proving deep models' capability to capture higher-order semantics. Finally, dual-stream feature contribution analysis using SHAP values demonstrated that manual rule features contributed minimally to grammatical/spelling error detection, while

deep semantic features played a significant role in content logic scoring. These ablation results confirmed the complementary innovation of the "feature perception-regulation constraint" architecture in the research methodology.

4.2 Practical application effect of English composition scoring model based on DL and artificial rules

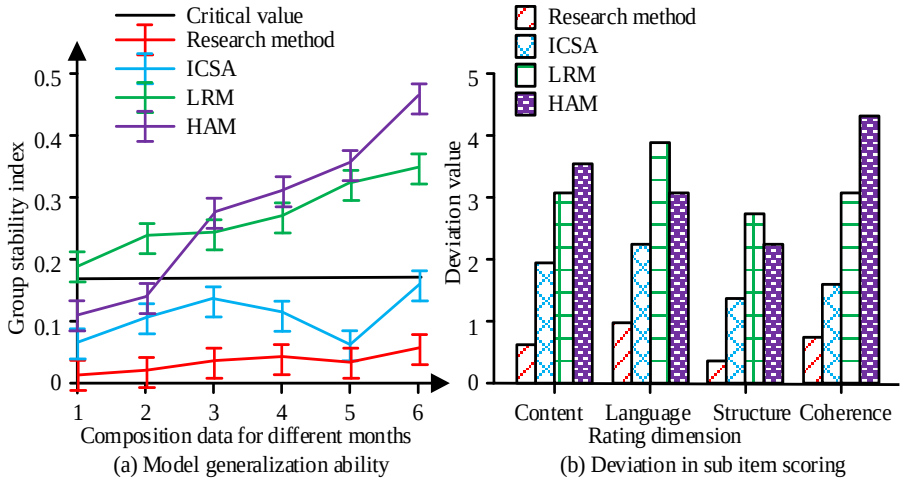
On the basis of verifying the performance of the English essay grading model based on DL and artificial rules, further research is conducted to ascertain the efficacy of the practical application of the research method. The study used the IELTS Writing Task 2 dataset to build a modular hierarchical architecture experimental platform. The research methods were compared with ICSA, LRM and HAM, and the semantic depth index was supplemented: The content dimension deviation of BERT in IELTS data set could be reduced to 0.42, which was better than the 0.67 of the research models, highlighting the advantages of lightweight. The study further compared Transformer-based pre-trained models. In the IELTS content dimension scoring, BERT exhibited lower deviation values than the research methodology. However, its reliance on billions of parameters resulted in significantly longer response times. Notably, the research methodology demonstrated markedly higher accuracy in detecting grammatical errors when incorporating rules, surpassing BERT. These findings indicated that compared to cutting-edge technologies, the research methodology demonstrated superior semantic understanding depth and error detection specificity. Then, the group stability index of the four methods for the English composition data in the first six months was scored, and the deviation values under different scoring dimensions were compared. The results are shown in Figure 9.



(a) The average absolute error of compositions of different lengths

(b) Rating efficiency

Figure 8: MAE and rating efficiency



(a) Model generalization ability

(b) Deviation in sub item scoring

Figure 9: Model generalization ability and sub-item scoring deviation

In Figure 9 (a), the critical value of the group stability index for English composition scoring was 0.17. The overall group stability index of the research method for scoring monthly composition data remained below

the critical value, with its highest group stability index being 0.07 in June and the lowest 0.02 in January. The stability indices of the other three methods for scoring monthly essay data were significantly higher than that of the research method. In Figure 9 (b), the deviation value of the research method in the dimension of composition content was 0.67, the deviation value in the dimension of composition language was 0.99, the deviation value in the dimension of composition structure was 0.33, and the deviation value in the dimension of composition coherence was 0.82. The bias values of the other three methods under different scoring dimensions were significantly greater than those of the research methods. Overall, compared to the comparative methods, the research methods had better generalization ability and higher accuracy. Comparing the sensitivity of four methods in identifying excellent compositions and their ability to capture advanced vocabulary in compositions, the results are presented in Figure 10.

As shown in Figure 10 (a), the misjudgment rates of the four methods for high - scoring essays with different score thresholds were not the same. The overall misjudgment rate of the research method for high-scoring essays was less than 20%. The highest misjudgment rate was 17.8% when the score threshold was 21 points, and the minimum misjudgment rate of the research method was 3.2% when the score threshold was 24-25 points. The misjudgment rates of the other three methods were significantly higher than that of the research method at different score thresholds. As shown in Figure 10 (b), the consistency between the vocabulary scoring results of the four methods and the manual vocabulary scoring was not the same. The distribution of the vocabulary scoring results of the research method was closely aligned with the diagonal, indicating it was highly consistent with the manual vocabulary scoring. However, the distribution of vocabulary scoring results for the other three methods differed significantly from that of manual vocabulary scoring, resulting in lower accuracy of their scoring results. Overall, compared to the comparative methods, the research method had a lower false positive rate and better scoring performance. The four methods were compared for the accuracy rate of scoring under different error types and the average response time under different concurrent request numbers, as shown in Figure 11.

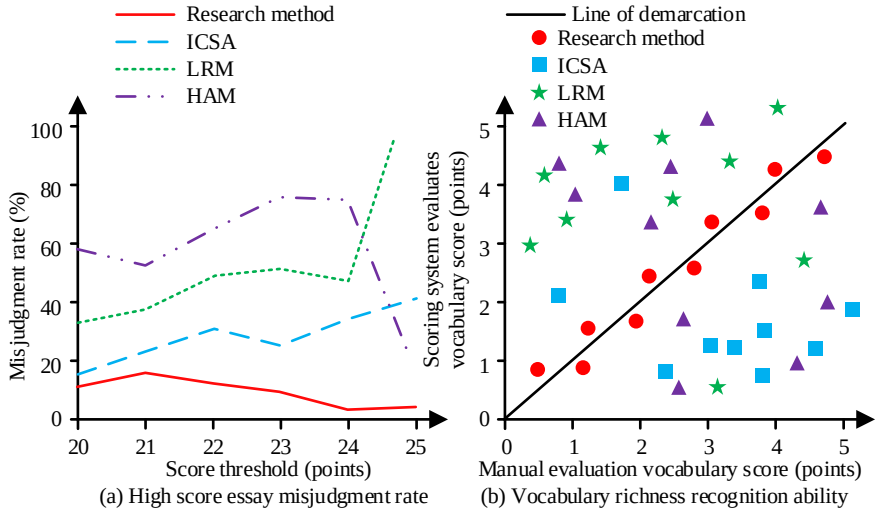


Figure 10: High score essay misjudgment rate and vocabulary richness recognition ability

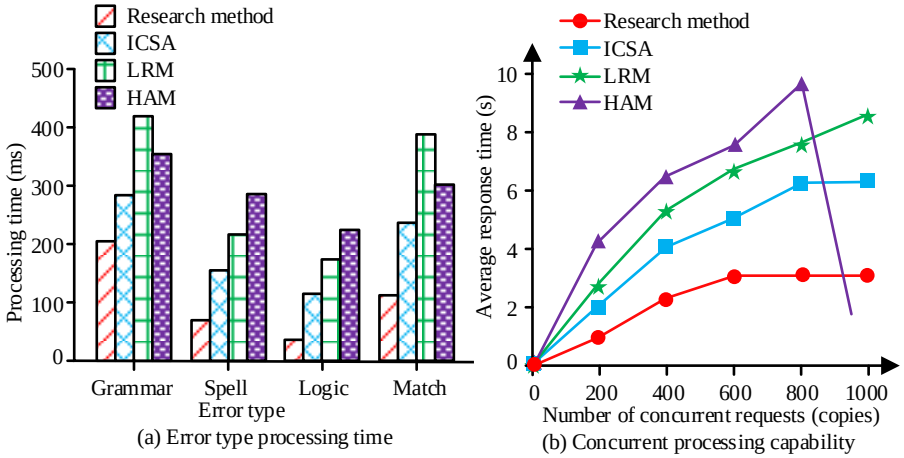


Figure 11: Error type processing time and concurrent processing capability

As shown in Figure 11(a), the research method demonstrated 99.2% accuracy in scoring grammatical errors, 98.7% in spelling errors, 99.0% in logical errors, and 97.9% in collocation errors. In contrast, the other three methods showed significantly lower accuracy rates for these error types compared to the research methodology. In Figure 11 (b), the average response time of the four methods gradually increased with the increase of concurrent requests, with the research method showing the smoothest trend of increase. When faced with 600 concurrent requests, the average response time of the research method reached a stable value of 3.4 seconds. However, the average response time of the other three methods showed a significantly greater increase trend than the research method. Overall, compared to comparative methods, research methods had better resource allocation capabilities and scoring performance. Overall, the English essay grading model proposed by the research based on DL and artificial rules had good generalization ability, accuracy, and performance.

5 Conclusion

To address the issues of high misjudgment rates and instability in existing English essay automatic scoring systems, this study innovatively proposes an English essay scoring model combining DL with manual rules. The research methodology extracts sequence features and semantic graph features from English essays, integrating them with manual rule features to construct a "feature perception-rule constraint-joint decision" fusion architecture for stable and accurate scoring. Experimental results show that when the essay contains 100 words, the average absolute error of the scoring method is 0.25; when the essay contains 350 words, the average absolute error increases to 0.52; and when the essay consists of 5 paragraphs, the scoring time reaches 42ms. In practical application tests, the method shows 0.67 deviation in content dimension scoring, 0.99 deviation in language dimension scoring, 0.33 deviation in structure dimension scoring, and 0.82 deviation in coherence dimension scoring. The method achieved 99.2% accuracy rate for grammatical errors, 98.7% accuracy rate for spelling errors, 99.0% accuracy rate for logical errors, and 97.9% accuracy rate for collocation errors. Overall, the proposed method demonstrated excellent scoring accuracy, robustness, and stability. The research findings failed to quantify the contribution ratios of DL and rule-based approaches to explainability. The test datasets were limited to IELTS/Kaggle materials, which did not validate the generalization capabilities of open-domain essays and consequently compromised practical applicability. Moreover, the methodology primarily relied on Word2Vec and traditional attention mechanisms for feature extraction. While effective in English essay scoring, the static embedding model of Word2Vec lacked contextual sensitivity, potentially limiting semantic depth comprehension and cross-linguistic transfer capabilities. Modern Transformer models, however, provide superior contextual representation and enhanced cross-linguistic application potential. Future studies could integrate Transformer pre-trained models to verify model stability and deviations across multilingual essay datasets (e.g., French, Chinese), evaluate cross-linguistic rule adaptability, and improve cross-linguistic performance and transferability. Additionally, the research could incorporate eye-tracking technology into multi-modal deep understanding frameworks. By recording eye movements during writing processes, it can analyze authors' attention allocation patterns. Combined with keystroke logs, this approach could quantify writing fluency and cognitive load, supplementing process dynamics that textual features cannot capture. However, this study is the first to migrate the Wide&Deep architecture from recommendation system to essay scoring field. Through semantic drift of DL with rule feature constraints, it provides a new idea for the interpretability of AI education products.

References

[1] Del Gobbo E, Guarino A, Cafarelli B, Grilli L. GradeAid: A framework for automatic short answers grading in educational contexts—design, implementation and evaluation. Knowledge and Information Systems, 2023, 65(10): 4295-4334. DOI: 10.1007/s10115-023-01892-9.

[2] Wang Q. The use of semantic similarity tools in automated content scoring of fact-based essays written by EFL learners. Education and Information Technologies, 2022, 27(9): 13021-13049. DOI: 10.1007/s10639-022-11179-1.

[3] Geçkin V, Kızıltaş E, Çınar Ç. Assessing second-language academic writing: AI vs. Human raters. Journal of Educational Technology and Online Learning, 2023, 6(4): 1096-1108. DOI: 10.31681/jetol.1336599.

[4] Theodosiou A A, Read R C. Artificial intelligence, machine learning and deep learning: Potential resources for the infection clinician. Journal of Infection, 2023, 87(4): 287-294. DOI: 10.1016/j. Jinf.2023.07.006.

[5] Wang J, Wang S, Zhang Y. Deep learning on medical image analysis. CAAI Transactions on Intelligence Technology, 2025, 10(1): 1-35. DOI:10.1049/cit2.12356.

[6] Ramesh D, Sanampudi S K. An automated essay scoring systems: A systematic literature review. Artificial Intelligence Review, 2022, 55(3): 2495-2527. DOI: 10.1007/s10462-021-10068-2.

[7] Fokides E, Peristeraki E. Comparing ChatGPTs correction and feedback comments with that of educators in the context of primary students short essays written in English and Greek. Education and Information Technologies, 2025, 30(2): 2577-2621. DOI: 10.1007/s10639-024-12912-8.

[8] Shahzad A, Wali A. Computerization of off-topic essay detection: a possibility? Education and Information Technologies, 2022, 27(4): 5737-5747. DOI: 10.1007/s10639-021-10863-y.

[9] Erturk S, van Tilburg W A P, Igou E R. Off the mark: Repetitive marking undermines essay evaluations due to boredom. Motivation and Emotion, 2022, 46(2): 264-275. DOI: 10.1007/s11031-022-09929-2.

[10] Sharma A, Katlaa R, Kaur G, Jayagopi D B. Full-page handwriting recognition and automated essay scoring for in-the-wild essays. Multimedia Tools and Applications, 2023, 82(23): 35253-35276. DOI: 10.1007/s11042-023-14558-z.

[11] Mohammed A, Kora R. A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University-Computer and Information Sciences, 2023, 35(2): 757-774. DOI: 10.1016/j. Jksuci.2023.01.014.

[12] Tropsha A, Isayev O, Varnek A, Schneider G, Cherkasov A. Integrating QSAR modelling and deep learning in drug discovery: The emergence of deep QSAR. Nature Reviews Drug Discovery, 2024, 23(2): 141-155. DOI: 10.1038/s41573-023-00832-0

[13] Whang S E, Roh Y, Song H, Lee J G. Data collection and quality challenges in deep learning: A data-centric ai perspective. The VLDB Journal, 2023, 32(4): 791-813. DOI: 10.1007/s00778-022-00775-9.

[14] Pereira T D, Tabris N, Matsliah A, Turner D M, Li J, Ravindranath S, et al. SLEAP: A deep learning system for multi-animal pose tracking. Nature methods, 2022, 19(4): 486-495. DOI: 10.1038/s41592-022-01426-1.

[15] Olan F, Spanaki K, Ahmed W, Zhao G. Enabling explainable artificial intelligence capabilities in supply chain decision support making. Production Planning & Control, 2025, 36(6): 808-819. DOI:10.1080/09537287.2024.2313514.

[16] Bhat M, Rabindranath M, Chara B S, Simonetto D A. Artificial intelligence, machine learning, and deep learning in liver transplantation. Journal of hepatology, 2023, 78(6): 1216-1233. DOI: 10.1016/j. Jhep.2023.01.006.

[17] Simon K, Vicent M, Addah K, Bamutura D, Atwiine B, Nanjebe D, Mukama A O. Comparison of deep learning techniques in detection of sickle cell disease. AIA, 2023, 1(4):252-259. DOI: https://doi.Org/10.47852/bonviewAIA3202853.

[18] K. Bhosle, V. Musande. Evaluation of deep learning CNN Model for recognition of Devanagari digit. Applied Artificial Intelligence. 2023, 1(2): 114-118. DOI: 10.47852/bonviewAIA3202441.

[19] Zamfiroiu A, Vasile D, Savu D. ChatGPT–a systematic review of published research papers. Informatica Economica, 2023, 27(1): 5-16. DOI: 10.24818/issn14531305/27.1.2023.01.

[20] Didimo W, Grilli L, Liotta G, Montecchiani F. Efficient and trustworthy decision making through human-in-the-loop visual analytics: A case study on tax risk assessment. Rivista italiana di informatica e diritto, 2022, 4(2): 15-21. DOI: 10.32091/RIID0092.

Integrating DDPG and QPSO for Multi-Objective Optimization in High Proportion Renewable Energy Power Dispatch Systems

Xu'an Qiao[1*], Chaofan Liu[2]
[1]School of Aeronautics, Chongqing City Vocational College, Chongqing 402160, China
[2]School of Physics Sciences, University of Science and Technology of China, Hefei 230026, China
E-mail: qiaoxuan109_2023@126.com, Liuchaofan1980_622@126.com
*Corresponding author

Keywords: Power system, dispatch optimization, renewable energy, DDPG, heuristic algorithm

Received: June 16, 2025

This study proposes a novel dispatch optimization model that integrates deep deterministic policy gradient (DDPG) and quantum particle swarm optimization (QPSO) to address the challenges posed by high proportions of renewable energy in power systems. The proposed multi-objective optimization framework considers system cost reduction, supply-demand balance, and dynamic adaptability to renewable energy fluctuations. The experimental results on the IEEE 30-bus and 118-bus systems demonstrated significant improvements. This method reduced total system costs by 13.6% and 11.4%, respectively. It also increased supply reliability to 97.1% and achieved an energy utilization rate of 94.85%. Additionally, it minimized frequency deviation to 1.25 Hz. The optimization time was also improved, achieving a reduction of 58.3 seconds in efficiency. The research results have important practical application value in improving power system economy, enhancing system reliability, and dynamic adaptability. It can provide efficient and reliable technical support for power dispatch planning, load management, and real-time control under high percentage renewable energy scenarios.

Povzetek:Študija predlaga hibridni model za optimizacijo razporejanja (dispečanja) v omrežjih z visokim deležem obnovljivih virov, ki združuje izboljšani DDPG in QPSO. Model zmanjša stroške, izboljša zanesljivost in stabilnost ter poveča izrabo energije. Preizkusi na IEEE sistemih potrjujejo visoko učinkovitost.

1 Introduction

With the continuous adjustment and optimization of the global energy structure, carbon peaking and carbon neutrality targets have become important strategies for countries around the world to cope with climate change and achieve sustainable development. Because of their clean and low-carbon benefits, renewable energy (RE) sources like solar and wind have been frequently used in this setting [1]. However, the operation and scheduling of the conventional power system (PS) have been severely hampered by the widespread use of wind, solar, and other high percentage (HP) RE sources. To begin with, RE is highly volatile and erratic. Their output is affected by natural conditions, including wind speed, light, etc., and there is a large uncertainty [2]. This uncertainty makes the load balance and stability of the system subject to shocks, and is prone to supply-demand imbalance in the PS in the case of peak power demand or insufficient wind and solar resources [3]. Second, the traditional PS, which relies on precise forecasts of load and generation capacity from the scheduling model, becomes irrelevant when the proportion of RE in the PS increases. Due to the significant impact of unstable factors on RE generation capacity, there is a substantial discrepancy between actual and forecasted values. This discrepancy makes short-term PS scheduling more challenging and complex [4-5]. In addition, current PS scheduling relies mainly on a phased sequential scheduling approach, i.e., unit commitment (UC) is performed first to determine the start/stop status of the units. Then economic dispatch (ED) is performed to optimize the unit output. Finally, real-time regulation is performed through automatic generation control (AGC) [6]. Although this staged dispatch model is simple to operate, there are problems such as response delays between different dispatch modules. In view of this, the study proposes a multilevel cooperative dispatch model for HP of renewable energy power systems (REPS) and introduces heuristic algorithms to accelerate the solution of complex systems. The study aims to solve the limitations of the traditional sequential scheduling method and improve the economy, reliability and feasibility of system operation through refined modeling and cooperative optimization.

This study's novel contribution is its proposal of a joint heuristic algorithm that combines improved deep deterministic policy gradient (DDPG) and quantum particle swarm optimization (QPSO) algorithms to optimize scheduling in high-proportion REPS. This method improves the convergence speed and stability of complex, multi-constraint, multi-timescale problems by introducing dual experience pooling and time-decaying exploration strategies. The proposed mathematical model addresses the inefficiencies and local optima of traditional models. It provides a more efficient and reliable solution for PS scheduling optimization.

2 Related works

The key to ensuring the PS operates steadily is PS schedule optimization. The optimization method directly impacts the stability and adaptability of the PS to RE fluctuations. These are essential to the functioning and advancement of contemporary PS, as well as its economic efficiency. Therefore, many scholars have carried out various researches on PS scheduling optimization. For the optimal reactive power scheduling problem in PS, M. Abd-El Wahab et al. suggested a hybrid method called augmented Jaya and artificial ecosystem-based optimization, which improved system stability, economic viability, and overall efficiency [7]. A nonconvex mixed integer and quadratic restricted planning technique was presented by Cox J L et al. to solve the challenge of optimizing a centralized solar power plant's profitability under changing solar resources. The method improved the solvability of the problem through exact and approximation techniques, thus enabling operational scheduling optimization in real-time decision support [8]. To address the issue of system security and economic cost over time in microgrid scheduling, Zhang et al. suggested a multi-timescale scheduling model that incorporated load voltage and frequency dynamics. To minimize economic cost while maintaining voltage and frequency stability, the study converted it into a multi-objective optimization problem that took into account economic cost, voltage deviation, and frequency stability. This improved the microgrid dispatch's efficiency and dependability. [9]. To address the global issues brought on by the recent explosive increase in the demand for electricity, Hou et al. developed an integrated day-ahead multi-objective microgrid optimization framework. The framework produced more affordable, dependable, and ecologically friendly power supply services by combining demand-side management, forecasting methods, and economic-environmental dispatch [10].

Large-scale access to the PS by RE sources affects the output characteristics of wind and photovoltaic energy sources. These sources exhibit strong intermittency, randomness, and volatility due to weather, climate, and other external natural factors. These challenges have led to the urgent need for innovation and optimization of existing dispatch methods to adapt to the new situation of HP of RE access. A mixed-integer linear programming method was put up by Shirzadi et al. to address the issue of enhancing the efficiency and dependability of RE systems. The study optimized the PS's daily operating expenses and system resilience by combining a unique hybrid model with deep learning and statistical modeling to forecast the load demand and wind power output (PO) for the ensuing three days [11]. Due to the impact of the volatility of wind and solar power generation on PS operation, Guo et al. proposed multi-stage optimization, online optimization, and multi-timescale optimization for RE integration. This study realized the strategic scheduling and control of energy storage units and improved the efficiency of RE integration in the power grid [12]. By proposing a new economic low-carbon clean PS dispatch model that incorporates power-to-gas technology, Cui et al. addressed the issue of increasing the grid's capacity to absorb wind power. This model integrated the effects of multiple price factors, resulting in low-carbon PS operation and cost optimization [13]. An enhanced jellyfish search optimization technique was presented by Gami et al. to solve the optimal reactive power dispatch problem in HP renewable PSs. By improving the algorithm's exploration and development stages, the study successfully optimized the PS's most secure and stable state [14]. This allowed the PS to operate in both deterministic and probabilistic load demands and RE resource states.

In summary, the existing research has made some significant progress in PS scheduling optimization. However, there are still deficiencies in the research for HP of RE access, such as insufficient consideration of the volatility and stochastic characteristics of RE, fewer studies on the collaborative scheduling of multi-timescale modules, and insufficiently perfect uncertainty handling methods. Therefore, the study proposes to construct a mathematical model of HP of REPS scheduling optimization and introduce a heuristic algorithm to solve it. The innovation of the study is to propose a multi-module cooperative optimization framework to accurately deal with uncertainty and extreme scenarios. Meanwhile, the optimization algorithm improves the solution efficiency and provides new ideas for complex PS scheduling.

3 Methods and materials

This section provides a detailed description of the PS scheduling optimization method proposed in the study. The method consists of scheduling optimization mathematical model and scheduling optimization heuristic algorithm. The combination of the two effectively improves the scheduling efficiency and stability of the PS with a HP of RE access.

3.1 Mathematical model construction for power system scheduling optimization

The PS suffers scheduling complexity issues brought on by volatility and uncertainty as a result of the extensive access to HP of RE sources. Moreover, the conventional phased scheduling approach finds it challenging to satisfy the needs of system stability and economy [15-16]. Therefore, the study proposes a PS scheduling optimization method for HP of RE access.

The two main cores of the method are a mathematical model that can achieve multi-module co-optimization by comprehensively considering system costs, constraints, and uncertainties. The heuristic algorithm can efficiently solve complex optimization problems, taking into account the global search and local fine optimization. Figure 1 depicts the method's general framework.

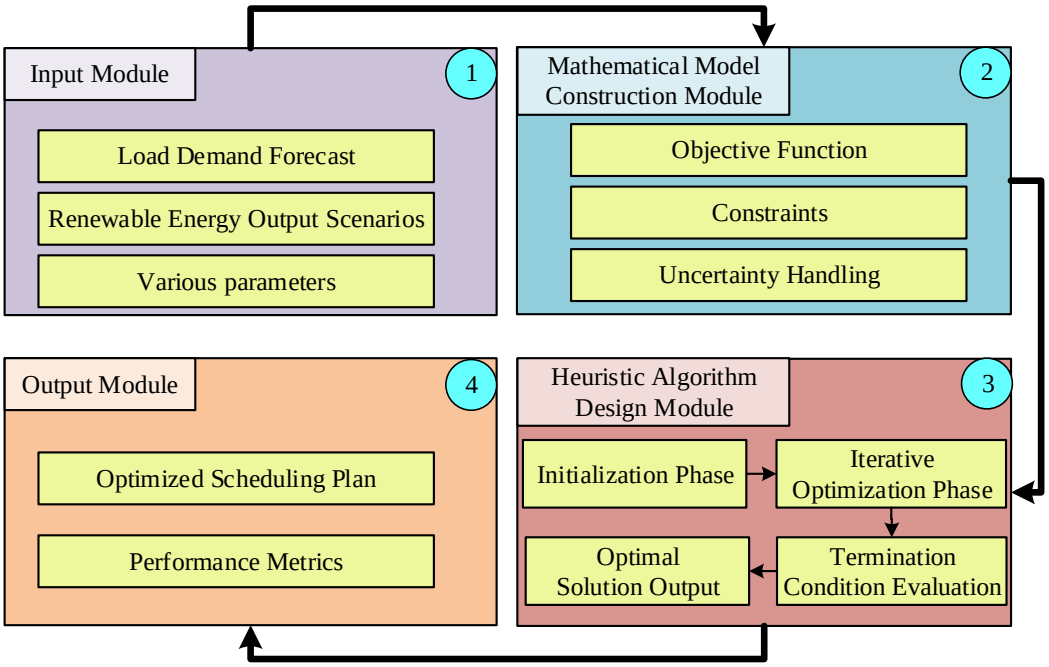


Figure 1: Overall framework of power system scheduling optimization method

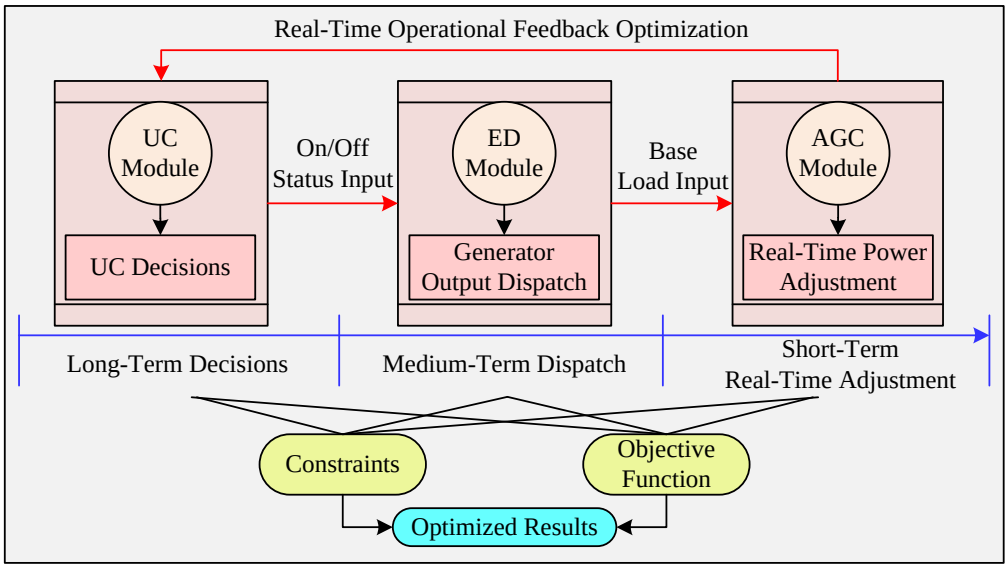


Figure 2: Closed-loop sequential optimization process among UC, ED, and AGC modules

Four components make up the general architecture of the PS scheduling optimization approach suggested in the study, as shown in Figure 1: the input module, the output module, the heuristic algorithm design module, and the mathematical model construction module. The input module contains load demand forecast, RE output scenarios, and various parameters as support. The mathematical model construction module is then responsible for constructing the PS scheduling optimization model based on three main foundations: objective function (OF), constraints, and uncertainty handling [17-18]. The scheduling optimization model based on the four steps is effectively solved using the heuristic algorithm design module. Finally, the output module generates the optimized scheduling plan, including unit start/stop status, power allocation, and standby capacity configuration. It also evaluates the economy and stability of the scheduling scheme through performance indicators.

The mathematical model developed in this study differs from the traditional stage-wise sequential scheduling model in terms of the scheduling optimization approach. The proposed model forms a closed-loop sequential scheduling framework with dynamic feedback coupling among modules by integrating the UC for start-stop decisions, the ED for cost minimization, and the AGC for real-time supply-demand balancing. These modules operate across different time scales and interact through feedback mechanisms to realize coordinated optimization.

The principle of inter-module coordination is illustrated in Figure 2.

As shown in Figure 2, the scheduling optimization process proposed in this study adopts a closed-loop sequential optimization mechanism, consisting of three main modules: UC, ED, and AGC. These modules interact across multiple time scales through real-time feedback to achieve dynamic coordination. During each scheduling cycle, the UC module first optimizes the on-and-off status of units based on current load forecasts, reserve requirements, and other system parameters. Then, it passes the results to the ED module. Then, the ED module performs power allocation and generates a base load profile for the AGC module, which makes real-time, short-term power adjustments. Unlike traditional dispatch models, which operate in isolated stages, the AGC module in this framework generates feedback information, such as load correction values and reserve margin stress levels, continuously during its adjustment process. Rather than discarding this data, it is fed back as correction inputs into the next UC scheduling cycle. Specifically, the system monitors the magnitude and frequency of AGC adjustments in the previous cycle. If frequent or significant real-time corrections are observed, it indicates potential deficiencies in load forecasting or reserve planning. In response, the system increases the reserve capacity settings for the next cycle to improve operational redundancy. At the same time, the load forecast is corrected by incorporating observed deviations into the predicted curve. This enables the UC module to make more accurate start and stop decisions that reflect actual system demand. This adaptive feedback mechanism is repeated in every cycle, progressively refining UC decisions to better match real-world operating conditions and improve overall dispatch responsiveness.

In the mathematical model, schedule optimization aims to reduce the system's overall running costs. The OF is set as shown in Equation (1).

$$\min Z = \sum_{t=1}^{T}\left(\sum_{i=1}^{N}\left(C_{fuel,i,t}+C_{start/stop,i,t}\right)+C_{resreve,t}+C_{EENS,t}\right) \quad (1)$$

In Equation (1), Z denotes the total system cost. T denotes the total quantity of scheduling time segments. N is the total quantity of units. $C_{fuel,i,t}$ is the fuel cost of the i th unit at time t. $C_{start/stop,i,t}$ is the startup and shutdown cost of the i th unit at time t. $C_{resreve,t}$ is the standby cost at time t. $C_{EENS,t}$ is the desired power deficit cost at time t, which mainly measures the supply-demand imbalance caused by RE fluctuations [19]. The UC module is responsible for optimizing the start-stop state (SSS) of the units. The SSS constraint $u_{i,t}$ is shown in Equation (2).

$$u_{i,t} \in \{0,1\}, \forall i,t \quad (2)$$

In Equation (2), a value of 1 for $u_{i,t}$ indicates that the unit is on and a value of 0 indicates that the unit is off. Equation (3) displays the unit start/stop time limitation.

$$\sum_{t=1}^{T} u_{i,t}\Delta t \geq T_{\text{min-on},i} \quad (3)$$

In Equation (3), Δt denotes the time period when the unit is turned on. $T_{\text{min-on},i}$ denotes the minimum continuous operation time of the i th unit. The output power constraint is shown in Equation (4).

$$P_{i,\min} \leq P_{i,t} \leq P_{i,\max} \quad (4)$$

In Equation (4), $P_{i,\min}$ and $P_{i,\max}$ are the maximum and minimum PO of the i th unit. $P_{i,t}$ denotes the PO of the i th unit at time t. Both Equation (3) and Equation (4) hold only when the value of $u_{i,t}$ is 1. The ED module is responsible for optimizing the power allocation of the turned-on units after the UC determines the SSS of the units [20]. The power balance constraints in the ED module are shown in Equation (5).

$$\sum_{i=1}^{N} P_{i,t} + P_{RES,t} = D_t, \forall t \quad (5)$$

In Equation (5), $P_{RES,t}$ denotes the RE output at time t. D_t denotes the load demand at time t. The climbing capacity constraint is shown in Equation (6).

$$P_{i,t} - P_{i,t-1} \leq R_i^{up}, P_{i,t-1} - P_{i,t} \leq R_i^{down}, \forall i,t \quad (6)$$

In Equation (6), R_i^{up} and R_i^{down} are the upper and lower climbing limits for unit i, respectively. The reserve capacity constraint is shown in Equation (7).

$$\sum_{i=1}^{N} R_{i,t} \geq R_{required,t}, R_{i,t} = P_{i,\max} - P_{i,t}, \forall t \quad (7)$$

In Equation (7), $R_{required,t}$ is the standby capacity requirement of the system at time t. $R_{i,t}$ denotes the standby capacity that the i th unit can provide at time t. Equation (8) illustrates how the AGC module regulates the fluctuations by modifying the power in real time depending on ED.

$$P_{i,t} = P_{i,t}^{base} + \Delta P_{i,t}, \forall i,t \quad (8)$$

In Equation (8), $P_{i,t}^{base}$ denotes the base point load provided by the ED module [21]. $\Delta P_{i,t}$ is the real-time power adjustment of the AGC module. The real-time balancing constraint is shown in Equation (9).

$$\sum_{i=1}^{N} \Delta P_{i,t} = \Delta D_t, \quad \Delta D_t = D_t - \sum_{i=1}^{N} P_{i,t}^{base} - P_{RES,t} \quad (9)$$

In Equation (9), ΔD_t represents the difference between the actual load demand and the base point load and RE output. The adjustment speed constraint is shown in Equation (10).

$$\Delta P_{i,t} \leq R_i^{response} \quad (10)$$

In Equation (10), $R_i^{response}$ denotes the upper limit of real-time regulation speed. As a result, the synergistic relationship among the UC, ED, and AGC modules is realized through the tight coupling of inputs and outputs. The UC provides the SSS for the ED, the ED provides the base point load for the AGC, and the feedback from the AGC optimizes the start-stop strategy of the UC.

The proposed model incorporates uncertainty in wind and solar output directly into the scheduling process to enhance adaptability to RE fluctuations. A limited number of representative renewable output scenarios are generated during each scheduling cycle by applying random deviations to forecasted values based on recent historical variation. These scenarios simulate possible short-term fluctuations in renewable generation. The reserve capacity constraint is adjusted accordingly based on the observed fluctuation range, ensuring sufficient buffer during high-variability periods. In the AGC stage, real-time control targets are fine-tuned using deviation trends derived from these scenarios. The proposed model maintains dispatch feasibility and system stability under uncertain renewable output conditions by dynamically updating reserve settings and AGC parameters.

3.2 Power system scheduling optimization heuristic algorithm design

The proposed mathematical model for optimizing PS scheduling takes into account total system operating costs, the synergistic optimization of multiple modules, and the uncertainty associated with a high proportion of RE sources. This provides a theoretical basis for scheduling. However, the simple model may be inefficient or susceptible to local optimization when dealing with complex, multi-constraint, multi-timescale optimization problems [22]. Therefore, heuristic algorithms are introduced to optimize the mathematical model and solve it. In deep reinforcement learning, the DDPG method effectively optimizes unit SSS and power allocation. However, it may converge slowly and become trapped in local optima when solving complex problems with multiple constraints [23]. QPSO overcomes the limitations of DDPG by improving particle diversity and global search capabilities using quantum behavioral mechanisms. Hence, this study combines improved DDPG and QPSO to propose a joint heuristic algorithm. Figure 3 illustrates the computational flow of the enhanced DDPG in this technique.

In Figure 3, the study introduces a dual experience pooling mechanism in DDPG, which balances exploration and utilization by storing diverse samples and high-value samples separately to improve training efficiency and policy quality. Second, to prevent falling into the local optimum, a time-decaying exploration noise technique is used to boost exploration at the beginning and improve stability at the end. Finally, the target network update strategy is optimized to dynamically adjust the target network parameters through the soft update method to enhance the training stability and convergence speed.

The improved DDPG workflow consists of four main stages. First, in the initialization phase, the Critic network, Actor network, and their target networks are randomly initialized. Additionally, two experience pools, B1 and B2, are established. B1 stores the initial experience samples. B2 stores the high-value samples that are selected using a filtering mechanism. The dual-experience pool design maintains diversity in the training data, which improves sample selection efficiency. Next, within the training iterations and time step loop, the agent generates actions via the policy network. This enhances the exploration of unknown strategies by adding exploration noise. After interacting with the environment, experience samples are generated and stored in B1. High-value samples are then selected based on reward values and stored in B2. In this way, the experience pool contains both common experience samples and high value samples. This ensures the samples are diverse and valuable for training, thereby improving the algorithm's learning efficiency.

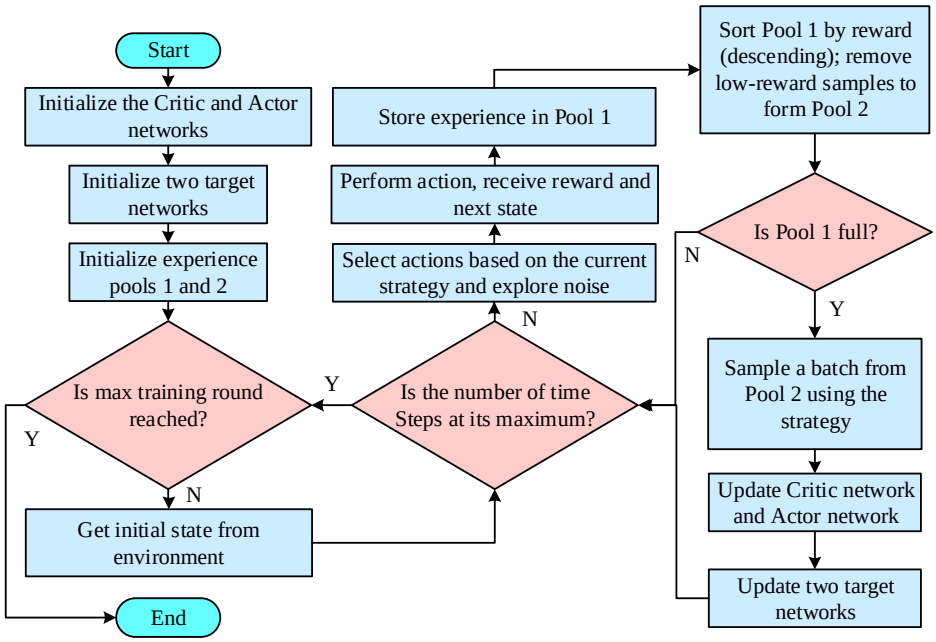


Figure 3: The framework of DDPG-QPSO joint heuristic algorithm

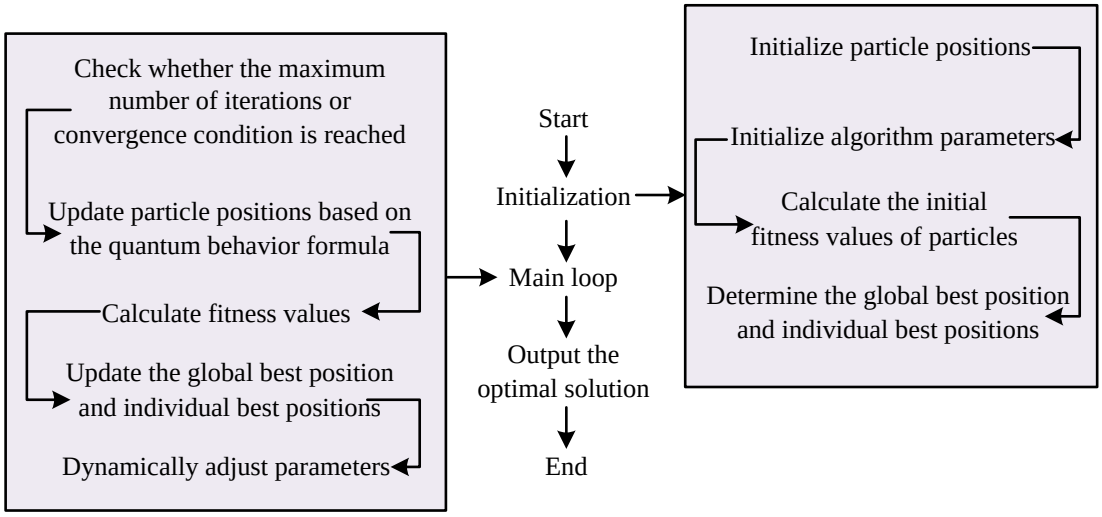


Figure 4: Schematic diagram of the algorithm flow of QPSO

Then, a small batch of high-value samples is sampled from B2 to update the Critic and Actor networks. The Critic network is updated using the error calculated from the target value. Meanwhile, the Actor network is updated using the policy gradient method to maximize the long-term cumulative reward. This optimization allows the model to continuously improve its policy and value function, thereby enhancing the quality of its decisions. Finally, the research employs a soft update method that dynamically adjusts the target network parameters. This further improves training stability and convergence speed. The soft update strategy smoothly adjusts the target network parameters. This prevents excessive fluctuations in the network during training and avoids instability caused by dramatic parameter updates. Figure 4 depicts the QPSO algorithm's flow.

In Figure 4, the overall process of QPSO algorithm is not much different from the traditional PSO algorithm. The steps are initializing particle positions and parameters, calculating fitness values, updating global optimal position (GOP) and individual optimal position (IOP), dynamically adjusting parameters, and iterative judgment to achieve the output optimal solution (OS). However, the core difference between the two is the way of updating the particle position. Traditional PSO is based on the iterative formula of velocity and position, while QPSO adopts the quantum behavioral formula, which constructs the quantum distribution of the particle position through the GOP and IOP. In the QPSO algorithm, the quantum modulation factor controls the randomness of the particle update process and regulates the particles' ability to explore the search space. This enhances search diversity and prevents the algorithm from getting trapped in local OSs. Quantum distribution describes the probabilistic characteristics of particle position updates. New particle positions are generated through formulas based on quantum behavior by combining global and IOPs. This reflects the non-deterministic update mode inspired by quantum mechanics. The quantum behavioral formulation updates the particle positions as shown in Equation (11).

$$x_{i,j}^{(t+1)} = P_{i,j} \pm \beta \cdot \left| x_{i,j}^{(t)} - P_{i,j} \right| \cdot \ln\left(\frac{1}{u}\right) \tag{11}$$

In Equation (11), $x_{i,j}^{(t)}$ and $x_{i,j}^{(t+1)}$ denote the updated position of particle i after t and $t+1$ iterations in the j th dimension. $P_{i,j}$ denotes the reference point of particle i in the j th dimension. β denotes the quantum modulation factor. u denotes the random number, which is used to introduce randomness to give the particle a non-deterministic update property. $P_{i,j}$ is obtained from the GOP and the IOP with certain weights, as shown in Equation (12).

$$P_{i,j} = \phi \cdot p_{best,i,j} + (1-\phi) \cdot g_{best,j} \tag{12}$$

In Equation (12), $p_{best,i,j}$ and $g_{best,j}$ denote the IOP and GOP, respectively. ϕ denotes the inertia factor, which is used to control whether the particle prefers the individual OS or the global OS. In the integration of DDPG and QPSO, the improved DDPG algorithm first generates an initial dispatch strategy based on the current environmental state and load demand information. This strategy includes the start-stop decisions and power allocation for each generation unit over all time periods. The output of DDPG is a deterministic decision vector, representing an executable scheduling solution. This comprehensive scheduling solution serves as a key reference for initializing the population in the QPSO algorithm. More specifically, the DDPG output is encoded as a particle position within the QPSO search space, which is then assigned as the initial position of at least one particle within the swarm. The remaining particles are initialized in the vicinity of this solution through random perturbations, ensuring that the initial population has both guidance and diversity. Based on this initialization, QPSO performs global search optimization. Its quantum-behavior mechanism further refines and adjusts the scheduling strategy, improving the solution's overall stability and adaptability.

Initially, the global search is favored and the local optimization is favored in the later stage. Combining the above, the final PS scheduling optimization flow designed by the study is shown in Figure 5.

In Figure 5, the final PS scheduling optimization process consists of inputting load demand forecasts, RE output scenarios, and related parameters to provide basic data for optimization. With the aim of reducing the overall system cost, a multi-module cooperative scheduling model comprising UC, ED, and AGC modules is built. The improved DDPG is utilized to generate the initial optimization strategy, which is further optimized by QPSO. The optimized scheduling plan covers unit start/stop status, power allocation, and standby capacity configuration. It ultimately achieves efficient and stable scheduling optimization.

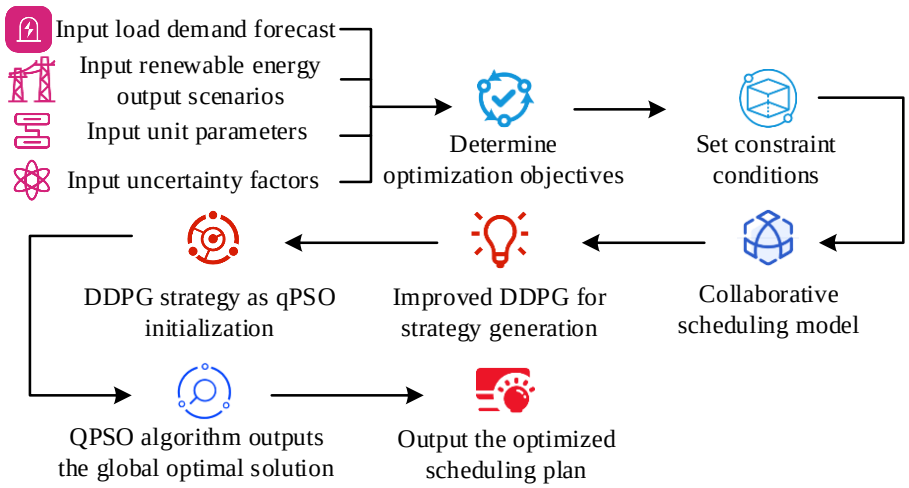


Figure 5: Final power system scheduling optimization process

Table 1: Experimental environment configuration

Hardware configuration		Software configuration	
CPU	Intel Core i9-12900K (16 cores, 3.2 GHz)	Operating system	Ubuntu 22.04 LTS
GPU	NVIDIA GeForce RTX 3090 (24GB VRAM)	Programming language	Python 3.10
Memory	32GB DDR4	Deep learning framework	TensorFlow 2.10
Storage	1TB SSD	Optimization algorithm library	NumPy, SciPy, Pyomo
Power supply	850W High-Efficiency Power Supply	Power system simulation tools	MATPOWER 7.1 (MATLAB Toolbox)
/	/	Data processing tools	Pandas

Table 2: Results of model scheduling performance differences

Indicator	IEEE 30-Bus Test System		IEEE 118-Bus Test System	
	Traditional sequential scheduling model	The proposed model	Traditional sequential scheduling model	The proposed model
Total cost	$12,500	$10,800	$48,200	$42,700
Fuel cost	$7,200	$6,500	$28,500	$26,000
Startup/shutdown cost	$3,000	$2,500	$12,000	$10,200
Reserve cost	$2,000	$1,500	$6,500	$5,500
Demand-supply Imbalance cost	$300	$300	$1,200	$1,000
Supply-demand deviation (MW)	5.5	4.2	25.0	18.5
Response time (s)	10.1	8.3	18.7	13.5

4 Results

The efficiency and superiority of the PS scheduling optimization methods suggested in the study are confirmed in this section using both heuristic algorithms and mathematical models. The focus is on verifying the effectiveness of multi-module collaboration and uncertainty handling, as well as multi-timescale optimization and the performance enhancement and comprehensive optimization capabilities of the improved DDPG, QPSO, and joint heuristic algorithms.

4.1 Validation of mathematical model for power system scheduling optimization

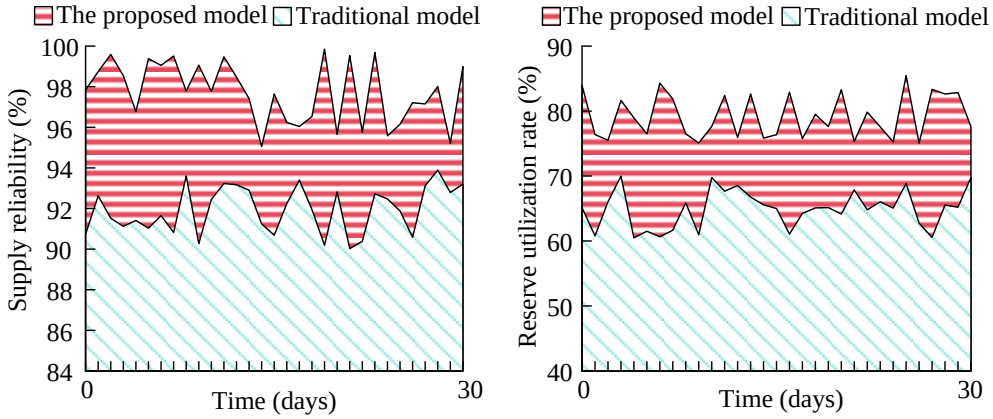
A multi-module cooperative scheduling model including UC, ED, and AGC modules is constructed with the goal of lowering the overall system cost.

Based on Table 1, the study selects the 30-node test system and the 118-node test system from the IEEE standard examples. The former includes 30 nodes, 41 transmission lines, 6 generators, and 20 load nodes, which are suitable for preliminary verification and experimentation. The latter includes 118 nodes, 186 transmission lines, 54 generators, and 99 load nodes, which can be used for in-depth research on the

optimization capabilities of multi module collaborative scheduling and heuristic algorithms. First, the performance of dispatches under the proposed closed-loop sequential scheduling model based on UC-ED-AGC feedback is compared with that under a traditional stage-wise sequential scheduling model. The traditional model is a dispatch process in which the UC, ED, and AGC modules run independently in a fixed order. This process does not consider RE uncertainty or provide feedback or coordination. To ensure a fair comparison, both models are solved using the same optimization algorithm (QPSO) under identical system configurations and forecast conditions. This setup ensures that performance differences are attributed to model structure rather than solver differences. Table 2 displays the findings.

In Table 2, the mathematical model proposed in the study demonstrates advantages in both IEEE 30 node and 118 node testing systems. In terms of economy, the total cost of the 30 node system has been reduced by $1700, and the 118 node system has been reduced by $5500. It optimizes fuel, start stop, and reserve capacity costs. In terms of supply-demand balance capability, the supply-demand deviation has been reduced by 1.3 MW and 6.5 MW respectively, effectively addressing the uncertainty of load demand and RE fluctuations. Meanwhile, the real-time adjustment response time is shortened by 1.8s and 5.2s respectively, improving the dynamic response capability. Overall, the mathematical model proposed in the study has achieved more efficient resource utilization in small-scale systems and demonstrated superior adaptability to complex problems in large-scale systems. The suggested model is compared to conventional PS scheduling models that do not take uncertainty handling into account since it takes into account the uncertainty of RE. The result is shown in Figure 6.

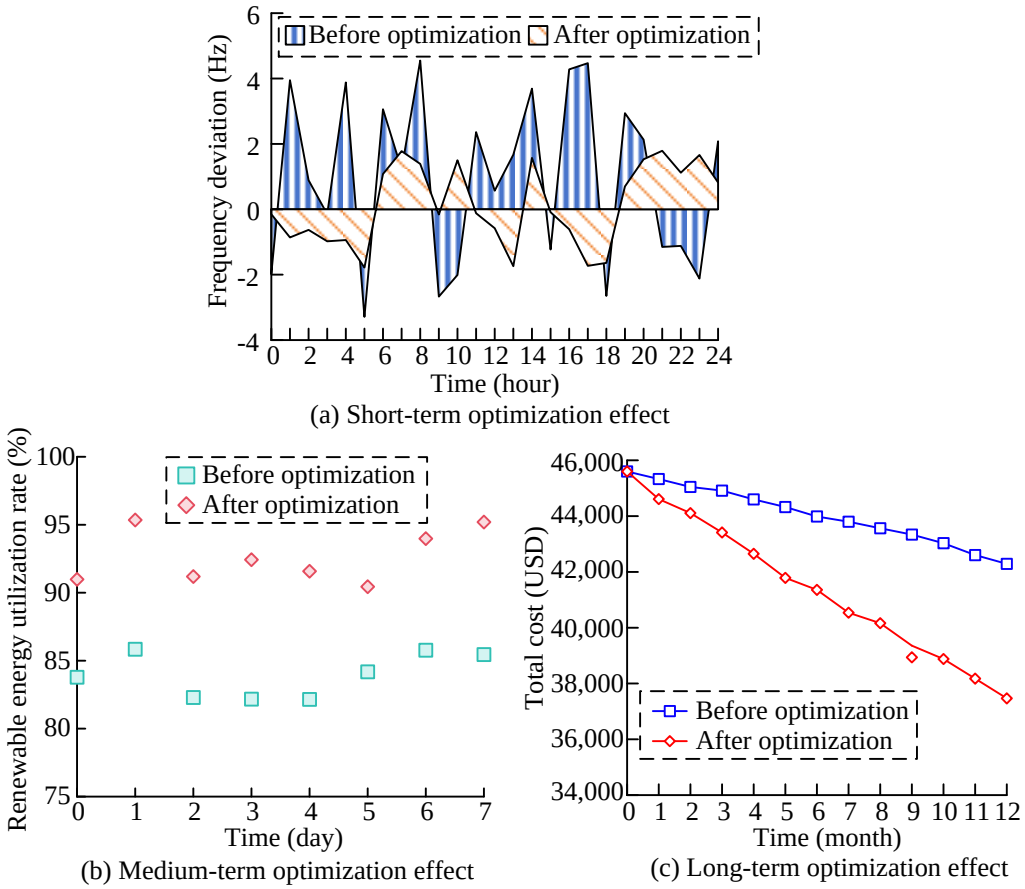
In Figure 6 (a), within 30 days of PS scheduling optimization, the proposed model achieves a power supply reliability of over 94%, with an average of 96.58%. However, traditional models that do not consider uncertainty processing have the highest power supply reliability of only 93.73%, with an average of only 92.16%. In Figure 6 (b), for the utilization rate of backup capacity, after 30 days of model operation, the proposed model increases the utilization rate to between 75% and 87%, while the utilization rate of the traditional model only fluctuated between 60% and 70%. The outcomes displays that the proposed model improves the adaptability of RE fluctuations in PS scheduling optimization. Finally, on various time scales, the impact of scheduling optimization of the study's suggested model is confirmed. The result is shown in Figure 7.



(a) Comparison of supply reliability (b) Comparison of reserve utilization rate

Figure 6: Results compared with traditional models that do not consider uncertainty processing

In Figure 7(a), in the short-term time (24 hours), the frequency deviation (FD) of the PS dispatch before optimization is much larger, up to more than 4Hz. Whereas, after optimization using the research model, the FD of the PS dispatch is effectively controlled and remains between -2Hz and 2Hz. In Figure 7(b), the energy consumption rate of the PS dispatch before optimization averages 84.15% during the interim time i.e., one week, whereas after optimization the energy consumption rate improves to 93.84%. In Figure 7(c), in the long-term time i.e., one year, the optimized PS dispatch significantly reduces the dispatch cost from $45,600 to $37,860 while the pre-optimization dispatch cost is $43,080. The outcomes reveal that the suggested model performs better in terms of long-term economics, medium-term efficiency, and short-term stability.

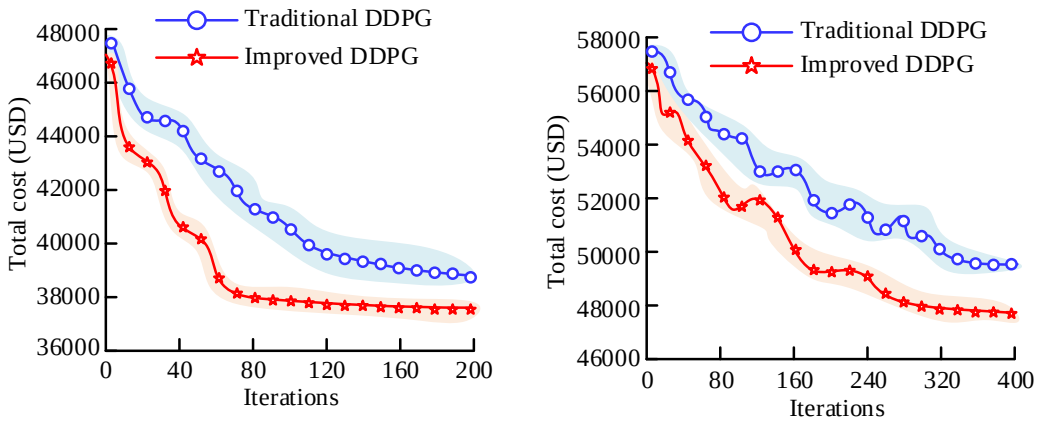


(a) Short-term optimization effect

(b) Medium-term optimization effect

(c) Long-term optimization effect

Figure 7: Scheduling optimization effect on different time scales



(a) IEEE 30-Bus Test System

(b) IEEE 118-Bus Test System

Figure 8: Comparison of DDPG algorithm before and after improvement

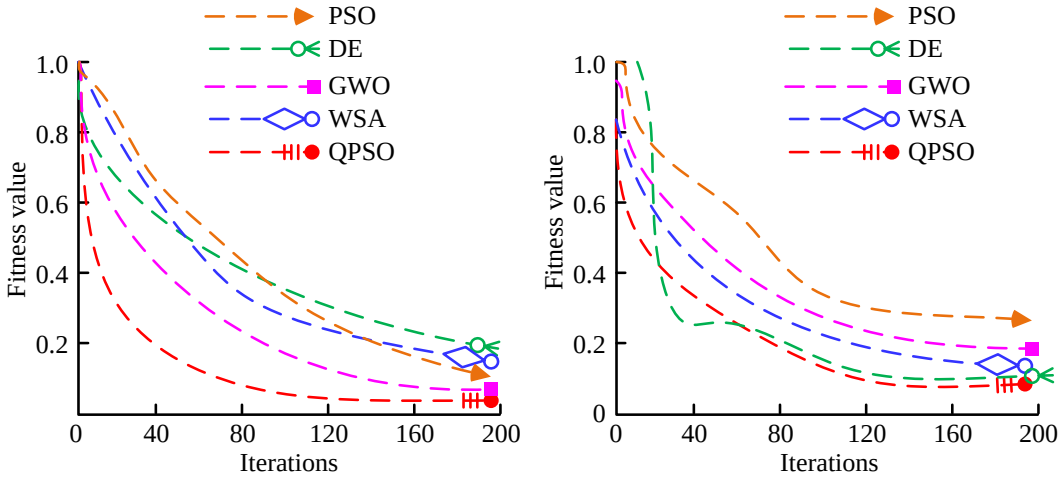
4.2 Validation of heuristic algorithms for power system scheduling optimization

After the validity and superiority of the mathematical model proposed by the study is verified, the study further validates the involved heuristic algorithms. Experiments are first conducted for the improvement of the DDPG algorithm. The DDPG before and after the improvement is applied to solve the mathematical model proposed by the study at the IEEE 30-node and 118-node test systems. When conducting the experiment, in the DDPG algorithm, the learning rates of the Critic network and the Actor network are set to 0.0001. The discount factor is 0.99, the batch size is 64, and the experience pool size is 1,000,000. The noise is examined through the use of an Ornstein-Uhlenbeck process, which has an initial standard deviation of 0.2 and undergoes attenuation during the training process. The target network adopts a soft update with an update parameter of 0.001 to ensure stability. The results are shown in Figure 8.

Figure 8(a) shows that the traditional DDPG algorithm converges after 160 iterations when using the IEEE 30-node system, resulting in a total cost of $39,560.

The improved DDPG converges after 80 iterations, and the total cost is reduced to $37,960. This improvement benefits from dual experience pooling. Storing low-quality samples and high-value samples separately optimizes the efficiency of sample utilization and improves the training speed. This accelerates the convergence process and reduces the cost. Meanwhile, time-decay exploration improves initial exploration capabilities and stabilizes strategy optimization in later stages, accelerating convergence and reducing costs. In As shown in Figure 8(b), both the traditional and improved DDPG require more iterations to converge when using the more complex IEEE 118-node system. However, the improved DDPG still performs better and has a lower convergence cost. The dual experience pool and time decay exploration strategy effectively improves the algorithm's adaptability and convergence efficiency in large-scale systems, demonstrating its superiority. Furthermore, the optimization effect of QPSO is validated through research, and differential evolution (DE), grey wolf optimizer (GWO), and wolf search algorithm (WSA) are selected for comparison. In the QPSO algorithm, the number of particles is 50, the maximum number of iterations is 1000, and the inertia factor is 0.9. The learning factors are set to 1.5 and 2.0, respectively, to control the global and local optimal attractive forces. Setting the quantum modulation factor to 0.5 enhances the flexibility of the particle position update. Figure 9 displays the findings.

In Figure 9(a), in the IEEE 30-node test system, QPSO has the fastest convergence speed among the five algorithms and the lowest final fitness value. In Figure 9(b), in the IEEE 118-node test system, again QPSO has the fastest convergence speed among the five algorithms and the lowest final fitness value. It can be concluded that QPSO effectively enhances the global search capability of particles through the quantum behavior mechanism. Both in the smaller-scale IEEE 30-node system and in the more complex IEEE 118-node system, QPSO shows superior performance, proving its adaptability to problems of different scales and complexities. Finally, the study applies the proposed mathematical model in combination with the joint DDPG-QPSO heuristic algorithm in HP RE scheduling optimization. The more advanced methods in reference [11], [12], 13], and [14] are selected as comparison methods. The algorithms from references [11] to [14] are re-implemented by the research team based on the original descriptions in the respective papers. Each method is tuned within a reasonable range of parameters based on the recommended settings. Then, it is validated to ensure optimal performance in the current test environment. All methods are evaluated under the same experimental conditions, which includes load forecast profiles, RE output scenarios, system topology, and a unified evaluation period. All performance metrics are kept consistent across experiments. The results are presented in Table 3.



(a) IEEE 30-Bus Test System (b) IEEE 118-Bus Test System

Figure 9: Optimization effect verification of QPSO

Table 3: Comprehensive comparison between research methods and reference methods

Method	Total cost ($)	Energy utilization rate (%)	Supply reliability (%)	Frequency deviation (Hz)	Optimization time (s)
Proposed method	37960	94.85	97.10	1.25	58.3
Reference [11]	40230	90.76	93.85	2.64	125.4
Reference [12]	39760	92.42	94.50	2.18	98.7
Reference [13]	38450	93.57	95.87	1.89	75.6
Reference [14]	38930	93.25	95.30	2.01	88.9

In Table 3, the proposed mathematical model and the joint DDPG-QPSO heuristic algorithm of the study show more obvious advantages in HP of REPS scheduling optimization. The proposed method outperforms other methods with the lowest total cost of $37960 and 94.85% energy consumption rate. Meanwhile, the reliability of power supply reaches 97.10%, the FD is only 1.25 Hz, and the optimization time is 58.3 s. The proposed algorithm

demonstrates excellent economy, system stability, and solution efficiency, and provides a highly efficient and reliable solution for PS scheduling optimization in high-percentage RE scenarios.

5 Discussion and conclusion

Targeting the scheduling issue brought on by the HP of RE access in the PS, the study put out a mathematical model for multi-module cooperative scheduling that brought together three main modules and used the enhanced DDPG and QPSO algorithms to address the issue. The efficacy of the study's suggested model and algorithm was confirmed by experimental findings. In the IEEE 30-node and 118-node test systems, the proposed model reduced the total scheduling cost by $1,700 and $5,500, respectively, compared with the traditional sequential scheduling model. It enhanced the energy consumption rate and power supply reliability. The improved DDPG algorithm increased the convergence speed by 50% from $39,560 to $37,960 in the 30-node system by introducing a dual experience pool and a time decay exploration strategy. QPSO exhibited a stronger global search capability, with the fastest convergence speed and the lowest final adaptation value in systems of different sizes compared to other algorithms. In addition, the study's optimization experiments on short-, medium-, and long-term time scales revealed that the FD was effectively reduced, the energy consumption rate was improved by 9.69%, and the total dispatch cost was reduced by 17.6%. The adaptability and superiority of the model in cooperative optimization over multiple time scales were demonstrated.

The DDPG and QPSO algorithms perform well in the test system. However, they may face challenges regarding scalability and adaptability in an actual power grid. As the power grid grows, its computational complexity will increase significantly. This is particularly relevant when working with large volumes of data and real-time dispatching. These factors can lead to a shortage of computing resources and excessively long training times. In addition, the diversity of power grid topologies and operating conditions may affect the algorithm's adaptability. The power grid itself contains complex generators, energy storage systems, and distributed energy resources. Corresponding adjustments to the algorithm are required to effectively address these. In terms of real-time performance, the algorithm functions well in a simulated environment. However, in a highly dynamic actual power grid, it may not respond promptly to load fluctuations and changes in RE. This affects the stability of the system. Therefore, although the algorithm performs well in the test system, it still needs further verification and optimization for practical applications to improve stability and response speed.

References

[1] Lei Gan, Tianyu Yang, Xingying Chen, Gengyin Li, and Kun Yu. Purchased power dispatching potential evaluation of steel plant with joint multienergy system and production process optimization. IEEE Transactions on Industry Applications, 58(2):1581-1591, 2022. https://doi.org/10.1109/TIA.2022.3144652

[2] A. A. Lebedev, A. A. Voloshin, and A. N. Lednev. Conceptual framework for developing highly-automated power distribution networks and micropower systems. Power Technology and Engineering, 58(1):163-168, 2024. https://doi.org/10.1007/s10749-024-01790-2

[3] Ehsan Naderi, Lida Mirzaei, Mahdi Pourakbari-Kasmaei, Fernando V. Cerna, and Matti Lehtonen. Optimization of active power dispatch considering unified power flow controller: application of evolutionary algorithms in a fuzzy framework. Evolutionary Intelligence, 17(3):1357-1387, 2024. https://doi.org/10.1007/s12065-023-00826-2

[4] Lilin Cheng, Haixiang Zang, Anupam Trivedi, Dipti Srinivasan, Zhinong Wei, and Guoqiang Sun. Mitigating the impact of photovoltaic power ramps on intraday economic dispatch using reinforcement forecasting. IEEE Transactions on Sustainable Energy, 15(1):3-12, 2023. https://doi.org/10.1109/TSTE.2023.3261444

[5] Yu Dong, Xin Shan, Yaqin Yan, Xiwu Leng, and Yi Wang. Architecture, key technologies and applications of load dispatching in China power grid. Journal of Modern Power Systems and Clean Energy, 10(2):316-327, 2022. https://doi.org/10.35833/MPCE.2021.000685

[6] Huating Xu, Bin Feng, Chutong Wang, Chuangxin Guo, Jian Qiu, and Mingyang Sun. Exact box-constrained economic operating region for power grids considering renewable energy sources. Journal of Modern Power Systems and Clean Energy, 12(2):514-523, 2023. https://doi.org/10.35833/MPCE.2023.000312

[7] Ahmed M. Abd-El Wahab, Salah Kamel, Mohamed H. Hassan, José Luis Domínguez-García, and Loai Nasrat. Jaya-AEO: an innovative hybrid optimizer for reactive power dispatch optimization in power systems. Electric Power Components and Systems, 52(4):509-531, 2024. https://doi.org/10.1080/15325008.2023.2227176

[8] John L. Cox, William T. Hamilton, Alexandra M. Newman, Michael J. Wagner, and Alex J. Zolan. Real-time dispatch optimization for concentrating solar power with thermal energy storage. Optimization and Engineering, 24(2):847-884, 2023. https://doi.org/10.1007/s11081-022-09711-w

[9] Huifeng Zhang, Dong Yue, Chunxia Dou, and Gerhard P. Hancke. PBI based multi-objective optimization via deep reinforcement elite learning strategy for micro-grid dispatch with frequency dynamics. IEEE Transactions on Power Systems, 38(1):488-498, 2022. https://doi.org/10.1109/TPWRS.2022.3155750

[10] Sicheng Hou, and Shigeru Fujimura. Day-ahead multi-objective microgrid dispatch optimization based on demand side management via particle swarm optimization. IEEJ Transactions on Electrical

and Electronic Engineering, 18(1):25-37, 2023. https://doi.org/10.1002/tee.23711

[11] Navid Shirzadi, Fuzhan Nasiri, Claude El-Bayeh, and Ursula Eicker. Optimal dispatching of renewable energy-based urban microgrids using a deep learning approach for electrical load and wind power forecasting. International Journal of Energy Research, 46(3):3173-3188, 2022. https://doi.org/10.1002/er.7374

[12] Zhongjie Guo, Wei Wei, Mohammad Shahidehpour, Zhaojian Wang, and Shengwei Mei. Optimisation methods for dispatch and control of energy storage with renewable integration. IET Smart Grid, 5(3):137-160, 2022. https://doi.org/10.1049/stg2.12063

[13] Dai Cui, Weichun Ge, Wenguang Zhao, Feng Jiang, and Yushi Zhang. Economic low-carbon clean dispatching of power system containing P2G considering the comprehensive influence of multi-price factor. Journal of Electrical Engineering & Technology, 17(1):155-166, 2022. https://doi.org/10.1007/s42835-021-00877-4

[14] Fatma Gami, Ziyad A. Alrowaili, Mohammed Ezzeldien, Mohamed Ebeed, Salah kamel, Eyad S. Oda, and Shazly A. Mohamed. Stochastic optimal reactive power dispatch at varying time of load demand and renewable energsy resources using an efficient modified jellyfish optimizer. Neural Computing and Applications, 34(22):20395-20410, 2022. https://doi.org/10.1007/s00521-022-07526-5

[15] Jatin Soni, and Kuntal Bhattacharjee. Multi-objective dynamic economic emission dispatch integration with renewable energy sources and plug-in electrical vehicle using equilibrium optimizer. Environment, Development and Sustainability, 26(4):8555-8586, 2024. https://doi.org/10.1007/s10668-023-03058-7

[16] Yukang Shen, Wenchuan Wu, Bin Wang, and Shumin Sun. Optimal allocation of virtual inertia and droop control for renewable energy in stochastic look-ahead power dispatch. IEEE Transactions on Sustainable Energy, 14(3):1881-1894, 2023. https://doi.org/10.1109/TSTE.2023.3254149

[17] Xiaojing Wang, Li Han, Mengjie Li, and Panpan Lu. A time-scale adaptive a forecasting and dispatching integration strategy of the combined heat and power system considering thermal inertia. IET Renewable Power Generation, 17(8):1966-1977, 2023. https://doi.org/10.1049/rpg2.12743

[18] Bing Sun, Ruipeng Jing, Leijiao Ge, Yuan Zeng, Shimeng Dong, and Luyang Hou. Quick hosting capacity evaluation based on distributed dispatching for smart distribution network planning with distributed generation. Journal of Modern Power Systems and Clean Energy, 12(1):128-140, 2023. https://doi.org/10.35833/MPCE.2022.000604

[19] Wei Liu, Tianhao Wang, Shuo Wang, Zhijun E, and Ruiqing Fan. Day-ahead robust optimal dispatching method for urban power grids containing high proportion of renewable energy. Process Safety and Environmental Protection, 178(1):715-727, 2023. https://doi.org/10.1016/j.psep.2023.08.025

[20] Maolin Li, Youwen Tian, Haonan Zhang, and Nannan Zhang. The source-load-storage coordination and optimal dispatch from the high proportion of distributed photovoltaic connected to power grids. Journal of Engineering Research, 12(3):421-432, 2024. https://doi.org/10.1016/j.jer.2023.10.042

[21] Junjie Rong, Ming Zhou, Zhi Zhang, and Gengyin Li. Coordination of preventive and emergency dispatch in renewable energy integrated power systems under extreme weather. IET Renewable Power Generation, 18(7):1164-1176, 2024. https://doi.org/10.1049/rpg2.12893

[22] Zhoujun Ma, Yizhou Zhou, Yuping Zheng, Li Yang, and Zhinong Wei Distributed robust optimal dispatch of regional integrated energy systems based on ADMM algorithm with adaptive step size. Journal of Modern Power Systems and Clean Energy, 12(3):852-862, 2023. https://doi.org/10.35833/MPCE.2023.000204

[23] Jian Hu, Yingjun He, Wenqian Xu, Yixin Jiang, Zhihong Liang, and Yiwei Yang. Anomaly detection in network access-using LSTM and encoder-enhanced generative adversarial networks. Informatica, 49(7):175-186, 2025. https://doi.org/10.31449/inf.v49i7.7246