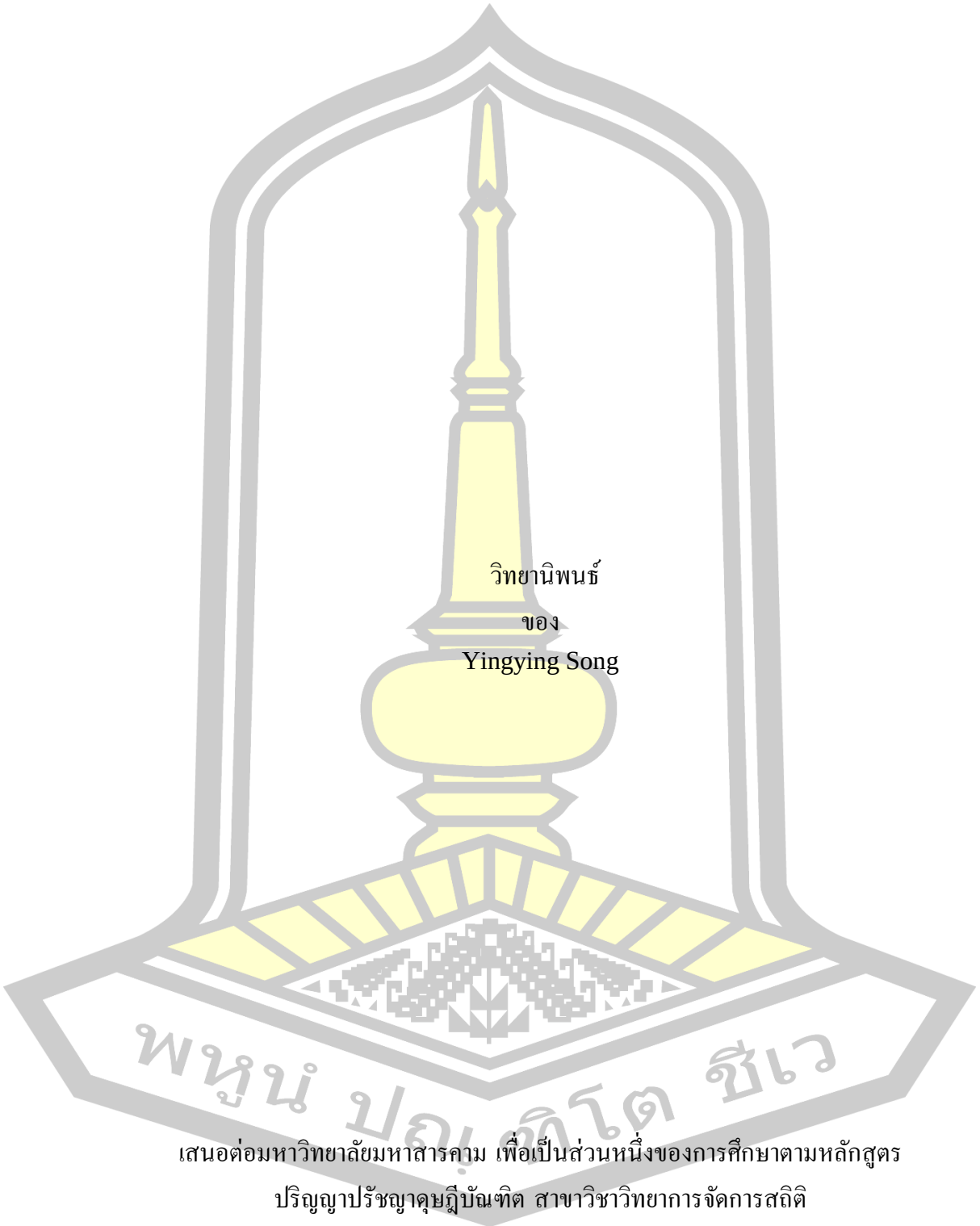
Financial Risk Early Warning Models Based on Machine Learning

Yingying Song

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Statistical Management Science
March 2025

แบบจำลองเตือนความเสี่ยงทางการเงินล่วงหน้าตามการเรียนรู้ของเครื่อง



วิทยานิพนธ์

ของ

Yingying Song

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการจัดการสถิติ

มีนาคม 2568

Financial Risk Early Warning Models Based on Machine Learning

Yingying Song

A Thesis Submitted in Partial Fulfillment of Requirements

for Doctor of Philosophy (Statistical Management Science)

March 2025

The examining committee has unanimously approved this Thesis, submitted by Ms. Yingying Song , as a partial fulfillment of the requirements for the Doctor of Philosophy Statistical Management Science at Mahasarakham University

Examining Committee

.....Chairman
(Assoc. Prof. Manad Khamkong , Ph.D.)

.....Advisor
(Assoc. Prof. Piyapatr Busababodhin , Ph.D.)

.....Co-advisor
(Asst. Prof. Monchaya Chiangpradit , Ph.D.)

.....Committee
(Assoc. Prof. Nipaporn Chutiman , Ph.D.)

.....External Committee
(Asst. Prof. Wuttichai Srisodaphol , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Statistical Management Science

.....
(Prof. Pairot Pramual , Ph.D.)
Dean of The Faculty of Science

.....
(Prof. Anongrit Kangrang , Ph.D.)
Acting Dean of Graduate School

TITLE Financial Risk Early Warning Models Based on Machine Learning

AUTHOR Yingying Song

ADVISORS Associate Professor Piyapatr Busababodhin , Ph.D.

Assistant Professor Monchaya Chiangpradit , Ph.D.

DEGREE Doctor of Philosophy **MAJOR** Statistical Management Science

UNIVERSITY Mahasarakham University **YEAR** 2025

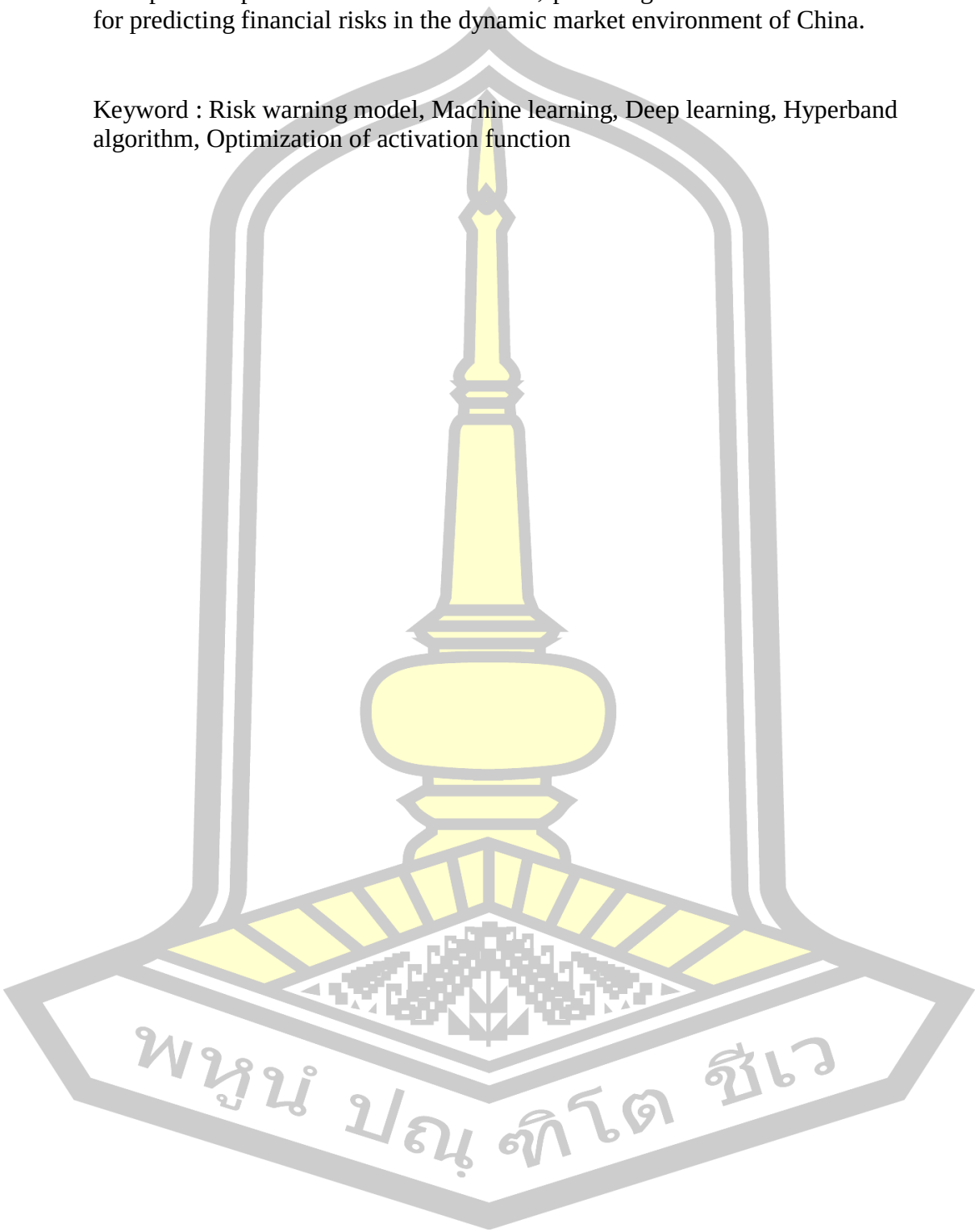
ABSTRACT

With the rapid development of the Chinese economy and the deepening of supply side structural reform, listed companies are facing increasingly fierce market competition and financial uncertainty. In this context, companies need to complete transformation and upgrading through technological innovation and intelligent management, accurately predict and respond to financial risks, in order to achieve sustainable and healthy development. This article takes the quarterly financial indicator data of Chinese listed companies from 2017 to 2024 as the research object. Window sliding and SMOTE techniques are used to handle sample imbalance and generate new samples. PCA and random forest methods are respectively used for feature extraction. In terms of model construction, this article combines traditional machine learning and deep learning techniques, covering traditional machine learning models (logistic regression, support vector machine, decision tree, BP neural network) and their ensemble models (random forest, XGBoost, Stacking model), deep learning models (CNN, BiLSTM, attention mechanism) and their ensemble models (CNN BiLSTM, CNN-AT, BiLSTM-AT, CNN-BiLSTM-AT). In addition, this article explores the optimization of hyperparameters for deep learning models using the Hyperband algorithm, as well as the optimization of activation functions for each layer.

The research results indicate that financial characteristics such as fixed asset ratio, working capital, and current ratio have a significant impact on financial risk prediction. Compared with the traditional machine learning model, the deep learning model has significant advantages in performance, while the integrated learning model outperforms the single model, and the introduction of attention mechanism further improves the performance of the model. Especially under the condition of a time step of 8, the CNN-BiLSTM-AT ensemble deep learning model achieved the highest accuracy (99.4%). By optimizing the activation function of the CNN-BiLSTM-AT model (custom ReLU_Tanh), the model's performance on the ROC curve is closer to the upper left corner, demonstrating a better balance of sensitivity and specificity. In addition, this article selected data from two listed companies for 100 repeated experiments to further verify the stability and effectiveness of the model in practical applications. The experimental results show

that integrated deep learning models can more effectively capture complex temporal and spatial dependencies in financial data, providing a robust and effective solution for predicting financial risks in the dynamic market environment of China.

Keyword : Risk warning model, Machine learning, Deep learning, Hyperband algorithm, Optimization of activation function



ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Associate Professor Dr. Piyapatr Busababodhin, and my co-supervisor, Assistant Professor Dr. Monchaya Chiangpradit, for their invaluable support, guidance, and encouragement throughout my research and thesis writing. As an international student, I am especially grateful for their continuous care and assistance, not only in my academic journey but also in adapting to a new environment. Their mentorship has been instrumental in my growth.

I would also like to extend my sincere appreciation to Associate Professor Dr. Nipaporn Chutiman, Associate Professor Dr. Manad Khamkong, and Assistant Professor Dr. Wuttichai Srisodaphol for their insightful comments and constructive suggestions, which have greatly enriched my research. Additionally, I am deeply thankful to all the academic staff members of the Department of Mathematics, Mahasarakham University, for their generous support and assistance. Their encouragement has provided me with a strong academic foundation and a welcoming community.

I am especially grateful to my family for their unwavering support, patience, and encouragement throughout my Ph.D. journey. Their love and belief in me have been my greatest source of strength and motivation.Finally, I would like to express my sincere gratitude to all the teachers and classmates who have guided and supported me through different stages of my education.Their kindness and companionship have made this experience even more meaningful.

Yingying Song

TABLE OF CONTENTS

Page

ABSTRACT ..... D

ACKNOWLEDGEMENTS ..... F

TABLE OF CONTENTS ..... G

List of tables L

List of figures ..... O

Chapter 1 Introduction 1

1.1 Research Background 1

1.2 Research Objectives and Significance 2

1.2.1 Research Objectives 2

1.2.2 Research significance 3

1.3 Research ideas and methods 4

1.3.1 Research ideas 4

1.3.2 Research methods 5

1.4 Research Content 5

1.5 Innovation points 7

Chapter2 Literature Review 8

2.1 Definition of Financial Risk Concept 8

2.2 Determination of financial risk warning indicators 10

2.3 Establishment of Financial Risk Warning Model 11

2.3.1 Early warning models of traditional statistical techniques 11

2.3.2 Early warning models of machine learning technology 12

2.3.3 Early warning models of optimization algorithm technology 15

2.4 Statistical method theory 17

2.4.1 Data Preprocessing methods 18

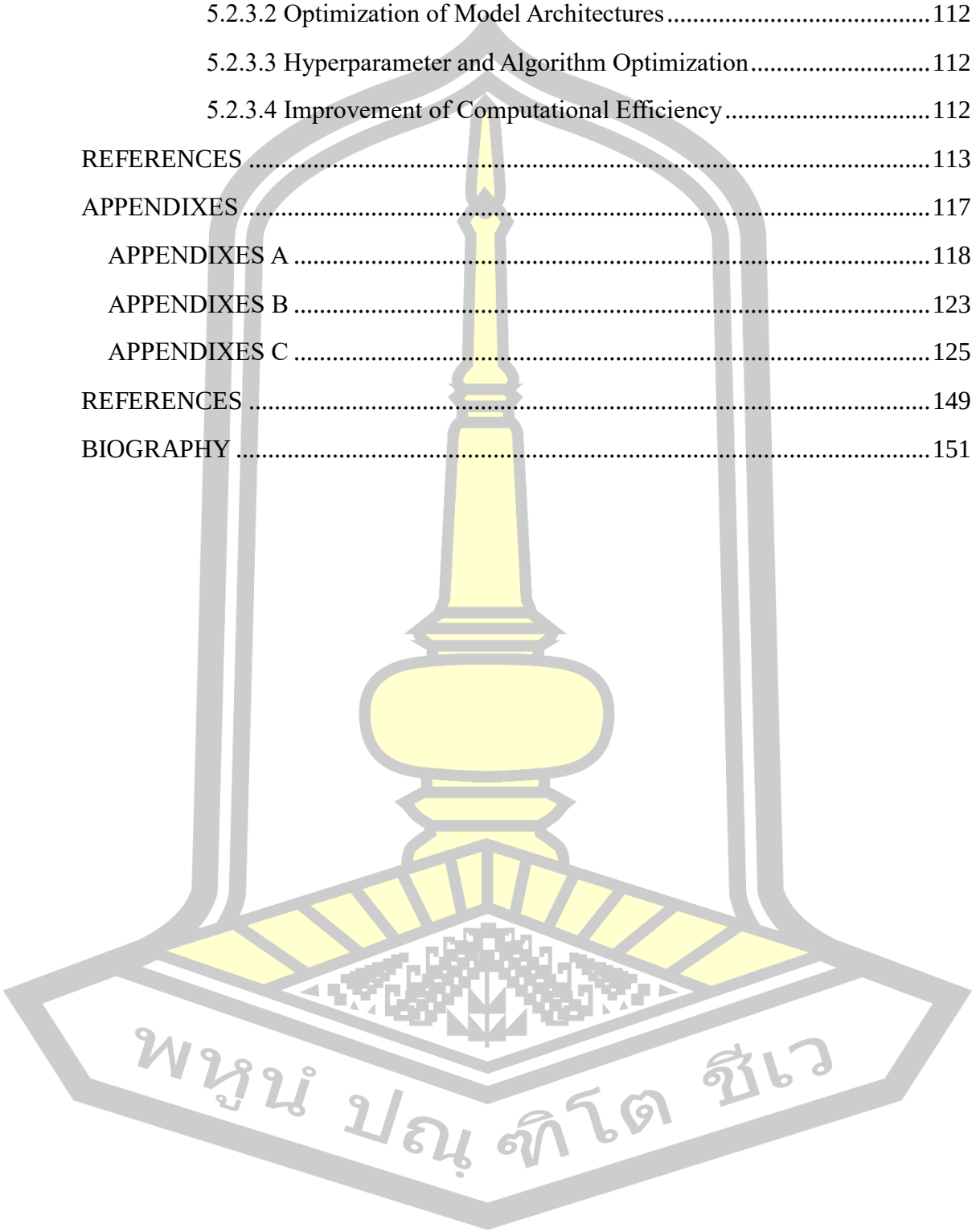
2.4.1.1 Smote Oversampling 18

2.4.2 Feature selection methods 18
2.4.2.1 Principal Component Analysis 19
2.4.2.2 Random Forest Feature Selection 20
2.4.4 Machine learning models 20
2.4.4.1 Logistic regression model 21
2.4.4.2 Decision Tree Model 22
2.4.4.3 Support Vector Machine Model 23
2.4.4.4 BP neural network model 24
2.4.5 Deep learning models 26
2.4.5.1 Convolutional Neural Network Model 27
2.4.5.2 Bidirectional Long Short-Term Memory Network Model 29
2.4.5.3 Attention Mechanism 32
2.4.6 Ensemble Learning Models 33
2.4.6.1 Bagging Algorithm - Random Forest Model 34
2.4.6.2 Boosting Algorithm - XGBoost Model 35
2.4.6.3 Stacking Algorithm - Layered Model 36
2.4.7 Activation function optimization algorithm 38
2.4.7.1 ReLU activation function 38
2.4.7.2 Sigmoid activation function 39
2.4.7.3 Tanh activation function 40
2.4.7.4 ReLU_Tanh activation function 40
2.4.7.5 ReLU_Sig activation function 41
2.4.7.6 Tanh_Sig activation function 42
2.4.8 Hyperband algorithm 43
2.4.9 Evaluation methods for binary classification models 44
2.4.9.1 Confusion Matrix 45
2.4.9.2 Accuracy, precision, recall, F1 value 45
2.4.9.3 ROC curve and AUC 46
2.4.9.4 Cross validation evaluation model 47

Chapter 3 Methodology 49
3.1 Research Scope 49
3.1.1 Data 49
3.1.1 Varaiables 50
3.2 Step of The Methodology 50
3.2.1 Data collection and preprocessing 51
3.2.1.1 Data collection 51
3.2.1.2 Data preprocessing 51
3.2.2 Financial indicator selection 52
3.2.2.1 Traditional indicator selection methods 52
3.2.2.2 Machine learning indicator selection methods 52
3.2.3 Establishing Single Machine Learning Models 53
3.2.4 Establishing ensemble learning models 53
3.2.5 Establishing an innovative optimization learning model 55
3.3 Work Flow Chart 57
Chapter 4 Results 58
4.1 Simulation 58
4.2.1 Indicator selection 58
4.2.2 Model performance comparison 61
4.2 Results 64
4.2.1 Data preprocessing 64
4.2.1.1 Sliding window 64
4.2.1.2 Smote oversampling 65
4.2.2 Indicator selection 66
4.2.2.1 PCA indicator selection 66
4.2.2.2 Random forest indicator selection 68
4.2.3 Model performance comparison 71
4.2.3.1 Traditional machine learning models 71
4.2.3.1.1 Traditional single machine learning models 71

4.2.3.1.2 Ensemble of traditional machine learning models.....77
4.2.3.1.3 Summary of Traditional Machine Learning Models 82
4.2.3.2 Deep learning models 82
4.2.3.2.1 Single deep learning models.....83
4.2.3.2.2 Ensemble of Deep Learning Models 90
4.2.3.2.3 Summary of Deep Learning Models 100
4.2.3.3 Model based on activation function optimization algorithm.....101
4.2.4 Case application analysis.....104
Chapter 5 Conclusion and Discussion 107
5.1 Conclusion 107
5.1.1 Importance of Financial Features 107
5.1.2 Impact of Time-Step Settings and Smote 108
5.1.3 Effectiveness of Deep Learning Models 108
5.1.4 Effectiveness of Ensemble Models 108
5.1.5 Effectiveness of the Hyperband Algorithm 109
5.1.6 Effectiveness of the Attention Mechanism.....109
5.1.7 Effectiveness of ReLU_Tanh Activation Function.....110
5.2 Discussion.....110
5.2.1 Research Significance 110
5.2.1.1 Advancements in Financial Risk Prediction Models.....110
5.2.1.2 Feature Importance and Data Handling Insights 110
5.2.1.3 Practical Implications for Ensemble and Deep Learning Models 111
5.2.2 Research Limitations.....111
5.2.2.1 Dataset Constraints.....111
5.2.2.2 Dependence on Hyperparameter Tuning.....111
5.2.2.3 Fixed Initialization Parameters.....111
5.2.2.4 Limitations of Model Architectures.....112
5.2.3 Future Research.....112

5.2.3.1 Expansion of Feature Dimensions 112
5.2.3.2 Optimization of Model Architectures 112
5.2.3.3 Hyperparameter and Algorithm Optimization 112
5.2.3.4 Improvement of Computational Efficiency 112
REFERENCES 113
APPENDIXES 117
APPENDIXES A 118
APPENDIXES B 123
APPENDIXES C 125
REFERENCES 149
BIOGRAPHY 151



List of tables

Page

Table 1 Binary Chaos Matrix 45
Table 2 Sample distribution table 50
Table 3 Financial risk warning indicator system 50
Table 4 Model performance of different feature extraction methods 58
Table 5 Accuracy of models (Non-smote) at different divide proportion 62
Table 6 Accuracy of models (Smote) at different divide proportion 63
Table 7 Accuracy of models (Non-smote) at different epochs and batches 63
Table 8 Accuracy of models (Smote) at different epochs and batches 64
Table 9 Distribution of sliding window samples 65
Table 10 Data distribution of training set before and after Smote 65
Table 11 Parameter settings for LR 71
Table 12 Performance of logistic regression models under different conditions 72
Table 13 Classification Report of Logistic Regression Models under Non-Smote-PCA 73
Table 14 Parameter settings for SVM 73
Table 15 Performance of SVM models under different conditions 74
Table 16 Classification Report of SVM Models under Non-Smote-PCA 74
Table 17 Parameter settings for DT 74
Table 18 Performance of Decision Tree models under different conditions 75
Table 19 Classification Report of DT Model under Non-Smote-RF 76
Table 20 Parameter settings for BP 76
Table 21 Performance of BP models under different conditions 77
Table 22 Classification Report of BP Model under Non-Smote-PCA 77
Table 23 Parameter settings for RF 78
Table 24 Performance of RF models under different conditions 78
Table 25 Classification Report of RF Model under Smote-RF 79

Table 26 Parameter settings for XGBoost 79
Table 27 Performance of XGBoost models under different conditions 80
Table 28 Classification Report of XGBoost Model under Non-Smote-RF 80
Table 29 Parameter settings for Stacking 80
Table 30 Performance of Stacking models under different conditions 81
Table 31 Classification Report of Stacking Model under Smote 81
Table 32 Best single and ensemble machine learning model accuracy 82
Table 33 Using hyperband algorithm to find the optimal parameters 83
Table 34 Parameter settings for CNN 83
Table 35 Performance of CNN under different conditions 84
Table 36 Classification Report of CNN under Smote-RF 84
Table 37 Parameter settings for BiLSTM 86
Table 38 Performance of BiLSTM under different conditions 86
Table 39 Classification Report of BiLSTM under Smote 87
Table 40 Parameter settings for AT 88
Table 41 Performance of AT under different conditions 88
Table 42 Classification Report of AT under Smote-RF 89
Table 43 Parameter settings for CNN-BiLSTM 90
Table 44 Performance of CNN-BiLSTM under different conditions 91
Table 45 Classification Report of CNN-BiLSTM under Smote-RF 91
Table 46 Parameter settings for CNN-AT 93
Table 47 Performance of CNN-AT under different conditions 93
Table 48 Classification Report of CNN-AT under Smote-PCA 94
Table 49 Parameter settings for BiLSTM-AT 95
Table 50 Performance of BiLSTM-AT under different conditions 96
Table 51 Classification Report of BiLSTM-AT under Smote-RF 96
Table 52 Parameter settings for CNN-BiLSTM-AT 98
Table 53 Performance of CNN-BiLSTM-AT under different conditions 98
Table 54 Classification Report of CNN-BiLSTM-AT under Smote 99

Table 55 Best single and ensemble machine learning model.....100

List of figures

Page

Figure 1 BP neural network structure diagram 25

Figure 2 CNN structure diagram..... 27

Figure 3 LSTM structure diagram 30

Figure 4 BiLSTM structure diagram..... 31

Figure 5 AT structure diagram 33

Figure 6 ReLU activation function curve graph 39

Figure 7 Sigmoid activation function curve graph 39

Figure 8 Tanh activation function curve graph 40

Figure 9 ReLU_Tanh activation function curve graph 41

Figure 10 ReLU_Sig activation function curve graph 42

Figure 11 ReLU_Sig activation function curve graph 42

Figure 12 Hyperband Algorithm Optimization Framework Diagram 44

Figure 13 Window sliding diagram..... 52

Figure 14 Step of The Methodology 56

Figure 15 Work Flow Chart 57

Figure 16 Correlation between important indicators extracted by PCA and results 60

Figure 17 Correlation between important indicators extracted by Random Forest and results 61

Figure 18 Correlation between principal components and original features 66

Figure 19 Box Plot Comparison of Key Financial Indicators with PCA Between ST and Non-ST Companies 68

Figure 20 Random forest selection of the important indicators..... 69

Figure 21 Box Plot Comparison of Key Financial Indicators with Random forest Between ST and Non-ST Companies 71

Figure 22 Traditional machine learning model accuracy..... 82

Figure 23 Trend chart of accuracy and loss value of CNN model..... 85

Figure 24 The ROC curve of the CNN model 85

Figure 25 Trend chart of accuracy and loss value of BiLSTM model 87
Figure 26 The ROC curve of the BiLSTM model 87
Figure 27 Trend chart of accuracy and loss value of AT model 89
Figure 28 The ROC curve of the AT model 90
Figure 29 Trend chart of accuracy and loss value of CNN-BiLSTM model 92
Figure 30 The ROC curve of the CNN-BiLSTM model 92
Figure 31 Trend chart of accuracy and loss value of CNN-AT model 94
Figure 32 The ROC curve of CNN-AT model 94
Figure 33 Trend chart of accuracy and loss value of BiLSTM-AT model 97
Figure 34 The ROC curve of BiLSTM-AT model 97
Figure 35 Trend chart of accuracy and loss value of CNN-BiLSTM-AT model 99
Figure 36 The ROC curve of CNN-BiLSTM-AT model 99
Figure 37 Ensemble of deep learning models accuracy 100
Figure 38 Structure diagram of CNN-BiLSTM-AT model before and after activation function optimization 101
Figure 39 Comparison of the results of different activation functions 102
Figure 40 The ROC curve of CNN-BiLSTM-AT under different activation function 104
Figure 41 Probability distribution of financial risk prediction for Company A 106
Figure 42 Probability distribution of financial risk prediction for Company B 106

Chapter 1

Introduction

This chapter introduces research background, research objectives and Significance, research ideas and methods, research content, and innovation points.

1.1 Research Background

Since the reform and opening up, China has transitioned from a planned economy to a market economy, and its economy has developed rapidly. With the continuous deepening of supply side structural reform, China's economy has begun to transition from a high-speed growth stage to a high-quality development stage. As the main part of the Chinese market, listed companies are facing more market competition and regulation due to the accelerated transformation and upgrading speed of listed companies in various industries led by the digital economy in the context of market economy transformation and global opening of financial markets. The outbreak of the epidemic in early 2020 has also led to an increase in the uncertainty of their financial situation, which has increased the company's default risk The difficulty of managing risks such as financial and operational risks.

In 1990, the Shanghai Stock Exchange and Shenzhen Stock Exchange were established and completed their first trading, and in 2021, the Beijing Stock Exchange was established, forming a basic pattern of three pillars in mainland China. As of December 2023, there were a total of 5335 listed companies in the Chinese A-share market, with a total market value of approximately 83.73 trillion yuan, a decrease of approximately 1.16 trillion yuan from the end of 2022. This decline in market value is largely due to issues such as macroeconomic uncertainty, financial market fluctuations, monetary policy adjustments, and insufficient investor confidence. The Chinese securities market started relatively late compared to European and American countries, and the regulatory system is still in the development stage, which to some extent affects the healthy development of the market. On the one hand, the company's operating performance, profit and loss situation, and debt situation have not received high attention from the company's management. On the other hand, the company may face problems such as blind expansion, opaque financial information, and inadequate internal control, or may fall into insufficient information disclosure in risk behaviors

such as financial fraud, these factors may lead to financial instability and ultimately evolve into financial crises. Generally speaking, the financial crisis of a listed company is not sudden, but a dynamic and continuous process. Firstly, it will go through a normal financial situation, problems will appear, and then gradually deteriorate. When it develops to a continuous and irreversible state, it will lead to the listed company falling into financial difficulties. Therefore, in order to avoid the above-mentioned problems, it is particularly important for the company's management, financial institutions, and relevant investors to detect potential crisis signals early and take corresponding preventive measures.

It is crucial to strengthen the construction of the financial risk warning system for listed companies, which can effectively prevent potential financial risks and improve the overall industry environment. Having a comprehensive understanding of the financial situation of listed companies and anticipating potential financial risks in advance is crucial for their healthy development during the transformation and upgrading process. By exploring and optimizing financial early warning models, creating high-precision models suitable for listed companies, providing more accurate risk assessments for listed companies, and promoting more stable and sustainable development of the capital market.

1.2 Research Objectives and Significance

1.2.1 Research Objectives

Based on the research of domestic and foreign scholars on financial crisis theory, this article aims to select a suitable financial risk warning model for Chinese listed companies through indicator selection and model comparison.

(1) To Determine Critical Indicators Influencing Financial Risk

To systematically identify and analyze the comprehensive system of relevant indicators that significantly impact a company's financial risk. This objective will involve an in-depth exploration of existing literature and empirical data to pinpoint the most critical factors affecting a company's financial stability.

(2) To Establish and Compare Financial Risk Early Warning Models Based on Machine Learning

To develop and critically compare various financial risk early warning models utilizing machine learning techniques. The aim is to discern the most effective and

accurate models capable of predicting a company's financial risk at an early stage. This objective encompasses a comparative analysis to evaluate the strengths and weaknesses of different machine learning models in predicting financial risks.

(3) To Create New Approach by Development of an Enhanced Combination Prediction Model

To conceptualize and establish a novel approach through the creation of a combination prediction model that integrates the strengths of the methods identified in objective 2. This objective aims to augment the predictive power and accuracy by leveraging the synergies of combined models, and subsequently compare its efficacy with single prediction models to determine the optimal solution for early financial risk detection.

1.2.2 Research significance

(1) Theoretical significance

To improve and enrich the selection methods of financial risk warning indicators for companies, this article will use traditional principal component analysis and feature selection methods in machine learning to analyze and compare their advantages and disadvantages in improving model performance and explaining feature relationships. Through comparative research, the selection method of financial risk warning indicators for companies can be expanded and improved, providing strong support for more accurate warning of financial risks.

To improve and enrich the construction methods of the company's financial risk warning model, this article will use traditional machine learning models, deep learning models, and ensemble learning models to analyze and compare their advantages and disadvantages in improving model performance, reducing overfitting risks, and adapting to different data distributions. Through in-depth research on the application of these two methods, it is expected to provide theoretical support for the design and improvement of financial risk warning systems.

(2) Realistic significance

The significance for the company and management: It helps to enhance the sensitivity and comprehensiveness of the company's financial situation. The management of the company is able to obtain key financial information more timely and accurately, thereby formulating effective risk management strategies, reducing the

adverse impact of potential risks on the business, improving their own risk management level, and promoting the high-quality and stable long-term development of the company.

The significance for investors: It helps investors to more accurately evaluate the financial condition and risk level of the company, estimate the potential value of the company, avoid the impact of information asymmetry, formulate investment strategies more rationally, improve the accuracy and reliability of investment decisions, reduce investment risks, improve long-term returns on investments, and protect the rights and interests of investors.

The significance for creditors: It helps creditors to comprehensively evaluate the company's repayment ability and debt risk, in order to formulate scientific and reasonable debt management strategies and credit decisions, reduce the potential risk of debt default, improve the monitoring level of creditors on the company's financial situation, and protect the legitimate rights and interests of creditors themselves.

The significance for regulatory authorities: Through financial risk warning, regulatory agencies can enhance their accurate monitoring ability of the financial status of listed companies, timely warn potential risks, identify market risks and abnormal fluctuations, reduce market uncertainty, provide scientific basis for regulatory policy formulation, and promote the development of the entire market towards a healthier and more transparent direction.

1.3 Research ideas and methods

This research has certain theoretical and practical significance in the field of financial risk warning research.

1.3.1 Research ideas

This article focuses on A-share listed companies in China's Shanghai and Shenzhen stock exchanges. Starting from both macro and micro perspectives, as well as financial and non-financial factors, a comprehensive financial risk early-warning indicator system is constructed. Python software is utilized for data processing, combined with principal component analysis and feature selection methods to achieve dimensionality reduction. Traditional machine learning models (logistic regression, SVM, decision tree, BP neural network), ensemble learning models (including Bagging algorithm (random forest), Boosting algorithm (XGBoost), and Stacking

algorithm), deep learning models (convolutional neural network, bidirectional long short-term memory network, attention mechanism), and ensemble deep learning models are employed to compare and analyze the financial risk prediction performance of each model across different time periods.

Based on these analyses, this study proposes a CNN-BiLSTM-AT financial risk early-warning model optimized with an activation function algorithm and validates its performance. Finally, the research findings are summarized, specific recommendations are provided, and limitations are identified to guide future improvements.

1.3.2 Research methods

(1) Literature research method

By conducting in-depth research and analysis of relevant literature and professional books at home and abroad, systematically reviewing the research progress of financial risk warning indicator systems and financial warning models, summarizing the advantages and disadvantages of various machine learning technologies, and providing sufficient theoretical support for the research in this article.

(2) Application of Machine Learning Algorithms

By integrating traditional machine learning models (logistic regression, decision tree, support vector machine, BP neural network), deep learning models (convolutional neural network, bidirectional long short-term memory network, attention mechanism), and ensemble learning models (random forest, XGBoost, Stacking) with corporate financial risk early warning, and incorporating principal component analysis and feature variable input techniques, the predictive performance and efficiency of the models have been significantly enhanced, improving their practicality and reliability in financial risk management.

1.4 Research Content

This research is divided into 5 chapters, including introduction, literature Review, methodology, results, and conclusion and discussion.

Chapter 1 is introduction. This chapter mainly introduces the research background, research objectives and significance, research ideas and methods, research content, and innovative points of this article.

Chapter 2 is literature review. This chapter systematically explains the concept and definition of financial risk, analyzes the research status of financial risk early warning for listed companies, and explores the development of machine learning-based early warning models, providing theoretical support for the targeted selection of financial risk early warning indicators and models in the future. Additionally, it provides detailed explanations, through formula derivations and theoretical descriptions, principal component analysis (PCA) for dimensionality reduction, random forest feature selection techniques, traditional machine learning models (logistic regression, decision tree, support vector machine, BP neural network), deep learning models (convolutional neural network, bidirectional long short-term memory network, attention mechanism), ensemble learning models (random forest, XGBoost, and stacking), and the principles and applications of activation functions. Furthermore, it offers an in-depth discussion of the confusion matrix and key evaluation metrics for binary classification tasks, laying a solid theoretical foundation for constructing and optimizing financial risk early warning models.

Chapter 3 is methodology. This chapter mainly introduces research scope, Then, laying a theoretical foundation and step of methodology for the empirical analysis in the following text. Finally, draw a work flow chart and provide an example.

Chapter 4 is results. In this chapter, principal component analysis (PCA) and feature selection methods are applied to reduce the dimensionality of financial risk early warning indicators, constructing an efficient early warning indicator system. Grid search is employed to fine-tune hyperparameters and optimize the parameter combinations of individual financial early warning models, with the prediction results on the test set comprehensively evaluated using relevant metrics. Building on this, various ensemble learning models are constructed to perform vertical comparisons of single models across different time periods and horizontal comparisons between different models. Based on the evaluation metrics, the model with the best predictive performance is selected. Finally, further optimization of the activation function is conducted on the optimal predictive model to enhance its forecasting performance.

Chapter 5 is conclusion and discussion. Summarize the applicability of various machine learning models and deep learning models in financial risk warning for listed

companies, and finally propose suggestions for optimizing future financial risk warning models.

1.5 Innovation points

This research has certain innovations in indicator selection, model comparison, and construction.

(1) Indicator Selection: This research employs principal component analysis (PCA) and feature selection(Random forest) methods based on random forest to filter and compare financial risk early warning indicators. Several key indicators with significant impact are identified and evaluated for their ability to predict financial crises. This approach is innovative and provides a more diverse solution for selecting financial crisis warning indicators for listed companies.

(2) Model Construction and Comparison: This research not only conducts a horizontal comparison of traditional machine learning, deep learning, and ensemble learning algorithm models but also performs a vertical comparison of the warning effectiveness of the same model across different time periods. Additionally, multiple deep learning algorithms are integrated to construct an improved ensemble model, enhancing prediction accuracy and demonstrating innovation in model research.

(3) Create New Approach: This research customizes and optimizes activation functions to improve the learning efficiency and predictive capability of the model. By designing a novel activation function tailored to financial risk early warning, the research further enhances model performance and robustness, adding another layer of innovation to the proposed approach.

Chapter2

Literature Review

This chapter studies and analyzes the literature in the field of financial risk warning from four aspects: definition of financial risk concepts, determination of financial risk warning indicators, establishment of financial risk warning models, and statistical method theory.

2.1 Definition of Financial Risk Concept

Fitzpatrick (1932) first identified bankruptcy as a sign of a company falling into financial crisis; Altman (1968) also believed that if a company enters legal bankruptcy proceedings, it can be judged as a financially distressed company, and Ohlson (1980) also used bankruptcy as a crisis indicator for predictive analysis. Many scholars regard corporate bankruptcy as the sole indicator of financial crisis, but some scholars have enriched the definition of financial crisis based on this. Beaver (1966) pointed out that in addition to bankruptcy applications, companies with issues such as credit overdraft, bond defaults, and inability to pay dividends can also be considered financially distressed. Gordon (1971) also pointed out that only when a company's profitability drops to a certain extent and it is unable to repay its principal and interest can the company be in a state of financial distress. Ross (1999) summarized corporate financial crises into four categories: ① business failure: inability to repay debts after financial liquidation; ② Statutory bankruptcy: unable to continue operating and filing a bankruptcy application with the court; ③ Technical bankruptcy: inability to repay contract principal and interest; ④ Meeting bankruptcy: net assets less than 0 [48]. In summary, most international scholars judge financial risks of enterprises based on situations such as bankruptcy, inability to repay debts, and default.

Due to the fact that this article focuses on Chinese enterprises, their operating environment and national conditions are different from those of Western countries. Even if Chinese enterprises face bankruptcy or operational difficulties, especially large and medium-sized enterprises, they can be protected through various forms such as other capital injections, mergers and acquisitions, custodial operations, and support for development. There are few cases of corporate bankruptcy. Gu Qi (1999) proposed that corporate bankruptcy is an extreme manifestation of financial crisis, Corporate

financial crisis is an economic phenomenon in which a company is unable to repay its debts and operate normally. Therefore, using corporate bankruptcy as a symbol of financial crisis for Chinese companies is not appropriate.

In March 1998, the China Securities Regulatory Commission issued a notice on the special treatment methods for stocks of listed companies during periods of abnormal financial conditions, requiring the stock exchange to implement special treatment (ST) for the trading of stocks of listed companies with abnormal conditions. Abnormal conditions mainly refer to the negative net profits in the last two accounting years and the net assets per share being lower than the face value of the stocks in the most recent accounting year. Therefore, most Chinese scholars regard abnormal financial conditions as a sign of a company's financial crisis. Chen Jing (1999) used whether a company is an ST company as a criterion to distinguish financial risk from a normal company. Scholars such as Zhang Ling (2000), Yang Shu'e (2003), Fang Kuangnan (2016), Wang Yudong (2018), and Li S (2021) have successively adopted the ST mark as the standard for a company's financial crisis. Some scholars also use other attributes to label financially distressed companies, enriching the categories of financial status indicators. For example, Liu Yanwen (2007) classified companies into good companies, unstable companies, and financially distressed companies based on financial indicators such as return on equity, and Song (2023) used K-means clustering algorithm to classify the financial status of companies into "healthy" and "early warning".

Due to the particularity of the socialist market economy with Chinese characteristics, scholars cannot fully apply the definition of financial crisis in Chinese enterprises to the international environment. Currently, the mainstream view is still to view companies being treated as special treatment (ST) as a criterion for evaluating financial crisis or risk. Given the transparency of financial statements and annual reports of Chinese listed companies, as well as the effective supervision of the China Securities Regulatory Commission, when a listed company is listed under ST, it indicates that the company has suffered serious losses or is on the brink of bankruptcy. Therefore, this article will also use whether listed companies are marked as ST as a criterion for predicting financial risk.

2.2 Determination of financial risk warning indicators

In the early days, scholars used a single financial indicator for early warning, and Fitzpatrick (1932) first proposed bankruptcy prediction using univariate indicators such as net profit/shareholder equity for 19 companies. Based on a single indicator model, Altman (1968) established a multivariate financial risk discrimination model (Z-Score model) to warn of instability and difficulty in comprehensively and objectively reflecting operating conditions. Five financial ratio indicators, including pre tax earnings/total assets, working capital/total assets, retained earnings/total assets, stock market value/debt value, and sales revenue/total assets, were selected for bankruptcy prediction, Subsequently, numerous scholars began to integrate multiple financial indicators. Aziz (1988) constructed a financial crisis warning model for cash flow indicators, while Ciampi (2013) screened 33 financial variables and ultimately obtained 9 financial indicators that could reflect the financial status of the enterprise, but with the infiltration of research, some scholars have improved the accuracy of predictions by introducing non-financial indicators. Pätäri (2014) used corporate social responsibility and competitiveness as early warning indicators, while Doumpos (2017) incorporated international and national policy, economic, and business information into early warning analysis.

Chinese listed companies regularly disclose annual and interim reports, so most scholars predict and analyze based on five major financial indicators in the disclosed financial reports: profitability, operating ability, growth ability, solvency, and cash flow (Lv Jun, 2014; Wang Yudong, 2018; Zhan Chen, 2023). In recent years, scholars have begun to explore the early warning of non-financial indicators. Li (2021) combined quantitative financial news texts with traditional financial indicators, revealing the important role of financial news in influencing corporate financial risks. Li Chenggang et al. (2023) focused on text analysis of the management discussion and analysis module in the annual reports of listed companies. They constructed text disclosure indicators from text similarity, text sentiment value, and text readability to conduct credit risk warnings, effectively supplementing the company's financial indicator information and improving warning accuracy. Therefore, based on the research of previous scholars, this article will choose a combination of financial indicators and non-financial indicators to construct a financial risk warning indicator system.

2.3 Establishment of Financial Risk Warning Model

2.3.1 Early warning models of traditional statistical techniques

Regarding the construction of financial risk early warning models, Fitzpatrick (1932) initially selected a single financial ratio indicator to establish an early warning model. Beaver (1966) selected samples of 79 bankrupt and normal companies, and introduced mathematical statistical methods to establish a univariate discriminant model. Altman (1968) selected bankrupt and normal companies with comparable asset sizes based on Beaver's research, construct a multivariate discriminant warning model with 5 financial indicators, calculate the Z-value through weighted financial indicators to predict financial distress. Altman (1977) optimized the Z-SCORE model by increasing the original model's five variables to seven, proposed the Zeta model, and improved the prediction accuracy. Ohlson (1980) first used logistic regression models to predict financial crises, and the results were better than those of discriminant analysis models. Chinese scholars Zhou Shouhua (1996) also considered the factor of cash flow based on the Z-SCORE model and proposed a multiple discriminant F-score model. Chen Xiao and Chen Zhihong (2000) constructed a multiple logistic regression model based on financial data of Chinese listed companies, identifying that six indicators, including retained earnings/total assets, working capital/total assets, have a significant impact on the prediction results. Yang Shue and Xu Weigang (2003) introduced principal component analysis, on this basis, a multivariate discriminant Y-value warning model was constructed. Wang Xuedan (2014) screened ST samples and normal samples in a 1:1 ratio based on the principle of the same category and similar listing years. Factor analysis was used for dimensionality reduction, and a warning model was established using logistic regression, achieving good prediction results. Although it is a traditional statistical method, many scholars still use the Z-Score model (Lu Xingyu, 2021), F-Score model (Rahman, 2021), and logistic regression model (Chaiyawat, 2011; Manodamrongsat, 2021) to warn companies of financial difficulties and achieve good prediction results.

2.3.2 Early warning models of machine learning technology

With the rapid development of information technologies such as artificial intelligence, big data, and cloud computing, the application of machine learning technology in early warning models is becoming increasingly widespread. Odam (1990) first proposed the application of artificial neural network technology in the field of financial early warning, and found that its predictive performance and stability are better than Z-value models. Vapnik (1999) used Support Vector Machine (SVM) method to predict financial distress with high accuracy. Gottlieb (2006) used financial data of listed companies in the United States as the research object, constructed logistic regression, SVM, and Bayesian models, and compared the results to show that logistic regression and SVM have better effects. Chen (2011) conducted a comparative analysis of the financial difficulties of Taiwanese listed companies using decision trees and logistic regression models, and found that the prediction accuracy of decision trees is high in the short term within one year, while the prediction accuracy of logistic regression models is high in the long term over one year. Yang Qinglong (2016) used the LASSO method to effectively screen indicators of corporate financial distress, and compared the performance of decision trees, random forests, SVM, nearest neighbor methods, and the most common logistic regression models. It was found that most machine learning algorithms outperformed logistic regression methods. Li Chenyao (2023) used nine machine learning algorithms, namely Bayesian Network (GBN), Naive Bayesian Network (NBN), Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN), Bagging (BA), k-Nearest Neighbor (KNN), and Random Forest (RF), to warn financial risks of listed real estate companies. He found that GBN-TAN, ANN, and KNN had the best results. Bian (2022) used a genetic algorithm optimized SVM model to improve the accuracy by 86.5% compared to traditional SVM models, from 79.2%. It can be seen that adding some algorithms to optimize a single machine learning model can also improve prediction performance.

With the rapid evolution of computer technology and the significant improvement of computing power, deep neural networks are a further development of artificial neural networks, many innovative algorithms and model architectures have emerged in the field of deep learning. Deep neural network structures enable models

to automatically learn hierarchical feature representations, thereby better adapting to the diversity and changes in the real world. This ability has made significant progress in areas such as image recognition, natural language processing, and speech recognition, and has surpassed traditional machine learning methods in many applications. Hochreiter (1997) proposed that compared to real-time recursive learning, backpropagation over time, recursive cascade correlation, Elman networks, and neural sequence partitioning, long short term memory (LSTM) runs faster and performs better. Kuen (2015) applied deep learning to visual tracking invariant representation learning, achieving good results through strong spatiotemporal constraints and stacked convolutional autoencoder tracking for visual description. Hosaka (2019) selected financial ratios from the financial statements of four fiscal periods, attempted to automatically generate images for training and testing, constructed a convolutional neural network model for bankruptcy prediction, and compared it with multiple methods such as Z-Score, decision tree, support vector machine, linear discriminant analysis, AdaBoost, etc. It was found that convolutional neural networks have better predictive performance. Li Sha (2021) used two restricted Boltzmann machines (RBMs) and one BP neural network to construct a deep learning model, and optimized the model parameters using the Whale Algorithm. Compared with the Least Squares Support Vector Machine (LSSVM), she found better predictive results. Li (2021) used the "Bert based Chinese" pre trained language model proposed by Google to represent word vectors, and combined the advantages of bidirectional recurrent neural network (Bi RNN) and LSTM to construct the BiLSTM model. It was found that this deep learning based model can have high predictive results in four aspects of emotions: capital market, stock market, internal business conditions, and politics.

With the continuous emergence of various models, there are certain advantages and disadvantages of models under certain conditions. Ensemble learning constructs a powerful model by combining multiple weak models to compensate for the shortcomings of each model and improve overall performance and generalization ability. Common methods for ensemble learning include Bagging, Boosting, Stacking, and so on. Liang Mingjiang (2012) used the AdaBoosting ensemble learning method with support vector machine as the base classifier, which has higher prediction

accuracy and stability than a single base classifier. Cao Wei (2018) used the Majority Voting method in ensemble learning Bagging to integrate five models: BPNN, SVM, KNN, LOG, and MDA, and found that the prediction results of the integrated model were better than those of a single model. Choi (2018) established a voting based ensemble model using six models: support vector machine (SVM), artificial neural network (ANN), decision tree (C4.5), naive Bayes (NB), LR, and k-nearest neighbor (KNN). The predictive performance of the model is higher than that of a single model. Chen Yufang (2022) addressed the issue of excessive indicators by using machine learning methods such as Lasso method, Mann Whitney nonparametric method, and random forest to screen indicators. To address the imbalance in the proportion of positive and negative samples, the Smote algorithm was used to solve the problem. A single financial fraud prediction model was constructed by constructing machine learning algorithms such as logistic regression, decision tree, and XGBoost, and a fusion model was constructed using the fusion method of Voting and Stacking, It was found that the fused model has better recognition ability. Abror (2023) used a genetic algorithm support vector machine feature selection algorithm to reduce the attributes of the dataset and trained it using a stacking method, which included multiple single classifier base learners such as k-nearest neighbors, naive Bayes, decision trees with classification and regression tree models, gradient boosting decision trees, and light gradient boosting, with an accuracy rate of up to 99.22%.In addition, in the field of ensemble models in deep learning, Ouyang (2021) developed a risk warning model based on an Attention-LSTM neural network and found that, compared to the BP neural network, SVM, and ARIMA models, the Attention-LSTM neural network achieved higher average prediction accuracy across short, medium, and long-term forecasts. Kavianpour(2023) integrated CNN, BiLSTM, attention mechanism, and ZOH preprocessing to predict earthquake magnitude and frequency for the next month. Testing on data from nine regions in China, the method outperformed SVM, MLP, DT, RF, CNN, LSTM, and CNN-BiLSTM in performance and generalization.

In summary, scholars have gone through an evolutionary process from traditional statistical techniques to modern machine learning techniques in the field of financial risk warning. They have achieved different results by establishing various financial crisis warning models and creating composite ensemble models, but the practical

significance of these models varies. In addition, research on sample selection, indicator selection, time span, and algorithms for constructing composite ensemble models still requires continuous exploration and improvement.

2.3.3 Early warning models of optimization algorithm technology

Optimization algorithms are integral to the advancement of early warning models, significantly improving their predictive accuracy, generalization capabilities, and adaptability to complex and imbalanced data. The correct selection and application of optimization algorithms directly affect the convergence speed and final performance of deep learning models. Selecting appropriate optimization algorithms based on task characteristics and model structure is a key step in deep learning development. Among the prominent techniques, activation function optimization and hyperparameter tuning stand out for their contributions to the development of robust and efficient models.

The activation function is a crucial part of deep learning, which has a profound impact on the nonlinear expression ability and gradient propagation efficiency of neural networks. In the early days, the Sigmoid activation function proposed by Rumelhart et al. (1986) was widely used due to its limitation of output values within the range of (0,1), but there were issues with non-zero centers and vanishing gradients. Subsequently, LeCun et al. (1998) introduced the Tanh activation function and adjusted the output range to (-1,1), partially alleviating the non-zero center problem but still limited by gradient vanishing. To address these issues, Nair et al. (2010) proposed the ReLU activation function, which has a constant gradient in the positive range, significantly improving the training efficiency of deep networks. However, it remains constant at zero in the negative range, which may lead to neuronal "death". In response, researchers have proposed various improvement schemes, such as Maas et al.'s (2013) Leaky ReLU and Parametric ReLU, which alleviate the problem of neuronal "death" by introducing negative slope values. The Swish activation function proposed by Ramachandran et al. (2017) optimizes gradient propagation through smooth non-monotonic properties. The Mish activation function proposed by Misra (2020), combined with Tanh and SoftPlus features, further enhances generalization performance. In addition, scholars have also explored activation function optimization for specific application scenarios, such as SPReLU

proposed by Wu Tingting (2022), which combines the advantages of ReLU, PReLU, and SoftPlus to significantly improve model performance. The LUTanh proposed by Shidik (2021) combines ReLU and Tanh, demonstrating strong robustness in tasks such as earthquake prediction. Jagtap et al. (2020) dynamically adjusted the topology of the loss function during the optimization process by introducing extensible hyperparameters into the activation function to form an adaptive activation function. The evolution of these improved activation functions has evolved from a single form to a parallel trend of dynamic adjustment and fusion strategies, providing strong support for the widespread application of deep learning in fields such as finance and healthcare by enhancing the model's feature extraction ability and generalization performance.

Hyperparameter tuning is a crucial aspect of deep learning optimization, directly influencing model architecture, training efficiency, and performance. Early traditional methods such as grid search and random search were commonly used for hyperparameter optimization. Bergstra and Bengio (2012) demonstrated that random search outperforms grid search in high-dimensional hyperparameter spaces, marking a significant improvement in model optimization strategies. However, as models became more complex, these methods showed inefficiency in exploring large search spaces. To address this, Snoek et al. (2012) introduced Bayesian optimization, which constructs surrogate models to predict performance and guide the search, significantly improving the efficiency of hyperparameter tuning in complex models. Building on this, Hyperband, introduced by Li et al. (2018), further advanced the field by combining a multi-armed bandit framework with dynamic resource allocation. Hyperband optimizes computational resource usage, enabling faster identification of high-performing configurations, especially in deep learning models with large hyperparameter spaces. Hyperband has been particularly successful in domains where computational resources are limited. For example, Falkner et al. (2018) combined Bayesian optimization with Hyperband to develop the BOHB algorithm, improving optimization efficiency by balancing exploration and exploitation. This method allows for more efficient hyperparameter optimization, especially for large-scale deep learning models, by allocating more resources to promising configurations while quickly discarding less effective ones. In financial forecasting, Chen and Liu (2021)

applied the Hyperband algorithm to optimize LSTM hyperparameters for stock price prediction. Their results showed that Hyperband outperformed Bayesian optimization, providing more stable and accurate predictions, thus offering a robust approach for dynamic financial data modeling. Similarly, in industrial diagnostics and healthcare, hyperparameter optimization techniques, such as those implemented by Hyperband, have been shown to improve performance in fault detection and disease prediction tasks. In these fields, optimizing parameters like learning rates and network architecture has proven effective for adapting to complex and diverse data distributions, enhancing model robustness. These advancements demonstrate that hyperparameter optimization not only enhances deep learning model performance but also accelerates model development by reducing tuning time and computational costs. Hyperband and other efficient optimization techniques have become essential tools in fields such as finance, healthcare, and industrial applications, where the complexity of data and models requires fast, resource-efficient tuning methods.

Optimization algorithms are crucial for improving the performance of deep learning models. While activation function optimization enhances the model's ability to express nonlinear relationships, hyperparameter optimization focuses on fine-tuning the overall parameter configuration. Together, they work towards improving both prediction accuracy and generalization capability. Therefore, this paper combines ReLU, Sigmoid, and Tanh custom activation functions with the Hyperband algorithm to optimize model performance, demonstrating their synergistic effect in enhancing the efficiency and effectiveness of deep learning models.

2.4 Statistical method theory

This section will introduce the multicollinearity detection method (VIF), principal component analysis (PCA) for data dimensionality reduction, feature selection based on decision tree and Lasso in machine learning, logistic regression model, decision tree model, SVM, BP neural network, and convolutional neural network model in deep learning algorithms used in the data processing stage of empirical analysis, And provide a method overview and content introduction for the random forest model in Bagging algorithm, XGBoost model in Boosting algorithm, and hierarchical model in Stacking algorithm among the three ensemble learning

algorithms, as well as the main evaluation indicators for machine learning binary classification models.

2.4.1 Data Preprocessing methods

2.4.1.1 Smote Oversampling

SMOTE (Synthetic Minority Oversampling Technique) was proposed by Chawla in 2002. It is a technique that uses synthetic minority oversampling to balance class distribution by adding minority samples, reducing the risk of overfitting while improving the model's generalization ability. Specific implementation steps:

(1) Randomly select minority class samples: Select sample x_i from the minority class samples as the benchmark sample for synthesizing new samples, calculate the Euclidean distance between this sample x_i and other minority class samples, and obtain the K nearest neighbors of sample x_i.

(2) Randomly select neighboring samples: Set a sampling rate of N % based on the imbalanced proportion of the samples, randomly select x_{ij} as the sample from the selected K nearest neighbors, and repeat the operation N times to obtain samples x_{11}, x_{12}, ..., x_{KN} .

(3) Composite new sample: Linear interpolation is performed between the benchmark sample x_i and the nearest neighbor sample x_{ij} to construct a new sample, generating a new sample

$$g_i = x_i + (x_{ij} - x_i). \qquad (2.1)$$

2.4.2 Feature selection methods

Principal component analysis (PCA) and random forest feature selection in machine learning are all related to dimensionality reduction to some extent. PCA focuses on maximizing the variance of the data by finding the most important principal components (linear combinations) in the data to reduce its dimensionality, while random forest feature selection focus more on selecting the features that make the greatest contribution to the prediction target, indirectly achieving dimensionality reduction effects.

2.4.2.1 Principal Component Analysis

Principal component analysis (PCA) is a data dimensionality reduction technique proposed by Pearson in 1901. Its goal is to find the principal components in the data and project the original data onto a new coordinate system, thereby achieving dimensionality reduction and preserving the main information of the data. The principal component is a linear combination of the original variables, arranged in descending order of variance. The specific implementation steps are as follows:

(1) Assuming there are n companies, each with m observation indicators, forming a matrix:

$$X = (\mathrm{x_{ij}})_{m\times n}. \tag{2.2}$$

(2) Standardized data: Normalize the original data to obtain a standardized matrix:

$$Z = (\mathrm{z_{ij}})_{m\times n}, \tag{2.3}$$

$$\mathrm{z_{ij}} = \frac{\mathrm{x_{ij}} - \overline{\mathrm{x}}_{\mathrm{j}}}{\mathrm{s_j}}. \tag{2.4}$$

(3) Calculate covariance matrix: Calculate the covariance matrix of the standardized data, using the following formula:

$$R = (\mathrm{r_{ij}})_{m\times n}\,, \tag{2.5}$$

$$\mathrm{r_{ij}} = \frac{1}{n-1}\sum_{i=1}^{n} z_{ij}\, z_{ik}\,. \tag{2.6}$$

(4) Calculate eigenvalues and eigenvectors: Perform eigenvalue decomposition on the covariance matrix to obtain eigenvalues and corresponding eigenvector eigenvectors:

$$\mathrm{u}_j = (u_{1j}, u_{2j}, \ldots, u_{mj})^T. \tag{2.7}$$

(5) Select Principal Component: Arrange the eigenvalues in descending order, and select the eigenvectors corresponding to the first k eigenvalues as the principal components :

$$y_j = u_{1j}z_1 + u_{2j}z_2 + \ldots + u_{mj}z_m. \tag{2.8}$$

The cumulative contribution rate is

$$\alpha = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda_i} y_j, \tag{2.9}$$

where k is the number of selected principal components.

2.4.2.2 Random Forest Feature Selection

The basic idea of random forest feature selection is to use the splitting process of each decision tree in the random forest model to evaluate the importance of features, proposed by Leo Breiman and Adele Cutler in 2001. By calculating the contribution of each feature in different trees, the impact of that feature on the model's prediction results can be measured, and the most useful feature can be selected.The specific calculation formula is as follows:

(1) Gini Importance

Gini importance measures the contribution of each feature to reducing node impurity in a tree model:

For a certain feature in a decision tree j, gini impurity is used to measure the impurity of nodes,

$$\mathrm{I(t)} = 1 - \sum_{k=1}^{K} p_k^2, \tag{2.10}$$

where p_k is the proportion of samples belonging to category k on node t.

Feature importance score: On node t, calculate the Gini importance GI (j) for feature j,

$$\mathrm{GI(T)} = \sum_{t \in splits\ \mathrm{on\ j}} \left(\frac{N_t}{N}\right) \cdot \left(I(t) - \frac{N_{tL}}{N} \cdot I(t_L) - \frac{N_{tR}}{N} \cdot I(t_R)\right), \tag{2.11}$$

where N_t is the number of samples on node t, N_{tL} and N_{tR} are the sample sizes of the left and right child nodes, respectively.$I(t)$ is the Gini impurity of node t.

(2) Feature Importance in Random Forest

In a random forest, the formula for calculating the importance of feature j is,

$$\mathrm{FI(t)} = \frac{1}{\mathrm{T}} \sum_{m=1}^{T} GI_m(\mathrm{j}), \tag{2.12}$$

where T is the number of trees in the forest, and $GI_m(j)$ is the Gini importance of feature j in the mth tree.

2.4.4 Machine learning models

This section mainly introduces four single models that are frequently used in machine learning models, namely, logical regression model, decision tree, support vector machine and BP neural network model, and describes the calculation formulas and theoretical points involved.

2.4.4.1 Logistic regression model

Logistic Regression was proposed by statistician David Cox in 1958 and has had a significant impact on the development of machine learning. Logistic regression is a linear single model used to perform classification tasks. It belongs to supervised learning in machine learning. Binary regression and multiple out of order regression are subdivisions of this model. Its name includes "regression" because its principle is similar to that of linear regression model. However, the application of logistic regression tends to handle classification problems rather than regression problems, with a focus on solving binary classification problems, that is, mapping input features to output as one of two categories. For multi class problems, it can be extended to polynomial logistic regression. Many scholars have used binary logistic regression in enterprise financial risk warning models.

Logistic regression maps the results of linear regression to between 0 and 1 using the sigmoid function, thereby obtaining the probability that the sample points belong to a certain category. The following is the calculation formula for the logistic regression model:

The linear regression form of the explanatory variable is as follows:

$$Z = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n, \tag{2.15}$$

where Z is the predicted value of linear regression, $\beta_0, \beta_1, \ldots, \beta_n$ is the coefficient to be estimated, and $x_1, x_2, \ldots, x_n$ is the explanatory variable.

(2) For the binary classification problem, the response variable $Y \in \{0,1\}$ is used to map the predicted value Z to [0,1] through the Sigmoid function, which is:

$$p = g(Z) = \frac{1}{1+e^{-z}}. \tag{2.16}$$

(3) Generate predicted response variable Y based on calculated probability p:

$$Y = \begin{cases} 1, p \geq 0.5 \\ 0, p < 0.5 \end{cases}. \tag{2.17}$$

Next, the maximum likelihood method can be used to solve the estimated coefficient β_i.

The advantages of logistic regression models are: (a) for binary classification problems, they generally perform well, are easy to implement, and have a fast speed; (b) The output probability can be interpreted as the occurrence of events under given

characteristic conditions, with good interpretability and practical significance; (c) Not requiring too much preprocessing, it can directly process various types of features.

2.4.4.2 Decision Tree Model

There are three important algorithms for the principle of decision trees. Breiman (1984) proposed the CART (Classification and Regression Trees) algorithm, which consists of feature selection, tree generation, and pruning. Quinlan (1986) proposed the ID3 (Iterative Dichotomizer 3) algorithm, which uses "information gain" to select partition attributes and can only handle discrete attributes. Quinlan (1993) further optimized it and proposed the C4.5 algorithm, using "gain rate" to select partitioning attributes can handle both continuous attributes and attributes with missing values. These algorithms can be used for both classification and regression.

This research will use the CART algorithm and Gini Index as an indicator to measure the purity of the dataset. When dealing with classification problems, the corresponding decision tree is called a classification tree. The Gini coefficient is used to measure the degree of mixing of samples in a dataset, that is, the probability of randomly selecting two samples from the dataset with different categories. The smaller the Gini coefficient, the higher the purity of the dataset, and the more likely the samples are to belong to the same category. The better the classification effect based on this feature. Assuming there are k categories of samples, Gini (T) is the Gini coefficient of dataset T, where the probability of a sample belonging to the i-th category is. The specific calculation formula is as follows:

$$\text{Gini(T)} = \sum_{i=1}^{k} p_i \ (1 - p_i) \ . \qquad (2.18)$$

In the process of constructing a decision tree, when selecting the optimal partitioning feature, the CART algorithm calculates the Gini coefficient of each feature and selects the feature with the smallest Gini coefficient for partitioning. The smaller the weighted sum of the Gini coefficients after partitioning, the better the partitioning effect. The process of constructing a classification tree is to recursively select the optimal features and partition points, continuously partition the dataset until the stop condition is reached. When the stop condition is met, a leaf node is generated, and its category is marked as the category with the highest number of samples in the dataset. The entire construction process will form a tree like structure, where each

internal node represents a feature and its partitioning points, and each leaf node represents a category, ultimately generating a CART classification tree.

The advantages of decision trees are: (a) they can be visualized and the generated prediction rules are easy to interpret; (b) The prediction process of decision trees is very efficient, requiring only a series of comparison operations along the path of the tree; (c) Decision trees are not sensitive to feature scaling and do not require normalization or normalization of input features.

2.4.4.3 Support Vector Machine Model

Support Vector Machine (SVM) is a machine learning model widely used in classification, regression, and anomaly detection. Cortes (1995) proposed how to use linear SVM for classification and introduced the concept of support vector networks. It has advantages in dealing with problems such as small samples, nonlinearity, and high dimensionality, especially suitable for classification problems of small and medium-sized complex datasets. The basic idea of SVM is to construct one or more hyperplanes in high-dimensional or infinite dimensional spaces, so that these hyperplanes can separate samples of different categories and ensure maximum separation intervals. In binary classification problems, the main objective is to find an optimal hyperplane, divide the dataset into two categories, so that the closest sample point in both categories has the maximum distance from the hyperplane, that is, to maximize the interval, while ensuring the accuracy of classification.

Linear Separable Case: For linearly separable datasets, SVM attempts to find an optimal hyperplane that maximizes the sum of distances from the closest point (called support vector) to the hyperplane in two types of samples. The objective function of SVM can be formalized as:

$$g(Z) = min_{w,b} \frac{1}{2} ||W||^2, \quad (2.19)$$

where W is the normal vector of the hyperplane, and b is the bias term. By minimizing this objective function, the hyperplane found correctly categorizes the samples into different categories and maximizes the interval.

The SVM decision function can be expressed as:

$$f(x) = \mathrm{sgn}(w^T \cdot x + b) , \quad (2.20)$$

where x is the input feature vector,sgn() is a sign function that returns the sign (+1 or -1) of its parameters.

(2) Non linearly separable case: For non linearly separable datasets, SVM maps the data from low dimensions to higher dimensions by introducing Kernel Function, making the sample data linearly separable and searching for the optimal hyperplane in this high-dimensional space. The commonly used kernel functions include linear kernel, polynomial kernel, Gaussian radial basis function (RBF) kernel, etc. The objective function of SVM can be formalized as:

$$\min_{w,b,\xi} \frac{1}{2}||W||^2 + C\sum_{i=1}^{n} \xi_i \,, \tag{2.21}$$

where ξ_i represents the relaxation variable of the i-th sample, and C is a regularization parameter used to balance the two objectives of minimizing interval and minimizing misclassified samples. This function simultaneously seeks to minimize the interval, minimize misclassified samples, and minimize the sum of slack variables in optimization. In the optimization process, the model will try to find a hyperplane as much as possible, so that most of the samples are on both sides of the hyperplane, and allow some samples to appear within the interval or on the misclassified side.

The SVM decision function can be expressed as:

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{n} \alpha_i \, y_i K(x, x_i) + b) \,, \tag{2.22}$$

The advantages of SVM are: (a) SVM can efficiently classify in high-dimensional space, suitable for processing data with a large number of features; (b) By using kernel functions, support vector machines can flexibly handle nonlinear relationships; (c) By maximizing the interval principle, the model has good generalization ability on data outside the training set, and has a significant anti overfitting effect.

2.4.4.4 BP neural network model

The BP neural network model was first proposed by Rumelhart (1986), focusing on the principle and application of error backpropagation algorithm. It consists of input layer, hidden layer (which can have multiple layers), and output layer, and is learned and predicted through the process of forward and backward propagation. Each layer of neuron state only affects the next layer of neuron state, making it one of the most widely used neural network models.

(1) Forward propagation: This article focuses on financial risk warning, so if it is ST company, the output value is 1, otherwise it is 0. Assuming there are n inputs and 2 outputs in the network, and s neurons in the hidden layer.

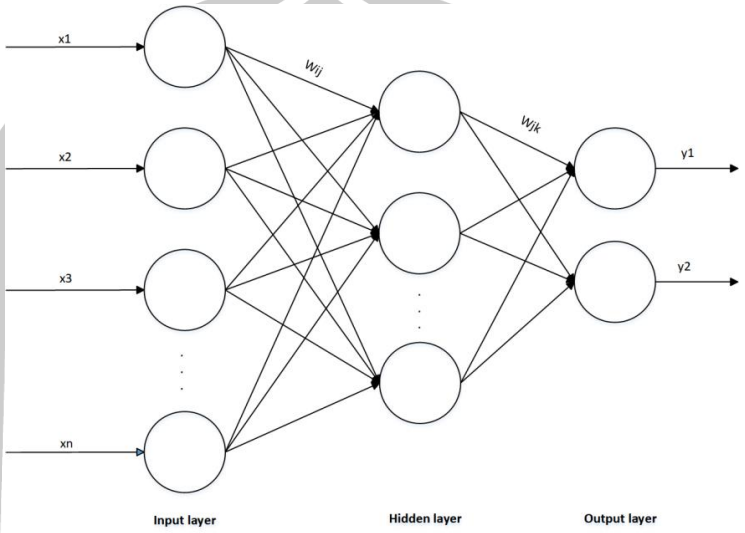


Figure 1 BP neural network structure diagram

Calculation formula for input layer to hidden layer:

$$z_j = f_1(\sum_{i=1}^{n} w_{ij} x_i - b_j) , \quad (2.23)$$

where f_1 is the activation function of the hidden layer (Sigmoid function or ReLu function), w_{ij} is the weight from the input layer to the hidden layer, x_i is the input value, and b_j is the threshold of the hidden layer j neuron.

Calculation formula from hidden layer to output layer:

$$y_k = f_2(\sum_{j=1}^{2} w_{jk} z_j - b_k) , \quad (2.24)$$

where f_2 is the activation function of the output layer, w_{jk} is the weight from the hidden layer to the output layer, y_k is the output value, and b_k is the threshold of the output layer.

Calculate the loss between the predicted output and the actual label. The common mean square error loss function is:

$$J = \frac{1}{2}\sum_{k=1}^{2}(y_k - t_k)^2 , \quad (2.25)$$

where y_k is the output value and t_k is the actual label of the sample.

(2) Backward propagation: Starting from the output layer, adjust the connection weights layer by layer in the direction of reducing the error between the expected

output and the actual output. Adjust the weights and biases of each neuron in the neural network to minimize the error between the network output and the actual labels.

Calculate the error term (gradient) of the output layer neurons:

$$\delta_{\mathrm{k}} = y_k - t_k \,. \tag{2.26}$$

Calculate the error term for hidden layer neurons:

$$\delta_{\mathrm{j}} = \mathrm{f}'(\mathrm{z_j}) \sum_{j=1}^{n} w_{jk}\, \delta_{\mathrm{k}} \,. \tag{2.27}$$

Update the weights and thresholds of the output layer:

$$w_{jk} = w_{jk} - \alpha \delta_{\mathrm{k}} z_j \,, \tag{2.28}$$

$$b_k = b_k - \alpha \delta_{\mathrm{k}} \,. \tag{2.29}$$

Update the weights and thresholds of hidden layers:

$$w_{jk} = w_{ij} - \alpha \delta_{\mathrm{j}} x_i \,, \tag{2.30}$$

$$b_j = b_j - \alpha \delta_{\mathrm{j}} \,, \tag{2.31}$$

where α is the learning rate, which is used to control the step size of parameter updates.

(3) Iterative optimization: Repeat the process of forward and backward propagation, continuously adjusting weights until a certain stopping criterion is met (such as an error less than the set value or an iteration reaching the set upper limit).

The advantages of the BP neural network model include: (a) the ability to learn and approximate any complex nonlinear relationship through activation functions and multi-layer network structures; (b) Automatically adjust weights and thresholds during the training process to adapt to different input data, with strong self-learning and adaptive abilities; (c) The damage of individual neurons or local noise in input data will not have a serious impact on the overall performance of the network, therefore it has a certain degree of fault tolerance and robustness.

2.4.5 Deep learning models

Deep learning models are a type of machine learning model based on Artificial Neural Networks (ANN), characterized by a deep structure that includes multiple layers of neural networks. Through multi-level nonlinear transformations, they can automatically learn complex features and representations in input data. Therefore, compared to traditional machine learning algorithms, deep learning performs better in

processing large-scale, high-dimensional data and solving complex tasks. The current deep learning models mainly include: deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory networks (LSTM), autoencoders, generative adversarial networks (GAN), etc. This article will use Convolutional Neural Networks (CNN) to effectively process financial data, extract relevant features, and achieve risk prediction and warning.

2.4.5.1 Convolutional Neural Network Model

LeCun (1998) applied convolutional neural networks to document recognition in the early stages, combining convolutional neural network character recognizers with global training techniques to improve the accuracy of recording commercial and personal checks. The core of CNN in financial risk warning is convolution operation, which uses a set of convolution kernels to perform sliding convolution on financial data to extract local features of the data. Through convolution operations, the model can locally perceive features in financial data, such as certain trends or fluctuations, without the need to globally consider the entire time series. This helps CNN better adapt to possible financial risks that may arise during different time periods. The basic structure of convolutional neural networks is shown in Figure 2.

Figure 2 CNN structure diagram

(1) Convolutional layer

The convolutional layer extracts features by convolving the convolutional kernel with input data. Through a series of convolution operations, these abstract feature representations can be used for subsequent tasks such as classification, regression, and object detection. The specific calculation formula for convolution operation is as follows:

$$z_{ij} = \sum_{m=1}^{M} \sum_{n=1}^{N} x_{i+m-1,j+n-1} \cdot w_{m,n} + \mathrm{b}) \,, \tag{2.32}$$

where z_{ij} is the output after convolution, $x_{i+m-1,j+n-1}$ is the input data, w_{mn} is the weight of the convolution kernel, and b is bias.

(2) Activation function

Applying activation functions to perform nonlinear mapping on the results of convolution operations significantly improves the expressive power of neural networks, enabling them to approximate any complex nonlinear function. The specific formula is as follows:

$$h_{ij} = f(z_{ij}) , \tag{2.33}$$

where f is the activation function, usually using ReLU (Corrected Linear Unit).

(3) Pooling layer

The pooling layer downsamples the output of the convolutional layer to reduce the dimensionality of the data, and performs pooling operations such as Max Pooling and Average Pooling. This article adopts max pooling, taking the maximum value in each pooling window. Maximum pooling is the process of dividing input data into non overlapping rectangular regions, selecting the maximum value within each rectangular region as the representative value for that region, and finally retaining the most prominent features within each region, i.e. outputting the pooled result. The specific formula is as follows:

$$p_{ij} = max(h_{2i,2j}, h_{2i,2j+1}, h_{2i+1,2j}, h_{2i+1,2j+1}) . \tag{2.34}$$

(4) Fully connected layer

Flatten the output of the pooling layer and connect it to the fully connected layer, where each node is connected to each node in the previous layer. The weights of each connection are trained and optimized using backpropagation algorithms, and comprehensive feature learning is performed through the fully connected layer to capture higher-level features. The specific formula is as follows:

$$y_k = \sigma(\sum_{i=1}^{N} w_{ki} \cdot p_i + b_k) , \tag{2.35}$$

where w_{ki} is the weight of the fully connected layer, p_i is the output of the last pooling layer, b_k is the bias, and σ is the activation function, usually using softmax or sigmoid.

(5) Loss function

The function of the loss function is to measure the difference between the predicted results of the model and the actual labels, providing an optimization direction for the model. By adjusting the parameters of the model, the value of the

loss function is minimized, which means that the model approximates or predicts actual data more accurately. This article adopts the commonly used cross entropy loss function for classification tasks, and the specific formula is as follows:

$$J(Y,T) = -(\sum_{i=1}^{C} T_i \log(Y_i)) \,, \tag{2.36}$$

where T is the actual label and Y is the predicted output result.

(6) Backpropagation optimizer

The optimizer updates the model parameters through backpropagation algorithm, calculates gradients based on the loss function, and updates the network parameters (convolutional kernel weights, fully connected layer weights, etc.) using gradient descent, Adam, Adagrad, and other methods.

The advantages of convolutional neural networks include: (a) effectively extracting local features from input data, reducing the number of model parameters through weight sharing mechanism, making it more effective in processing images, sequences, and other data, reducing the risk of overfitting, and improving the model's generalization ability; (b) Automatically learn useful feature representations from input data without the need for manual feature extractor design; (c) Convolutional and pooling operations have high parallelism and computational efficiency.

2.4.5.2 Bidirectional Long Short-Term Memory Network Model

Bidirectional Long Short-Term Memory Network(BiLSTM) was proposed by Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins in 1997 as an extension of the Long Short-Term Memory Time Network (LSTM), BiLSTM combines two LSTM networks that process sequences in a forward manner and in a reverse manner, respectively, and this bidirectional architecture Capable of capturing the backward and forward dependencies of time series data well, BiLSTM is able to show better performance in many tasks by learning data from the past to the future and never to the past.

The LSTM unit is composed of input gates, output gates, forgetting gates and memory units that can process and preserve the long-term dependencies of the time series.

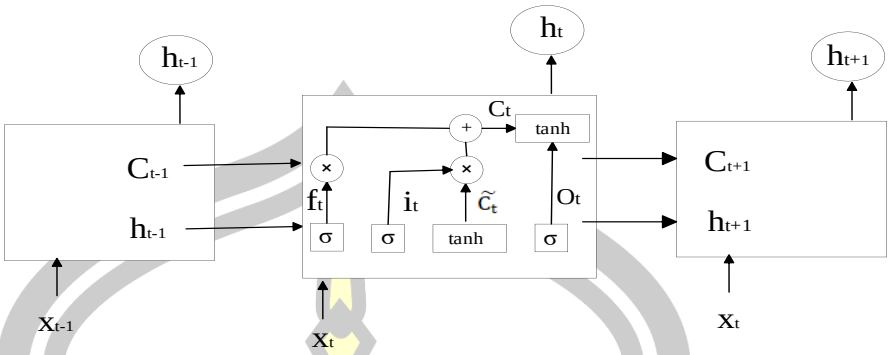


Figure 3 LSTM structure diagram

(1) Forget Gate:The forget gate determines how much of the previous memory cell C_{t-1} should be retained or discarded. It selectively forgets irrelevant information from past time steps, enabling the network to focus on important patterns.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (2.37)$$

where f_t is forget gate output, values range between [0,1];W_f,b_f represent weight matrix and bias vector of the forget gate;$h_{t-1}$is hidden state from the previous time step;x_t is current input;σ is sigmoid activation function, which outputs probabilities.

(2) Input Gate:The input gate decides which parts of the current input x_t are useful for updating the memory cell. It acts as a filter to allow relevant information into the memory.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2.38)$$

where i_t is input gate output, values range between [0,1];W_i,x_t represent weight matrix and bias vector of the input gate;σ is sigmoid activation function, which outputs probabilities.

(3) Candidate Memory Cell:The candidate memory cell $\tilde{C}_t$ computes potential new information to add to the memory cell. This step synthesizes information from the current input $x_t$and the hidden state h_{t-1}, creating a candidate update for the memory.

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \quad (2.39)$$

where $\tilde{C}_t$ is candidate memory values, range between [-1,1];W_c,b_c represent weight matrix and bias vector of the candidate memory cell;$tanh$is Hyperbolic tangent activation function.

(4) Update Memory Cell:The memory cell $C_t$is updated by combining the results of the forget gate and input gate. The forget gate removes irrelevant parts of C_{t-1}, while the input gate determines the amount of new information to be added from $\widetilde{C_t}$.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t\,, \tag{2.40}$$

where C_t is updated memory cell state at time t.

(5) Output Gate:The output gate controls how much of the updated memory cell C_t contributes to the hidden state h_t, which will be used as output and passed to the next time step. It ensures that only the most relevant information is passed forward.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{2.41}$$

where $o_t$is output gate values, range between [0,1];W_o,b_o represent weight matrix and bias vector of the output gate;σ is sigmoid activation function.

(6) Compute the Hidden State:The hidden state h_t is generated using the memory cell C_t and the output gate o_t. It represents the network's interpretation of the current input in the context of past information and is passed to the next time step.

$$h_t = o_t * tanh(C_t), \tag{2.42}$$

where h_t is hidden state at the current time step, used for output or passed to the next time step;tanh is hyperbolic tangent activation function,ensuring smooth and bounded outputs.

BiLSTM combines two LSTM networks to comprehensively capture the dependencies of the sequence data by processing the sequences in both forward and reverse directions.

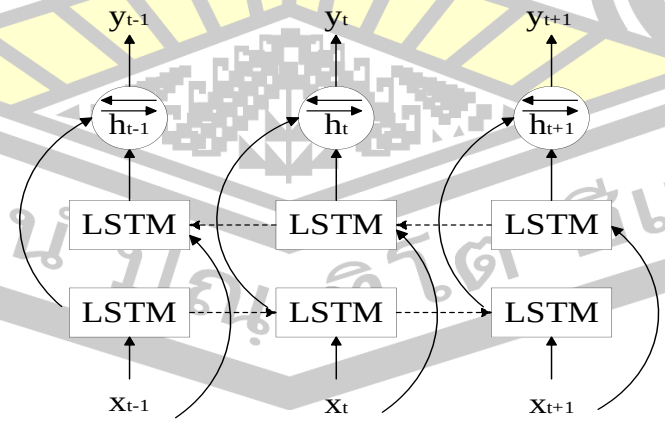


Figure 4 BiLSTM structure diagram

(1) Positive LSTM processing sequence, where the input sequence is processed step by step from the first time step to the last time step of the time sequence, and for each time step t, the positive hidden layer state $\overrightarrow{h_t}$ is computed according to the positive LSTM formula.

$$\overrightarrow{h_t} = LSTM\left(x_t, \overrightarrow{h_{t-1}}, \overrightarrow{C_{t-1}}\right). \tag{2.43}$$

(2) The reverse LSTM processes the sequence, processing the input sequence step by step from the last time step to the first time step of the time sequence, and for each time step t, the reverse hidden layer state $\overleftarrow{h_t}$ is computed according to the reverse LSTM formula.

$$\overleftarrow{h_t} = LSTM(x_t, \overleftarrow{h_{t+1}}, \overleftarrow{C_{t+1}}). \tag{2.44}$$

(3)Connect the forward and backward hidden states, and for each time step t, connect the forward hidden layer state and the backward hidden layer state, and pass the connected hidden state through the fully connected layer to get the final output y_t.

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}], \tag{2.45}$$

$$y_t = \sigma(W \cdot h_t + b). \tag{2.46}$$

2.4.5.3 Attention Mechanism

Attention Mechanism (AT) was proposed by Bahdanau et al. in 2014, and was initially intended to be used mainly to improve the performance of machine translation by allowing the model to dynamically select and learn the relevant information in the original sentence as it translates each word, instead of relying on a fixed context vector. Unlike BiLSTM, the attention mechanism not only focuses on the input of the current time step when processing each time step of the input sequence, but also adjusts its weights based on the information of other time steps, so as to better capture the important information in the input sequence. In this paper, we use the implementation of a customized layer based on Additive Attention.

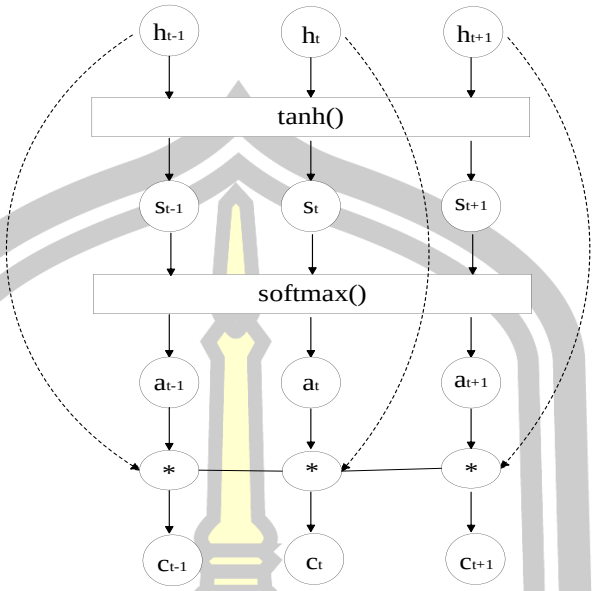


Figure 5 AT structure diagram

(1)Calculate the attention score, for each time step t, the attention score s_t is calculated via the tanh activation function.

$$s_t = tanh(W \cdot h_t + b). \tag{2.47}$$

(2)Calculate the attention weights, for each time step t, calculate the attention weights a_t by softmax function, these weights represent the importance of each time step.

$$a_t = \frac{exp(s_t)}{\sum_{k=1}^{T} exp(s_k)}. \tag{2.48}$$

(3) Calculate the weighted summation, based on the calculated attention weights a_t, the input sequence h_t is weighted and summed to obtain the context vector c_t of the output of the current time step.

$$c_t = \sum_{i=1}^{T} a_t \cdot h_t. \tag{2.49}$$

2.4.6 Ensemble Learning Models

Ensemble learning is a method of improving overall model performance by combining the prediction results of multiple weak learners. This strong model can take the length of all base learners and achieve relative optimal performance. These base learners can be of the same type or different types, presenting a "diverse yet different" characteristic. The main ensemble learning algorithms include three categories: Bagging, Boosting, and Stacking. In this paper, we propose a random

forest model based on the Bagging ensemble framework, an XGBoost model based on the Boosting ensemble framework, and a hierarchical model based on the Stacking ensemble framework.

2.4.6.1 Bagging Algorithm - Random Forest Model

The Bagging algorithm was proposed by Breiman in 1996, which randomly selects sub samples from the training set with dropouts. Each base learner is trained on different sub samples, and the prediction results of each sub learner are voted or averaged to obtain the final ensemble model.

Random forest is one of the most practical algorithms in bagging ensemble learning. Breiman (2001) was the first to propose random feature selection and the use of Bootstrap method to construct multiple decision trees. The results of each tree are integrated through voting (classification problem) or averaging (regression problem), resulting in high performance and robustness of the entire model. The specific implementation steps are as follows:

(1) Random sampling: Use the Bootstrap method to randomly select k sub training sets with replacement from the original training set.

(2) Feature random selection: For each node in each decision tree, a portion of the features are randomly selected from all the features during the training process for node splitting, and the selected features are used as candidate feature sets.

(3) Build a decision tree using decision tree algorithms (such as CART) based on candidate feature sets. During the construction process, certain splitting rules and evaluation indicators (such as Gini impurity or information gain) are adopted.

(4) Repeat steps 2 and 3 to construct multiple decision trees and form a random forest.

(5) Voting: The financial warning in this article is a classification problem. The prediction results of multiple decision trees are voted on to determine the final category. The category with the most votes is used as the final prediction result. The specific formula is as follows:

$$Y = \text{mode}(T_1(\text{x}), T_2(\text{x}), \ldots, T_n(\text{x})) \,, \tag{2.50}$$

where Y is the final prediction result, $T_i(x)$ is the prediction result of the i-th decision tree, and mode is the mode taken.

The advantages of the random forest model: (a) By integrating the prediction results of multiple decision trees, it can effectively reduce the risk of overfitting of a single model and improve the overall prediction accuracy; (b) Capable of processing datasets with high-dimensional features and performing well in high-dimensional spaces; (c) Being able to provide the contribution of each feature to model performance helps to understand the data and model.

2.4.6.2 Boosting Algorithm - XGBoost Model

The Boosting algorithm was first proposed by Schapire (1990), which iteratively trains base learners. Each base learner's training focuses on the errors of previous learners, aiming to correct errors and improve overall performance. AdaBoost and Gradient Boosting Machines (GBM) are common boosting algorithms that improve model performance by combining weighted base learners. XGBoost and LightGBM are both Boosting algorithms based on GBM optimization and improvement. This article uses the XGBoost algorithm.

XGBoost (Extreme Gradient Boosting) was proposed by Chen (2016). It is an improved version of the Gradient Boosting Decision Tree (GBDT), which combines multiple weak learners (decision trees) to construct a strong learner. During the training process, a gradient descent algorithm is used to minimize the loss function, and regularization terms are added to control the complexity of the model and prevent overfitting. The base learner used in the XGBoost model in this article is the CART decision tree, and the specific implementation steps are as follows:

(1) Initialization: Calculate the initial predicted value (usually the mean of the training data labels).

(2) Calculate gradient and loss: For each sample, calculate the model's prediction error on the target value, which is the negative gradient. Calculate the gradient of the loss function with respect to the predicted value based on its form. This article is a binary classification problem, with a gradient of $y_i - P(y_i = 1)$ for logistic loss.

(3) Build objective function: Fit a regression tree with negative gradients and prune the tree using regularization terms to prevent overfitting. The objective function consists of a loss function and a regularization term, which measures the difference between predicted and true values and controls the complexity of the model. The specific formula is as follows:

$$bj = \sum_{i=1}^{n} L(y_i, \hat{y}_i)) + \sum_{k=1}^{K} \Omega(f_k)\,, \tag{2.51}$$

where L is the loss function, y_i is the true value, $\hat{y}_i$ is the predicted value, Ω is the regularization term, f_k is the k-th base learner (decision tree), and K is the number of base learners.

The fitting of the objective function to the tree: At each node, select feature j and segmentation point s to minimize the objective function:

$$\mathrm{Obj}_{\mathrm{split}} = \frac{1}{2}\left[\frac{G_L^2}{\lambda + H_L} + \frac{G_R^2}{\lambda + H_R} - \frac{(G_L + G_R)^2}{\lambda + H_R}\right] - \gamma\,, \tag{2.52}$$

where G_L and G_R are the sum of gradients for the left and right child nodes, H_L and H_R are the sum of Hessian matrices for the left and right child nodes, respectively, and λ are regularization terms, Y It is the regularization of tree complexity.

If the objective function value decreases (gain $Obj_{split} > 0$) after splitting a node, it indicates that the splitting is beneficial and the result of this splitting can be considered. In order to select the optimal splitting feature and splitting point, the calculated feature gain values can be sorted. The greedy algorithm is used to traverse all partition points on all features, select the optimal feature and splitting point that cause the maximum decrease in the objective function value, and then recursively construct the CART decision tree. This process is iterated multiple times to obtain multiple decision trees, and a new tree is added in each iteration until the predetermined number of iteration rounds or stop conditions are met, gradually improving the performance of the model.

The advantages of XGBoost model include: (a) efficient capture of nonlinear relationships and interaction effects in data, and strong learning ability; (b) Introduced regularization terms, including tree complexity and regularization of leaf node weights, effectively preventing model overfitting; (c) With fast training speed and lower memory usage, it is suitable for processing large-scale datasets.

2.4.6.3 Stacking Algorithm - Layered Model

The Stacking algorithm was first proposed by Wolpert (1992), which takes the prediction results of multiple base models as new input features and uses two training modes, multi generalization and single generalization, to train a meta model for prediction, improving the prediction accuracy and generalization ability of the model.

Unlike traditional bagging and boosting, the key idea of Stacking is to input the predicted results of different models as new features into the final meta model.

Stacking is a hierarchical ensemble learning algorithm that consists of two levels. In the first layer, multiple different types of base learners are trained on different subsets, and the output results form a new dataset. The second layer's meta learner uses this new dataset and the original training set labels for training, thereby integrating the output of the primary learner to obtain the final prediction result. This structure allows different learners to collaborate and improve model performance.

In Stacking, the primary learners of the first layer are usually designed to use complex nonlinear transformations to extract high-order features of the data, and may choose better performing learners to better capture the complex patterns of the data. This design makes the first layer learner more capable of adapting to training data. Due to the complexity of primary learners, there is a risk of overfitting. In order to reduce the risk of overfitting, the second layer of meta learners usually choose simple and robust models, such as linear regression or logistic regression. This article constructs a hierarchical model based on financial risk warning:

(1) Data preparation: Divide the original dataset into training and testing sets.

(2) First level base learner training: Select multiple different types of base learners, such as decision trees, support vector machines, BP neural networks, random forests, etc. Train these base learners on different subsets of the training set to obtain their respective prediction results.

(3) Generate a new training set: Use the predicted results of each base learner as new features to form a new training set.

(4) Second layer meta learner training: This article selects logistic regression as the meta learner and trains the meta learner using labels from the new and original training sets to learn how to combine the output of the primary learner.

(5) Generate final prediction result: Use the prediction results of the first layer base learner on the test set as input to the second layer meta learner to obtain the final model prediction.

The advantages of hierarchical models include: (a) The use of multiple different types of primary learners, which can integrate the advantages of various models and improve their performance; (b) Beginner learners use complex nonlinear

transformations to extract features, making them more adaptable to training data; (c) By selecting simple and robust meta learners, such as logistic regression, it helps to reduce the risk of overfitting and improve the model's generalization ability.

2.4.7 Activation function optimization algorithm

With the development of deep learning, optimizing activation functions has become an important direction for improving model performance. The traditional ReLU activation function performs well in many tasks due to its simplicity, computational efficiency, and ability to alleviate gradient vanishing. However, it suffers from the Reaper Meridian problem, which makes the model unable to learn effectively in certain situations. To overcome this issue, researchers have proposed various improved versions of ReLU functions, such as Leaky ReLU and Parametric ReLU (PReLU), which avoid the dead zone problem of ReLU by introducing negative slopes or trainable parameters. In addition, traditional activation functions such as Tanh and Sigmoid are also commonly used for different tasks. Tanh has smoothness by limiting the output between -1 and 1, but may be affected by gradient vanishing. Sigmoid maps the output to the interval of 0 to 1, which is commonly used in binary classification problems, but may also be limited by gradient vanishing. To address these issues, novel activation functions such as Swish and Mish have been proposed, which combine the characteristics of functions such as Sigmoid and Tanh to provide a smoother activation process, further improving the training performance and generalization ability of deep neural networks. This article will propose an improved ReLU_Tanh activation function in order to achieve the effect of optimizing the model.

2.4.7.1 ReLU activation function

ReLU is a simple and efficient activation function that sets the part of the input value that is less than zero to zero and retains the part that is greater than zero as the original value. This feature can introduce non-linearity while avoiding gradient vanishing problems, and is therefore widely used in deep learning. It has high computational efficiency and can significantly accelerate the convergence speed of the network, but it may lead to the "Grim Reaper" problem, where the gradient of some neurons is always zero and cannot be updated.The specific calculation formula and curve graph are as follows,

$$ReLU(x) = \max(0, x). \tag{2.53}$$

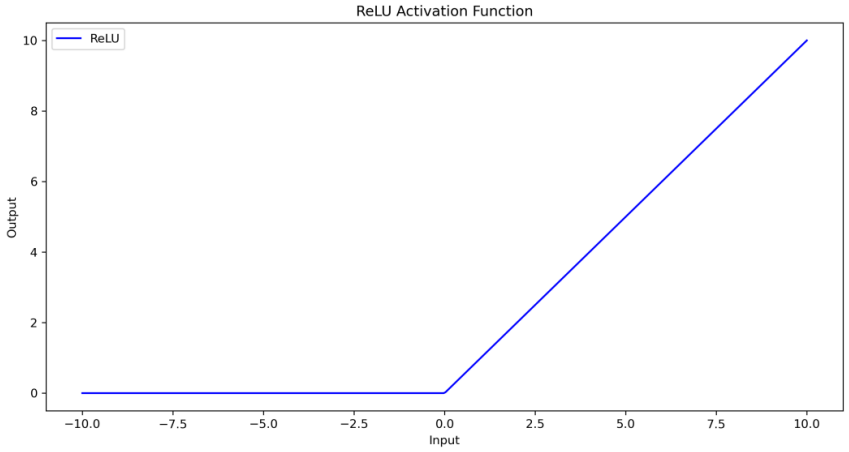


Figure 6 ReLU activation function curve graph

2.4.7.2 Sigmoid activation function

Sigmoid compresses input values between 0 and 1, making it suitable for tasks that represent probabilities, especially in binary classification problems[63]. Its output is a smooth "S" - shaped curve that can compress larger input values to near 1 and smaller input values to near 0. However, due to its gradient tending towards zero at extreme values, it can easily lead to the problem of gradient vanishing. In addition, Sigmoid's index calculation is relatively complex, which may increase training costs.The specific calculation formula and curve graph are as follows.

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{2.54}$$

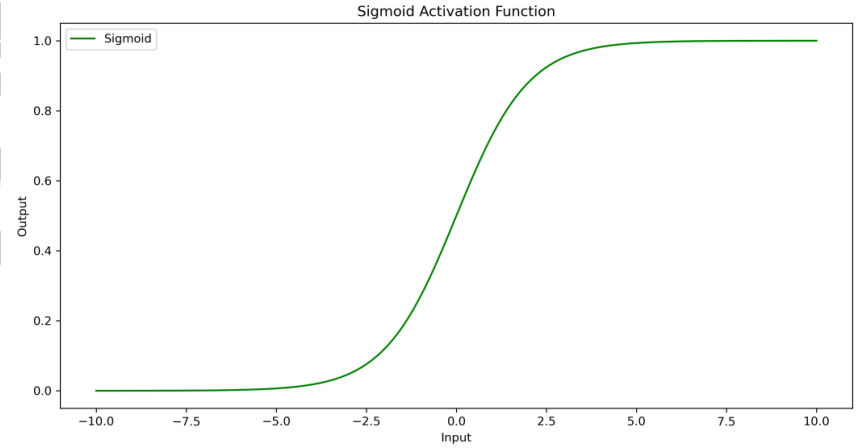


Figure 7 Sigmoid activation function curve graph

2.4.7.3 Tanh activation function

Tanh maps input values between -1 and 1, which is an enhanced version of Sigmoid with symmetry and is more suitable for handling zero mean data[59]. Its output range is wider and it can capture more nonlinear features. However, Tanh still faces the problem of gradient vanishing when the input values are large or small, and the computational complexity is high. Nevertheless, it performs excellently in deep learning tasks that require symmetric outputs.The specific calculation formula and curve graph are as follows,

$$Tanh(x) = 2\sigma(x) - 1 = \frac{e^{x}+e^{-x}}{e^{x}+e^{-x}}. \quad (2.55)$$

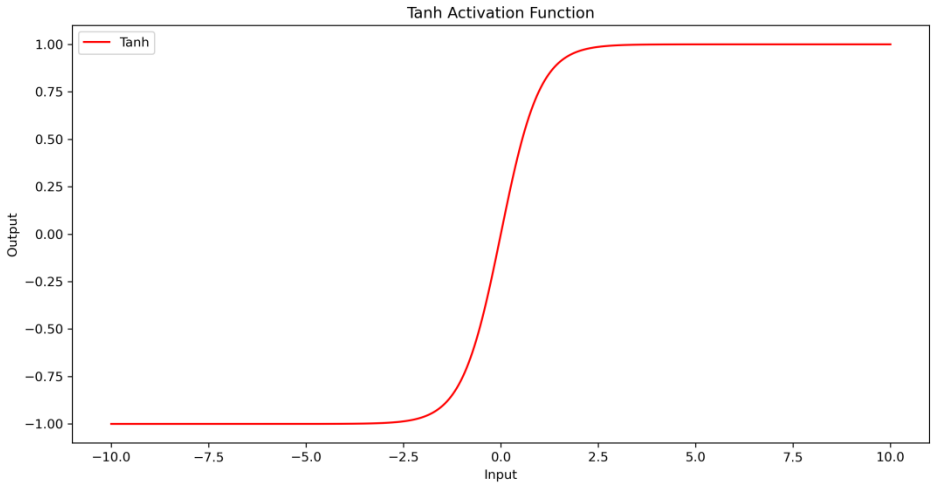


Figure 8 Tanh activation function curve graph

2.4.7.4 ReLU_Tanh activation function

The ReLU_Tanh activation function can rely on ReLU to accelerate training and feature extraction in positive regions, and use Leaky ReLU to ensure gradient flow in negative regions, thereby avoiding dead neuron problems. At the same time, Tanh maintains smoothness and output centralization, avoiding gradient explosion and improving the convergence and stability of the model. Ultimately, it can effectively improve the learning ability of neural networks when processing complex data, especially in deep networks, helping neurons better participate in the training process and avoid training stagnation or overfitting.The specific calculation formula and curve graph are as follows,

$$ReLU\_Tanh(x) = \mathrm{x} * \mathrm{ReLU}(\mathrm{x}, \alpha = 0.01) * Tanh(x)$$

$$= \begin{cases} x^2 * \frac{e^x + e^{-x}}{e^x + e^{-x}} & if\ x \geq 0 \\ 0.01x^2 * \frac{e^x + e^{-x}}{e^x + e^{-x}} & if\ x < 0 \end{cases} . \quad (2.56)$$

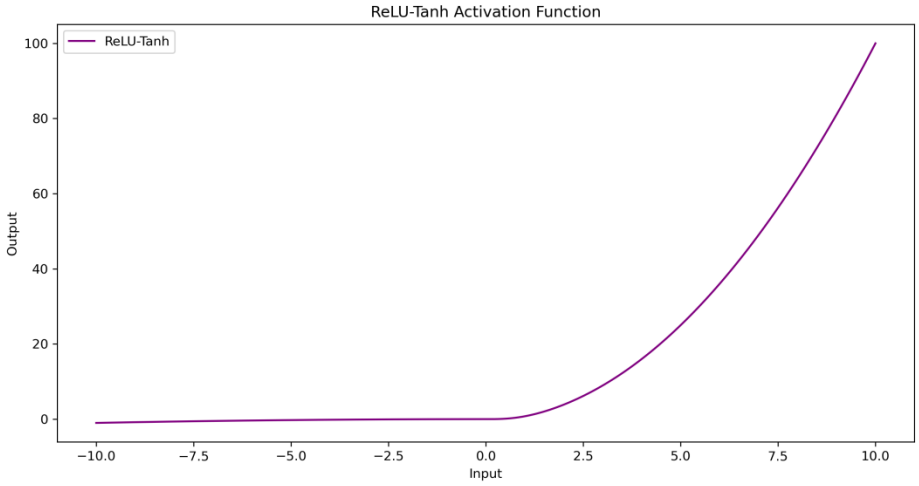


Figure 9 ReLU_Tanh activation function curve graph

2.4.7.5 ReLU_Sig activation function

The ReLU_Sig activation function integrates the advantages of the Rectified Linear Unit (ReLU) and Sigmoid functions, along with an additional scaling factor x. ReLU facilitates rapid feature extraction in the positive region, while Leaky_ReLU addresses the "dead neuron" issue by maintaining a small gradient flow in the negative region. The Sigmoid component ensures smooth and normalized outputs within the range [0,1], enhancing non-linear transformation. The inclusion of the scaling factor xxx further improves gradient flow and feature representation, making the function particularly effective for handling complex data in deep networks by preventing training stagnation and enhancing overall efficiency.The specific calculation formula and curve graph are as follows,

$$ReLU\_Sig(x) = x * ReLU(x, \alpha = 0.01) * Sigmoid(x)$$

$$= \begin{cases} \frac{x^2}{1+e^{-x}} & if\ x \geq 0 \\ \frac{0.01x^2}{1+e^{-x}} & if\ x < 0 \end{cases} . \quad (2.57)$$

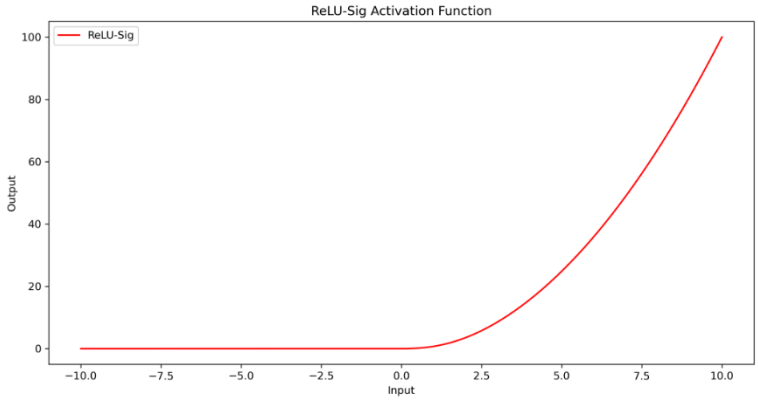


Figure 10 ReLU_Sig activation function curve graph

2.4.7.6 Tanh_Sig activation function

The Tanh_Sig activation function integrates the benefits of the hyperbolic tangent (tanh) and sigmoid functions, enhanced by a scaling factor xxx. The smooth and centralized characteristics of tanh(x) help mitigate gradient explosion and accelerate convergence, while sigmoid(x) adds non-linearity and normalizes outputs to the [0,1] range. The scaling factor xxx further promotes efficient gradient flow, allowing neurons to contribute more effectively during training. This synergistic design improves the neural network's ability to learn complex patterns, making it highly suitable for deep networks by reducing the risks of stagnation and overfitting. The specific calculation formula and curve graph are as follows,

$$Tanh\_Sig(x) = x * Tanh(x) * Sigmoid(x) = \frac{e^{x}+e^{-x}}{e^{x}+e^{-x}} * \frac{x}{1+e^{-x}} \quad (2.58)$$

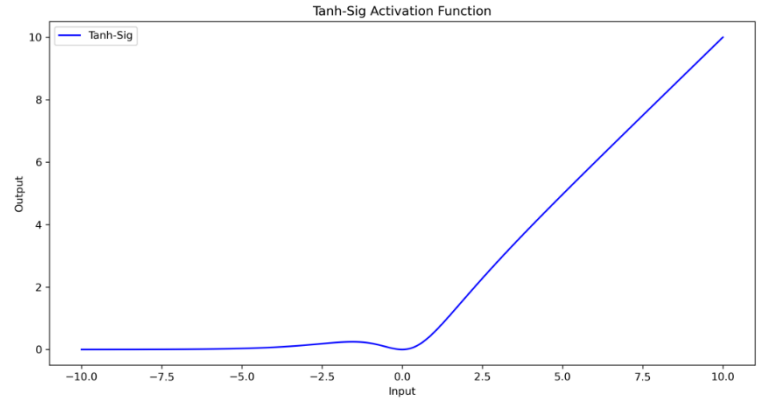


Figure 11 ReLU_Sig activation function curve graph

2.4.8 Hyperband algorithm

The Hyperband algorithm, proposed by Lisha Li and Kevin Jamieson in 2016, eliminated the underperforming hyperparameter configurations in each round by combining the Successive Halving algorithm in the multi-armed bandit strategy and efficiently selects the best-performing hyperparameter configurations from the final configuration to optimize the model training process.

(1) Initialization parameters: Set the resource budget R (the total number of training rounds), set the bandwidth factor η (a parameter controlling the elimination speed of hyper-parameter configurations, the default is 3), and set the initial amount of resources r0 (the amount of resources used by each configuration in the initial evaluation).

(2) Calculate maximum rounds:s_{max} is the maximum number of rounds that can be performed with the current resource budget R and initial resource amount r0.

$$s_{max} = \left\lfloor \log_{\eta}\left(\frac{R}{r_0}\right) \right\rfloor, \tag{2.59}$$

where⌊ ⌋ stands for the downward rounding sign.

(3)Perform multiple rounds of hyperparameter configuration and elimination process: Multiple rounds are performed from s=0 to s=s_{max}.

For each rounds, the number of configurations and the amount of resources are initialized, and n hyperparameter configurations are generated, each of which receives a training resource amount r at the beginning.

$$n = \left\lceil \frac{R}{r_0(s+1)\eta^s} \right\rceil, \tag{2.60}$$

$$r = r_0 \cdot \eta^s. \tag{2.61}$$

In each rounds, obtain the number of hyperparameter configurations ni and the amount of resources ri used for each configuration in the ith subphase (from 0 to s).

$$n_i = \left\lceil \frac{n}{\eta^i} \right\rceil, \tag{2.62}$$

$$r = r_0 \cdot \eta^i. \tag{2.63}$$

Allocate resources: Allocate resources evenly to all hyperparameter configurations and perform initial training.

Eliminate configurations: According to the Successive Halving algorithm, eliminate the underperforming configurations and concentrate the resources on the best-performing configurations, at which point the elimination ratio is $1-\frac{1}{\eta}$, the best-performing $\frac{1}{\eta}$ configurations are retained.

(3)Selection of the optimal configuration: At the end of all rounds, the best performing hyperparameter configuration is selected from the final configuration.

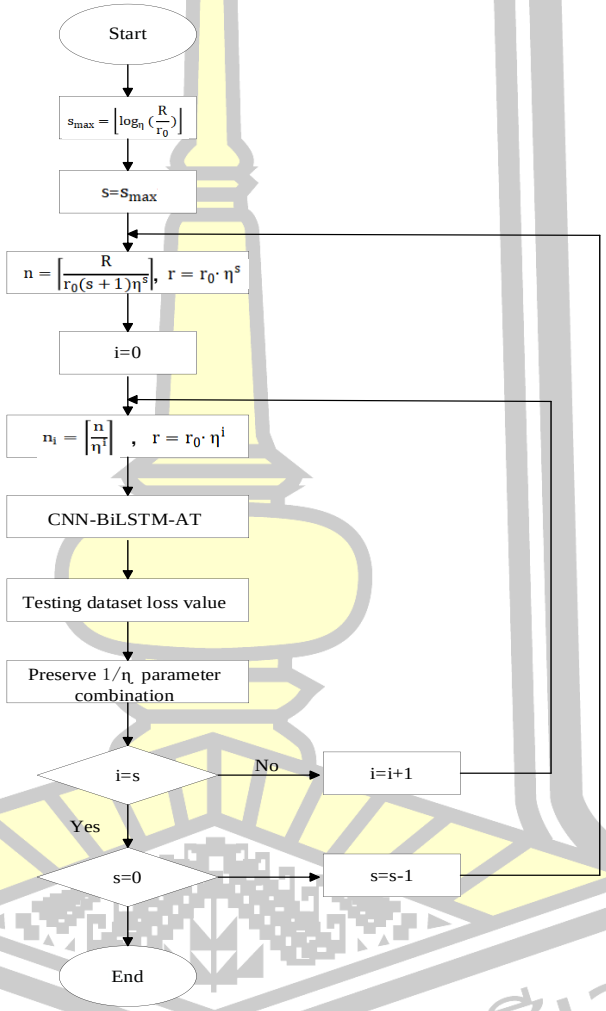


Figure 12 Hyperband Algorithm Optimization Framework Diagram

2.4.9 Evaluation methods for binary classification models

In binary classification tasks, a confusion matrix and multiple evaluation metrics derived from the confusion matrix (accuracy, precision, recall, F1 value, ROC curve, and AUC) are commonly used to evaluate the performance of machine learning classification models.

2.4.9.1 Confusion Matrix

The confusion matrix is the foundation of various evaluation indicators for machine learning classification models. The confusion matrix for binary classification models is shown in Table 1. The letters T and F represent correct and incorrect predictions, respectively, while the letters P and N represent positive and negative predictions. Therefore, the total number of samples is equal to the sum of TN, FP, FN, and TP. When TN (true and negative examples) and TP (true and negative examples) are larger, FP (false positive examples) and FN (false negative examples) are smaller, the classification ability of the model is stronger.

Table 1 Binary Chaos Matrix

Prediction category	**Real category**	
	1（True）	**0（False）**
1（True）	TP	FP
0（False）	FN	TN

Among them, True Positive (TP): positive category samples are correctly predicted as positive categories; True Negative (TN): Negative category samples are correctly predicted as negative categories; False Positive (FP): Negative category samples are incorrectly predicted as positive categories; False Negative (FN): Positive category samples are incorrectly predicted as negative categories.

2.4.9.2 Accuracy, precision, recall, F1 value

(1) Accuracy

Accuracy is a key evaluation indicator in classification problems, which measures the overall accuracy of the model's classification and represents the proportion of correct predictions made by the model across all samples. Accuracy considers the prediction of positive and negative categories, including correctly predicting samples that were originally negative as negative examples and correctly predicting samples that were originally positive as positive examples. High accuracy means that the overall classification performance of the model is good, but in the presence of data imbalance, accuracy may not be an effective evaluation indicator.

$$\text{Accuracy} = \frac{\text{TN+TP}}{\text{TN+FP+FN+TP}}. \tag{2.64}$$

(2) Precision

Precision is an important indicator for measuring the accuracy of a model in predicting positive samples in classification problems. Precision focuses on positive examples that have a relatively small proportion in the sample and poor performance. Specifically, it represents how many samples predicted as positive by the model are correctly predicted, that is, samples that were originally positive are actually predicted as positive. The higher the precision, the stronger the model's classification ability for positive samples.

$$Precision = \frac{TP}{FP+TP}. \tag{2.65}$$

(3) Recall

Recall is an important indicator for evaluating the model's ability to recognize positive samples in classification tasks. In practical applications, compared to negative examples, we are more concerned about the situation of positive examples. For example, in financial risk scenarios, we do not want to miss any financial crisis company (positive example), because misclassifying a financial crisis company as a normal operating company (negative example) may lead to greater losses. Therefore, the recall rate represents how many samples were successfully predicted by the model as positive cases among all the samples that were originally positive.

$$Recall = \frac{TP}{FN+TP}. \tag{2.66}$$

(4) F1 score

F1 score is the harmonic average of precision and recall, taking into account the predictive accuracy and coverage of the model in positive samples. The range of F1 values is between 0 and 1, and the closer it is to 1, the better the model performance. When the model reaches a balance between accuracy and recall, the F1 value reaches its maximum. Therefore, the superiority of F1 value lies in its ability to comprehensively evaluate the classification performance of the model under a single indicator.

$$F1\ value = \frac{2\times Precision\times Recall}{Precision+Recall}. \tag{2.67}$$

2.4.9.3 ROC curve and AUC

The Receiver Operating Characteristic Curve (ROC) is a graphical tool used to evaluate the performance of binary classification models. It shows the performance of

the model under different classification thresholds, with True Positive Rate (recall rate) as the vertical axis and False Positive Rate as the horizontal axis. In the ROC curve, the closer the graph is to the upper left corner, the better the model performance. The working principle is to help select a threshold that balances sensitivity and specificity, which helps optimize the classification performance of the model.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}}, \tag{2.68}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{FN+TP}}. \tag{2.69}$$

AUC (Area Under the Curve) is the area under the ROC curve used to quantify the overall performance of a model at different thresholds. The calculation of AUC value involves the trapezoidal area of each point under the ROC curve. The AUC value range is between 0 and 1, where 0.5 represents model performance equivalent to random guessing, and 1 represents model perfect prediction. AUC provides a single value for comparing the overall performance of different models, and a high AUC value typically indicates that the model can balance recall and false positives under various classification thresholds.

$$AUC = \int_0^1 TPR(FPR^{-1}(t))dt\,. \tag{2.70}$$

2.4.9.4 Cross validation evaluation model

Cross Validation is a statistical method used to evaluate the performance of machine learning models. Its concept was first proposed by statisticians in the 1970s to reduce the risk of models performing too well on training data, known as overfitting. Stone proposed the idea of cross validation in 1974, and Geisser introduced the "Leave One Out Cross Validation" (LOO-CV) method in 1975, which is a special form of cross validation.

(1) Dataset partitioning: Divide the original dataset into k equally sized subsets (usually k=5 or k=10, known as 5-fold or 10 fold cross validation).

(2) Model training and validation: Each iteration selects one subset as the validation set (test set), and the remaining k-1 subsets as the training set. Train the model on the training set, then test the model on the validation set and record the validation error.

(3) Repeat k times: Repeat the above process k times, selecting a different subset

as the validation set each time.

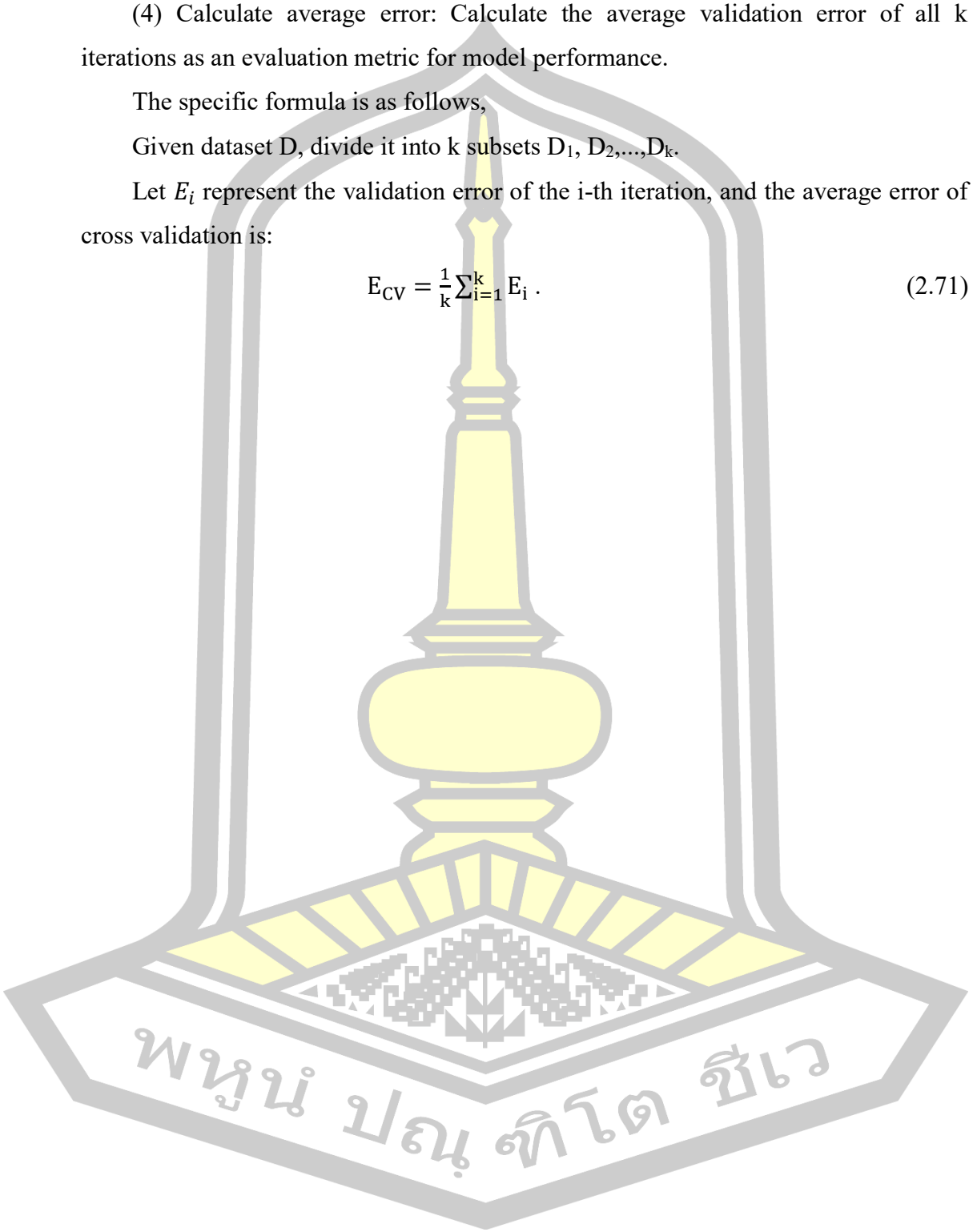
(4) Calculate average error: Calculate the average validation error of all k iterations as an evaluation metric for model performance.

The specific formula is as follows,

Given dataset D, divide it into k subsets D_1, D_2,...,D_k.

Let E_i represent the validation error of the i-th iteration, and the average error of cross validation is:

$$E_{CV} = \frac{1}{k}\sum_{i=1}^{k} E_i \ . \tag{2.71}$$



Chapter 3

Methodology

This chapter introduces research scope, step of the methodology, and work flow chart.

3.1 Research Scope

The scope of this research is to investigate whether Chinese listed companies will experience financial crises and how to provide early warning. This section will focus on introducing the data used and the variables selected.

3.1.1 Data

The focal population for this study encompasses the 5052 companies listed on the Shanghai and Shenzhen stock exchanges. In line with the prevalent criteria adopted by a majority of Chinese scholars, this study utilizes the special treatment status (inclusive of ST classification) assigned to listed companies due to irregular financial circumstances as a marker of financial distress, special treatment was implemented due to abnormal financial conditions (such as continuous losses in the past two years) as the sample data of ST companies. Consequently, the research meticulously identifies and selects financially distressed companies adhering to this principle as prime subjects for ST analysis.

In pursuit of this, the research aims to concentrate on ST companies that first encountered special treatment during the period of 2017-2024, positioning them as the primary subjects for ST sample analysis. will identify companies marked as ST as financial risk companies, and other companies not marked as ST as normal operating companies.

135 ST companies and 755 non-ST companies are selected, and the financial indicator data for 12 quarters of 3 years are extracted for each company, so there are 1630 time series points for ST companies and 9060 time series points for non-ST companies. The criteria for selecting the non-ST companies mandates a parallel in terms of asset volume, industry sector, and scale, akin to their ST counterparts, within the same fiscal year.The year marked as ST is recorded as year T, and the previous year of ST is recorded as year T-1. In order to predict the financial difficulties in advance in the normal operation of the company, this paper selects the quarterly

financial index data from year T-3 to year T-5 as the sample characteristic data for prediction analysis.

Table 2 Sample distribution table

Company Type	Number of company samples	Number of Time Series Data Points
ST	135	1620
Non-ST	755	9060
Total	**890**	**10680**

3.1.1 Varaiables

Based on the actual situation of listed agricultural companies in China and the new policy guidelines, 34 indicators have been set up as a warning indicator system from the aspects of debt service ratios, ratio structure, operating capacity, earnings capacity, cash flow ratios, development capability and per share metrics.

Table 3 Financial risk warning indicator system

Financial indicators	Indicator name
Debt service capacity	Current ratio、quick ratio、cash ratio、working capital、debt to asset ratio、equity ratio、multiplier equity ratio
Ratio structure	Current assets ratio、cash assets ratio、working capital ratio、non current assets ratio、current liabilities ratio、fixed assets ratio、operating profit margin
Operating capacity	Accounts receivable turnover、inventory turnover、accounts payable turnover、current asset turnover、fixed asset turnover、total asset turnover
Earnings capacity	Return on assets、net profit margin on total assets 、net profit margin on current assets、net profit margin on fixed assets、return on equity、operating profit rate
Cash flow capacity	Operating index
Development capability	Capital preservation and appreciation rate、fixed asset growth rate、revenue growth rate、sustainable growth rate、owners' equity growth rate
Per share metrics	Earnings per share、earnings before interest and taxes per share

3.2 Step of The Methodology

This research will use Python to preprocess data, screen indicators, construct models, optimize models, etc., in order to obtain a high-performance financial risk warning model.

3.2.1 Data collection and preprocessing

This study aims to collect and preprocess data on financial and non-financial indicators of listed companies.

3.2.1.1 Data collection

Determine the data source from the CSMAR database and obtain financial and non- financial indicator data through API calls.

The selection of ST companies is primarily based on the special labeling (ST) by stock exchanges for listed companies with abnormal operating conditions. The companies initially marked as ST are chosen as samples for financial risk (ST) companies. The selection of normally operating companies involves resampling from companies that have not been labeled as ST, aiming for a 2:1 ratio to obtain samples of normally operating companies. During the resampling process, matching is conducted based on certain features such as industry, size, and profitability to ensure that the distributions of these features are similar between ST and nomal companies.

3.2.1.2 Data preprocessing

Preliminary data exploration, including data overview and descriptive statistics, in order to understand the basic characteristics of the data. In the data cleaning stage, handle missing and abnormal values, remove duplicates, and ensure data quality. Subsequently, data conversion is carried out, including variable standardization, encoding, etc. Finally, store the cleaned and transformed data as a new file for data quality validation, ensuring compliance with business rules and logic, and providing a high-quality data foundation for subsequent analysis and modeling.

(1) Sliding window

In this article, a total of 135 ST companies and 755 non-ST companies were selected, with financial indicator data extracted for 12 quarters over 3 years for each company. This resulted in 1,630 time series points for ST companies and 9,060 time series points for non-ST companies. Using the sliding window technique, feature windows and corresponding labels can be generated for subsequent analysis and model training. As shown in Figure 13 as an example of a sliding feature window of 6, windows are slid over the time series data to create overlapping windows. Each window contains six consecutive data points, and for each feature window, a label was assigned from the subsequent time step T, i.e., whether it was an ST company,

which can effectively utilize the time series data to generate more training samples and improve the generalization ability of the model.

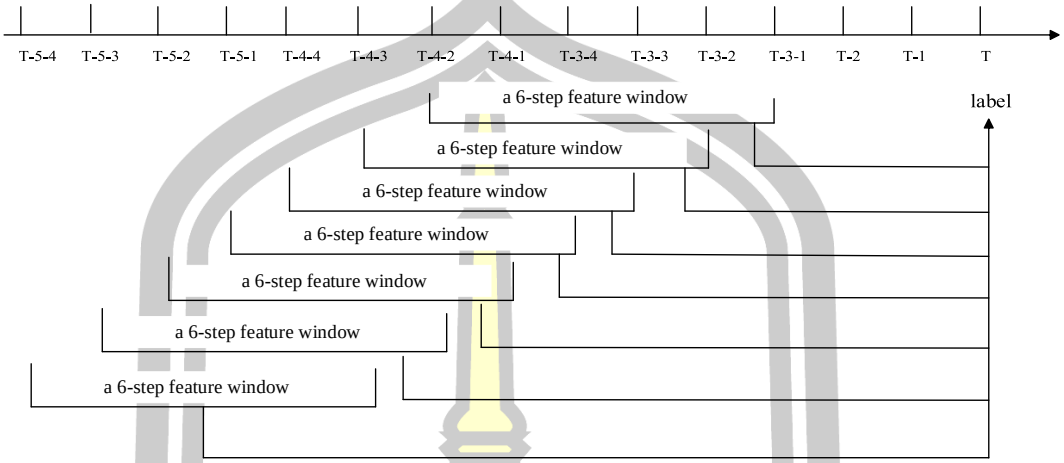


Figure 13 Window sliding diagram

(2) Smote oversampling

The training set data was augmented using the Synthetic Minority Over-sampling Technique (Smote) method.Smote increases the number of minority samples by generating new synthetic samples around the minority class samples for the purpose of data balancing.

3.2.2 Financial indicator selection

The feature engineering of financial indicators is aimed at extracting the most informative features for subsequent modeling purposes. Using traditional indicator screening methods such as principal component analysis (PCA), as well as machine learning indicator screening methods such as random forest methods.

3.2.2.1 Traditional indicator selection methods

(1) PCA: Fit the PCA method to obtain the principal components, view the variance interpretation ratio of each principal component, and select the principal components whose cumulative interpretation ratio meets the predetermined threshold of 95%.

3.2.2.2 Machine learning indicator selection methods

(2) Random forest feature selection: Using the feature importance index of the random forest, evaluate the impact of each financial indicator on the target variable, and select financial indicators with importance higher than the threshold.

3.2.3 Establishing Single Machine Learning Models

Establishing a single machine learning model involves multiple steps, including data preparation, model selection, training, evaluation, etc.

(1) To build and train models like Logistic Regression, Decision Tree, and SVM, use Scikit-learn for straightforward implementation. Logistic Regression models the relationship between features and outcomes, Decision Trees split data based on feature values, and SVM find the optimal hyperplane for classification. For neural networks, use Keras or TensorFlow to design and train models using the backpropagation algorithm. Evaluate model performance with metrics such as accuracy, precision, recall, and F1-score. Optimize by tuning hyperparameters like regularization in Logistic Regression, tree depth in Decision Trees, kernel type in SVMs, and learning rate or layer configuration in neural networks. Use grid search or other optimization methods to fine-tune the models for the best performance.

(2) To build and train CNN, BiLSTM, and Attention models using TensorFlow, you start by constructing the models. For the CNN, use convolutional layers to extract spatial features, followed by pooling layers for dimensionality reduction, and fully connected layers for classification. The BiLSTM captures bidirectional temporal dependencies, while dropout layers prevent overfitting. The attention mechanism is integrated to focus on key parts of the input sequence, enhancing feature prioritization. During training, use cross-entropy as the loss function and the Adam optimizer to adapt the learning rate for faster convergence. Monitor accuracy and loss to assess model performance. Hyperparameters such as learning rate, number of hidden units, and convolution kernel size should be adjusted to optimize the model, while dropout rates are modified to balance overfitting and generalization. After training, evaluate the model using metrics like accuracy, precision, and recall on a test set. Finally, hyperparameter optimization techniques like grid search can be used to fine-tune the model further, ensuring it achieves the best possible predictive performance.

3.2.4 Establishing ensemble learning models

Establishing ensemble learning models includes Bagging algorithm (Random Forest Model), Boosting algorithm (XGBoost Model), and Stacking algorithm (Layered Model). This article will combine the aforementioned single machine

learning model as a base classifier to establish an ensemble learning model, such as logistic regression, decision tree, SVM, BP neural network, etc.

(1) Establish a random forest model for the Bagging algorithm by scikit-learn library, adjust hyperparameters such as the number of trees and the maximum depth of each tree, and evaluate and optimize the model through indicators such as accuracy, precision, and recall.

(2) Establish an XGBoost model for the Boosting algorithm by xgboost library, adjust hyperparameters, such as learning rate and number of trees, and evaluate and optimize the model through indicators such as accuracy, precision, and recall.

(3) Establish a layered model for the Stacking algorithm, select base classifiers such as decision trees, SVM, etc., use the output of the base classifier as input, and choose logistic regression as the meta model. Combining methods such as random search, Bayesian optimization, or grid search to find faster hyperparameter combination search strategies, evaluating and optimizing the model through indicators such as accuracy, precision, and recall.

(4) Establish a layered model for the Stacking algorithm, select base classifiers such as decision trees, SVM, etc., use the output of the base classifier as input, and choose logistic regression as the meta model. Combining methods such as random search, Bayesian optimization, or grid search to find faster hyperparameter combination search strategies, evaluating and optimizing the model through indicators such as accuracy, precision, and recall.

(5) Establish a CNN-AT model, by constructing a Convolutional Neural Network (CNN) to extract spatial features from the input data. After the CNN layers, integrate an Attention mechanism (AT) to focus on the most relevant parts of the extracted features. The output of the Attention layer is then passed through fully connected layers to produce the final classification. Hyperparameter tuning, such as adjusting the convolution kernel size, number of filters, and attention layer parameters, can be performed using techniques like random search, Bayesian optimization, or grid search. The model is evaluated and optimized using metrics such as accuracy, precision, and recall.

(6) Establish a BiLSTM-AT model,constructing a Bidirectional Long Short-Term Memory (BiLSTM) network to capture temporal dependencies in the input sequences.

After the BiLSTM layers, add an Attention mechanism to allow the model to focus on important time steps in the sequence. The Attention layer output is then connected to fully connected layers for final prediction. Hyperparameter tuning, including the number of LSTM units, learning rate, and attention layer settings, can be optimized using hyperband. The model's performance is evaluated using accuracy, precision, and recall.

(7) Establish a CNN-BiLSTM model, starting with a CNN to extract spatial features, followed by a BiLSTM layer to capture temporal dependencies from the CNN output. The combination of CNN and BiLSTM layers allows the model to learn both spatial and sequential patterns. After the BiLSTM layer, fully connected layers are used for classification. Hyperparameters such as the number of CNN filters, LSTM units, and learning rate can be optimized using hyperband optimization, or grid search. The model is evaluated based on accuracy, precision, and recall.

(8) Establish a CNN-BiLSTM-AT model, constructing a model by first using a CNN to extract spatial features, followed by a BiLSTM layer to capture temporal dependencies. After the BiLSTM layer, integrate an Attention mechanism to focus on important parts of the sequence. The output from the Attention layer is passed through fully connected layers for classification. This model combines the strengths of CNN, BiLSTM, and Attention to effectively capture spatial, temporal, and contextual information. Hyperparameter tuning for CNN, BiLSTM, and Attention layers can be done using hyperband optimization. The model is optimized and evaluated using accuracy, precision, and recall as key performance metrics.

3.2.5 Establishing an innovative optimization learning model

Establish a CNN-BiLSTM-AT model with optimized activation functions and perform a comparative analysis using various activation functions, including ReLU, Tanh, Sigmoid, and the newly designed ReLU_Tanh function. This optimization significantly improves the model's prediction accuracy and stability.

Based on the performance indicators of each model, taking into account both prediction accuracy and generalization ability, select the optimal model.

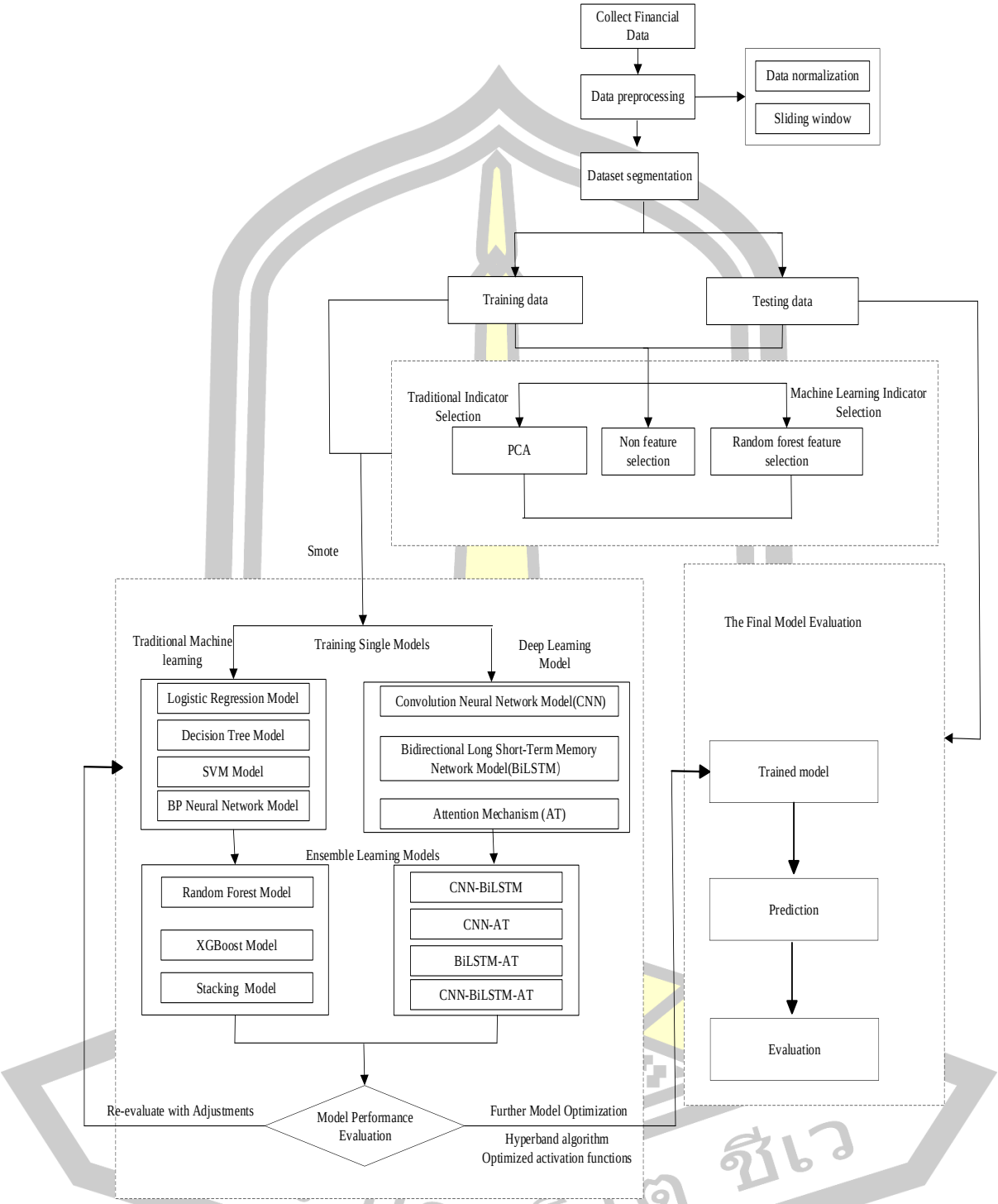


Figure 14 Step of The Methodology

3.3 Work Flow Chart

The main work flow chart and general idea of this article are shown in figure 15.

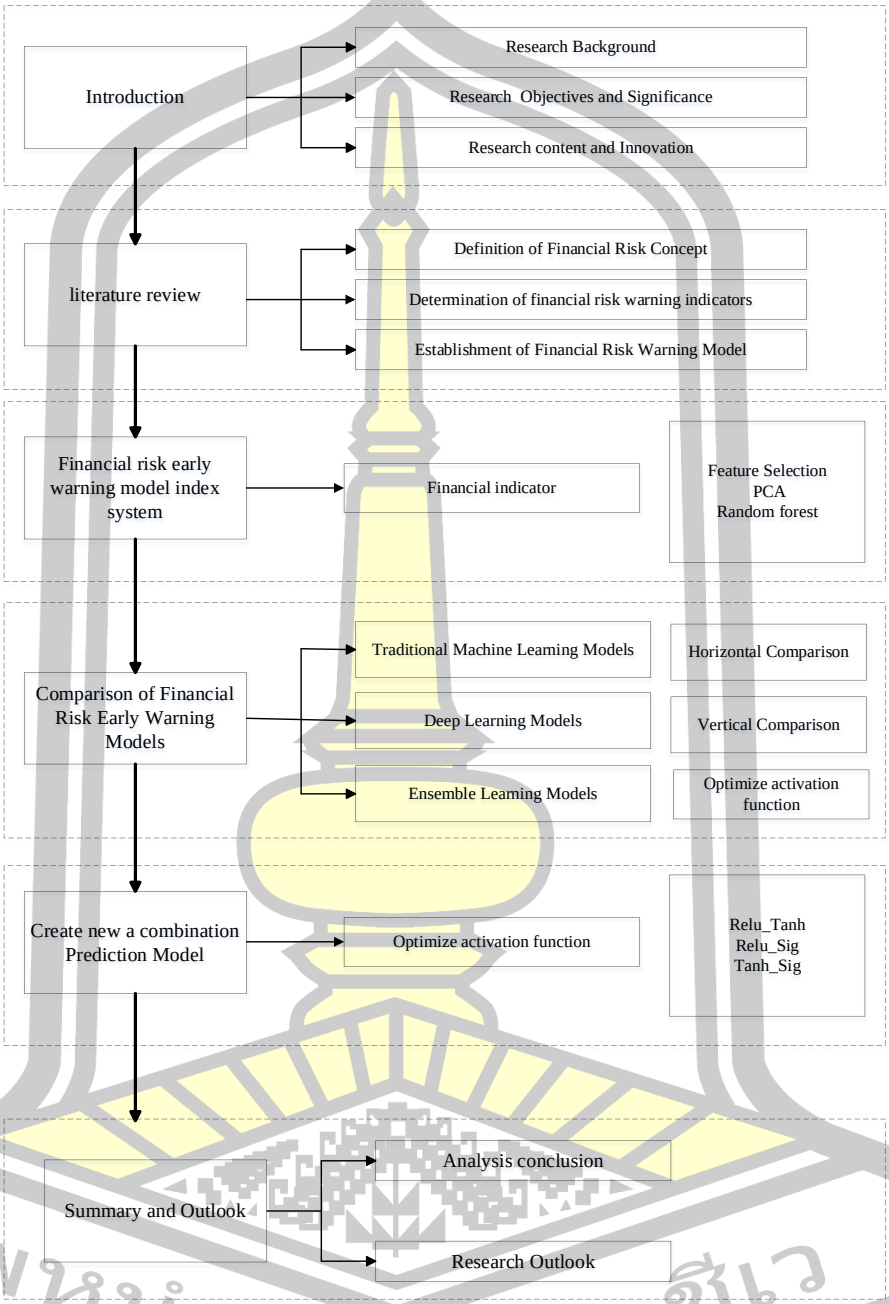


Figure 15 Work Flow Chart

Chapter 4

Results

This chapter introduces simulation and results around three research objectives.The results section mainly includes indicator selection, model performance comparison and case application analysis.

4.1 Simulation

4.2.1 Indicator selection

This article will compare PCA, random forest feature selection methods, and the impact of raw feature input on machine learning and deep learning models, and explore the key roles of different methods in model prediction. This article selected 135 ST companies and 755 non-ST companies, each with financial indicator data for 3 years and 12 quarters. Therefore, ST companies have 1630 sequential data points, while non-ST companies have 9060 sequential data points. This article conducted simulation experiments on these raw data with PCA, RF and raw feature.

From Table 4, different feature extraction methods have varying impacts on the model. For example, ensemble learning models perform better in RF feature extraction. Therefore, this article will use PCA, RF, and raw feature input methods to identify key financial indicators and evaluate their impact on model predictions.

Table 4 Model performance of different feature extraction methods

Model			**Feature extraction**		**Raw feature**
			PCA	**RF**	
Machine learning model	Single model	Logistic Regression	0.775	0.809	0.865
		SVM	0.764	0.798	0.854
		Decision Tree	0.719	0.798	0.854
		ANN	0.865	0.888	0.865
	Ensemble model	Random Forest	0.820	0.854	0.899
		XGBoost	0.854	0.899	**0.910**
		Stacking	0.865	**0.910**	0.888
Deep learning model	Single model	CNN	0.832	0.843	0.876
		BiLSTM	0.865	0.820	0.854
		Attention	**0.876**	0.843	0.831

Model			Feature extraction		Raw feature
			PCA	RF	
Deep learning model	Ensemble model	CNN-BiLSTM	0.809	0.843	0.843
		CNN-AT	0.787	0.832	0.820
		BiLSTM-AT	0.753	0.854	0.820
		CNN-BiLSTM-AT	0.854	0.820	0.798

In the feature selection simulation, this study defined the target variable as whether a company is an ST company, assigning a value of 1 to ST companies and 0 to non-ST companies. The data was standardized, and principal components were selected based on a cumulative contribution rate of 95%. Features with a correlation higher than 0.5 with the principal components were extracted, and the Pearson correlation coefficient was used to assess the relationship between these key features and the target variable. The analysis showed that when current liabilities ratio, total asset turnover, and revenue growth rate are higher, the company is more likely to be an ST company. For ST companies, a higher current liabilities ratio indicates a greater short-term debt burden, while higher total asset turnover and revenue growth rate might result from aggressive short-term strategies, such as high leverage financing and using asset turnover or low-profit sales to maintain short-term growth, reflecting the financial distress and survival tactics of these companies. On the other hand, when cash assets ratio and fixed assets ratio are higher, the company is more likely to be a non-ST company. Non-ST companies typically exhibit stronger cash reserves and higher long-term asset investments, indicating more stable financial conditions, with stronger debt repayment capacity and long-term viability. In conclusion, ST companies often rely on short-term financing, asset turnover, and revenue growth strategies to manage their financial difficulties, but these strategies do not represent long-term sustainability, whereas non-ST companies demonstrate more stable and healthy financial structures.

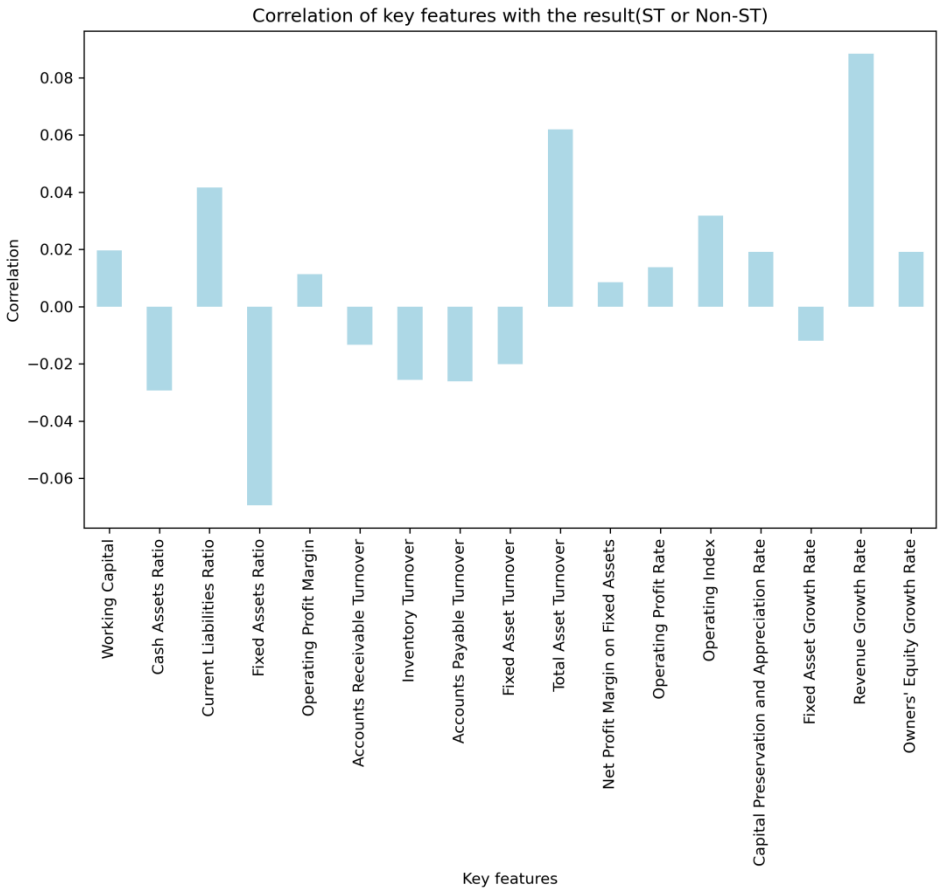


Figure 16 Correlation between important indicators extracted by PCA and results

During the feature selection process, Random Forest was used to identify key features, retaining those that contributed to 95% of the cumulative importance. The top 15 most important original features were then extracted, and their correlation with the target variable was analyzed using the Pearson correlation coefficient. The results indicate that a higher working capital and current liabilities ratio are more likely associated with ST companies, while a higher fixed assets ratio, cash asset ratio, inventory turnover, and accounts payable turnover are more indicative of non-ST companies. The analysis revealed that when working capital and current liabilities ratio are higher, the company is more likely to be an ST company. A higher working capital suggests that ST companies may struggle with liquidity management, relying on short-term financing to sustain operations. Meanwhile, an elevated current liabilities ratio indicates a heavier short-term debt burden, reflecting financial distress and an increased risk of default. On the other hand, when fixed assets ratio, cash asset ratio, inventory turnover, and accounts payable turnover are higher, the company is

more likely to be a non-ST company. A higher fixed assets ratio suggests greater long-term investments, indicating financial stability and a focus on sustainable growth. A higher cash asset ratio reflects stronger liquidity reserves, enabling non-ST companies to meet short-term obligations with greater ease. Additionally, higher inventory turnover and accounts payable turnover indicate efficient operational management and strong supplier relationships, contributing to a healthier financial structure.In conclusion, ST companies often rely on short-term financing and struggle with debt management, increasing their financial risk. In contrast, non-ST companies demonstrate stronger financial resilience, characterized by stable asset investments, higher liquidity, and efficient operational processes, ensuring long-term financial sustainability.

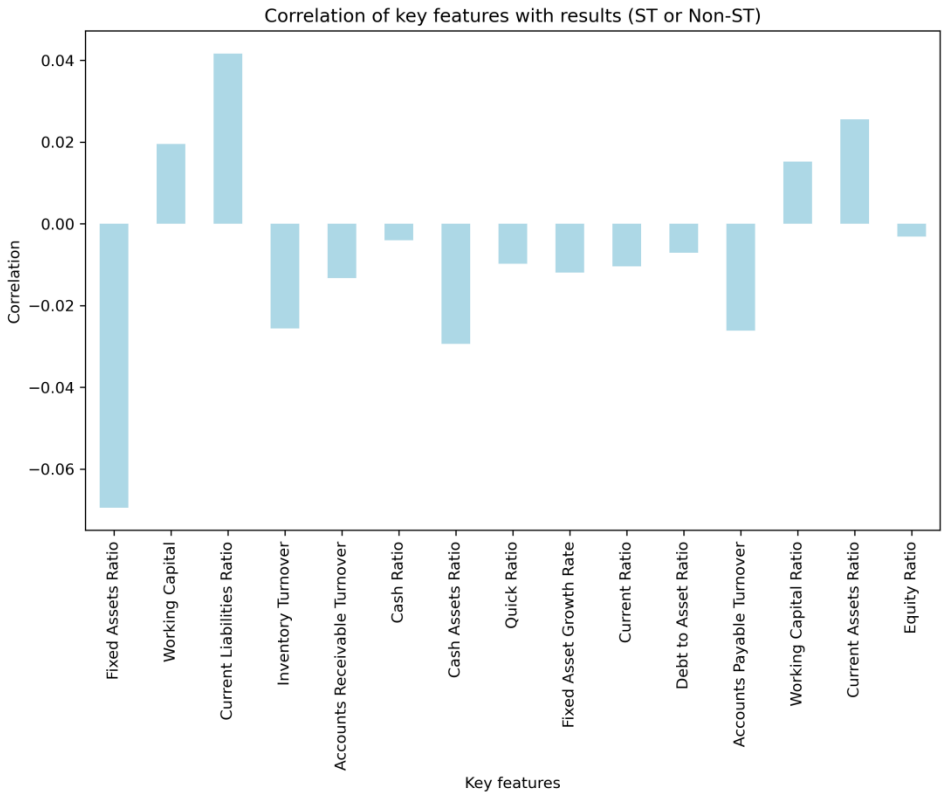


Figure 17 Correlation between important indicators extracted by Random Forest and results

4.2.2 Model performance comparison

This article will compare the advantages and disadvantages of traditional machine learning models, deep learning models, and ensemble learning models in financial risk warning. For the same set of data, what ratio should be used to divide the training and testing sets for optimal model performance? We will conduct

simulation experiments, using ratios of 9:1, 8:2, and 7:3 to divide the training and testing sets respectively.

This article conducted simulation experiments on these raw data, and found that when the division ratio was 9:1, the accuracy of each model was the highest.

Table 5 Accuracy of models (Non-smote) at different divide proportion

Model(Non-smote)			**Accuracy**		
			Ratio 9:1	**Ratio 8:2**	**Ratio 7:3**
Machine learning model	Single model	Logistic Regression	0.865	0.865	0.843
		SVM	0.854	0.848	0.824
		Decision Tree	0.854	0.798	0.775
		ANN	0.865	0.837	0.820
	Ensemble model	Random Forest	0.899	0.854	0.843
		XGBoost	0.910	0.848	0.839
		Stacking	0.888	0.854	0.843
Deep learning model	Single model	CNN	0.876	0.826	0.809
		BiLSTM	0.854	0.781	0.809
		Attention	0.831	0.781	0.790
	Ensemble model	CNN-BiLSTM	0.843	0.831	0.846
		CNN-AT	0.820	0.809	0.824
		BiLSTM-AT	0.820	0.815	0.805
		CNN-BiLSTM-AT	0.798	0.787	0.816

Due to the sample size ratio of 1:5.6 between ST and non ST companies, there is a significant difference in sample size between the two categories. Therefore, Smote is used in this article to balance the sample size. After Smote operation on the raw data, this article applies different types of models in a ratio of 9:1, 8:2, and 7:3 for simulation experiments. The experiment found that the accuracy of each model is highest when the dataset partition ratio is 9:1.

Table 6 Accuracy of models (Smote) at different divide proportion

Model(Smote)			Accuracy		
			Ratio 9:1	**Ratio 8:2**	**Ratio 7:3**
Machine learning model	Single model	Logistic Regression	0.809	0.792	0.764
		SVM	0.831	0.775	0.749
		Decision Tree	0.809	0.753	0.779
		ANN	0.888	0.831	0.813
	Ensemble model	Random Forest	0.854	0.798	0.794
		XGBoost	0.888	0.837	0.820
		Stacking	0.899	0.837	0.824
Deep learning model	Single model	CNN	0.854	0.809	0.805
		BiLSTM	0.820	0.792	0.813
		Attention	0.876	0.775	0.779
	Ensemble model	CNN-BiLSTM	0.820	0.803	0.798
		CNN-AT	0.831	0.815	0.813
		BiLSTM-AT	0.820	0.787	0.801
		CNN-BiLSTM-AT	0.831	0.831	0.794

Therefore, based on the above simulation experiments, this article will use a 9:1 ratio to divide the training set and test set to establish and evaluate the model.

Since neural network models need to set epoch and batch parameters, differences in model performance can be observed by setting different epoch and batch parameters when simulation experiments are conducted for network models (Non-smote) such as ANN, CNN, BiLSTM, and Attention. The experimental results show that different batch sizes at epoch=100 do not affect the performance of the models, and except for the BiLSTM model, the accuracy of the other models at epoch=180 and batch=32 is higher than or equal to the accuracy at epoch=180 and batch=64.

Table 7 Accuracy of models (Non-smote) at different epochs and batches

Model(Non-smote)	epoch=100		epoch=180	
	batch=32	**batch=64**	**batch=32**	**batch=64**
ANN	0.865	0.865	0.865	0.865
CNN	0.876	0.876	0.888	0.843
BiLSTM	0.854	0.854	0.854	0.865
Attention	0.831	0.831	0.831	0.820
CNN-BiLSTM	0.843	0.843	0.843	0.809
CNN-AT	0.820	0.820	0.820	0.809
BiLSTM-AT	0.820	0.820	0.820	0.820
CNN-BiLSTM-AT	0.798	0.798	0.865	0.831

When simulation experiments are conducted for each neural network model (Smote), differences in model performance can be observed by setting different epoch and batch parameters. The experimental results show that the different batch sizes at epoch=100 do not affect the performance of the models, and except for the Attention model, the accuracy of the other models at epoch=180 and batch=32 is higher than or equal to the accuracy at epoch=180 and batch=64.

Table 8 Accuracy of models (Smote) at different epochs and batches

Model(Smote)	**epoch=100**		**epoch=180**	
	batch=32	**batch=64**	**batch=32**	**batch=64**
ANN	0.888	0.888	0.888	0.888
CNN	0.854	0.854	0.876	0.865
BiLSTM	0.820	0.820	0.843	0.787
Attention	0.876	0.876	0.809	0.809
CNN-BiLSTM	0.820	0.820	0.843	0.798
CNN-AT	0.831	0.831	0.865	0.854
BiLSTM-AT	0.820	0.820	0.865	0.820
CNN-BiLSTM-AT	0.831	0.831	0.876	0.865

Finally, this article selected a division ratio of 9:1 through simulation experiments, and set the epoch parameters for each neural network model to 180. At epoch=180, all models except BiLSTM achieved higher or equal accuracy with batch=32 compared to batch=64 on Non-Smote data. On Smote data, all models except the Attention model showed higher or equal accuracy with batch=32 compared to batch=64.

4.2 Results

4.2.1 Data preprocessing

4.2.1.1 Sliding window

The main purpose of sliding windows is to increase the sample size by creating different time steps. Each company has 12 quarters of financial indicator data, and w=12 in the table represents the use of data points from the original 12 consecutive quarters,that means there are a total of 890 companies, and according to the simulation, the dataset is divided into a 9:1 ratio, with 801 training sets and 89 testing sets. As the time step decreases (such as w=11, w=10, etc.), the sliding window creates multiple samples from different starting points, thereby increasing the total sample size of the dataset while maintaining partial time series information. The

specific distribution of data after window sliding can be seen in the table.

Table 9 Distribution of sliding window samples

time step	Training data			Testing data		
	ST	Non-ST	Total	ST	Non-ST	Total
w=4	1082	6127	**7209**	133	668	**801**
w=5	982	5426	**6408**	98	614	**712**
w=6	866	4741	**5607**	79	544	**623**
w=7	716	4090	**4806**	94	440	**534**
w=8	619	3386	**4005**	56	389	**445**
w=9	484	2720	**3204**	56	300	**356**
w=10	365	2038	**2403**	40	277	**317**
w=11	243	1359	**1602**	27	151	**178**
w=12	127	674	**801**	8	81	**89**

4.2.1.2 Smote oversampling

For the purpose of data balancing,taking time step 6 (w=6) as an example, the number of samples of ST companies in the original training data was 716, and the number of samples of non-ST companies was 4090. Using the Smote method, new synthetic samples were generated to increase the number of samples of ST companies from 716 to 4090, which was equal to the number of samples of non-ST companies, and it can effectively increase the number of samples of the minority class (ST companies) and make the data more balanced and reduce the bias due to sample imbalance.

Table 10 Data distribution of training set before and after Smote

time step	Training data			Testing data		
	ST	Non-ST	Total	ST	Non-ST	Total
w=4	6127	6127	**12254**	133	668	**801**
w=5	5426	5426	**10852**	98	614	**712**
w=6	4741	4741	**9482**	79	544	**623**
w=7	4090	4090	**8180**	94	440	**534**
w=8	3386	3386	**6772**	56	389	**445**
w=9	2720	2720	**5440**	56	300	**356**
w=10	2038	2038	**4076**	40	277	**317**
w=11	1359	1359	**2718**	27	151	**178**
w=12	674	674	**1348**	8	81	**89**

4.2.2 Indicator selection

4.2.2.1 PCA indicator selection

This article focuses on PCA feature extraction of data after sliding windows, and obtains 20 principal components based on a cumulative variance contribution rate of 95%. Therefore, this article selects 20 principal component indicators to represent all indicators of company operations. Figure 18 shows a heatmap of the correlation coefficients between the extracted 20 principal components and the original 34 indicators. Among them, indicators with high correlation coefficients (absolute value>0.5) include Working Capital, Cash Assets Ratio, Current Liabilities Ratio, Fixed Assets Ratio, Operating Profit Margin, Accounts Receivable Turnover, Inventory Turnover, Accounts Payable Turnover, Fixed Asset Turnover, Total Asset Turnover, Net Profit Margin on Fixed Assets, Operating Profit Rate, Operating Index, Capital Preservation and Appreciation Rate, Fixed Asset Growth Rate, Revenue Growth Rate, Owners' Equity Growth Rate.

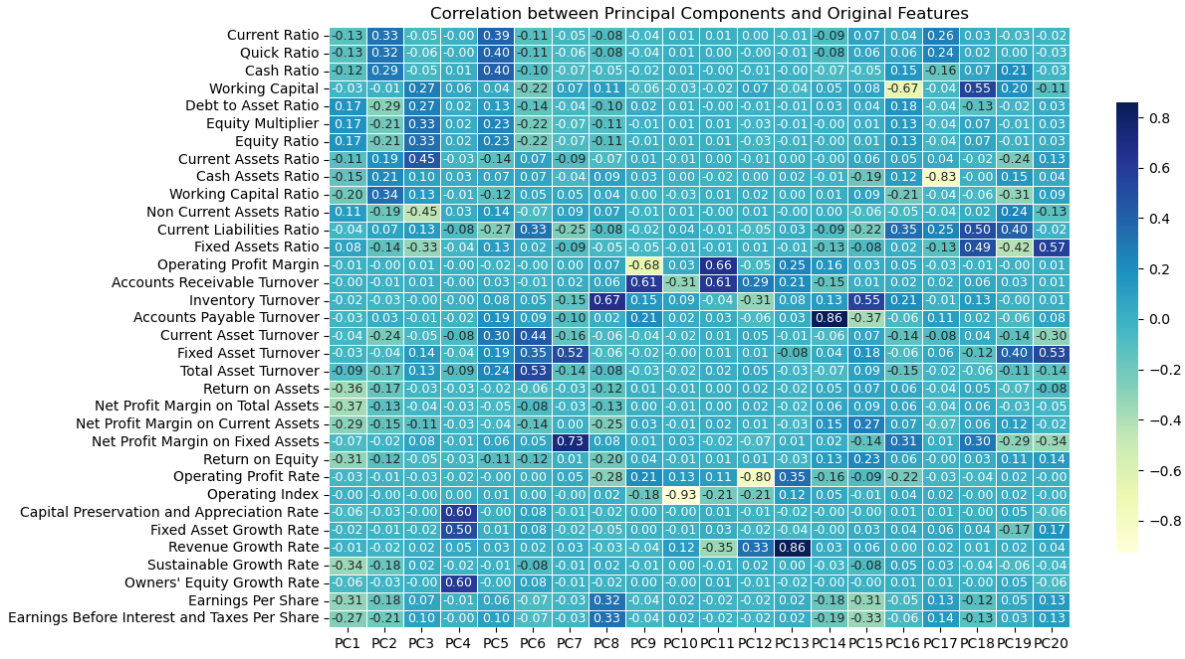


Figure 18 Correlation between principal components and original features

In Figure 19, we analyze the box plot distributions of the top 9 indicators most correlated with the principal components, comparing ST companies and non-ST companies. The analysis highlights key financial differences between the two groups:

(1) Operating Index (-0.93): The box plot shows a wider range of values for non-

ST companies, with a higher median (0.54 vs. 0.38 for ST companies). The lower median and greater negative outliers in ST companies suggest poor operational performance and higher financial instability.

(2) Revenue Growth Rate (0.86): The box plot shows that both groups have a narrow range, with the median for non-ST companies at 0.04 and for ST companies at 0.03. The slight difference suggests weaker growth potential in ST companies, but with relatively low variability.

(3) Accounts Payable Turnover (0.86): The box plot for non-ST companies is slightly higher, with a median of 2.44 compared to 2.10 for ST companies. The smaller spread in ST companies may indicate difficulty in managing short-term liabilities, leading to financial stress.

(4) Cash Assets Ratio (-0.83): The box plot shows that ST companies have a lower median (0.09) compared to non-ST companies (0.12). The lower interquartile range (IQR) and higher presence of outliers in ST companies indicate weaker liquidity and cash reserves.

(5) Operating Profit Rate (-0.80): The box plot distribution for both groups is relatively similar, but ST companies have a lower median (0.06 vs. 0.08 for non-ST companies). This suggests lower profitability and operational efficiency in ST firms.

(6) Net Profit Margin on Fixed Assets (0.73): The box plot indicates that ST companies have a significantly lower median (0.09) compared to non-ST companies (0.13), with a narrower IQR. This highlights inefficient asset utilization and weaker profitability in ST firms.

(7) Operating Profit Margin (-0.68): The box plots for both groups are nearly identical, with a median of 0.98 for non-ST and 0.99 for ST companies. This suggests that, while overall profitability margins are similar, ST companies may struggle in other financial areas.

(8) Working Capital (-0.67): The box plot shows significant variation, with non-ST companies having a much higher median (732,059,561.71 vs. 564,166,915.39 for ST companies). The wider spread in ST companies suggests greater financial instability and capital management issues.

(9) Inventory Turnover (0.67): The box plot shows a slightly higher median for ST companies (1.86) compared to non-ST companies (1.73). However, the presence

of extreme outliers in ST companies suggests inventory fluctuations, which may indicate poor demand forecasting or liquidity issues.

ST companies generally exhibit weaker financial performance across multiple indicators, particularly in liquidity (Cash Assets Ratio, Working Capital), profitability (Operating Profit Rate, Net Profit Margin on Fixed Assets), and operational efficiency (Operating Index). The box plot distributions further highlight the increased financial instability in ST companies, as evidenced by wider spreads, lower medians, and a higher presence of outliers. These factors may play a crucial role in determining why a company is classified as ST.

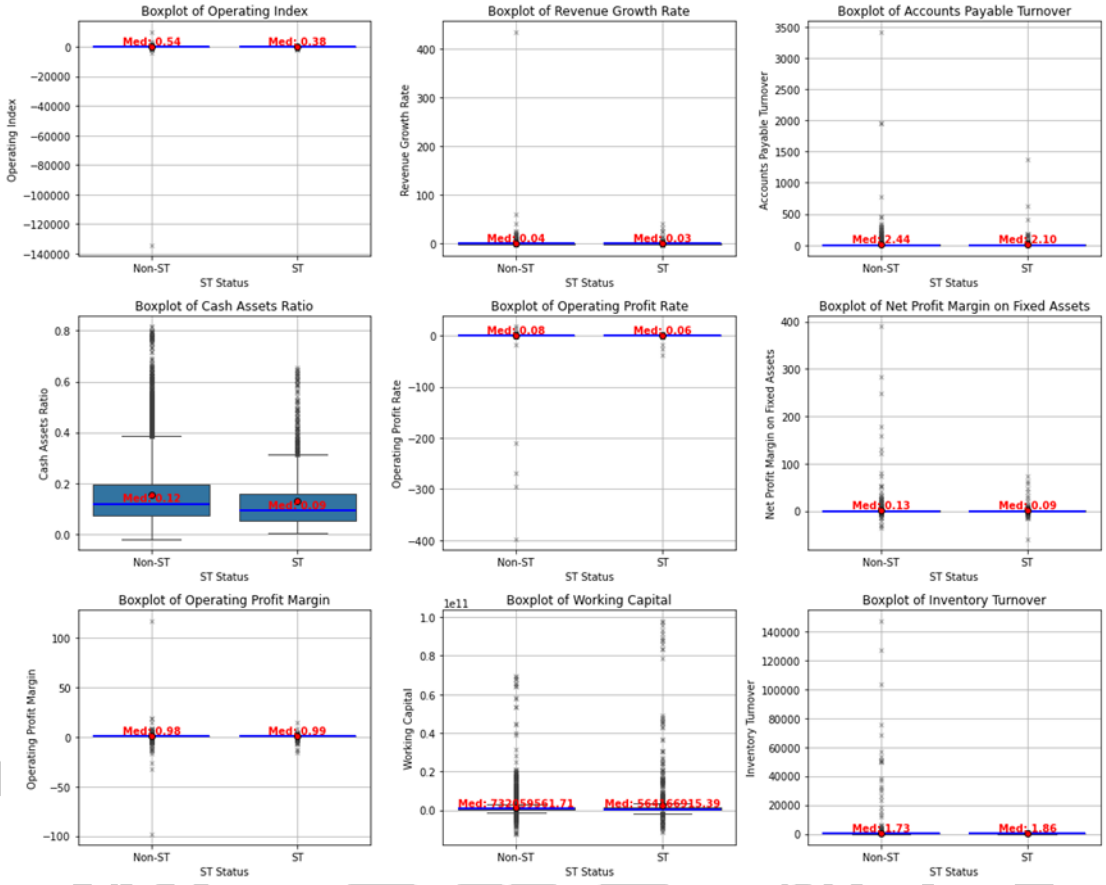


Figure 19 Box Plot Comparison of Key Financial Indicators with PCA Between ST and Non-ST Companies

4.2.2.2 Random forest indicator selection

When using Random Forest for feature extraction, the model selects the most predictive feature for the target variable by evaluating the importance of each feature.

This article also sets a cumulative importance threshold of 95%, and the model retains the most critical features, reducing the number of features from 34 to 31. Among them, the top 10 indicators that contribute the most to the predictive performance of the model are fixed assets ratio, working capital, current liabilities ratio, inventory turnover, accounts receivable turnover, cash ratio, cash assets ratio and quick ratio.

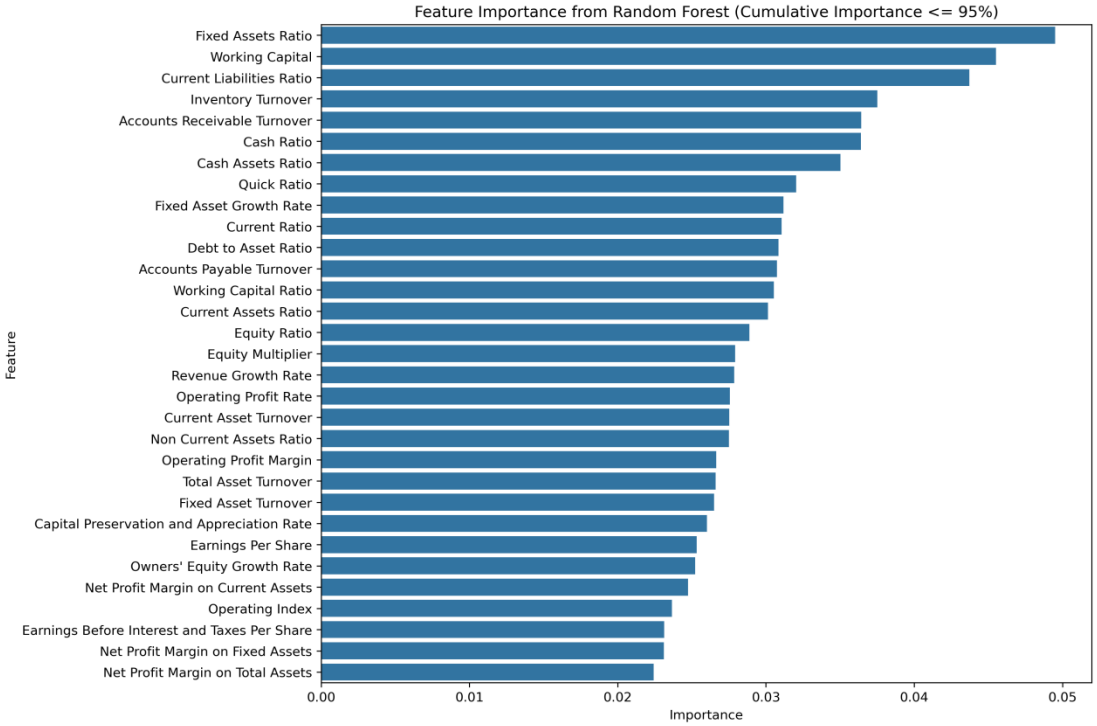


Figure 20 Random forest selection of the important indicators

In Figure 21, we analyze the box plot distributions of the top 9 indicators ranked by importance in the Random Forest model, comparing ST companies and non-ST companies. This analysis highlights key financial differences between the two groups:

(1) Fixed Assets Ratio: The box plot shows a lower median for ST companies (0.12) compared to non-ST companies (0.17), with a wider spread among non-ST firms. This suggests that ST companies invest less in fixed assets, possibly indicating financial instability or a lack of long-term capital investment.

(2) Working Capital: The median for non-ST companies (732,059,561.71) is significantly higher than for ST companies (564,166,915.39). The wider distribution among ST companies indicates greater financial instability and inconsistent capital management, making them more vulnerable to liquidity risks.

(3) Current Liabilities Ratio: Both groups have similar distributions, with a

median of 0.87 for ST companies and 0.89 for non-ST companies. The similar values suggest that ST firms carry a comparable level of short-term liabilities but may struggle more with repayment due to weaker liquidity.

(4) Inventory Turnover: ST companies have a slightly higher median (1.86) than non-ST companies (1.73), but the presence of more extreme outliers in ST firms suggests erratic inventory management. This could indicate poor demand forecasting or pressure to sell inventory at lower margins.

(5) Accounts Receivable Turnover: The box plot shows a lower median for ST companies (1.98) compared to non-ST companies (2.22), indicating slower collection of receivables. This suggests cash flow issues and potential difficulties in managing credit sales.

(6) Cash Ratio: The median for ST companies (0.26) is significantly lower than for non-ST companies (0.41), with a narrower interquartile range (IQR) in ST firms. This highlights weaker cash reserves and a reduced ability to cover short-term obligations, increasing the risk of liquidity crises.

(7) Cash Assets Ratio: The lower median for ST companies (0.09) compared to non-ST companies (0.12) suggests that ST firms maintain lower cash holdings. The increased presence of outliers in ST companies reflects financial distress and inconsistent cash management.

(8) Quick Ratio: ST companies exhibit a lower median (1.18) compared to non-ST companies (1.48), emphasizing their weaker short-term financial health. The narrower IQR in ST firms suggests that most of them face similar liquidity challenges.

(9) Fixed Asset Growth Rate: Both groups show negative median values, with ST companies at -0.02 and non-ST companies at -0.01. The lower growth rate in ST firms suggests limited reinvestment in fixed assets, potentially reflecting financial constraints or business contraction.

ST companies generally exhibit weaker financial performance across multiple dimensions, particularly in liquidity (Cash Ratio, Cash Assets Ratio, Quick Ratio, Working Capital), profitability and efficiency (Accounts Receivable Turnover, Fixed Assets Ratio, Fixed Asset Growth Rate), and operational stability (Inventory Turnover). The box plot distributions highlight the increased financial instability in ST companies, as evidenced by lower medians, wider spreads, and a greater presence

of extreme outliers. These financial weaknesses may contribute significantly to a company's classification as ST.

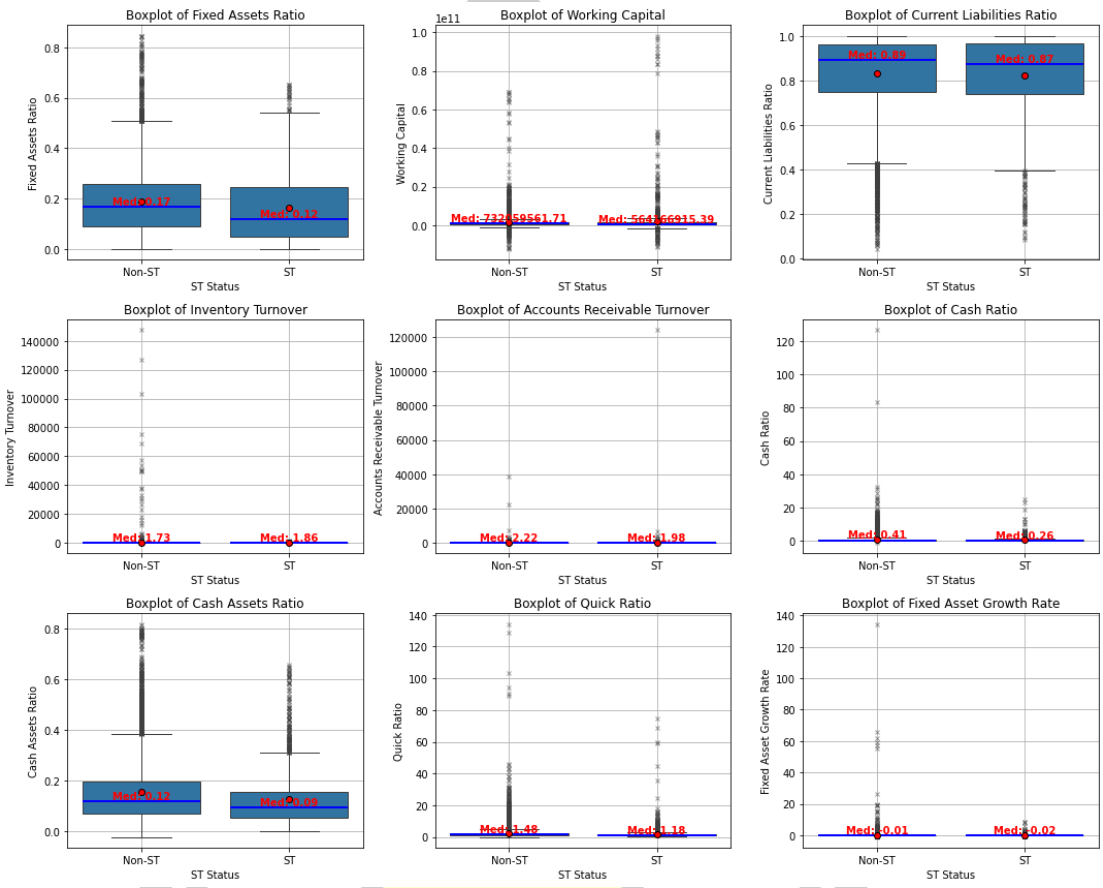


Figure 21 Box Plot Comparison of Key Financial Indicators with Random forest Between ST and Non-ST Companies

4.2.3 Model performance comparison

4.2.3.1 Traditional machine learning models

4.2.3.1.1 Traditional single machine learning models

1.Logistic regression model(LR)

The logistic regression model is set up with a maximum of 500 iterations to ensure convergence and a random seed of 42 to guarantee reproducibility.

Table 11 Parameter settings for LR

Parameters	**Values**
max_iter	500
random_state	42

Table 12 shows the performance of the logistic regression model at different time steps (w=4 to w=12), with or without smote, and using different feature selection methods (PCA and random forest).The overall performance is better when Smote is not used, especially when combined with PCA dimensionality reduction. The accuracy of the model is highest at longer time steps, reaching 0.910. In contrast, the performance of random forest feature extraction is slightly inferior to PCA, but still better than the scheme using Smote.

(1) Smote: Smote was applied to the dataset and the overall performance was relatively low, with accuracy fluctuating between 0.693 and 0.809. The highest accuracy of 0.809 was achieved at a time step of w=12.

(2) Non-Smote: Using only the raw data, the comparative Smote accuracy is significantly higher, ranging from 0.824 to 0.865. The highest accuracy (0.865) is achieved when the time steps are w=8 and w=12, indicating that the model performs best at these time steps without using Smote.

(3) Non-Smote-PCA: Without the use of Smote, the model's accuracy performance improved slightly after PCA was applied to the raw data, ranging from 0.831 to 0.910. The best performance was achieved at w=12 with an accuracy of 0.910, this indicates that PCA has a positive impact on the performance of logistic regression at longer time steps.

(4) Non-Smote-RF: Without using Smote and after applying Random Forest feature extraction, the model performs inferior to PCA, with accuracy fluctuating between 0.826 and 0.868. the highest accuracy of 0.868 is achieved at w=5. The overall performance is better than the Smote and Non-Smote schemes but not as good as the Non-Smote-PCA scheme.

Table 12 Performance of logistic regression models under different conditions

LR	**w=4**	**w=5**	**w=6**	**w=7**	**w=8**	**w=9**	**w=10**	**w=11**	**w=12**
Smote	0.695	0.761	0.693	0.760	0.733	0.705	0.704	0.770	0.809
Non-Smote	0.835	0.861	0.859	0.824	0.865	0.834	0.831	0.848	0.865
Non-Smote-PCA	0.834	0.869	0.865	0.831	0.870	0.843	0.858	0.848	**0.910**
Non-Smote-RF	0.831	0.868	0.851	0.826	0.863	0.834	0.843	0.837	0.854

From Table 13, it can be seen that although the logistic regression model has the highest accuracy of 0.910 under Non-Somte-PCA and performs well in distinguishing

Non ST samples, its performance on ST samples is very weak. Only 50% of the predicted ST samples are correct, and only 12.5% of all actual ST samples are correctly identified.

Table 13 Classification Report of Logistic Regression Models under Non-Smote-PCA

LR	precision	recall	f1-score	accuracy	support
Non-ST	0.92	0.988	0.952	0.910	81
ST	0.5	0.125	0.2		8

2.Support Vector Machine model (SVM)

The support vector machine model is set up with a linear kernel to ensure simplicity and interpretability, with probability estimation enabled and a random seed of 42 to guarantee reproducibility.

Table 14 Parameter settings for SVM

Parameters	Values
kernel	linear
probability	True
random_state	42

Table 15 shows the performance of the SVM model at different time steps (w=4 to w=12), with or without smote, and using different feature selection methods (PCA and random forest). The overall performance is better when Smote is not used, especially when combined with PCA dimensionality reduction. The accuracy of the model is highest at longer time steps, reaching 0.875. In contrast, the performance of random forest feature extraction is slightly inferior to PCA, but still better than the scheme using Smote.

(1) Smote: Smote was applied to the dataset and the overall performance was relatively low, with accuracy fluctuating between 0.695 and 0.831. The highest accuracy of 0.831 was achieved at a time step of w=12.

(2) Non-Smote: Using only the raw data, the comparative Smote accuracy is significantly higher, ranging from 0.830 to 0.867. The highest accuracy (0.867) is achieved when the time step is w=8, indicating that the model performs best at these time steps without using Smote.

(3) Non-Smote-PCA: Without the use of Smote, the model's accuracy

performance improved slightly after PCA was applied to the raw data, ranging from 0.824 to 0.875. The best performance was achieved at w=6 with an accuracy of 0.875, this indicates that PCA has a positive impact on the performance of SVM at longer time steps.

(4) Non-Smote-RF: Without using Smote and after applying Random Forest feature extraction, the model performs inferior to PCA, with accuracy fluctuating between 0.815 and 0.873. the highest accuracy of 0.873 is achieved at w=6. The overall performance is better than the Smote and Non-Smote schemes but not as good as the Non-Smote-PCA scheme.

Table 15 Performance of SVM models under different conditions

SVM	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Smote	0.702	0.757	0.695	0.747	0.73	0.719	0.685	0.787	0.831
Non-Smote	0.834	0.862	0.865	0.83	0.867	0.843	0.835	0.837	0.854
Non-Smote-PCA	0.834	0.862	**0.875**	0.824	0.872	0.84	0.85	0.854	0.843
Non-Smote-RF	0.834	0.862	0.873	0.826	0.872	0.843	0.85	0.815	0.865

From Table 16, it can be seen that although the SVM model has the highest accuracy of 0.875 under Non-Smote-PCA and performs well in distinguishing Non-ST samples, its performance on ST samples is very weak. 66.7% of the predicted ST samples are correct, and only 2.5% of all actual ST samples are correctly identified.

Table 16 Classification Report of SVM Models under Non-Smote-PCA

SVM	precision	recall	f1-score	accuracy	support
Non-ST	0.876	0.998	0.933	0.875	544
ST	0.667	0.025	0.049		79

3.Decision Tree model (DT)

The decision tree model is set up with the gini impurity criterion for splitting, the "best" splitter for selecting the best feature at each node, and a random seed of 42 to ensure consistent results during tree building.

Table 17 Parameter settings for DT

Parameters	Values
criterion	gini
splitter	best
random_state	42

Table 18 shows the performance of the DT model at different time steps (w=4 to w=12), with or without smote, and using different feature selection methods (PCA and random forest).The overall performance is better when Smote is not used, especially when combined with random forest indicator selection. The accuracy of the model is highest at longer time steps, reaching 0.899. In contrast, the performance of PCA is slightly inferior to random forest, but still better than the scheme using Smote.

(1) Smote: Smote was applied to the dataset and the overall performance was relatively low, with accuracy fluctuating between 0.723 and 0.820. The highest accuracy of 0.820 was achieved at a time step of w=11.

(2) Non-Smote: Using only the raw data, the comparative Smote accuracy is significantly higher, ranging from 0.778 to 0.879. The highest accuracy (0.879) is achieved when the time step is w=8, indicating that the model performs best at these time steps without using Smote.

(3) Non-Smote-PCA: Without the use of Smote, the model's accuracy performance improved slightly after PCA was applied to the raw data, ranging from 0.764 to 0.896. The best performance was achieved at w=5 with an accuracy of 0.896, this indicates that PCA has a positive impact on the performance of DT at longer time steps.

(4) Non-Smote-RF: Without using Smote and after applying Random Forest feature extraction, the performance of this model exceeds that of PCA, with accuracy fluctuating between 0.781 and 0.899. the highest accuracy of 0.899 is achieved at w=12. The overall performance is better than other schemes.

Table 18 Performance of Decision Tree models under different conditions

Decision Tree	**w=4**	**w=5**	**w=6**	**w=7**	**w=8**	**w=9**	**w=10**	**w=11**	**w=12**
Smote	0.765	0.788	0.796	0.79	0.76	0.781	0.723	0.82	0.809
Non-Smote	0.834	0.827	0.854	0.854	0.879	0.778	0.839	0.809	0.854
Non-Smote-PCA	0.819	0.896	0.835	0.822	0.874	0.803	0.801	0.764	0.798
Non-Smote-RF	0.839	0.834	0.835	0.83	0.843	0.812	0.843	0.781	**0.899**

From Table 19, it can be seen that although the decision tree model has the highest accuracy of 0.899 under Non-Smote-RF and performs well in distinguishing Non-ST samples, its performance on ST samples is very weak. Only 44.4% of the

predicted ST samples are correct, and 50% of all actual ST samples are correctly identified.

Table 19 Classification Report of DT Model under Non-Smote-RF

Decision Tree	precision	recall	f1-score	accuracy	support
Non-ST	0.95	0.938	0.944	0.899	81
ST	0.444	0.5	0.471		8

4.BP neural network model (BP)

BP neural network is set up with two hidden layers containing 128 and 64 neurons respectively, with a maximum of 300 iterations for training and a random seed of 42 to guarantee reproducibility.

Table 20 Parameter settings for BP

Parameters	Values
1st hidden_layer	128
2nd hidden_layer	64
max_iter	300

Table 21 shows the performance of the BP neural network model at different time steps (w=4 to w=12), with or without smote, and using different feature selection methods (PCA and random forest).The overall performance is better when Smote is not used, especially when combined with PCA indicator selection. The Non-Smote-PCA and Non-Smote-RF schemes perform the best at w=8, achieving accuracies of 0.919 and 0.912, respectively.

(1) Smote: Smote was applied to the dataset and the overall performance was relatively low, with accuracy fluctuating between 0.823 and 0.905. The highest accuracy of 0.905 was achieved at a time step of w=6.

(2) Non-Smote: Using only the raw data, the comparative Smote accuracy is significantly higher, ranging from 0.837 to 0.908. The highest accuracy (0.908) is achieved when the time step is w=8, indicating that the model performs best at these time steps without using Smote.

(3) Non-Smote-PCA: Without the use of Smote, the model's accuracy performance improved slightly after PCA was applied to the raw data, ranging from

0.844 to 0.919. The best performance was achieved at w=8 with an accuracy of 0.919, this indicates that PCA has a positive impact on the performance of BP at longer time steps.

(4) Non-Smote-RF: Without using Smote and after applying Random Forest feature extraction, the model performs inferior to PCA, with accuracy fluctuating between 0.868 and 0.912. the highest accuracy of 0.912 is achieved at w=8. The overall performance is better the Smote and Non-Smote schemes but not as good as the Non-Smote-PCA scheme.

Table 21 Performance of BP models under different conditions

BP	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Smote	0.895	0.889	0.905	0.873	0.901	0.823	0.85	0.86	0.888
Non-Smote	0.894	0.9	0.902	0.878	0.908	0.837	0.861	0.865	0.865
Non-Smote-PCA	0.895	0.844	0.9	0.878	**0.919**	0.851	0.869	0.871	0.854
Non-Smote-RF	0.885	0.879	0.883	0.91	0.912	0.868	0.869	0.876	0.888

From Table 22, it can be seen that although the BP neural network model has the highest accuracy of 0.919 under Non-Smote-PCA and performs well in distinguishing Non-ST samples, its performance on ST samples is weak.70% of the predicted ST samples are correct, and 65% of all actual ST samples are correctly identified.

Table 22 Classification Report of BP Model under Non-Smote-PCA

BP	precision	recall	f1-score	accuracy	support
Non-ST	0.947	0.961	0.954	0.919	389
ST	0.700	0.625	0.660		56

In summary, among the machine learning models used in this article, the BP model performs better than the logistic regression model, SVM model, and decision tree model, but its accuracy in predicting ST companies is still not high.

4.2.3.1.2 Ensemble of traditional machine learning models

1.Random forest model (RF)

Random forest model is set up with 100 estimators (trees), using the Gini impurity criterion for splitting nodes, and a random seed of 42 to ensure reproducibility and consistent results across runs.

Table 23 Parameter settings for RF

Parameters	Values
n_estimators	100
criterion	gini
random_state	42

Table 24 shows the performance of the random forest model at different time steps (w=4 to w=12), with the highest accuracy of 0.926 at w=8 and Smote-RF as the optimal configurations.

(1) Non-Smote: uses only raw data and has relatively low overall performance, ranging from 0.820 to 0.899. The highest accuracy of 0.899 is achieved when the time step is w=8.

(2) Non-Smote-PCA: After applying PCA to the raw data without using Smote, the accuracy of the model is not improved and remains at 0.899.

(3) Non-Smote-RF: After not using Smote and applying Random Forest feature extraction, the model outperforms Non-Smote-PCA (0.921). overall performance is better than the Smote and Non-Smote schemes but not as good as the Smote-RF scheme.

(4) Smote: The Smote algorithm is applied to the dataset and the overall performance is better than Non-Smote, with accuracy fluctuating between 0.854 and 0.917. The highest accuracy of 0.917 is achieved at a time step of w=8.

(5) Smote-PCA: After using Smote algorithm and applying PCA extraction to the dataset, the accuracy of the model is not improved and remains at 0.899.

(6) Smote-RF: After using Smote and applying Random Forest feature extraction, the model performs (0.926) with the highest accuracy and outperforms the other schemes.

Table 24 Performance of RF models under different conditions

Random Forest	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Smote	0.9	0.904	0.91	0.899	0.917	0.899	0.876	0.876	0.854
Smote-PCA	0.878	0.885	0.889	0.886	0.899	0.879	0.869	0.843	0.82
Smote-RF	0.903	0.913	0.913	0.895	**0.926**	0.89	0.88	0.893	0.854
Non-Smote	0.869	0.897	0.91	0.865	0.899	0.868	0.865	0.871	0.899
Non-Smote-PCA	0.853	0.881	0.878	0.841	0.874	0.843	0.85	0.848	0.899
Non-Smote-RF	0.873	0.895	0.9	0.871	0.894	0.882	0.876	0.882	0.921

From Table 25, it can be seen that although the Random Forest model has the highest accuracy of 0.926 under the Smote-RF scheme and performs well in distinguishing non-ST samples, it performs weakly on ST samples, with 79.5% of the predicted ST samples being correct, and 65% of the actual ST samples being correctly identified.

Table 25 Classification Report of RF Model under Smote-RF

Random Forest	precision	recall	f1-score	accuracy	support
Non-ST	0.938	0.979	0.958	0.926	389
ST	0.795	0.554	0.653		56

2.XGBoost model

The XGBoost model is set up with gbtree as the booster, with 100 trees (n_estimators), logloss as the evaluation metric, and a learning rate of 0.3.

Table 26 Parameter settings for XGBoost

Parameters	Values
booster	gbtree
n_estimators	100
eval_metric	logloss
learning_rate	0.3

Table 27 shows the performance of the XGBoost model at different time steps (w=4 to w=12), with the highest accuracy of 0.920 at w=6 and Non-Smote-RF as the optimal configurations.

(1) Non-Smote: only raw data is used, ranging from 0.854 to 0.912. Accuracy 0.912 is highest when the time step is w=6.

(2) Non-Smote-PCA: After applying PCA to the raw data without using Smote, the accuracy of the model is not improved and remains at 0.912.

(3) Non-Smote-RF: After not using Smote and applying Random Forest feature extraction, the model outperforms the other schemes (0.920).

(4) Smote: applying the Smote algorithm to the dataset outperforms Non-Smote overall. the highest accuracy of 0.919 is achieved at a time step of w = 8.

(5) Smote-PCA: After using Smote algorithm on the dataset and applying PCA extraction, the accuracy of the model did not improve and decreased to 0.917.

(6) Smote-RF: After using Smote and applying Random Forest feature extraction, the accuracy of the model did not improve and decreased to 0.916.

Table 27 Performance of XGBoost models under different conditions

XGBoost	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Smote	0.903	0.893	0.913	0.884	0.919	0.89	0.884	0.882	0.888
Smote-PCA	0.883	0.882	0.884	0.88	0.917	0.874	0.88	0.837	0.854
Smote-RF	0.895	0.888	0.899	0.901	0.912	0.916	0.891	0.871	0.899
Non-Smote	0.886	0.904	0.912	0.895	0.906	0.904	0.884	0.876	0.91
Non-Smote-PCA	0.884	0.893	0.912	0.882	0.897	0.862	0.854	0.876	0.899
Non-Smote-RF	0.899	0.899	**0.920**	0.888	0.912	0.907	0.884	0.854	0.865

From Table 28, although the XGBoost model has the highest accuracy under the Non-Smote-RF scheme, reaching 0.920, and performs well in distinguishing non-ST samples, its performance on ST samples is weak. 83.7% of predicted ST samples are correct, while only 45.6% are correctly identified in actual ST samples.

Table 28 Classification Report of XGBoost Model under Non-Smote-RF

XGBoost	precision	recall	f1-score	accuracy	support
Non-ST	0.926	0.987	0.956	0.920	544
ST	0.837	0.456	0.590		79

3.Stacking Algorithm - Layered Model (Stacking)

The Stacking model defines four base learners: logistic regression, SVM, random forest, and XGBoost. Logistic regression is used as the Meta Learner, with a maximum iteration of 200 and a random seed of 42.

Table 29 Parameter settings for Stacking

Parameters	Values
base learners	LR,SVM,RF,XGBoost
Meta learner	LR
max_iter	200
random_state	42

Table 30 shows the performance of the Stacking model at different time steps (w=4 to w=12), with the highest accuracy of 0.926 at w=6 and SMOTE as the optimal configurations.

(1) Non-Smote: only raw data is used, ranging from 0.880 to 0.915. Accuracy

0.921 is the highest when the time step is w=6.

(2) Non-Smote-PCA: After applying PCA to the raw data without using Smote, the accuracy of the model is not improved and drops to 0.915.

(3) Non-Smote-RF: After not using Smote and applying Random Forest feature extraction, the model outperforms (0.925) the Non-Smote and Non-Smote-PCA schemes.

(4) Smote: using Smote algorithm on the dataset, the overall performance is better than Non-Smote. the highest accuracy of 0.926 is achieved at a time step of w=6, which outperforms the other schemes.

(5) Smote-PCA: After using Smote algorithm on the dataset and applying PCA extraction, the accuracy of the model did not improve and decreased to 0.915.

(6) Smote-RF: After using Smote and applying Random Forest feature extraction, the accuracy of the model is not improved and decreases to 0.917.

Table 30 Performance of Stacking models under different conditions

Stacking	**w=4**	**w=5**	**w=6**	**w=7**	**w=8**	**w=9**	**w=10**	**w=11**	**w=12**
Smote	0.905	0.899	**0.926**	0.888	0.917	0.896	0.884	0.893	0.899
Smote-PCA	0.89	0.9	0.907	0.884	0.915	0.888	0.884	0.86	0.865
Smote-RF	0.906	0.897	0.915	0.899	0.917	0.907	0.884	0.888	0.91
Non-Smote	0.893	0.907	0.918	0.906	0.921	0.919	0.888	0.899	0.888
Non-Smote-PCA	0.893	0.899	0.915	0.88	0.89	0.865	0.854	0.876	0.91
Non-Smote-RF	0.906	0.923	0.925	0.891	0.917	0.916	0.891	0.882	0.888

From Table 31, it can be seen that although the Stacking model has the highest accuracy of 0.926 under the Somte scheme and performs well in distinguishing non-ST samples, it performs weakly on ST samples, with 77.0% of the predicted ST samples being correct, and only 59.5% of the actual ST samples being correctly identified.

Table 31 Classification Report of Stacking Model under Smote

Stacking	**precision**	**recall**	**f1-score**	**accuracy**	**support**
Non-ST	0.943	0.974	0.958	0.926	544
ST	0.770	0.595	0.671		79

4.2.3.1.3 Summary of Traditional Machine Learning Models

Table 32 shows the highest accuracy of different single and ensemble machine learning models under specific conditions. Among the individual models, the BP neural network achieved the highest accuracy of 0.919 under the "Non-Smote-PCA" and w=8 condition. For ensemble models, both the Random Forest (RF) and Stacking models reached the highest accuracy of 0.926 under different conditions, with RF under "Smote-RF" and w=8, and Stacking under "SMOTE" and w=6. This indicates that ensemble models generally outperform individual models, and different preprocessing methods and time steps significantly impact model performance.

Table 32 Best single and ensemble machine learning model accuracy

Model		**Conditions**	**Highest accuracy**
Single machine learning model	LR	Non-Smote-PCA and w=12	0.910
	SVM	Non-Smote-PCA and w=6	0.875
	DT	Non-Smote-RF and w=12	0.899
	BP	Non-Smote-PCA and w=8	0.919
Ensemble machine learning models	RF	Smote-RF and w=8	**0.926**
	XGBoost	Non-Smote-RF and w=6	0.920
	Stacking	Smote and w=6	**0.926**

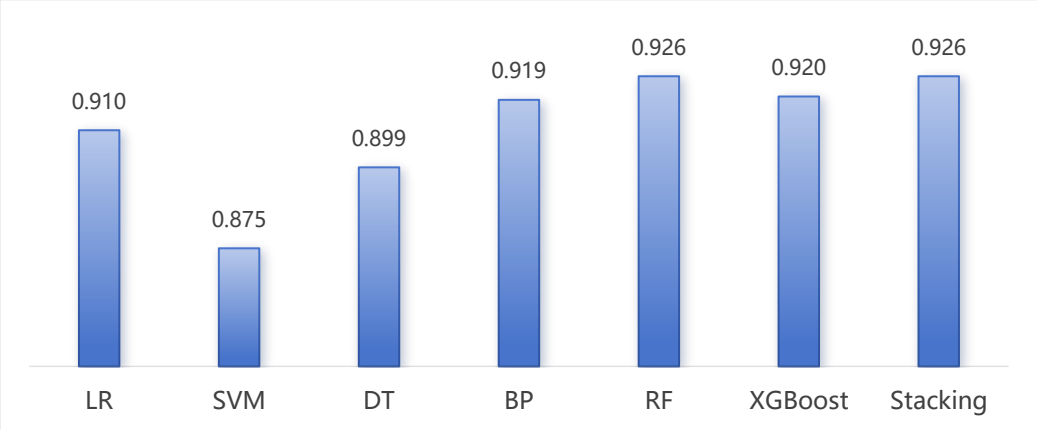


Figure 22 Traditional machine learning model accuracy

4.2.3.2 Deep learning models

The hyperband algorithm was used to explore the hyperparameter space of various layers in the deep learning model. The specific search ranges for each parameter are outlined in the table 33:

Table 33 Using hyperband algorithm to find the optimal parameters

Layers	Parameters	Values Scope	Step
Convolutional Layer	filters	[16,64]	16
Dropout1 Layer	rate	[0.1,0.5]	0.1
BiLSTM Layer	units	[32,128]	32
Dropout2 Layer	rate	[0.1,0.5]	0.1
Dense1 Layer	units	[32,256]	32
Dropout3 Layer	rate	[0.1,0.5]	0.1
Dense2 Layer	units	[32,256]	32
Dropout4 Layer	rate	[0.1,0.5]	0.1

4.2.3.2.1 Single deep learning models

1.Convolutional Neural Networks model (CNN)

The CNN model is set up with 64 filters with a kernel size of 3 for feature extraction and a pool size of 2 for dimensionality reduction. Dropout rates of 0.5, 0.1, and 0.3 are applied to prevent overfitting. Two dense layers, each with 128 units, process high-level features, and the Adam optimizer ensures efficient and stable training.

Table 34 Parameter settings for CNN

Parameters	Values	Parameters	Values
filters	64	dropout2	0.1
kernel_size	3	dense2_units	128
pool_size	2	dropout3	0.3
dropout1	0.5	optimizer	adam
dense1_units	128		

The specific parameter settings are shown in the table below. The five fold cross validation method is used to compare CNN models under different conditions. Table 35 shows the performance of the CNN model at different time steps (w=4 to w=12). When w=8 and Smote-RF is used as the optimal configurations, the accuracy is highest at 0.981.

(1) Non-Smote: Using only raw data, ranging from 0.880 to 0.963. When the time step is w=7, the accuracy is highest at 0.963.

(2) Smote: Using the Smote algorithm on the dataset, the overall performance is better than Non-Smote. When the time step is w=6, the accuracy is highest at 0.978.

(3) Smote-PCA: After applying the Smote algorithm and PCA extraction to the dataset, the accuracy of the model was decreased to 0.974 at a time step of w=10.

(4) Smote-RF: After using Smote and applying random forest feature extraction, the accuracy of the model was improved to 0.981, which is superior to other schemes.

Table 35 Performance of CNN under different conditions

CNN	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Non-Smote	0.95	0.951	0.958	0.963	0.957	0.959	0.934	0.911	0.888
Smote	0.944	0.955	0.978	0.974	0.974	0.971	0.966	0.966	0.876
Smote-PCA	0.955	0.957	0.97	0.964	0.973	0.963	0.974	0.966	0.843
Smote-RF	0.939	0.962	0.975	0.977	**0.981**	0.966	0.963	0.955	0.832

From Table 36, it can be seen that the CNN model has the highest accuracy under the Smote-RF scheme, reaching 0.981. Choosing this model as the optimal model, the test set data was used to train and predict the optimal model, with an accuracy of 0.981. Compared to traditional machine learning models, it performs better in distinguishing between non-ST and ST samples, with 98.1% being correct in predicting ST samples and 87.3% being correctly identified in actual ST samples.

Table 36 Classification Report of CNN under Smote-RF

CNN	precision	recall	f1-score	accuracy	support
Non-ST	0.981	0.997	0.989	0.981	389
ST	0.981	0.873	0.924		56

The figure 23 shows that the loss value trend and accuracy trend of the training iterations are relatively stable. From figure 24, an AUC of 0.99 indicates that the CNN model has excellent ability to distinguish between classes, with nearly perfect classification performance.

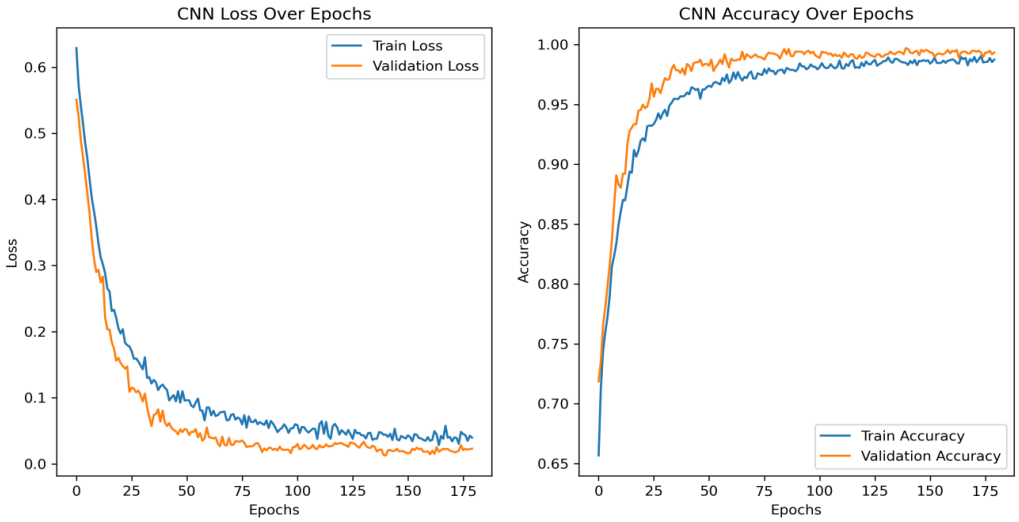


Figure 23 Trend chart of accuracy and loss value of CNN model

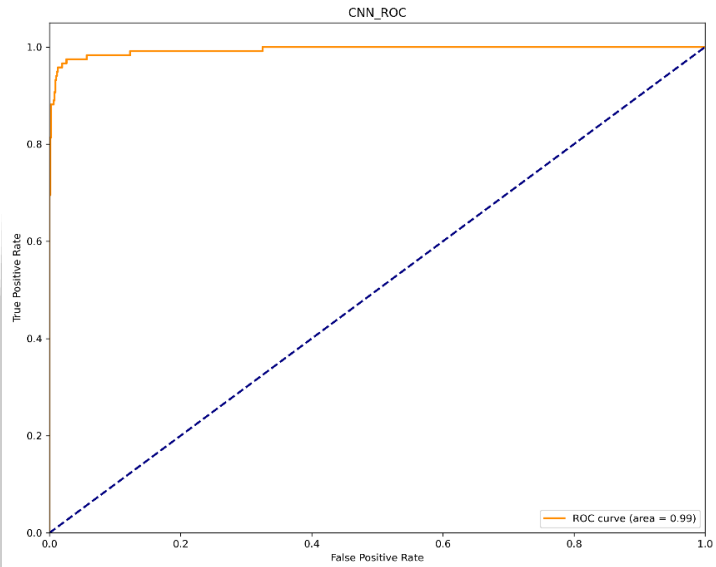


Figure 24 The ROC curve of the CNN model

2.Bi-directional Long Short-Term Memory model (BiLSTM)

The BiLSTM model is configured with two LSTM layers (64 and 128 units) to capture temporal patterns, with dropout rates of 0.2 and 0.1 to reduce overfitting. It includes two fully connected layers (128 units each) with dropout rates of 0.1 and 0.3, ensuring regularization. The Adam optimizer is used for efficient training. These settings aim to balance model complexity and enhance performance on sequential data.The specific parameter settings are shown in the table below.

Table 37 Parameter settings for BiLSTM

Parameters	Values	Parameters	Values
lstm1_units	64	dropout3	0.1
dropout1	0.2	dense2_units	128
lstm2_units	128	dropout4	0.3
dropout2	0.1	optimizer	adam
dense1_units	128		

The five fold cross validation method is used to compare BiLSTM models under different conditions. Table 38 shows the performance of the BiLSTM model at different time steps (w=4 to w=12), with the highest accuracy of 0.991 at w=7 and Smote as the optimal configurations.

Non-Smote: Using only raw data, ranging from 0.854 to 0.977. When the time step is w=9, the accuracy is highest at 0.977.

Smote: Using the Smote algorithm on the dataset, the overall performance is better than Non Smote. When the time step is w=8, the accuracy is highest at 0.991.

Smote-PCA: After applying the Smote algorithm and PCA extraction to the dataset, the accuracy of the model was improved to 0.988 at a time step of w=6.

Smote-RF: After using Smote and applying random forest feature extraction, the accuracy of the model did not improve, and the accuracy of the model remained at 0.989.

Table 38 Performance of BiLSTM under different conditions

BiLSTM	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Non-Smote	0.976	0.963	0.975	0.970	0.975	0.977	0.963	0.939	0.854
Smote	0.980	0.979	0.989	0.989	**0.991**	0.988	0.970	0.946	0.843
Smote-PCA	0.973	0.983	0.988	0.978	0.978	0.983	0.970	0.967	0.865
Smote-RF	0.980	0.978	0.989	0.987	0.989	0.986	0.981	0.977	0.820

From Table 39, it can be seen that the BiLSTM model has the highest accuracy under the Smote scheme. Choosing this model as the optimal model and using the test set data for training and prediction, the accuracy is 0.991. Compared with traditional machine learning models and CNN models, it performs better in distinguishing between non ST samples and ST samples, with 97.5% of predicted ST samples being correct and 96.3% correctly identified in actual ST samples.

Table 39 Classification Report of BiLSTM under Smote

BiLSTM	precision	recall	f1-score	accuracy	support
Non-ST	0.993	0.996	0.994	0.991	389
ST	0.975	0.963	0.969		56

The figure 25 shows that the loss value trend and accuracy trend of the training iterations are relatively stable. From figure 26, an AUC of 0.99 indicates that the BiLSTM model has excellent ability to distinguish between classes, with nearly perfect classification performance.

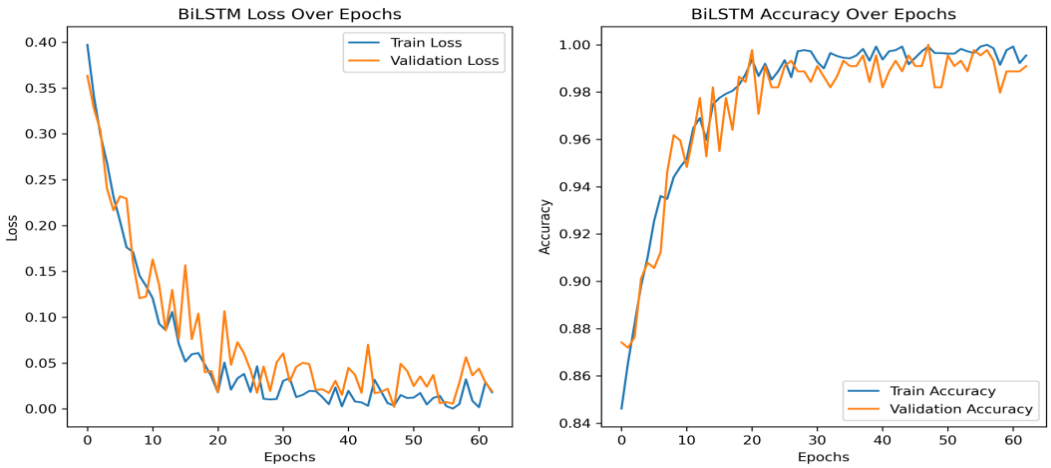


Figure 25 Trend chart of accuracy and loss value of BiLSTM model

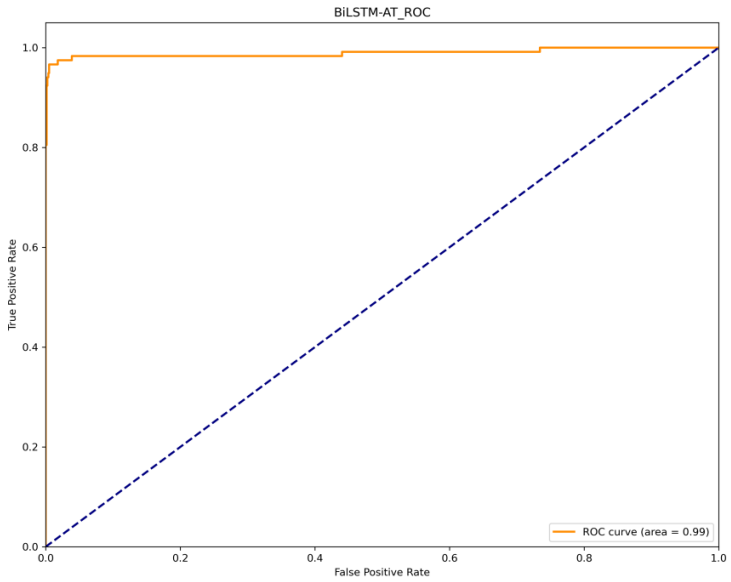


Figure 26 The ROC curve of the BiLSTM model

3.Attention mechanism model (AT)

The Attention (AT) model is set up with scaling in the attention mechanism and calculates attention scores with the dot product. It has two dense layers with 128 units each, with dropout rates of 0.1 and 0.3. The adam optimizer is used for training. These settings help optimize performance and reduce overfitting. The specific parameter settings are shown in the table 40.

Table 40 Parameter settings for AT

Parameters	Values	Parameters	Values
attention_use_scale	TRUE	dense2_units	128
attention_score_mode	dot	dropout2	0.3
dense1_units	128	optimizer	adam
dropout1	0.1		

The five fold cross validation method is used to compare AT models under different conditions. Table 41 shows the performance of the AT model at different time steps (w=4 to w=12), with the highest accuracy of 0.949 at w=6 and Smote as the optimal configurations.

Non-Smote: Using only raw data, ranging from 0.831 to 0.937. When the time step is w=5, the accuracy is highest at 0.937.

Smote: Using the Smote algorithm on the dataset, the overall performance is better than Non Smote. When the time step is w=6 the accuracy is highest at 0.949.

Smote-PCA: After applying the Smote algorithm and PCA extraction to the dataset, the accuracy of the model was decreased to 0.930 at a time step of w=6.

Smote-RF: After using Smote and applying random forest feature extraction, the accuracy of the model was at 0.948.

Table 41 Performance of AT under different conditions

Attention	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Non-Smote	0.930	0.937	0.929	0.932	0.928	0.905	0.897	0.877	0.831
Smote	0.934	0.933	**0.949**	0.944	0.930	0.944	0.914	0.899	0.809
Smote-PCA	0.925	0.923	0.936	0.930	0.918	0.919	0.906	0.862	0.876
Smote-RF	0.933	0.928	0.945	0.948	0.939	0.934	0.906	0.896	0.843

From Table 42, it can be seen that the AT model has the highest accuracy under the Smote scheme. Choosing this model as the optimal model and using the test set

data for training and prediction, the accuracy is 0.949. However, compared to the CNN and BiLSTM models, the prediction accuracy is lower. Compared with traditional machine learning models, it performs better in distinguishing between non-ST samples and ST samples, with 87.0% of predicted ST samples being correct and 74.3% correctly identified in actual ST samples.

Table 42 Classification Report of AT under Smote-RF

Attention	precision	recall	f1-score	accuracy	support
Non-ST	0.960	0.982	0.971	0.949	544
ST	0.870	0.743	0.801		79

The figure 27 shows that the loss value trend and accuracy trend of the training iterations are relatively stable. From figure 28, an AUC of 0.94 indicates that the AT model has excellent discriminatory ability, with a high capability to distinguish between the positive and negative classes. However, it is lower than the AUC values achieved by the CNN and BiLSTM models.

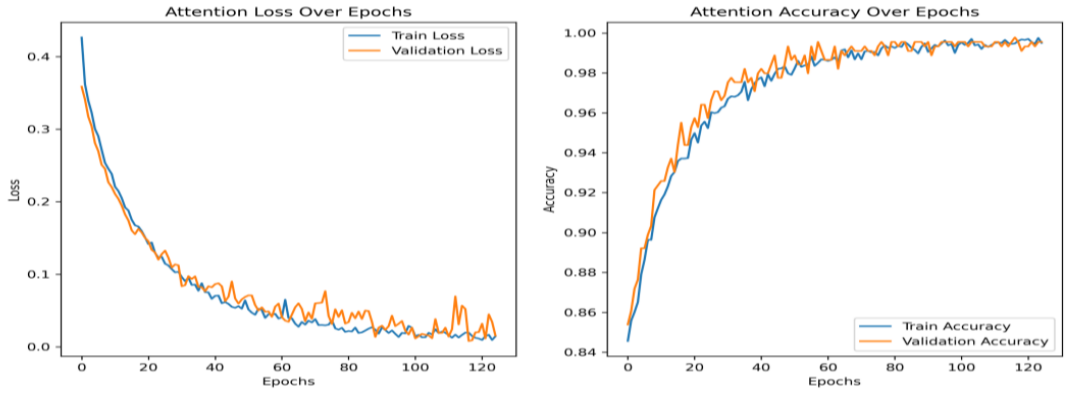


Figure 27 Trend chart of accuracy and loss value of AT model

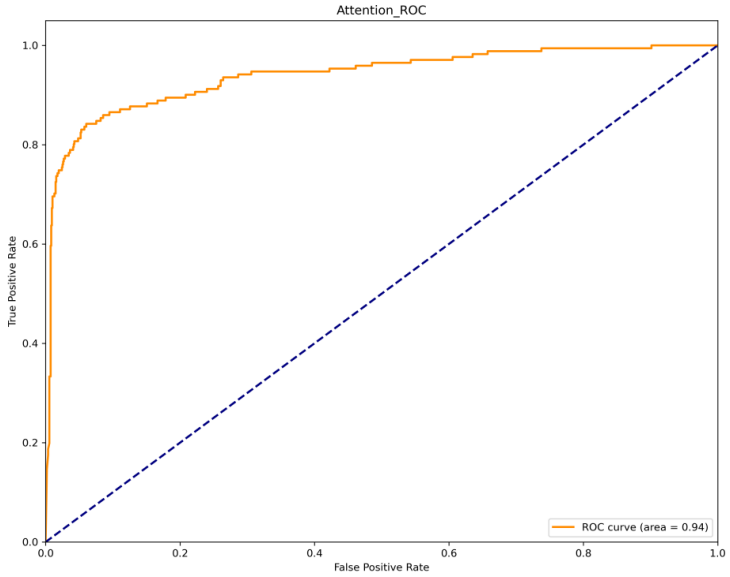


Figure 28 The ROC curve of the AT model

4.2.3.2.2 Ensemble of Deep Learning Models

1.CNN-BiLSTM model

The CNN-BiLSTM model is set up with 64 convolutional filters, a kernel size of 3, and a pooling size of 2, along with a 50% dropout to prevent overfitting. It includes two bidirectional LSTM layers with 64 and 128 units, applying 20% and 10% dropout, respectively. The fully connected layers have 128 neurons each, with 10% and 30% dropout. The model uses the Adam optimizer for efficient and effective optimization. The specific parameter settings are shown in the table 43.

Table 43 Parameter settings for CNN-BiLSTM

Parameters	**Values**	**Parameters**	**Values**
filters	64	dropout3	0.1
kernel_size	3	dense1_units	128
pool_size	2	dropout4	0.1
dropout1	0.5	dense2_units	128
lstm1_units	64	dropout5	0.3
dropout2	0.2	optimizer	adam
lstm2_units	128		

Table 44 shows the performance of the CNN-BiLSTM model at different time steps (w=4 to w=12), with the highest accuracy of 0.993 at w=9 and Smote, Smote-PCA and Smote-RF as the optimal configurations.

Non-Smote: Using only raw data, ranging from 0.843 to 0.977. When the time step is w=8, the accuracy is highest at 0.977.

Smote: Using the Smote algorithm on the dataset, the overall performance is better than Non Smote. When the time step is w=8, the accuracy is highest at 0.993.

Smote-PCA: After applying the Smote algorithm and PCA extraction to the dataset, the accuracy of the model was improved to 0.993 at a time step of w=8.

Smote-RF: After using Smote and applying random forest feature extraction, the accuracy of the model improved to 0.993 at a time step of w=8.

Table 44 Performance of CNN-BiLSTM under different conditions

CNN-BiLSTM	**w=4**	**w=5**	**w=6**	**w=7**	**w=8**	**w=9**	**w=10**	**w=11**	**w=12**
Non-Smote	0.95	0.956	0.971	0.973	0.977	0.971	0.963	0.932	0.843
Smote	0.964	0.955	0.986	0.978	0.987	**0.993**	0.981	0.979	0.843
Smote-PCA	0.963	0.964	0.987	0.981	0.992	**0.993**	0.981	0.978	0.809
Smote-RF	0.959	0.962	0.984	0.986	0.990	**0.993**	0.985	0.961	0.843

Taking Smote-RF conditions as an example, from Table 45, it can be seen that the CNN-BiLSTM model has the highest accuracy under the Smote-RF scheme. Choosing this model as the optimal model, the test set data was used to train and predict the optimal model with an accuracy of 0.993. Compared with traditional machine learning models, single deep learning models, it performs better in distinguishing between non ST samples and ST samples. Among them, it is 99% correct in predicting ST samples and 96.2% correctly recognized in actual ST samples.

Table 45 Classification Report of CNN-BiLSTM under Smote-RF

CNN-BiLSTM	**precision**	**recall**	**f1-score**	**accuracy**	**support**
Non-ST	0.993	0.998	0.996	0.993	300
ST	0.990	0.962	0.976		56

The Figure 29 shows that the loss and accuracy trends during training iterations are relatively stable, indicating good convergence of the model. From Figure 30, an AUC of 1 demonstrates that the CNN-BiLSTM model achieves perfect discrimination between classes, reflecting its exceptional performance in the classification task.

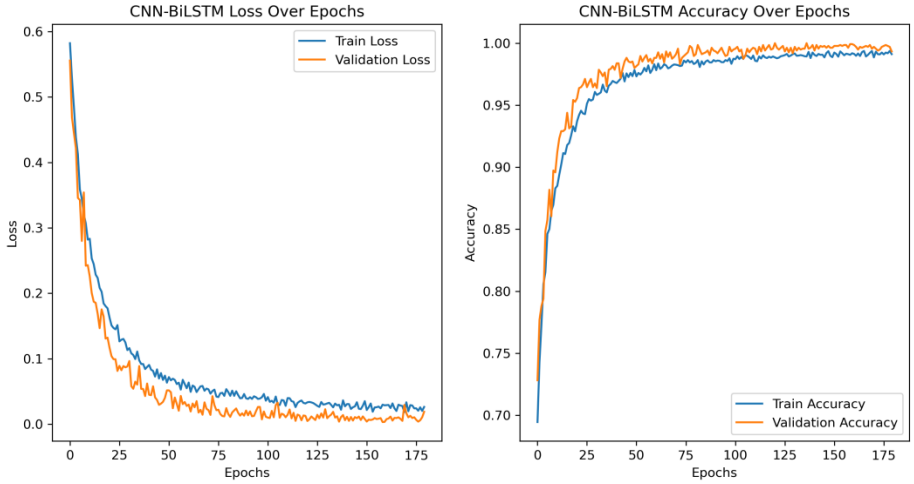


Figure 29 Trend chart of accuracy and loss value of CNN-BiLSTM model

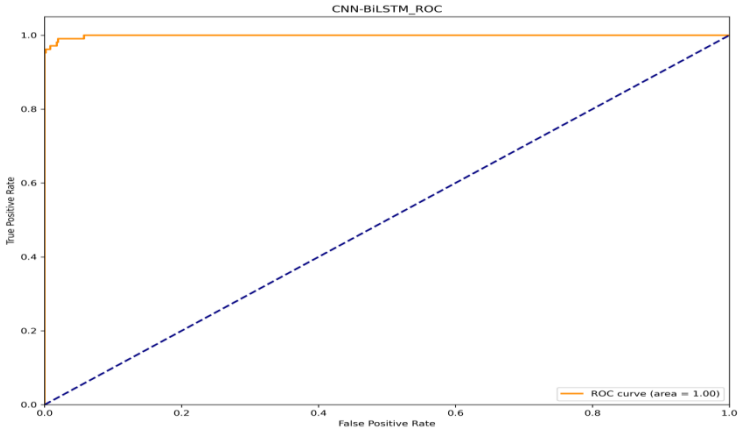


Figure 30 The ROC curve of the CNN-BiLSTM model

2.CNN-AT model

The CNN-AT model is set up with 64 convolutional filters, a kernel size of 3, and a pooling size of 2, along with a 50% dropout to prevent overfitting. It incorporates an attention mechanism with weights and biases initialized uniformly to enhance focus on important features. The fully connected layers have 128 neurons each, applying 10% and 30% dropout, respectively. The model uses the Adam optimizer for efficient and effective optimization.The specific parameter settings are shown in the table 46.

Table 46 Parameter settings for CNN-AT

Parameters	Values	Parameters	Values
filters	64	dense1_units	128
kernel_size	3	dropout2	0.1
pool_size	2	dense2_units	128
dropout1	0.5	dropout3	0.3
attention_weight	uniform	optimizer	adam
attention_bias	uniform		

Table 47 shows the performance of the CNN-AT model at different time steps (w=4 to w=12), with or w=8 and Smote-PCA as the optimal configurations, achieving the highest accuracy of 0.992.

Non-Smote: Using only raw data, ranging from 0.820 to 0.985. When the time step is w=8, the accuracy is highest at 0.985.

Smote: Using the Smote algorithm on the dataset, the overall performance is better than Non Smote. When the time step is w=9, the accuracy is highest at 0.991.

Smote-PCA: After applying the Smote algorithm and PCA extraction to the dataset, the accuracy of the model was 0.992 at a time step of w=8.

Smote-RF: After using Smote and applying random forest feature extraction, the accuracy of the model decreased to 0.990 at a time step of w=9.

Table 47 Performance of CNN-AT under different conditions

CNN-AT	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Non-Smote	0.960	0.972	0.978	0.981	0.985	0.979	0.966	0.928	0.82
Smote	0.965	0.968	0.989	0.984	0.985	0.991	0.985	0.978	0.865
Smote-PCA	0.956	0.965	0.984	0.980	**0.992**	0.989	0.978	0.975	0.787
Smote-RF	0.965	0.972	0.990	0.984	0.988	0.990	0.981	0.952	0.832

From Table 48, it can be seen that the CNN-AT model has the highest accuracy under the Smote-PCA scheme. Choosing this model as the optimal model, the test set data was used to train and predict the optimal model, with an accuracy of 0.992. Compared with traditional machine learning models and deep learning models alone, it performs better in distinguishing between non ST and ST samples, with 96.3% of predicted ST samples being correct and 98.1% correctly identified in actual ST samples.

Table 48 Classification Report of CNN-AT under Smote-PCA

CNN-AT	precision	recall	f1-score	accuracy	support
Non-ST	0.997	0.993	0.995	0.992	389
ST	0.963	0.981	0.972		56

The figure 31 shows that the loss and accuracy trends during training iterations are relatively stable, indicating good convergence of the model. From figure 32, an AUC of 1 demonstrates that the CNN-AT model achieves perfect discrimination between classes, reflecting its exceptional performance in the classification task.

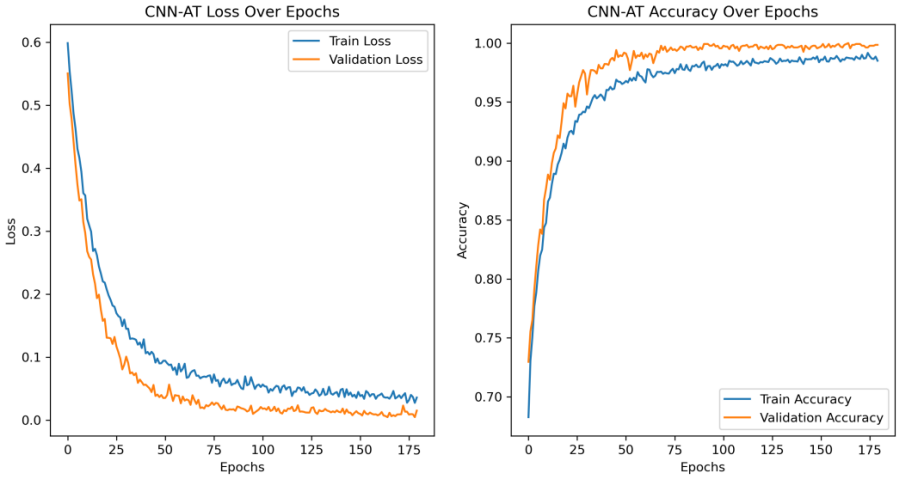


Figure 31 Trend chart of accuracy and loss value of CNN-AT model

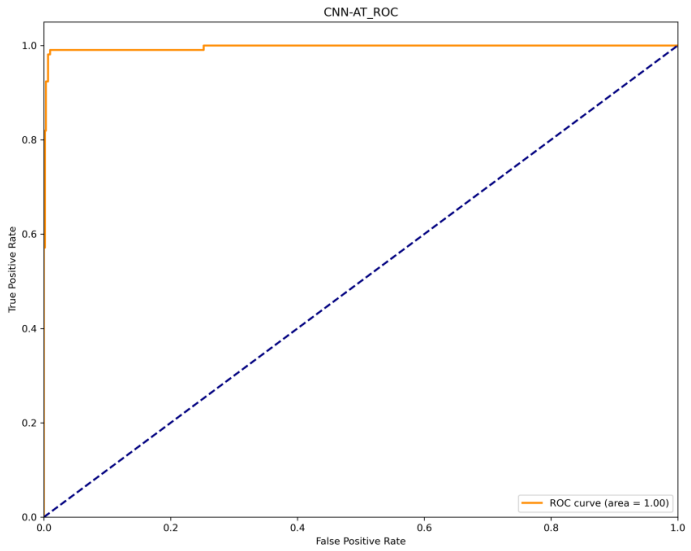


Figure 32 The ROC curve of CNN-AT model

3.BiLSTM-AT model

The BiLSTM-AT model is set up with a bidirectional LSTM layer consisting of 64 units and a 20% dropout to prevent overfitting. It incorporates an attention mechanism with weights and biases initialized uniformly, enabling the model to focus on significant features. The fully connected layers include two layers with 128 neurons each, applying 10% and 30% dropout, respectively. The adam optimizer is used for efficient and adaptive optimization. The specific parameter settings are shown in the table 49.

Table 49 Parameter settings for BiLSTM-AT

Parameters	**Values**	**Parameters**	**Values**
lstm1_units	64	dropout2	0.1
dropout1	0.2	dense2_units	128
attention_weight	uniform	dropout3	0.3
attention_bias	uniform	optimizer	adam
dense1_units	128		

Table 50 shows the performance of the BiLSTM-AT model at different time steps (w=4 to w=12), with the highest accuracy of 0.993 at w=8 and Smote-PCA as the optimal configurations.

Non-Smote: Using only raw data, ranging from 0.820 to 0.977. When the time step is w=8 and 9, the accuracy is highest at 0.977.

Smote: Using the Smote algorithm on the dataset, the overall performance is better than Non-Smote. When the time step is w=6, the accuracy is at 0.991.

Smote-PCA: After applying the Smote algorithm and PCA extraction to the dataset, the accuracy of the model was decreased to 0.989 at a time step of w=8.

Smote-RF: After using Smote and applying random forest feature extraction, the accuracy of the model improved to 0.993 which is highest at a time step of w=8.

Table 50 Performance of BiLSTM-AT under different conditions

BiLSTM-AT	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Non-Smote	0.971	0.962	0.976	0.975	0.977	0.977	0.97	0.932	0.82
Smote	0.978	0.979	0.991	0.989	0.990	0.982	0.976	0.978	0.865
Smote-PCA	0.966	0.965	0.986	0.981	0.989	0.985	0.981	0.973	0.753
Smote-RF	0.975	0.973	0.982	0.989	**0.993**	0.985	0.983	0.955	0.854

From Table 51, it can be seen that the BiLSTM-AT model has the highest accuracy under the Smote-RF scheme. Choosing this model as the optimal model, the test set data was used to train and predict the optimal model, with an accuracy of 0.993. Compared with traditional machine learning models and deep learning models alone, it performs better in distinguishing between non ST and ST samples, with 97.5% of predicted ST samples being correct and 97.5% correctly identified in actual ST samples.

Table 51 Classification Report of BiLSTM-AT under Smote-RF

BiLSTM-AT	precision	recall	f1-score	accuracy	support
Non-ST	0.996	0.996	0.996	0.993	544
ST	0.975	0.975	0.975		79

The figure 33 shows that the loss and accuracy trends during training iterations are relatively stable, indicating good convergence of the model. From Figure 34, an AUC of 0.99 demonstrates that the BiLSTM-AT model has excellent classification performance, with a high ability to distinguish between classes effectively, but lower than CNN-BiLSTM and CNN-AT.

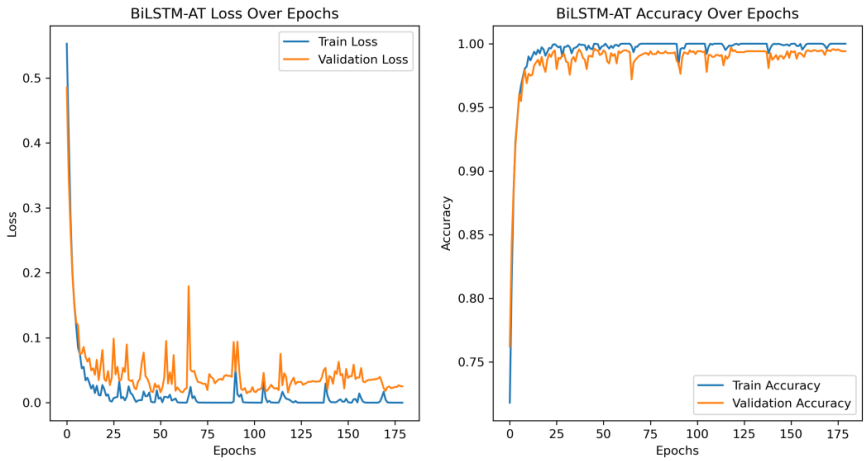


Figure 33 Trend chart of accuracy and loss value of BiLSTM-AT model

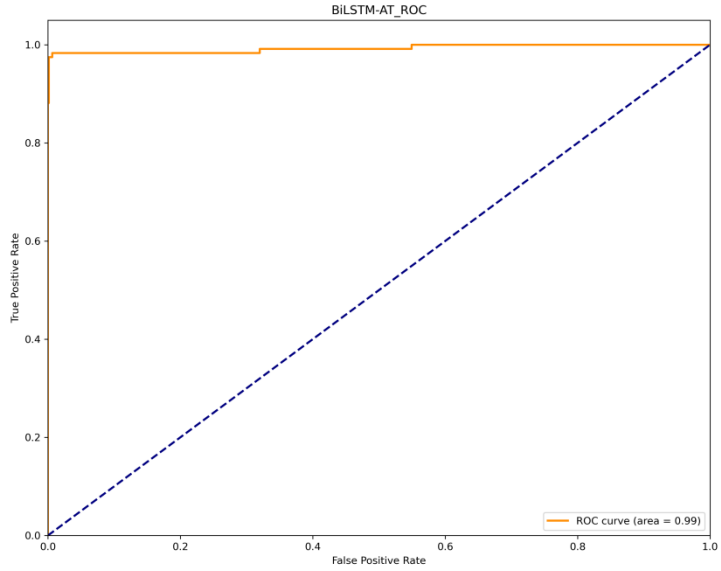


Figure 34 The ROC curve of BiLSTM-AT model

4.CNN-BiLSTM-AT model

The CNN-BiLSTM-AT model is configured with 64 convolutional filters, a kernel size of 3, and a pooling size of 2, along with a 50% dropout to prevent overfitting. It includes a bidirectional LSTM layer with 64 units, applying a 20% dropout to reduce overfitting. An attention mechanism with uniformly initialized weights and biases enhances the model's focus on critical features. The fully connected layers consist of two layers with 128 neurons each, applying 10% and 30% dropout, respectively. The Adam optimizer is employed for efficient and adaptive optimization. The specific parameter settings are shown in the table 52.

Table 52 Parameter settings for CNN-BiLSTM-AT

Parameters	Values	Parameters	Values
filters	64	attention_bias	uniform
kernel_size	3	dense1_units	128
pool_size	2	Dropout3	0.1
dropout1	0.5	dense2_units	128
lstm1_units	64	Dropout4	0.3
Dropout2	0.2	optimizer	adam
attention_weight	uniform		

Table 53 shows the performance of the CNN-BiLSTM-AT model at different time steps (w=4 to w=12), with the highest accuracy of 0.994 at w=8 and Smote as the optimal configurations.

Non-Smote: Using only raw data, ranging from 0.865 to 0.978. When the time step is w=8, the accuracy is highest at 0.978.

Smote: Using the Smote algorithm on the dataset, the overall performance is better than Non-Smote. When the time step is w=8, the accuracy is highest at 0.994.

Smote-PCA: After applying the Smote algorithm and PCA extraction to the dataset, the accuracy of the model was decreased to 0.990 at a time step of w=8.

Smote-RF: After using Smote and applying random forest feature extraction, the accuracy of the model was improved to 0.992 at time steps w=9.

Table 53 Performance of CNN-BiLSTM-AT under different conditions

CNN-BiLSTM-AT	w=4	w=5	w=6	w=7	w=8	w=9	w=10	w=11	w=12
Non-Smote	0.959	0.948	0.973	0.973	0.978	0.976	0.965	0.933	0.865
Smote	0.966	0.975	0.986	0.982	**0.994**	0.993	0.983	0.974	0.876
Smote-PCA	0.958	0.958	0.982	0.982	0.990	0.989	0.985	0.975	0.854
Smote-RF	0.956	0.968	0.989	0.983	0.990	0.992	0.978	0.952	0.820

From Table 54, it can be seen that the CNN-BiLSTM-AT model has the highest accuracy under the Smote scheme. Choosing this model as the optimal model, the test set data was used to train and predict the optimal model, with an accuracy of 0.994. Compared with traditional machine learning models and deep learning models alone, it performs better in distinguishing between non ST and ST samples, with 99.1% of predicted ST samples being correct and 96.6% correctly identified in actual ST samples.

Table 54 Classification Report of CNN-BiLSTM-AT under Smote

CNN-BiLSTM-AT	precision	recall	f1-score	accuracy	support
Non-ST	0.995	0.999	0.997	0.994	389
ST	0.991	0.966	0.979		56

The figure 35 shows that the loss and accuracy trends during training iterations are relatively stable, indicating good convergence of the model. From Figure 36, an AUC of 1.00 demonstrates that the CNN-BiLSTM-AT model has excellent classification performance, with a high ability to distinguish between classes effectively.

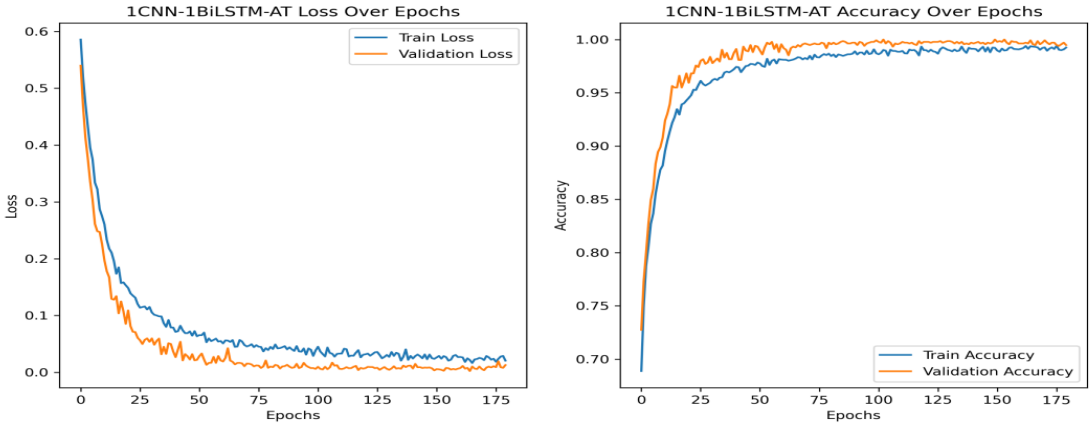


Figure 35 Trend chart of accuracy and loss value of CNN-BiLSTM-AT model

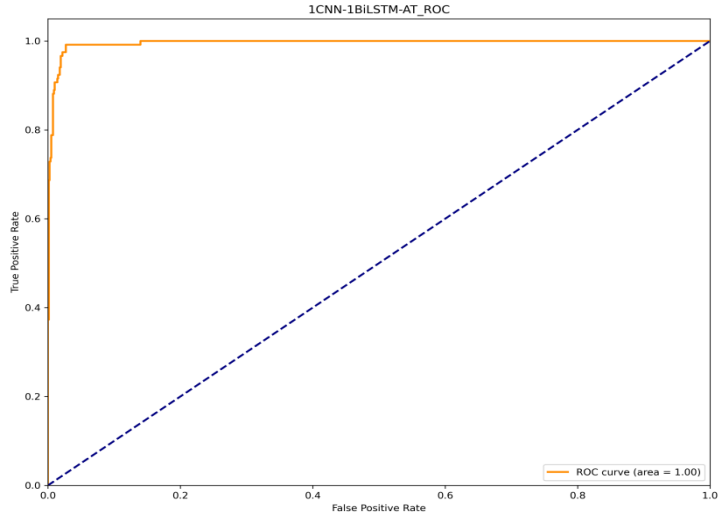


Figure 36 The ROC curve of CNN-BiLSTM-AT model

4.2.3.2.3 Summary of Deep Learning Models

The results highlight that ensemble deep learning models outperform single deep learning models in accuracy. Among single models, BiLSTM achieved the highest accuracy of 0.991 with SMOTE and w=8, followed by CNN at 0.981 and AT at 0.977. In terms of training time, BiLSTM required the longest time at 909.81 seconds, while CNN and AT had shorter training times of 84.77 seconds and 7.22 seconds, respectively. Ensemble models showed even better performance, with CNN-BiLSTM and BiLSTM-AT reaching 0.993, CNN-AT achieving 0.992, and CNN-BiLSTM-AT delivering the best accuracy of 0.994 under SMOTE and w=8. In terms of training time, CNN-BiLSTM took 359.52 seconds, BiLSTM-AT took 537.33 seconds, CNN-AT took 85.82 seconds, and CNN-BiLSTM-AT required 400.27 seconds. These results demonstrate the superior feature extraction and integration capabilities of ensemble models, particularly CNN-BiLSTM-AT, which balances high accuracy and reasonable training time.

Table 55 Best single and ensemble machine learning model

Model		**Conditions**	**Training time(seconds)**	**Highest accuracy**
Single deep learning model	CNN	Smote-RF and w=8	84.77	0.981
	BiLSTM	Smote and w=8	909.81	0.991
	AT	Smote-RF and w=6	7.22	0.977
Ensemble of deep learning models	CNN-BiLSTM	Smote(PCA/RF) and w=8	359.52	0.993
	CNN-AT	Smote-PCA and w=8	85.82	0.992
	BiLSTM-AT	Smote-RF and w=8	537.33	0.993
	CNN-BiLSTM-AT	Smote and w=8	**400.27**	**0.994**

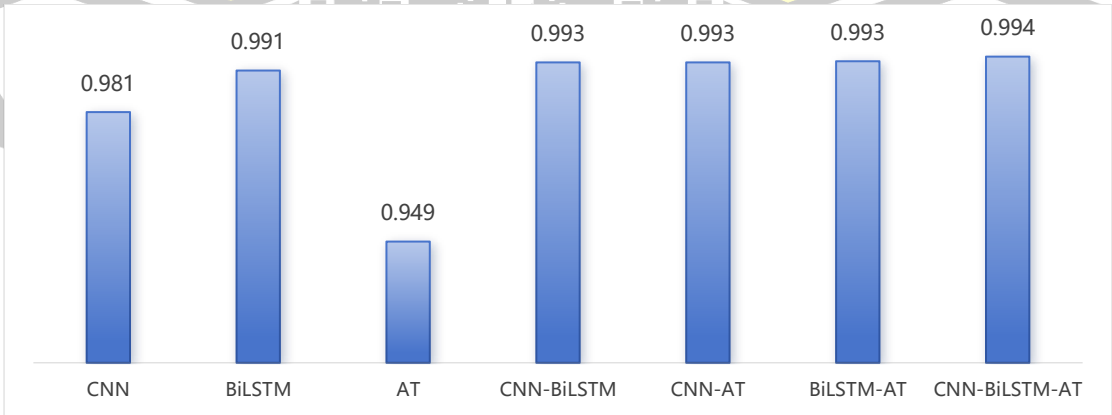


Figure 37 Ensemble of deep learning models accuracy

4.2.3.3 Model based on activation function optimization algorithm

The convolutional and Dense layers in the original CNN-BiLSTM-AT used ReLU activation functions. In this paper, a custom activation function will be used to replace the original activation function (see Figure 38) to compare the model performance before and after activation function optimization.

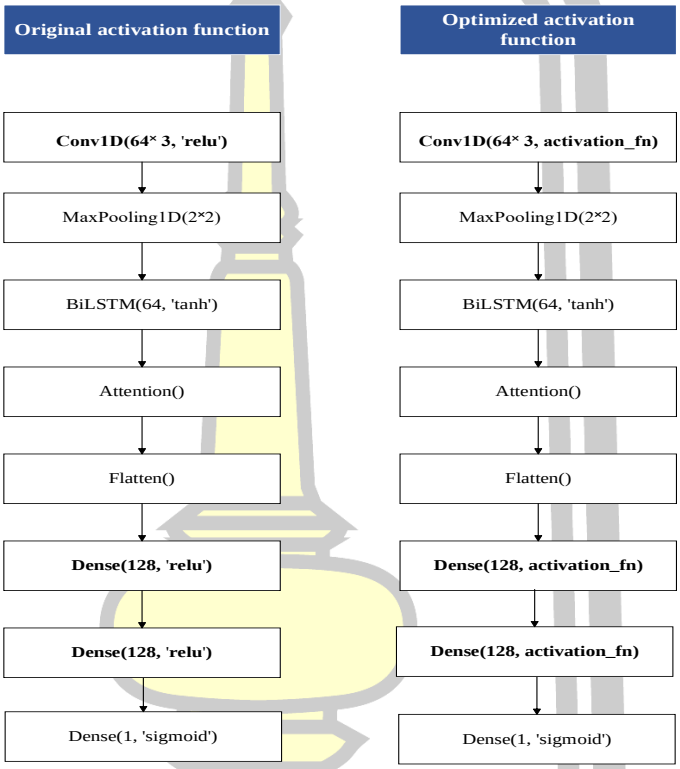


Figure 38 Structure diagram of CNN-BiLSTM-AT model before and after activation function optimization

The parameter configuration for activation function optimization combines the characteristics of classical activation functions (e.g., ReLU, Tanh, and Sigmoid) to create new hybrid forms, such as ReLU_Sig, Tanh_Sig, and ReLU_Tanh, with performance enhancement achieved through parameter tuning. For example, in ReLU_Sig and ReLU_Tanh, the introduction of ReLU's leakage coefficient α, set to 0.1 and 0.01 respectively, balances gradient flow and avoids issues like gradient explosion or vanishing. Additionally, the use of the linear multiplier x expands the dynamic range of the activation output. This design leverages ReLU's sparse activation, Tanh's normalization, and Sigmoid's smooth mapping characteristics,

enabling deep learning models to exhibit enhanced nonlinear representation capabilities and training stability in complex tasks.

Table 56 Activation function forms for CNN-BiLSTM-AT

Activation function	Functional form
ReLU	K.relu(x)
Tanh	K.tanh(x)
Sigmoid	K.sigmoid(x)
ReLU_Sig	x * K.relu(x, alpha=0.1) * K.sigmoid(x)
Tanh_Sig	x*K.tanh(x)*K.sigmoid(x)
ReLU_Tanh	x*K.relu(x, alpha=0.01)*K.tanh(x)

The experimental results demonstrate that different activation functions significantly impact model accuracy. Among them, ReLU and ReLU_Tanh achieved the highest accuracy of 0.994, highlighting their superior non-linear characteristics and computational efficiency in handling complex tasks. Tanh, with an accuracy of 0.991, ranked slightly below ReLU and ReLU_Tanh but outperformed Sigmoid, which achieved an accuracy of 0.985, indicating that Tanh's symmetric output is advantageous for certain data distributions. In contrast, the composite activation functions ReLU_Sig and Tanh_Sig had accuracies of 0.980 and 0.976, respectively, which were lower than the standalone activation functions, possibly due to their sensitivity to parameters or excessive complexity.

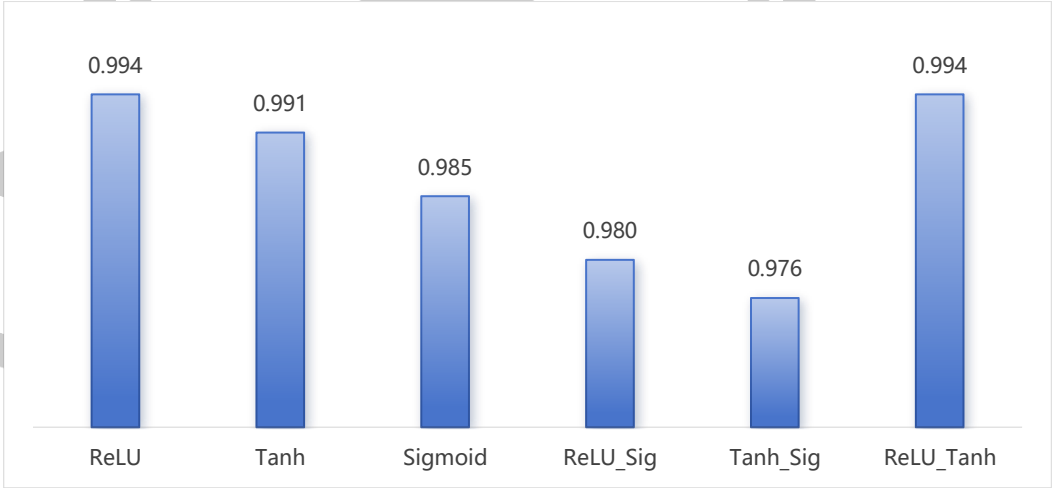


Figure 39 Comparison of the results of different activation functions

Figure 40 presents a comparison of the classification performance of the CNN-BiLSTM-AT model using different activation functions, assessed through ROC curves and AUC values. The results clearly demonstrate that the choice of activation function has a significant impact on the model's ability to distinguish between financially distressed (ST) and stable (non-ST) companies.

Among all tested activation functions, ReLU_Tanh exhibited the strongest performance, achieving an AUC of 1.00 while positioning its ROC curve closest to the top-left corner. This placement indicates that ReLU_Tanh provides the best trade-off between sensitivity and specificity, meaning it effectively identifies financial risk cases (ST companies) while minimizing false positives (misclassifying non-ST companies as high-risk). This suggests that ReLU_Tanh enhances the model' s predictive reliability and stability, making it particularly suitable for financial risk prediction.Although ReLU and Tanh also achieved AUC values of 1.00, their ROC curves indicate slightly lower sensitivity and specificity compared to ReLU_Tanh. This implies that while they are effective, they may not perform as consistently across different datasets or experimental conditions.In contrast, the Sigmoid activation function recorded a slightly lower AUC of 0.99, showing strong classification performance but slightly weaker discriminatory ability compared to ReLU_Tanh. The Softmax activation function performed the worst, with an AUC of 0.98, indicating reduced effectiveness in distinguishing between high-risk and low-risk companies, though it still maintained reasonable classification capability.

Overall, these results highlight the critical role of activation function selection in optimizing model performance. The ReLU_Tanh activation function demonstrated superior classification accuracy, sensitivity, and specificity, making it the most effective choice for financial risk prediction in this scenario. Its ability to produce stable and high-confidence predictions across different experiments reinforces its potential as an optimal activation function for financial risk assessment models.

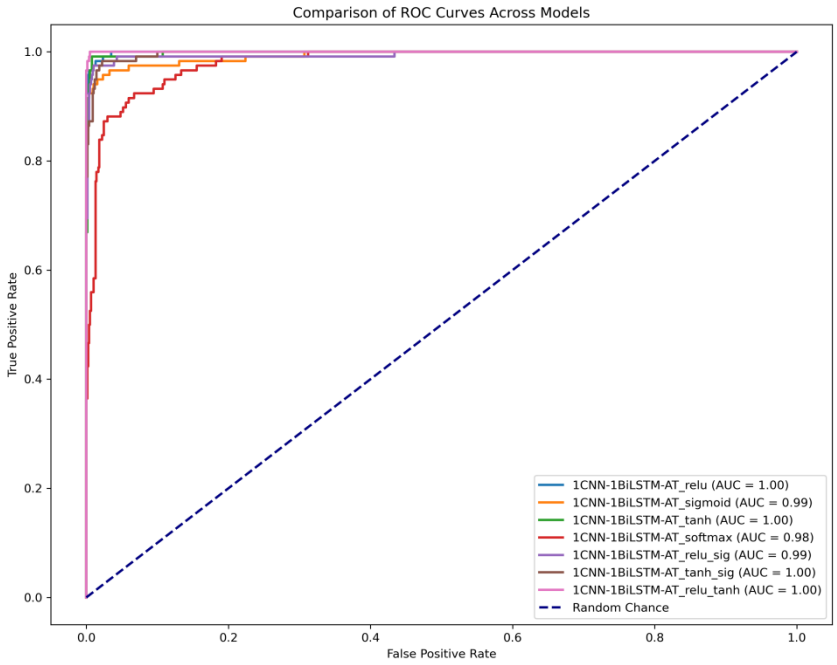


Figure 40 The ROC curve of CNN-BiLSTM-AT under different activation function

4.2.4 Case application analysis

To achieve dynamic monitoring of financial risks for listed companies and validate the stability and reliability of the proposed model, this study employed a CNN-BiLSTM-AT financial risk prediction model optimized with a ReLU_Tanh activation function algorithm. The model was trained using financial data from Company A and Company B over the past 12 quarters. To ensure robustness, 100 independent experiments were conducted with varying random seed values.

The implementation process is as follows:

(1)Repeated Experiments: The model was trained in 100 iterations, resetting the random seed in each iteration to introduce variability in model initialization, thereby enhancing the diversity of training outcomes and predictive performance.

(2) Independent Training and Prediction: In each iteration, the model was independently trained on the same dataset and then used to predict financial risks for the selected companies, recording their probability scores.

(3) Probability Distribution Visualization: Based on the prediction results from 100 independent experiments, probability distribution curves were plotted for selected companies. These visualizations illustrate the probability fluctuations across different

trials, providing insights into the model's stability and the financial risk trends for each company.

In each experiment, the model calculated the probability p of financial risk for the selected companies, where p ranged from [0,1]. According to the model's decision rule, a probability $p>0.5$ indicates that the company is likely facing financial risk(ST), while $p\leqslant 0.5$ indicates that the company is operating normally(non-ST).

The experimental results, presented in Figures 41 and 42, illustrate the predicted financial risk probabilities for selected companies(Company A and Company B) across 100 independent experiments. These results provide a comprehensive assessment of the model's ability to distinguish between financially distressed (ST) and stable (non-ST) companies.

For Company A, the predicted probability of financial risk remained consistently above 0.5 in all experiments, with the minimum recorded probability exceeding the threshold. This strong and stable prediction suggests that the model confidently identifies Company A as experiencing financial distress. The lack of fluctuation below the 0.5 threshold indicates that the model maintains high reliability in detecting financial risk, regardless of variations in initial random seed values.In contrast, for Company B, the predicted probability of financial risk was consistently 0 across all 100 experiments. This means that in every trial, the model classified Company B as non-ST, reinforcing the conclusion that the company is in a stable operational state. The absence of any deviation above 0.5 further validates the robustness of the model in distinguishing financially sound companies from those at risk.

These findings substantiate the accuracy, stability, and robustness of the CNN-BiLSTM-AT model optimized with a ReLU_Tanh activation function algorithm in identifying financial risks across different companies. By demonstrating clear and consistent classification results under varying experimental conditions, the model proves to be a reliable tool for early financial risk warning. Its capability to maintain stable predictions across multiple trials highlights its practical value in financial risk assessment and proactive risk management for listed companies.

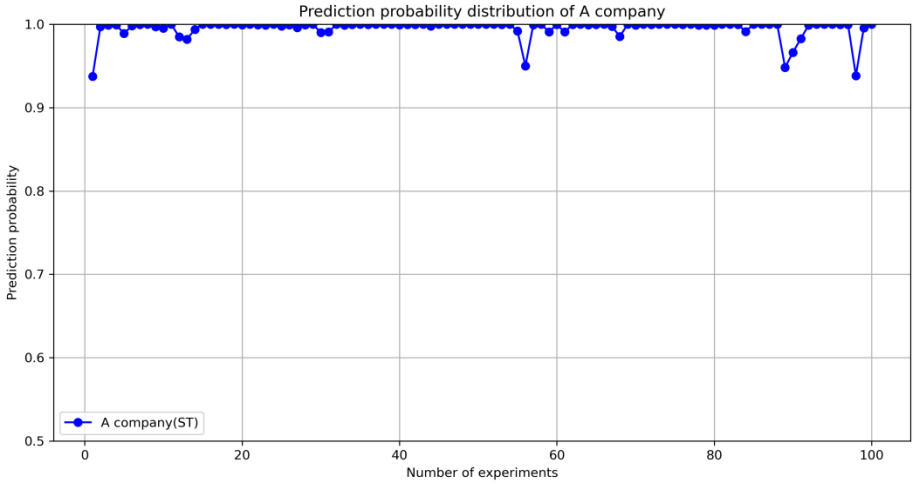


Figure 41 Probability distribution of financial risk prediction for Company A

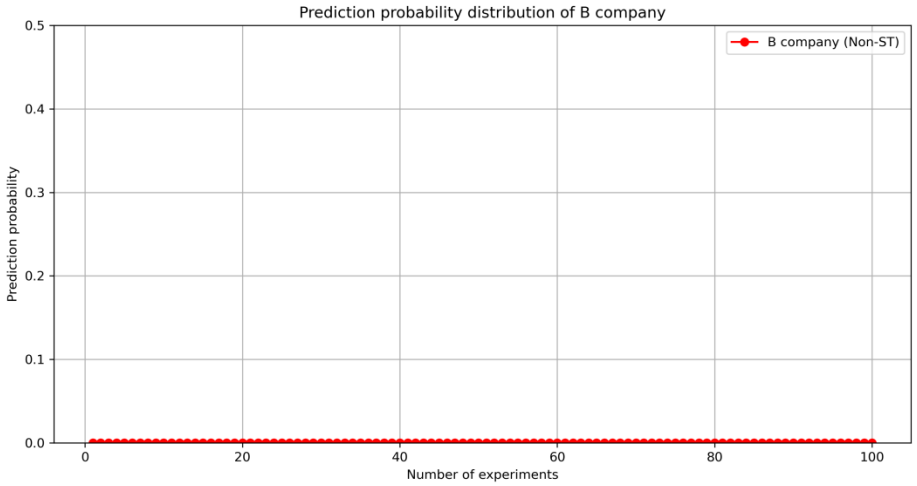


Figure 42 Probability distribution of financial risk prediction for Company B

Chapter 5
Conclusion and Discussion

This chapter introduces the research conclusions and discussions.

5.1 Conclusion

This article uses financial indicators data from the past three years and 12 quarters of listed companies to conduct a comparative study between traditional machine learning models and deep learning models. It proposes a CNN-BiLSTM-AT deep learning ensemble model based on an activation function optimization algorithm. The empirical results validate the effectiveness of this model, and the optimal hyperparameters of the model are determined using the hyperband algorithm. The conclusions of this study are summarized as follows.

5.1.1 Importance of Financial Features

The features selected by PCA indicate that ST companies have significant weaknesses in liquidity(Cash Assets Ratio, Working Capital), profitability (Operating Profit Rate, Net Profit Margin on Fixed Assets), and operational efficiency(Operating Index). They have lower cash reserves, weaker short-term solvency, slow revenue growth, and inefficient asset utilization. These factors contribute to financial instability, making companies more vulnerable to operational difficulties. Improving cash flow management, optimizing cost control, and enhancing asset returns is crucial for financial stability.

The features identified by Random Forest indicate that ST companies have significant weaknesses in liquidity (Cash Ratio, Quick Ratio, Working Capital), profitability and efficiency (Accounts Receivable Turnover, Fixed Assets Ratio, Fixed Asset Growth Rate), and operational stability (Inventory Turnover). They face tight cash flow, higher short-term debt pressure, and lower fixed asset investment, reflecting financial constraints. Additionally, poor debt management and slow accounts receivable collection further increase financial risks. Optimizing capital allocation, increasing cash reserves, and improving debt management are essential steps to strengthen financial health.

In summary, both PCA and Random Forest highlight key financial weaknesses of ST companies, particularly in liquidity, profitability, efficiency, and stability. These

firms struggle with cash flow, debt pressure, and asset utilization, making them more vulnerable to financial distress. Addressing these issues through better cash management, cost control, and capital optimization is essential for long-term financial stability.

5.1.2 Impact of Time-Step Settings and Smote

Larger time steps (w=6 to w=10) generally improve model accuracy by capturing temporal dependencies more effectively. However, excessively long time steps can degrade performance, likely due to the reduced number of usable samples generated, making it harder for the model to learn effectively. This highlights the importance of optimizing the time-step settings to balance capturing temporal information and maintaining a sufficient dataset size.

Smote oversampling significantly improves model performance by addressing class imbalance and small dataset sizes. By generating synthetic samples for minority classes, Smote enables the model to better learn the characteristics of these classes. After applying Smote, the overall model accuracy increased from 0.978 to 0.994, demonstrating a marked improvement in predicting ST risks and enhancing the model’ s ability to handle imbalanced data.

5.1.3 Effectiveness of Deep Learning Models

Experimental results demonstrate that deep learning models, including CNN (accuracy:0.981), BiLSTM(0.991), and Attention Mechanism (AT,0.977), consistently outperform traditional machine learning models such as Logistic Regression (LR, 0.910), Support Vector Machine (SVM, 0.875), Decision Tree (DT, 0.899), and BP Neural Network (BP, 0.919).

These results highlight the superior capability of deep learning models in analyzing and predicting financial time-series data.Deep learning models excel in capturing complex temporal dependencies and feature interactions, which are crucial in financial data characterized by non-linear patterns and high dimensionality.

5.1.4 Effectiveness of Ensemble Models

Experimental results demonstrate that ensemble models consistently outperform single models, showcasing their effectiveness in improving predictive performance.

Among traditional machine learning models, ensemble approaches such as Random Forest (RF) and Stacking achieved the highest accuracy of 0.926, exceeding

the best-performing single model, BP Neural Network (BP, 0.919). This indicates that ensemble methods can better capture the complex relationships in data by combining the predictive strengths of multiple models.

Similarly, ensemble deep learning models—CNN-BiLSTM (0.993), CNN-AT (0.992), BiLSTM-AT (0.993), and CNN-BiLSTM-AT (0.994)—outperformed single deep learning models like CNN (0.981), BiLSTM (0.991), and Attention Mechanism (AT, 0.977). These results highlight the advantages of ensemble methods in leveraging diverse model architectures to capture temporal dependencies and intricate feature interactions more effectively.

In particular, the CNN-BiLSTM-AT model not only achieved the highest accuracy but also maintained a reasonable training time, demonstrating a balanced trade-off between performance and computational efficiency. These findings underscore the potential of ensemble models in both traditional and deep learning contexts, particularly for complex tasks like financial risk prediction, where accuracy and robustness are crucial. The ability to integrate complementary strengths of different models makes ensemble approaches a powerful tool for handling high-dimensional, non-linear financial datasets.

5.1.5 Effectiveness of the Hyperband Algorithm

The Hyperband algorithm efficiently searches for optimal hyperparameter combinations by using a resource allocation strategy that dynamically adjusts the search process. It explores hyperparameter spaces across various layers, such as the convolutional layer filters, BiLSTM units, and dropout rates, with values ranging from 16 to 128 for filters and units, and 0.1 to 0.5 for dropout rates. The algorithm automates hyperparameter tuning, reducing the need for manual adjustments and significantly saving time. This process improves model performance by ensuring that the best hyperparameter configurations are selected, leading to enhanced accuracy and robustness.

5.1.6 Effectiveness of the Attention Mechanism

The integration of the attention mechanism significantly enhances the performance of the CNN, BiLSTM, and CNN-BiLSTM models. Specifically, adding the attention mechanism improves the accuracy of the single CNN model from 0.981 to 0.992, the BiLSTM model from 0.991 to 0.993, and the combined CNN-BiLSTM

model from 0.993 to 0.994. These improvements demonstrate that the attention mechanism not only boosts accuracy but also increases the model's ability to focus on the most relevant features, leading to more efficient learning. By highlighting key temporal and spatial patterns in the data, the attention mechanism provides better feature weighting, enhancing both model effectiveness and training efficiency.

5.1.7 Effectiveness of ReLU_Tanh Activation Function

The experimental results highlight the significant impact of activation functions on model performance. The ReLU_Tanh activation function not only achieves the same accuracy as ReLU (0.994), but also shows a ROC curve that is positioned further to the left. This suggests that ReLU_Tanh improves the model's ability to differentiate between classes more effectively, making it particularly valuable in scenarios where precision and recall are crucial. While the Tanh function also performed well with an accuracy of 0.991, its symmetric output offers advantages for certain data distributions. In contrast, composite activation functions like ReLU_Sig and Tanh_Sig yielded lower accuracies of 0.980 and 0.976, likely due to their added complexity and sensitivity to parameters. ROC curve analysis supports these findings, with ReLU_Tanh achieving the best balance of sensitivity and specificity, reflected in an AUC of 1.00. Overall, the ReLU_Tanh activation function stands out as the most effective choice, enhancing both model performance and class discrimination.

5.2 Discussion

5.2.1 Research Significance

5.2.1.1 Advancements in Financial Risk Prediction Models

The proposed CNN-BiLSTM-AT ensemble model, combined with innovative techniques such as the ReLU_Tanh activation function and the Hyperband algorithm for hyperparameter optimization, significantly advances the field of financial risk prediction. By effectively integrating convolutional layers, bidirectional LSTMs, and attention mechanisms, the study demonstrates the ability to capture complex temporal and spatial patterns in financial data, offering a robust solution for predictive modeling.

5.2.1.2 Feature Importance and Data Handling Insights

Highlighting the importance of specific financial indicators (e.g., fixed assets ratio, inventory turnover) offers valuable insights for stakeholders, such as investors

and analysts, to prioritize critical features in financial decision-making. Additionally, demonstrating the impact of SMOTE oversampling and optimal time-step settings underscores the importance of handling imbalanced data and temporal dependencies in financial datasets.

5.2.1.3 Practical Implications for Ensemble and Deep Learning Models

The findings reveal the effectiveness of ensemble models in improving prediction accuracy, emphasizing their potential for real-world applications. The success of ensemble approaches in both traditional and deep learning contexts provides a framework for deploying such methods in industries where predictive accuracy and robustness are critical, such as banking, auditing, and investment analysis.

5.2.2 Research Limitations

5.2.2.1 Dataset Constraints

The article is based on financial data from Chinese listed companies over the past three years and 12 quarters. While this provides a comprehensive temporal view, it may limit the generalizability of the findings to other industries, countries, or timeframes. Additionally, the dataset primarily focuses on structured financial indicators, potentially overlooking critical unstructured data, such as annual reports, media sentiment, or industry-specific contextual information, which could offer richer insights into financial risks and performance.

5.2.2.2 Dependence on Hyperparameter Tuning

Although the hyperband algorithm automates hyperparameter tuning, its effectiveness depends on the predefined search space. This may leave room for suboptimal configurations, particularly in the absence of domain-specific insights into ideal hyperparameter ranges.

5.2.2.3 Fixed Initialization Parameters

The article employs standard weight initialization techniques but does not explore advanced or dynamic strategies, such as layer-specific initialization or data-driven approaches. These could potentially enhance model convergence and stability, particularly in deep learning architectures.

5.2.2.4 Limitations of Model Architectures

The research primarily focuses on the CNN-BiLSTM-AT ensemble model and compares it with a limited set of traditional and deep learning models. While effective, the study does not explore other promising architectures, such as Transformers, GRUs, or hybrid configurations, which may further enhance predictive performance.

5.2.3 Future Research

5.2.3.1 Expansion of Feature Dimensions

Future research will explore additional structured and unstructured data sources, such as annual reports, media sentiment, and macroeconomic indicators. Incorporating these diverse dimensions may provide richer insights and improve the model's ability to predict financial risks and performance across various industries and regions.

5.2.3.2 Optimization of Model Architectures

Advanced deep learning architectures, including Transformers, GRUs, and hybrid configurations, will be investigated. Additionally, the integration of novel mechanisms, such as attention variants and multi-scale feature extraction, will be explored to enhance the ability to capture complex temporal and spatial dependencies.

5.2.3.3 Hyperparameter and Algorithm Optimization

Alternative optimization algorithms, such as genetic algorithms and Bayesian optimization, will be applied to improve hyperparameter tuning. Further, the refinement of loss functions and neuron initialization parameters, such as layer-specific or adaptive initialization strategies, will be explored to improve model convergence and performance.

5.2.3.4 Improvement of Computational Efficiency

Strategies to optimize computational efficiency will be developed, focusing on reducing the resource requirements of ensemble models. Techniques such as model pruning, quantization, and distributed training will be applied to ensure scalability and real-time applicability in practical financial scenarios.

REFERENCES

[1] A Aziz，DC Emanuel，GH Lawson. (1988).Bankruptcy Prediction-An Investigation of Cash Flow Based Models.Journal of Management Studies,25（5）：419-437.

[2] Abror, W. F., Alamsyah, A., & Aziz, M. (2023). Bankruptcy Prediction Using Genetic Algorithm-Support Vector Machine (GA-SVM) Feature Selection and Stacking. Journal of Information System Exploration and Research, 1(2).

[3] Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The journal of finance, 23(4), 589-609.

[4] Altman, E. I., Haldeman, R. G., & Narayanan, P. (1977). ZETATM analysis A new model to identify bankruptcy risk of corporations. Journal of banking & finance, 1(1), 29-54.

[5] Beaver, W. H. (1966). Financial ratios as predictors of failure. Journal of accounting research, 71-111.

[6] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).

[7] Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140.

[8] Breiman, L., Gordon, A., Friedman, J., Olshen, R., and Stone, C. (1984).Classification and Regression Trees. Biometrics, 40(3):874.

[9] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[10] Bian, X., Lv, X., & Tian, J. (2022). Research on Agricultural Economic Early Warning Based on Genetic Algorithm and SVM. Journal of Sensors, 2022.

[11] Cao Wei, Li Can, Zhu.(2018). Weidong Research on Financial Crisis Warning for Small and Medium sized Enterprises from the Perspective of Multi source Information Fusion: A Data Mining Method Based on Ensemble Learning. Financial and Accounting Communication: Zhong, (2): 95-99.

[12] Chaiyawat, T., & Samranruen, P. (2011). Delisting Risk Analysis: Empirical Evidence from the Thai Listed Companies. Technology, 2(2,212,584), 12.

[13] Chen Jing. (1999). Empirical analysis of financial deterioration prediction for listed companies. Accounting Research, (4): 31-38.

[14] Chen Xiao, Chen Zhihong. (2000).Theoretical, methodological, and applied research on corporate financial distress. Investment Research, (06): 29-33.

[15] Chen, M. Y. (2011). Predicting corporate financial distress based on integration of decision tree classification and logistic regression. Expert systems with applications, 38(9), 11261-11272.

[16] Choi, H., Son, H., & Kim, C. (2018). Predicting financial distress of contractors in the construction industry using ensemble learning. Expert Systems with Applications, 110, 1-10.

[17] Chen Yufang, Zhao Yingjie, Wu Xinjie. (2022). Research on Machine Learning Algorithms for Identifying Financial Fraud Based on Fusion Models: A Case Study of Manufacturing Industry. Statistics and Management, 37 (07): 116-121.

[18] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[19] Chen, Jian & Liu, Weiji. (2023). A study on stock price prediction based on Hyperband-LSTM model. Financial Management Research (01), 65-85.

[20] Ciampi, F., & Gordini, N. (2013). Small Enterprise Default Prediction Modeling through Artificial Neural Networks: An Empirical Analysis of I talian Small Enterprises. Journal of Small Business Management, 51(1), 23-45.

[21] Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society Series B: Statistical Methodology, 20(2), 215-232.

[22] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20, 273-297.

[23] Doumpos, M., Andriosopoulos, K., Galariotis, E., Makridou, G., & Zopounidis, C.

(2017). Corporate failure prediction in the European energy sector: A multicriteria approach and the effect of country characteristics. European Journal of Operational Research, 262(1), 347-360.
[24] Fang Kuangnan, Fan Xinyan, Ma Shuangge. (2016). Enterprise credit risk warning based on network structure logistic model. Statistical Research, 33 (4): 50-55.
[25] Falkner, S., Klein, A., & Hutter, F. (2018, July). BOHB: Robust and efficient hyperparameter optimization at scale. In International conference on machine learning (pp. 1437-1446). PMLR.
[26] Fitzpatrick, P. J. (1932). A comparison of the ratios of successful industrial enterprises with those of failed companies, 23(8): 57-65.
[27] Gordon, M. J. (1971). Towards a theory of financial distress. The journal of finance, 26(2), 347-356.
[28] Gottlieb, O., Salisbury, C., Shek, H., & Vaidyanathan, V. (2006). Detecting corporate fraud: An application of machine learning. A publication of the American Institute of Computing, 100-215.
[29] Gu Qi, Liu Shulian. (1999).Analysis and Countermeasures of Investment Behavior of Enterprises in Financial Crisis. Accounting Research, (10): 28-31.
[30] Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In Proceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 315-323). JMLR Workshop and Conference Proceedings.
[31] Geisser, S. (1975). The predictive sample reuse method with applications. Journal of the American Statistical Association, 70(350), 320-328.
[32] Hosaka, T. (2019). Bankruptcy prediction using imaged financial ratios and convolutional neural networks. Expert systems with applications, 117, 287-299.
[33] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[34] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).
[35] Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). Automated machine learning: methods, systems, challenges (p. 219). Springer Nature.
[36] Jagtap, A. D., Kawaguchi, K., & Karniadakis, G. E. (2020). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics, 404, 109136.
[37] Kavianpour, P., Kavianpour, M., Jahani, E., & Ramezani, A. (2023). A CNN-BiLSTM model with attention mechanism for earthquake prediction. The Journal of Supercomputing, 79(17), 19194-19226.
[38] Kuen, J., Lim, K. M., & Lee, C. P. (2015). Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle. Pattern recognition, 48(10), 2964-2982.
[39] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[40] Li Chenggang, Jia Hongye, Zhao Guanghui. (2023). Credit Risk Warning of Listed Companies Based on Information Disclosure Text: Empirical Evidence from Management Discussion and Analysis of Chinese Annual Reports. Chinese Management Science,31 (2): 18-29.
[41] Li Chenyao. (2023). Comparative study of financial risk warning models for real estate enterprises based on machine learning. Shanghai Commercial, (10): 205-207.
[42] Li S, Shi W, Wang J. (2021). A deep learning-based approach to constructing a domain sentiment lexicon: a case study in financial distress prediction. Information Processing & Management,58(5): 102673.
[43] Li Sha, Chen Xuan.(2021). Research on Enterprise Financial Crisis Warning Model

Based on Deep Learning Neural Network. Modern Information Technology, 5 (24): 101-103+107.
[44] Liang Mingjiang, Zhuang Yu.(2012). The application of ensemble learning methods in enterprise financial crisis warning. Soft Science, 26 (4): 114-117.
[45] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18(185), 1-52.
[46] Liu Yanwen, Dai Hongjun. (2007). Empirical study on financial distress warning model based on ternary logistic theory. Journal of Dalian University of Technology: Social Sciences Edition, (2): 60-66.
[47] Lu Xingyu, Liang Shuo, Qi Bei. (2021). Research on Financial Early Warning Evaluation of Agricultural Listed Enterprises - Based on Z-Model. Economic Research Guide, (36): 71-73+89.
[48] Lv Jun. (2014).Comparative Study on the Symptoms and Predictions of Corporate Financial Crisis Based on Different Indicator Types. Journal of Shanxi University of Finance and Economics,36 (01): 103-113.
[49] Manodamrongsat, P., Tongkong, S., & Boonyanet, W. (2021). Early warning of problematic firms listed in the stock exchange of Thailand:an investigation of financial ratios. Phranakhon Rajabhat Research Journal: Humanities and Social Sciences, 16(2), 34-51.
[50] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In Proc. icml (Vol. 30, No. 1, p. 3).
[51] Misra, D. (2019). Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681.
[52] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5, 115-133.
[53] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807-814).
[54] Odom, M. D., & Sharda, R. (1990, June). A neural network model for bankruptcy prediction. In 1990 IJCNN International Joint Conference on neural networks (pp. 163-168). IEEE.
[55] Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. Journal of accounting research, 109-31.
[56] Ouyang, Z. S., & Lai, Y. (2021). Systemic financial risk early warning of financial market in China using Attention-LSTM model. The North American Journal of Economics and Finance, 56, 101383.
[57] Pätäri, S., Arminen, H., Tuppura, A., & Jantunen, A. (2014). Competitive and responsible? The relationship between corporate social and financial performance in the energy sector. Renewable and Sustainable Energy Reviews, 37, 142-154.
[58] Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin philosophical magazine and journal of science, 2(11), 559-572.
[59] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1, 81-106.
[60] Quinlan, J. R. (1993).C4.5: Programs for Machine Learning. San Mateo, Calif: Morgan Kaufmann, 35-55.
[61] Rahman, M., Sa, C. L., & Masud, M. A. K. (2021). Predicting firms' financial distress: an empirical analysis using the F-score model. Journal of Risk and Financial Management, 14(5), 199.
[62] Ross, S., Westerfield, R., and Jaffe, J. (1971). Corporate Finance. Homewood, IL., 420-424.
[63] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[64] Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.
[65] Schapire, R. E. (1990). The strength of weak learnability. Machine learning, 5, 197-227.
[66] Shidik, G. F., Pramunendar, R. A., Kusuma, E. J., Saraswati, G. W., Winarsih, N. A. S., Rohman, M. S., ... & Andono, P. N. (2024). LUTanh Activation Function to Optimize BI-LSTM in Earthquake Forecasting. International Journal of Intelligent Engineering & Systems, 17(1).
[67] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, 25.
[68] Song, X., Jing, Y., & Qin, X. (2023). BP neural network-based early warning model for financial risk of internet financial companies. Cogent Economics & Finance, 11(1), 2210362.
[69] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society: Series B (Methodological), 36(2), 111-133.
[70] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1), 267-288.
[71] Vapnik, V. (1999). The nature of statistical learning theory. Springer science & business media.
[72] Wang Xuedan. (2014). Research on Financial Early Warning Models for Listed Companies. Zhejiang University of Business and Technology.
[73] Wang Yudong, Wang Di, Wang Shanshan. (2018). Comparison of Financial Crisis Warning Models Based on PSO-BP and FOA-BP Neural Networks. Statistics and Decision Making, (15): 177-179.
[74] Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.
[75] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5, 115-133.
[76] Wu, T., Xu, X., & Wu, Y. (2021). Research on optimization of SPReLU activation function in convolutional neural network. Computer & Digital Engineering, 49(8), 1637–1641.
[77] Xie, X., & Seung, H. S. (2003). Equivalence of backpropagation and contrastive Hebbian learning in a layered network. Neural computation, 15(2), 441-454.
[78] Yang Qinglong, Tian Xiaochun, Hu Peiyuan.(2016). Financial distress prediction of enterprises based on LASSO method. Statistics and Decision Making, (23): 170-173.
[79] Yang Shu'e, Xu Weigang. (2003). An Empirical Study on the Financial Warning Model of Listed Companies - Y Score Model. Chinese Soft Science, (01): 56-60.
[80] Zhan Chen. (2023). Research on Financial Crisis Warning Model Based on Machine Learning: Empirical Analysis from Science and Technology Innovation Enterprises. Financial Management, 5 (2): 16.
[81] Zhang Ling. (2000). A Discriminant Model for Financial Crisis Warning Analysis and Its Application. Prediction, 19 (6): 38-40.
[82] Zhou Shouhua, Yang Jihua, Wang Ping. (1996). On the Early Warning Analysis of Financial Crisis - F-score Model.Accounting Research, (8): 8-11.

APPENDIXES

APPENDIXES A

Table A.1 Comparison of Model Prediction Results for Non-ST Companies (Partial Sample)…….....119
Table A.2 Comparison of Model Prediction Results for Non-ST Companies (Partial Sample)…….....121

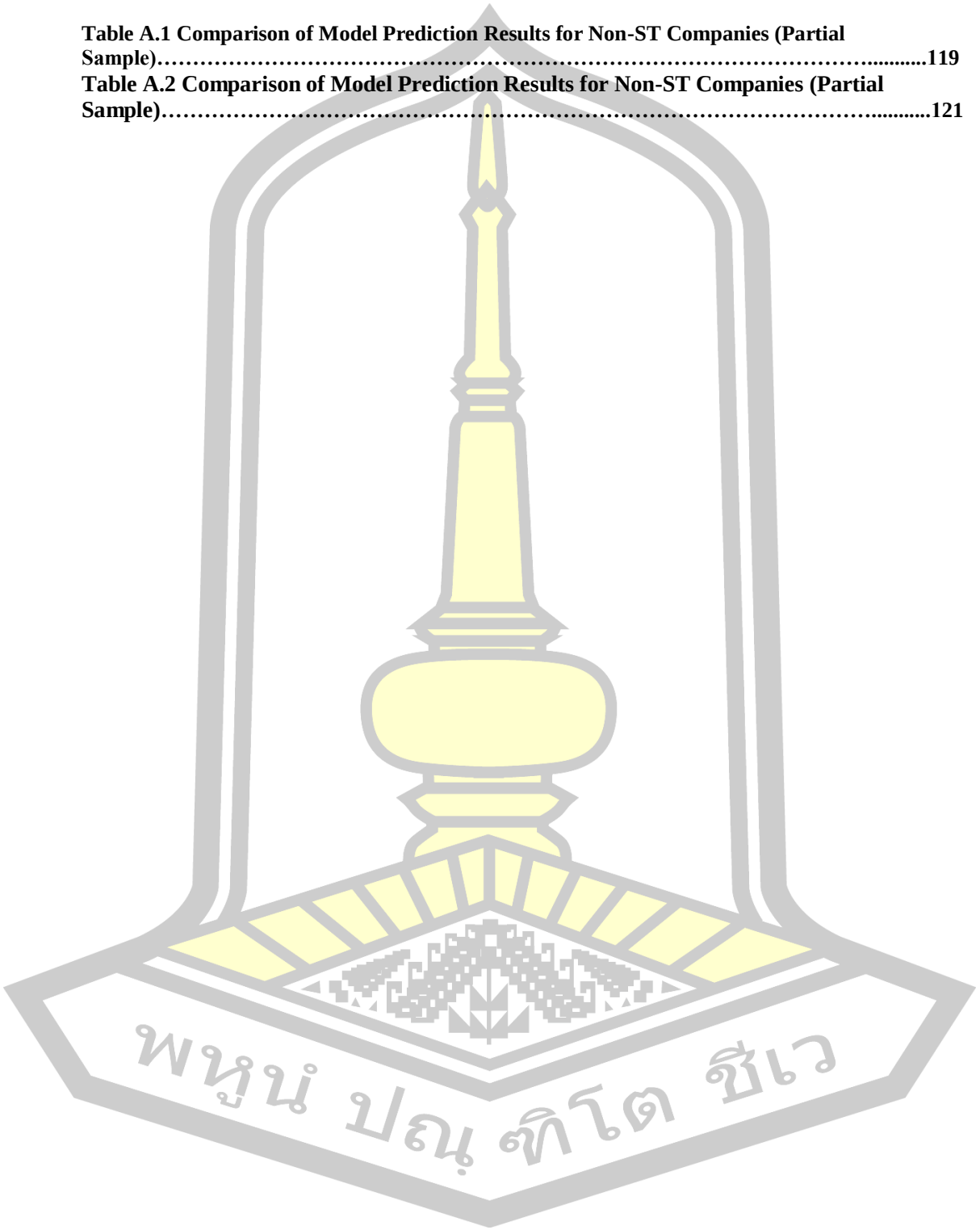


Table A.1 Comparison of Model Prediction Results for Non-ST Companies (Partial Sample)

ST Identifier	Actual_Class	Predicted_Probability	Predicted_Class
002822	0	0.000135955	0
002829	0	2.40E-12	0
002830	0	7.88E-05	0
002841	0	6.64E-11	0
002845	0	3.27E-09	0
002851	0	2.51E-09	0
002855	0	3.17E-10	0
002898	0	1.44E-05	0
002906	0	7.80E-13	0
002913	0	3.72E-13	0
300014	0	0.000619001	0
300018	0	1.01E-05	0
300036	0	5.38E-12	0
300048	0	1.92E-10	0
300096	0	3.08E-06	0
300109	0	6.75E-12	0
300125	0	1.43E-13	0
300147	0	8.60E-13	0
300163	0	8.39E-12	0
300167	0	2.32E-10	0
300174	0	6.31E-06	0
300177	0	2.50E-06	0
300199	0	6.17E-10	0
300205	0	0.2398335	0
300209	0	9.29E-12	0
300213	0	1.28E-06	0
300220	0	2.97E-06	0
300222	0	4.58E-08	0
300238	0	1.55E-06	0
300239	0	9.42E-05	0
300240	0	9.99E-13	0
300247	0	1.35E-08	0
300250	0	2.13E-09	0
300275	0	2.45E-10	0
300292	0	3.96E-09	0
300300	0	6.52E-06	0
300319	0	6.92E-08	0
300331	0	1.94E-10	0
300333	0	5.09E-06	0
300352	0	2.19E-06	0
300357	0	2.47E-08	0
300366	0	4.64E-08	0

ST Identifier	Actual_Class	Predicted_Probability	Predicted_Class
300368	0	4.11E-09	0
300373	0	0.000834949	0
300376	0	1.83E-12	0
300385	0	4.08E-06	0
300389	0	2.24E-09	0
300393	0	6.15E-10	0
300397	0	3.91E-06	0
300398	0	4.44E-08	0
300403	0	1.75E-10	0
300436	0	1.90E-09	0
300443	0	1.13E-06	0
300451	0	7.44E-08	0
300456	0	4.84E-07	0
300457	0	7.53E-11	0
300458	0	9.84E-05	0
300465	0	6.96E-10	0
300475	0	3.89E-12	0
300476	0	7.19E-09	0
300479	0	2.65E-14	0
300486	0	2.39E-11	0
300496	0	1.63E-11	0
300499	0	6.44E-05	0
300501	0	0.007480376	0
300509	0	4.09E-09	0
300517	0	2.40E-05	0
300534	0	1.33E-07	0
300546	0	8.56E-14	0
300555	0	6.23E-15	0
300563	0	0.85654867	1
300570	0	2.13E-11	0
300600	0	3.55E-13	0
300657	0	1.71E-10	0
300660	0	7.71E-11	0
300686	0	5.88E-11	0
300723	0	2.48E-10	0

Table A.2 Comparison of Model Prediction Results for ST Companies (Partial Sample)

ST Identifier	Actual_Class	Predicted_Probability	Predicted_Class
002835	1	1	1
300211	1	0.99988	1
300379	1	1	1
300382	1	1	1
300387	1	0.9999996	1
300519	1	0.9987622	1
300548	1	0.9999992	1
300573	1	0.9949867	1
300582	1	0.99990135	1
300623	1	1	1
600127	1	0.9999968	1
600165	1	0.9999948	1
600351	1	0.9999995	1
600488	1	0.9999714	1
600565	1	0.65800047	1
600590	1	0.999997	1
600724	1	0.9994336	1
600829	1	0.9990444	1
603015	1	1	1
603067	1	0.9999997	1
603131	1	0.99868673	1
603160	1	1	1
603308	1	1	1
603515	1	0.99998146	1
603678	1	0.9997452	1
603828	1	1	1
603861	1	0.99999905	1
603959	1	0.6835272	1
42	1	0.99999577	1
70	1	0.9999999	1
2341	1	0.99999595	1
2402	1	1	1
2403	1	0.9999998	1
2528	1	0.99929905	1
2791	1	0.9628967	1
2822	1	0.99999344	1
2841	1	0.9999973	1
2906	1	0.9999887	1
300331	1	1	1
300393	1	0.9980362	1
300398	1	0.9999551	1
300501	1	0.9999997	1

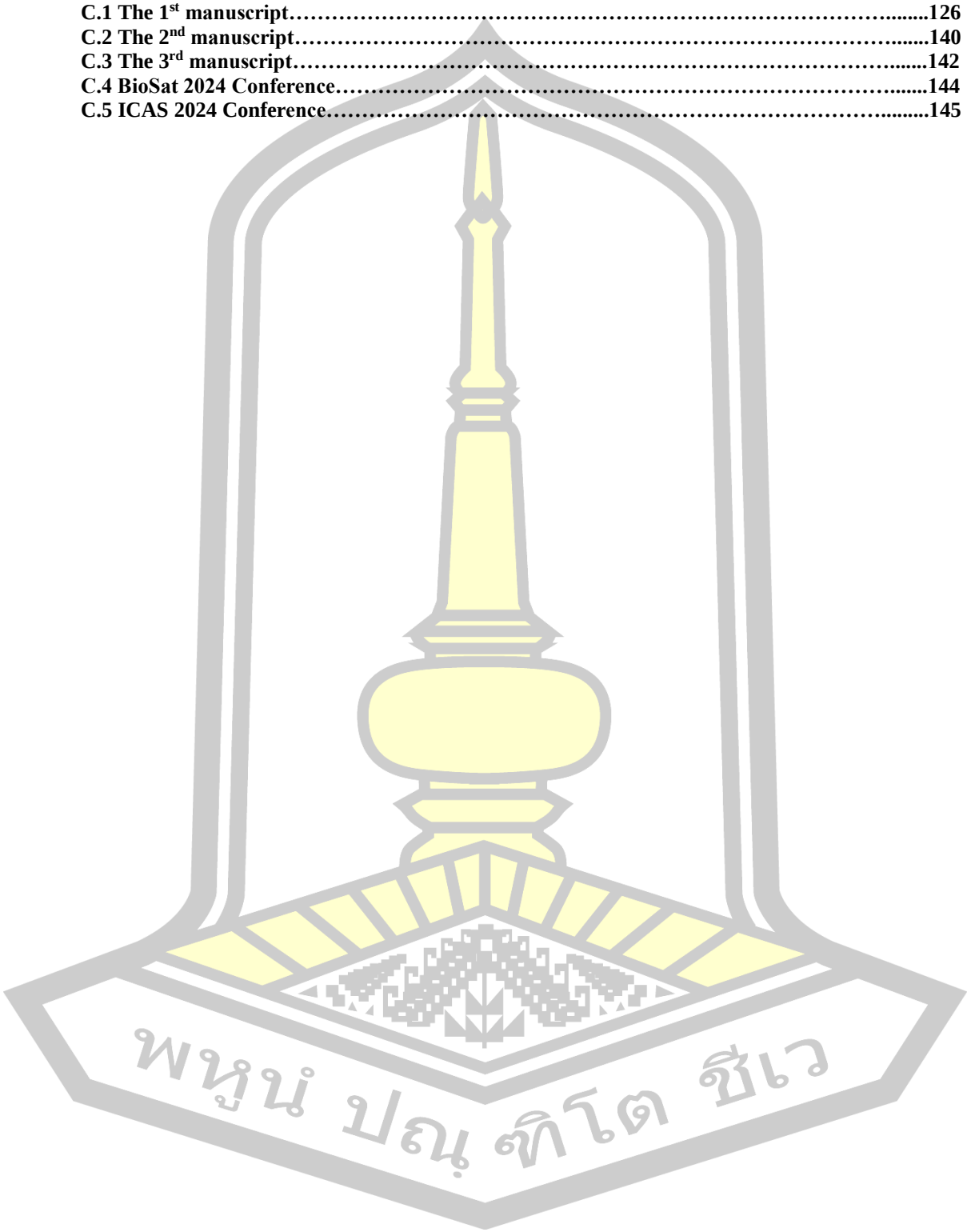
ST Identifier	Actual_Class	Predicted_Probability	Predicted_Class
600770	1	0.9998668	1
600777	1	0.9999917	1
603318	1	0.96307665	1
603628	1	0.99988085	1
603986	1	0.9999829	1
000919	1	0.9999999	1
002034	1	0.99999946	1
002223	1	0.99999243	1
002480	1	0.9999816	1
002528	1	1	1
002898	1	0.99996984	1
300048	1	0.999702	1
300240	1	0.000909373	0
300393	1	0.9999907	1
300456	1	0.99761605	1
300600	1	0.8487532	1
600220	1	0.99988574	1
600351	1	0.9999921	1
600488	1	0.9993933	1
600777	1	0.999985	1
603380	1	0.9999984	1
603618	1	0.9937391	1
603959	1	0.9997362	1
000070	1	0.9999943	1
000836	1	1	1
002424	1	0.99999267	1
002815	1	0.99990034	1
002822	1	0.9978923	1
002845	1	0.9991095	1
002906	1	0.99963313	1
300048	1	0.9999875	1
300205	1	0.9523313	1
300222	1	0.9999997	1
300250	1	1	1
300389	1	0.99999934	1

APPENDIXES B

```
##### Main Program ###############
### Sliding window
time_steps =8
X, y = [], []
for company in data['Security Name'].unique():
    company_data = data[data['Security Name'] == company].sort_values(by='ST Time')
    company_features = features_scaled[company_data.index]
    company_target = company_data['ST Identifier'].values
    for i in range(len(company_features) - time_steps + 1):
        X.append(company_features[i:i + time_steps])
        y.append(company_target[i + time_steps - 1])
X = np.array(X)
y = np.array(y)
### Custom attention mechanism
class Attention(Layer):
    def __init__(self, **kwargs):
        super(Attention, self).__init__(**kwargs)
    def build(self, input_shape):
        self.W=self.add_weight(name='attention_weight',shape=(input_shape[-1],          input_shape[-
1]),initializer='uniform', trainable=True)
        self.b = self.add_weight(name='attention_bias', shape=(input_shape[-1],),
initializer='uniform', trainable=True)
        super(Attention, self).build(input_shape)
    def call(self, x):
        e = K.tanh(K.dot(x, self.W) + self.b)
        a = K.softmax(e, axis=1)
        output = x * a
        return K.sum(output, axis=1)
    def compute_output_shape(self, input_shape):
        return input_shape[0], input_shape[-1]
### Define activation function optimization model
def create_model(structure, activation_fn=None, loss_fn=None):
    if isinstance(activation_fn, str):
        activation_fn = get_activation(activation_fn)
    elif not callable(activation_fn):
        raise ValueError(f"Invalid activation function: {activation_fn}")
    model = Sequential()
    if structure == 'CNN-BiLSTM-AT':
        model.add(Conv1D(64, kernel_size=3, activation=activation_fn, padding='same'))
        model.add(MaxPooling1D(pool_size=2))
        model.add(Dropout(0.5))
        model.add(Bidirectional(LSTM(64,return_sequences=True)))
        model.add(Dropout(0.2))
        model.add(Attention())
        model.add(Flatten())
        model.add(Dense(128, activation=activation_fn))
        model.add(Dropout(0.1))
        model.add(Dense(128, activation=activation_fn))
        model.add(Dropout(0.3))
        model.add(Dense(1, activation='sigmoid'))
    else:
        raise ValueError(f"Invalid structure: {structure}")
    model.compile(optimizer='adam', loss=loss_fn, metrics=['accuracy'])
        return model
```

APPENDIXES C

C.1 The 1st manuscript…….....126
C.2 The 2nd manuscript…….....140
C.3 The 3rd manuscript…….....142
C.4 BioSat 2024 Conference…….....144
C.5 ICAS 2024 Conference…….....145



C.1 The 1st manuscript

Lobachevskii Journal of Mathematics
ISSN: 1995-0802 (Print), 1818-9962 (Online)
Publisher: Pleiades Publishing, Ltd. Distributed by Springer.
Lobachevskii Institute of Mathematics and Mechanics,
Kazan Federal University, Kremlevskaya ul. 18,
Kazan, Tatarstan, 420008 Russia. Phone: +7 (843) 233-72-46;
e-mail: ljmeditor@gmail.com; http://ljm.kpfu.ru

30 September, 2024

Dear Professors Yingying Song, Monchaya Chiangpradit, Kamon Budsaba, and Piyapatr Busababodhin,

Ref: The Application of Genetic Algorithm Optimized Neural Network in Financial Risk Early Warning Model of Agricultural Listed Companies.

Our referees have now considered your paper and have recommended publication in Lobachevskii Journal of Mathematics. We are pleased to accept your paper in its current form which will now be forwarded to the publisher for copy editing and typesetting. Our plan is to publish your paper in Issue 12 (December) of Volume 45 (year 2024).

You will receive proofs for checking, and instructions for transfer of copyright in due course.

The publisher also requests that proofs are checked and returned within 48 hours of receipt.

Thank you for your contribution to Lobachevskii Journal of Mathematics and we look forward to receiving further submissions from you.

Sincerely,
Andrei Volodin
Editor, Lobachevskii Journal of Mathematics

ISSN 1995-0802, Lobachevskii Journal of Mathematics, 2025, Vol. 46, No. 2, pp. 888–900. © Pleiades Publishing, Ltd., 2025.

The Application of Genetic Algorithm Optimized Neural Network in Financial Risk Early Warning Model of Agricultural Listed Companies

Yingying Song[1*], Monchaya Chiangpradit[1], Kamon Budsaba[2***], and Piyapatr Busababodhin[1****]**

(Submitted by A. I. Volodin)

[1] *Department of Mathematics, Faculty of Science, Mahasarakham University, Maha Sarakham, 44150 Thailand*

[2] *Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Rangsit Center, Pathumthani, 12120 Thailand*

Received September 1, 2024; revised September 20, 2024; accepted October 30, 2024

Abstract—In China, a major agricultural country, the agricultural economy plays a crucial role in the national economy. As a representative of the agricultural industry chain, agricultural listed companies not only bear the responsibility of support, but also face various financial risks such as market fluctuations, climate change, and policy adjustments. This research is based on agricultural listed companies on the Shanghai and Shenzhen A-shares from 2010 to 2023, taking into account both financial and non-financial indicators from three years ago. Principal Component Analysis (PCA) dimensionality reduction and oversampling techniques (SMOTE) were used to handle sample imbalance. Through the BP neural network model optimized by genetic algorithm, the financial risk of agricultural listed companies was successfully predicted, and the overall accuracy rate reached 94%. Compared with the original neural network, the optimized neural network had better performance. This research provides important decision support for agricultural enterprises and investors and establishes an empirical foundation for further optimization of financial risk warning models.

2000 Mathematics Subject Classification: Primary 62P12, Secondary 62F99.

DOI: 10.1134/S199508022560030X

Keywords and phrases: *Neural Network; Genetic algorithm; Financial risk; Agricultural listed companies*

I. INTRODUCTION

China is a large agricultural country, and the development of agriculture is not only the foundation of national construction, but also an important guarantee for people's clothing, food, housing, and transportation. As a leading enterprise in agricultural industrialization, agricultural listed companies guide the transformation of agriculture towards market orientation, promote the optimization of resource allocation, upgrade the agricultural industry chain and value chain, promote the construction of smart agriculture and agricultural digitization, and play a crucial role in the development of the national economy. Due to the particularity and high risk of the industry, agricultural listed companies still face problems such as poor profitability, excessive dependence on national policies, diversified production and operation, long return cycles, unreasonable asset structure, and industrial structure upgrading. As a

*E-mail:yingyingsong565@gmail.com
**E-mail:monchaya.c@msu.ac.th
***E-mail:kamon@mathstat.sci.tu.ac.th
****Corresponding Author E-mail:piyapatr.b@msu.ac.th

result, from 2015 to 2022, the number of agricultural listed companies has remained at 40–50, and their number have not shown a rapid increase trend compared to other industries. This does not match the background of being a major agricultural country. The operating performance of some agricultural listed companies has been seriously affected, and the probability of financial risk occurrence has increased, even falling into a passive position of ST.

Therefore, constructing a precise and effective financial crisis warning model for agricultural listed companies will enhance the risk identification ability of investors and creditors, provide strong decision-making support, and encourage companies to take appropriate risk management and governance measures before financial risks escalate. In addition, it helps to strengthen government regulation and promote industry compliance and stability.

Scholars have conducted varying degrees of research on financial risk warning. In 1932, Fitzpatrick [1] first proposed the concept of a univariate warning model, which uses the level of a single financial indicator as the criterion for judging a company's risk. However, due to limited capacity of a single indicator to comprehensively and objectively reflect the operational situation, Altman [2] proposed using multivariate discriminant warning models to improve prediction accuracy and introduced new financial indicators to establish the Z-Score model. Then, Ohlson first used the logistic regression model for financial risk warning in 1980 and found that the results were better than the discriminant analysis model [3]. Subsequently, Odom and Sharda [4] applied artificial neural network technology to the field of financial warning and found that the prediction accuracy of the BP neural network model was superior to traditional multivariate warning models. Vapnik [5] introduced the support vector machine (SVM) method in 1995, which addresses classification prediction problems by seeking the optimal hyperplane to maximize the margin between different classes. This approach achieved higher accuracy. In recent years, there has been a growing trend of comparing multiple machine learning and deep learning models for financial warning,For example, Ma Xuhui [6] compared decision trees, random forests, and XGBoost models for financial warning, Ouyang and Lai [7] also found that the Long Short Term Memory Network (LSTM) model had higher accuracy than other neural network models by combining deep learning techniques with the Chinese financial market.

Previous research has mainly focused on the general field of financial warning, with relatively little attention paid to agricultural listed companies. Ma Xiaoli [8] and Yu Jingxuan [9] used Z-score and F models for early warning research, while other scholars used logistic regression models to reconstruct financial risk prediction models for agricultural listed companies[10–12]. Compared with logistic regression models, using neural network models, the accuracy of financial crisis prediction for agricultural listed companies was higher [13]. In addition, Xueyong Bian [14] used a genetic algorithm optimized SVM model to improve the accuracy by 86.5% compared to traditional SVM models, from 79.2%. The current financial warning for agricultural listed companies mainly focuses on Z-score model, F-model, and logistic regression model. There are still relatively few mainstream prediction models used in other industries, such as neural networks, SVM, XGBoost, etc. At present, the use of principal component analysis, SMOTE oversampling, genetic algorithm, and BP neural network model for financial risk warning of agricultural listed companies is still in the exploratory stage. This study describes the use of Python software to comprehensively apply the above methods for effective warning classification of agricultural listed companies.

2. DATA AND INDICATORS

2.1. Data

This study selected 42 listed companies in the agricultural, forestry, animal husbandry, and fishery sectors of the Shanghai and Shenzhen A-shares from 2010 to 2023. The sample company data is from the CSMAR database. We used the commonly used definition criteria by most Chinese scholars, taking the special treatment (including ST and * ST) implemented by listed companies due to abnormal financial conditions (such as consecutive losses in the past two years) as a sign of their financial distress, and selecting appropriate financially distressed companies as research samples based on this principle. The year of was recorded as T by ST, the previous year of being recorded as T-1 by ST, and the previous two years of were recorded as T-2 by ST. Choosing data from T-2 to T as a warning is not very meaningful. This study takes 9 ST agricultural companies that were first subjected to special treatment that year as ST samples, selects financial and non-financial data from T-3 years as sample data to analyze for early warning, and the remaining 33 agricultural companies that have not been specially treated were considered as normal operating samples.

Table 1. Early warning indicator system for agricultural listed companies

Primary indicators	Number	Secondary indicators	Primary indicators	Number	Secondary indicators
Debt paying ability	X1	Asset liability ratio	Ratio structure	X23	Cash flow to capital expenditures ratio
	X2	Cash ratio		X24	Capital expenditure ratio
	X3	Operating working capital		X25	Bank borrowing ratio
	X4	Equity multiplier		X26	Current asset ratio
	X5	Equity ratio		X27	Fixed asset ratio
	X6	Quick ratio		X28	Working capital to current asset ratio
	X7	Current ratio		X29	Current liability ratio
Business capability	X8	Total asset turnover		X30	Non-Current liability ratio
	X9	Inventory turnover		X31	Retained earnings to asset ratio
	X10	Current asset turnover		X32	Working capital ratio
	X11	Fixed asset turnover		X33	Non-Current asset ratio
Development capability	X12	Total asset growth rate	Cash flow capability	X34	Operating index
	X13	Owner's equity growth rate		X35	Total cash recovery rate
Profitability	X14	Basic earnings per share		X36	Cash net content of operating income
	X15	Return on assets (ROA)	Non-financial indicators	X37	Independent director proportion
	X16	Return on equity (ROE)		X38	Property right nature
	X17	Return on invested capital (ROIC)		X39	Ownership percentage of largest shareholder
	X18	Operating cost ratio		X40	Sum of ownership percentages of top 10 shareholders
	X19	Operating profit margin		X41	Analyst coverage
	X20	Net profit margin		X42	Media coverage
Business indicators	X21	Company size		X43	Audit opinion
	X22	Net profit			

2.2. Indicators

Based on the actual situation of listed agricultural companies in China and the new policy guidelines, 43 indicators were set up as a warning indicator system from the aspects of debt paying ability,

development capability, profitability, business capability, cash flow capability, ratio structure, non-financial indicators, etc.

3. METHODOLOGY

3.1. Smote Oversampling

SMOTE (Synthetic Minority Oversampling Technique) was proposed by Chawla in 2002 [15]. It is a technique that uses synthetic minority oversampling to balance class distribution by adding minority samples, reducing the risk of overfitting while improving the model's generalization ability. Specific implementation steps are as follows.

(1) Minority class samples were randomly selected: Sample x_i from the minority class samples was selected as the benchmark sample for synthesizing new samples and for calculating the Euclidean distance between this sample x_i and other minority class samples to obtain the K nearest neighbors of sample x_i.

(2) Neighboring samples were randomly selected: Sampling rate was set as $N\%$ based on the imbalanced proportion of the samples, x_{ij} was randomly selected as the sample from the selected K nearest neighbors, and the operation was repeated N times to obtain samples $x_{11}, x_{12}, \ldots, x_{kN}$.

(3) Composite new sample: linear interpolation was performed between the benchmark sample x_i and the nearest neighbor sample x_{ij} to construct a new sample, generating a new sample $g = x_i + e(x_{ij} - x_i)$, where e is a random number of [0,1].

3.2. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a commonly used data dimensional reduction technique proposed by Pearson [16], which transforms the original data into a set of data represented by linearly independent variables through orthogonal transformation with minimal information loss, and transforms multiple indicators into multiple comprehensive indicators using multivariate statistical methods. Smith [17] provides a detailed description of the calculation steps involved in PCA, which include computing the covariance matrix, determining the eigenvalues and eigenvectors of the covariance matrix, and selecting the principal components. The specific implementation steps are:

(1) Assuming there are n companies, each with m observation indicators, forming a matrix

$$X = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1m} \\ x_{21} & x_{22} & \ldots & x_{2m} \\ x_{n1} & x_{n2} & \ldots & x_{nm} \end{pmatrix},$$

where x_{ij} represents the jth indicator of the ith company.

(2) Standardization processing: normalize the original variable x_{ij} to obtain $y_{ij} = (x_{ij} - x_i)/s_j$, where x_j is the sample mean $1/n$, s_j is the sample standard deviation $\sum_{i=1}^{n} x_{ij}$, and output the normalization matrix $Y = (y_{ij})_{n\times m}$.

(3) Calculate the correlation coefficient matrix

$$R = \begin{pmatrix} r_{11} & r_{12} & \ldots & r_{1m} \\ r_{21} & r_{22} & \ldots & r_{2m} \\ r_{m1} & r_{m2} & \ldots & r_{mm} \end{pmatrix},$$

where r_{jk} is the correlation coefficient between the jth indicator and the kth indicator of the company $\frac{1}{n-1}\sum_{i=1}^{n} y_{ij}y_{ik}, i = 1, 2, \ldots, n; j, k = 1, 2, \ldots, m$.

(4) Calculate the eigenvalues of the correlation coefficient matrix λ and eigenvector A: for eigenvalues λ, sort from top to bottom $\lambda_1 > \lambda_2 > \ldots > \lambda_m$. The corresponding feature vectors $\mathbf{a_j} = (a_{1j}, a_{2j}, \ldots, a_{mj})^T, j = 1, 2, \ldots, m$.

(5) Calculate the principal component contribution rate and cumulative contribution rate: contribution rate $\lambda_j / \sum_{i=1}^{m} \lambda_i$, cumulative contribution rate $\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{m} \lambda_i$, where k is the number of selected principal components.

(6) Determine the principal components: $z_j = a_{1j}y_1 + a_{2j}y_2 + \ldots + a_{mj}y_m$, $j = 1, 2, \ldots, k$, where z_1 to z_k represent the 1st to kth principal components.

3.3. Genetic Algorithm(GA)

GA (Genetic Algorithm) was developed by Professor J. Holland who first proposed it in 1975[18]. It is an iterative optimization algorithm that simulates species evolution patterns. It calculates fitness on the generated initial population according to the rule of survival of the fittest, and iteratively optimizes the population through selection, crossover, mutation, and other operations, thereby outputting the global optimal solution [18, 19]. This article uses genetic algorithm to optimize the initial weights and thresholds of neural networks to avoid falling into local minima. The specific operation is as follows:

(1) Calculate fitness value: the fitness function $f(x)$ is used to evaluate the quality of the solution, where x is the binary or real string corresponding to the solution. The calculation formula for the fitness function depends on the specific problem.

(2) Selection operation $p_i = f(x_i) / \sum_{i=1}^{n} f(x_i)$, where $f(x_i)$ is the probability of individual i being selected, and is the fitness value of individual i. Evaluate the probability of each individual being selected, and individuals with higher probabilities will have a greater chance of being selected into the next generation.

(3) Cross operation: cross operation is the exchange of partial genes between two individuals to generate new individuals. The preset fixed value of p_c represents the probability of cross operation, which determines the frequency of cross operation. A higher cross probability helps to increase population diversity.

(4) Mutation operation: mutation operation randomly changes individual genes, which helps to increase population diversity and avoid getting stuck in local optima. The preset fixed value p_m represents the probability of mutation operation, which determines the frequency of mutation operation. A higher mutation probability can help jump out of local optima, but may also damage better solutions.

3.4. BP Neural Network

BP (back propagation) neural network is a multi-layer feedforward neural network trained using the error backpropagation algorithm proposed by Rumelhart and McClelland in 1986[20]. By using a simple gradient descent method, it minimizes the total or average error of the output calculated by the network. In neural networks, there are input layers, hidden layers, and output layers. The specific implementation steps are as follows.

(1) Forward propagation: assuming there are n inputs and m outputs in the network, and s neurons in the hidden layer.

The output of the jth neuron in the hidden layer is $h_j = f_1(\sum_{i=1}^{n} w_{ij}x_i - \theta_j)$, where f_1 is the activation function of the hidden layer, w_{ij} is the connection weight from the input layer to the hidden layer, x_i is the input value, and θ_j is the threshold of the jth neuron.

The output layer $y_k = f_2(\sum_{j=1}^{m} w_{jk}h_j - \theta_k)$, where f_2 is the activation function of the output layer, w_{jk} is the connection weight from the hidden layer to the output layer, and θ_k is the threshold of the output layer.

The error function $\frac{1}{2}\sum_{k=1}^{m}(y_k - t_k)^2$, where t_k is the expected output value. Common error functions include mean square error, cross entropy error, etc.

(2) Backpropagation: using the chain rule to adjust the weights and biases of each neuron in the network, minimizing the error between the network output and the actual label, reducing the error to a pre-set minimum value or stopping at a preset training step, achieving convergence [20, 21].

3.5. GA-BP Neural Network

GA-BP neural network (BP neural network optimized by genetic algorithm) is a hybrid algorithm that combines the genetic algorithm (GA) and backpropagation neural network (BPNN). This algorithm aims to utilize the global search capability of genetic algorithms to optimize the weights and thresholds of BP neural networks, thereby improving the performance of neural networks. This study used genetic algorithm to adjust the hyperparameters of neural networks, continuously evolving to generate new individuals, in order to maximize the accuracy of the neural network on the test set [22, 23]. The specific implementation steps are as follows.

(1) Initialization:

① Neural network initialization: random initialization of weights and thresholds in a backpropagation neural network.

② Genetic algorithm initialization: set parameters such as population size, maximum evolution generations, crossover probability, and mutation probability. Randomly generate an initial population, where each individual in the population represents a set of weights and thresholds for a neural network.

(2) Fitness function calculation:

For each individual in the population, calculate its fitness value using a fitness function. The fitness function typically evaluates how well an individual performs the given task or problem. Using error or a certain function (such as the sum of squares of errors or the accuracy of neural networks) as an individual's fitness value.

The error function: $MSE = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, where N refers to the number of samples, y_i represents the desired or expected output, and $\hat{y}_i$ represents the predicted output.

The accuracy of neural networks: $\text{Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Total number of samples}}$.

(3) Genetic operations:

Genetic operations encompass selection, crossover, and mutation. In the selection phase, excellent individuals are filtered based on their fitness values, with common methods including roulette selection and tournament selection. In the crossover phase, individuals are randomly chosen and their genes are exchanged with a certain probability to generate offspring. In the mutation phase, genes of individuals are subject to minor changes with a certain probability, often achieved by randomly adding or subtracting a small value.

(4) Iteration and termination condition:

① Iteration: repeat selection, crossover, and mutation operations to generate a new population.

② Termination condition: determine whether the maximum evolutionary generation has been reached or whether the fitness value has reached a preset threshold. If satisfied, stop the iteration.

(5) BP neural network optimization and prediction

Train the BP neural network using the optimal solution (i.e., weight and threshold) found by genetic algorithm, and use the trained neural network for prediction or classification.

4. RESULTS

4.1. SMOTE Oversampling

Before oversampling, this study divided 42 sample data into training and testing sets in a 6:4 ratio. The training set consisted of 25 samples, with 19 samples from normally operating companies and 6 samples from ST companies. The testing set consisted of 17 samples. After applying SMOTE oversampling to the training set, the sample size increased to 38, with both ST companies and normally operating companies balanced at 19 samples each. This indicates that SMOTE successfully generated synthetic samples to balance the class distribution, which helps improve the model's performance on imbalanced datasets.

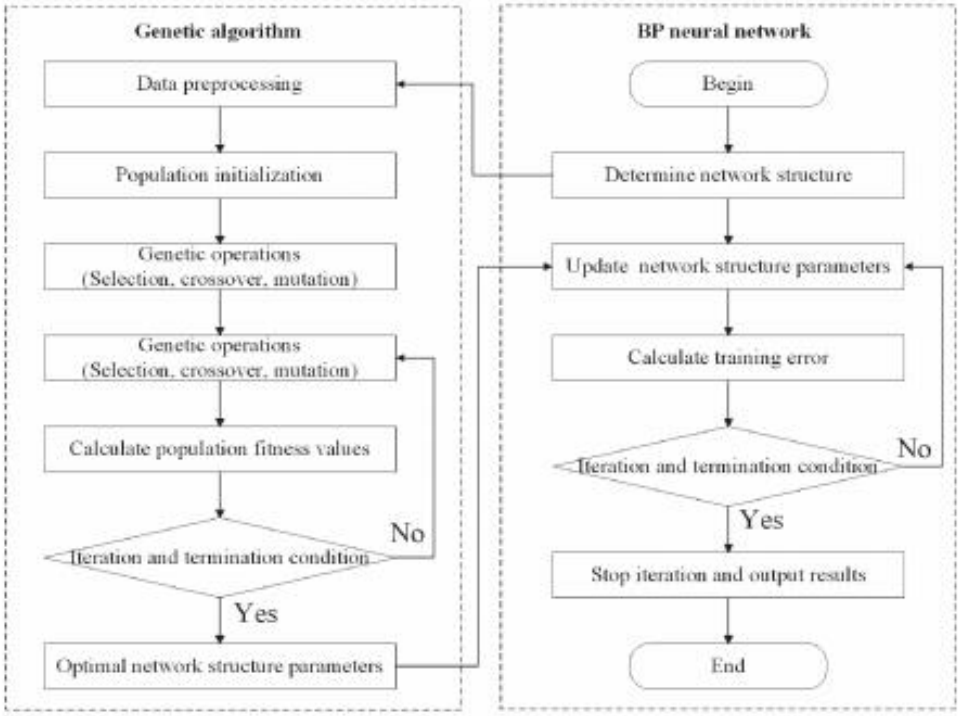


Fig. 1. Process diagram for constructing GA-BP model.

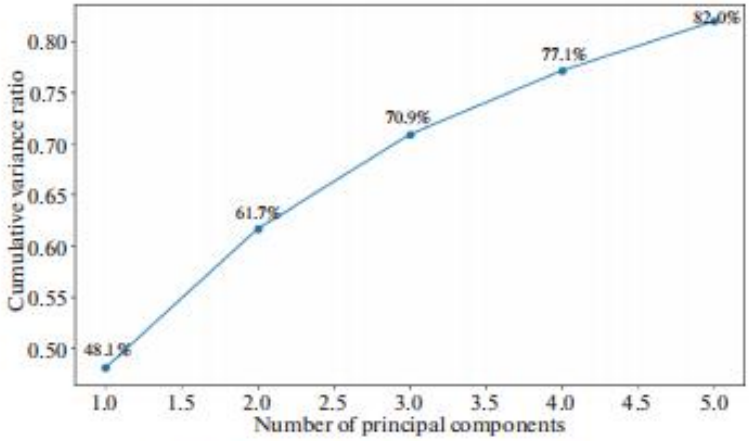


Fig. 2. Curve of PCA cumulative contribution rate.

4.2. Principal Component Analysis

By using Python software for data standardization and PCA, the cumulative contribution rate of variance of the first 5 principal component components was 82.0%. Therefore, this study selected the first 5 principal component indicators to represent all indicators of company operation. Figure 3 shows the heatmap of the correlation coefficients between the extracted 5 principal components and the original 43 indicators. Among them, the indicators with higher correlation coefficients (absolute

Table 2. Number of datasets before and after oversampling

Financial risk identification	Original training set	Over sampled training set
Normal company	19	19
ST company	6	19
Total	25	38

value 0.3) includes: total asset to net profit margin (ROA), return on equity (ROE), operating cost ratio, operating net profit margin, enterprise size (total assets), operating profit margin, basic earnings per share, fixed asset turnover rate representing operating ability indicator, equity ratio representing debt paying ability indicator, owner's equity growth rate representing development ability indicator, cash flow capital expenditure ratio representing ratio of ratio structure indicator, non-current liability ratio, retained earnings asset ratio, capital expenditure ratio, sum of ownership ratio representing non-financial indicator,percentages of top 10 shareholders and analyst attention representing non-financial indicator.

4.3. Results before and after GA Optimization

The study initially involved configuring the parameters of a genetic algorithm to determine the optimal hyperparameters for a neural network. Subsequently, the neural network underwent training, and a comparative analysis was conducted between the results obtained before and after optimization.

(1) Fitness functions

The fitness function measures the accuracy of a neural network on the test set. For each individual, a set of hyperparameters of the neural network, the fitness value was calculated by training the

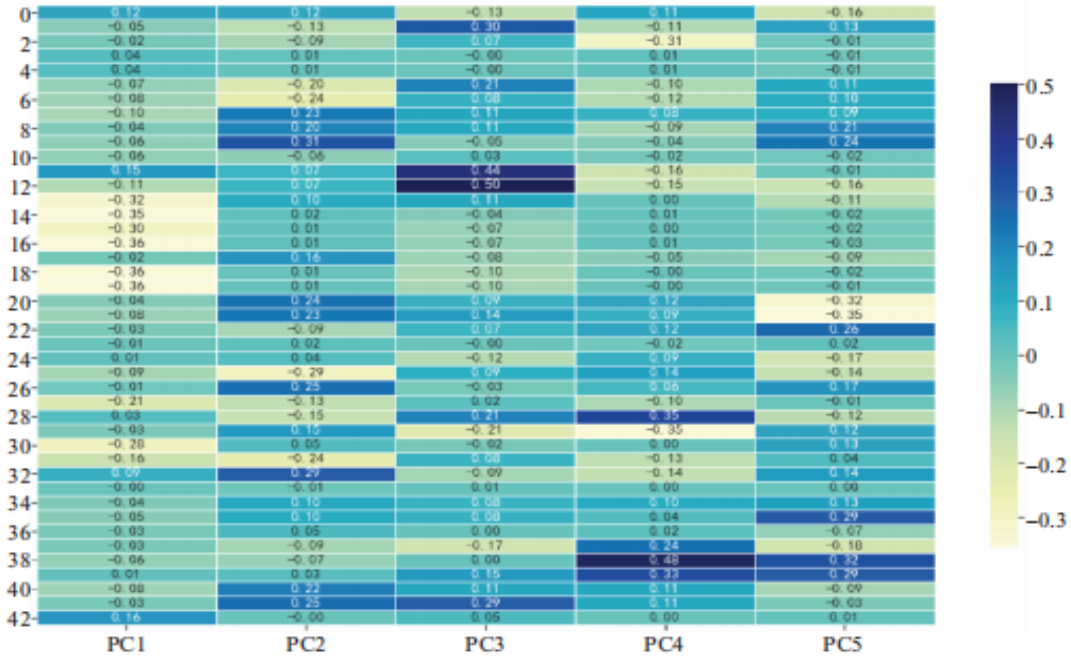


Fig. 3. Heat map of the relationship between principal components and original features.

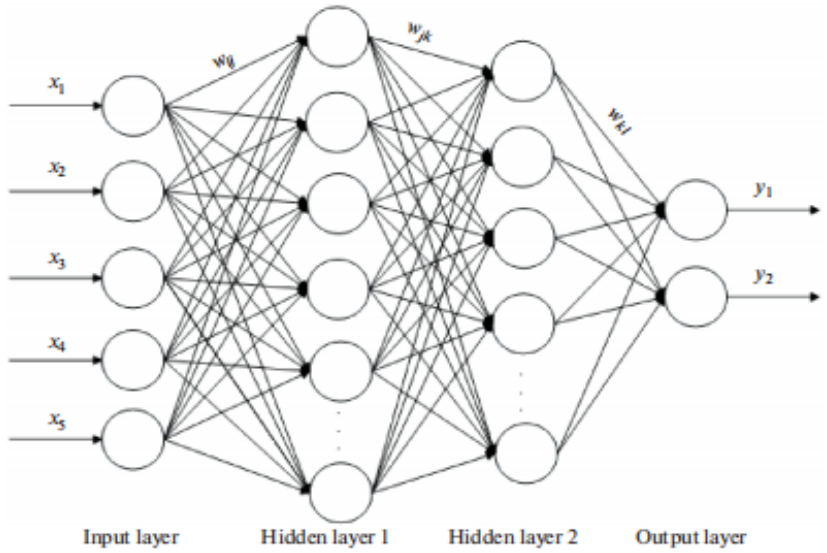


Fig. 4. The structure diagram of BP neural network optimized by genetic algorithm.

neural network and evaluating the accuracy on the test set. Assuming an individual (neural network configuration) had an accuracy of 85% on the test set. This accuracy was used as a fitness value.

(2) Initialize population

Use standard genetic algorithms to initialize the population. The encoding of each individual includes all connection weights of the neural network. The length of the colorizer is equal to the total number of connections in the neural network. This study used a toolbox to create an initial population of 10 individuals, each consisting of three genes representing learning rate, the number of neurons in the first hidden layer, and the number of neurons in the second hidden layer.

(3) The iterative process of genetic algorithm

① Select operation: Using the accuracy of neural networks as fitness values for roulette wheel selection, selecting the better individuals as parents, and using the NSGA-II method to select 10 individuals from each generation to pass on to the next generation.

Table 3. Parameter Values of Genetic Algorithm

Parameter name	Parameter value	Parameter name	Parameter value
Population size	20	Selection mechanism	NSGA-II: Each generation chooses to retain 10 individuals for the next generation
Cross probability	0.8	Cross operator	Mixing intensity 0.5: The genes of two parental individuals are evenly mixed
Mutation probability probability of individual being mutated 0.5	0.2	Mutation operator	Gaussian variation: mean 0, standard deviation 1,
Iterations	10	Generate new individuals	20 (New individuals generated by crossover and mutation)

Table 4. Hyperparameter values of neural networks

Hyperparameter name	Hyperparameter value
Learning rate	0.043
Number of neurons in hidden layer 1	344
Number of neurons in hidden layer 2	294
Iterations	10

Table 5. Comparison of results before and after smote and genetic algorithm optimization

	Model	Type	Precision	Recall	F1-score	Accuracy
No-Smote	BP neural network	0 (normal)	1.00	0.57	0.73	0.65
		1 (ST)	0.33	1.00	0.50	
	GA-BP neural network	0 (normal)	0.92	0.79	0.85	0.76
		1 (ST)	0.40	0.67	0.50	
Smote	BP neural network	0(normal)	1.00	0.86	0.92	0.88
		1 (ST)	0.60	1.00	0.75	
	GA-BP neural network	0 (normal)	1.00	0.93	0.96	0.94
		1 (ST)	0.75	1.00	0.86	

② Cross operation: by crossing the genes (connection weights) of individual parents in some way, a new individual is formed. This study used a mixed crossover operator, where the genes of two parent individuals were uniformly mixed and crossed to generate new individuals.

③ Mutation operation: Perform mutation operations on individuals. This study used a Gaussian mutation operator to perform small mutations on genes (connection weights) and generated 20 new individuals through crossover and mutation operations.

④ Iteration termination: calculate the fitness value of the new individual, that is, train the neural network with new hyperparameters and evaluate the accuracy on the test set. Based on the fitness value, select new individuals to form the next generation population, and repeat the above steps until the predetermined number of iterations is reached. Select the Pareto frontier of non-dominated sorting from the final population and select an individual with the best accuracy.

(4) Building the final neural network model.

The GA-BP neural network model adopted two hidden layers, with five neurons set in the input layer: five principal component indicators processed by PCA, and two neurons set in the output layer ST company and normal company.

Construct the final neural network model using hyperparameters extracted from the optimal individual, such as learning rate (0.043) and number of hidden layer neurons(hidden layer 1 : 344; hidden layer 2 : 294). Train the model on the entire training set using the hyperparameters of the final neural network model. Predict the test set, calculate the accuracy of the final neural network model on the test set at 0.94, and output performance indicators.

(5) Comparison of results before and after SMOTE and genetic algorithm optimization.

In the case of without oversampling, the overall accuracy of the neural network before optimization was 0.65, with the accuracy rate for predicting financial risk companies at only 0.33. After optimization using a genetic algorithm, the overall accuracy of the GA-BP neural network improved to 0.76, with the accuracy rate for predicting financial risk companies increasing to 0.40. Thus, compared to the ordinary neural network, the GA-BP neural network demonstrated better overall performance.

In the case of oversampling, the overall accuracy of the neural network before optimization was 0.88. However, the precision rate for predicting financial risk companies was only 0.60, while the

recall rate for predicting normal business companies was 0.86. After optimization using a genetic algorithm, the overall accuracy of the GA-BP neural network increased to 0.94. Moreover, the precision rate for predicting financial risk companies improved to 0.75, and the recall rate for predicting normal companies rose to 0.93. Therefore, the adoption of SMOTE significantly enhanced the model's prediction performance, particularly for certain categories like financial risk. Additionally, the GA-BP neural network model, optimized using a genetic algorithm, exhibited superior performance compared to the BP neural network model.

5. DISCUSSION

In the field of financial risk prediction and management for agricultural listed companies, we will put forward viewpoints and suggestions regarding data collection, algorithm optimization, and other machine learning methods, aiming to provide insights and guidance for further research and practice in financial risk prediction.

In terms of data collection, this study selected sample data from the T-3 years for analysis and early warning. Furthermore, it is suggested that one considers extracting time series data for analysis and modeling to predict the changing trends of financial risk for listed companies over time. Additionally, leveraging natural language processing techniques to extract and analyze semantic and sentiment information from financial texts can provide richer dimensions for financial risk prediction and management.

In terms of algorithm optimization, although this study has demonstrated the effectiveness of genetic algorithms in optimizing neural networks, there is still room for exploration and optimization of the number and structure of hidden layers. Therefore, techniques such as automatic parameter tuning and Bayesian optimization could be employed to further improve the prediction accuracy and stability of the models.

In terms of other machine learning methods, besides neural networks, there are various other machine learning algorithms that can be utilized for financial risk prediction, such as decision trees, support vector machines, XGBoost, and ensemble learning algorithms. Future research might compare the predictive performance of different methods and explore their applicability in various scenarios, thereby providing more comprehensive guidance for practical applications.

Overall, these suggestions aim to contribute to advancements in the field of financial risk prediction and management for agricultural listed companies, facilitating better decision-making and risk mitigation strategies.

6. CONCLUSIONS

In summary, this study provides a perspective of neural network optimization based on genetic algorithms for early warning of financial risks in agricultural listed companies. A summary of this study is as follows.

Effectiveness of Smote oversampling.This approach can effectively solve the problem of small amount of data and category imbalance problems, which helps the model to better learn the features of a few categories. After Smote oversampling, the overall accuracy of the BP neural network was improved from 0.65 to 0.88, and the accuracy in predicting the ST risk category was improved from 0.33 to 0.60, and the overall GA-BP neural network's accuracy improved from 0.76 to 0.94 and from 0.40 to 0.75 in predicting ST risk categories, all of which can significantly improve the model's prediction performance for ST risk.

Correlation of variables. This study used PCA to select five principal component indicators to represent all the indicators of the company's operation, and further analyzed the heat map of correlation coefficients between the 5 principal components and the original 43 indicators and it discovered a number of indicators with high correlation coefficients, including ROA, ROE, operating cost ratio, operating net interest rate, enterprise size, operating profitability, basic EPS, fixed asset turnover, etc. These indicators may be of some importance and should be focused on when forecasting financial risk.

High performance of GA-BP prediction. Genetic algorithm selection, crossover and mutation operations were used to calculate the optimal hyper-parameters of the BP neural network, which exhibited higher prediction accuracy. Without applying SMOTE, the overall accuracy of the BP neural network was 0.65, however, when using the GA-BP neural network, the overall accuracy was improved

to 0.76. In the case of Smote situation, the overall accuracy of the BP neural network was improved from 0.65 to 0.88, the overall accuracy of the GA-BP neural network was improved from 0.76 to 0.94. This indicates that the optimization of neural network parameters using genetic algorithms can significantly improve the prediction performance of the model, especially when dealing with unbalanced data.

The BP neural network method based on SMOTE oversampling, PCA, and genetic algorithm provides decision-makers with strategies for identifying high-risk companies.The ability to accurately predict financial crises can lead to more timely and targeted response measures, minimizing the potential risk of bankruptcy.

FUNDING

This research project was financially supported by Mahasarakham University.

ACKNOWLEDGMENTS

We would like to thank the referees for their comments and suggestions on the manuscript. The authors are grateful to the reviewers for their valuable and constructive comments. Observational data in China were provided by Shenzhen GTA Education Tech Ltd. (CSMAR Database accessed on 10 August 2023) at https://data.csmar.com/.

CONFLICT OF INTEREST

The authors declare no conflict of interest in this area.

REFERENCES

1. P. J. Fitzpatrick, "A comparison of the ratios of successful industrial enterprises with those of failed firm," Certif. Publ. Account. **6**, 727–731 (1932).
2. E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," J. Finance **23**, 589–609 (1968).
3. J. A. Ohlson, "Financial ratios and the probabilistic prediction of bankruptcy," J. Account. Res. **18**, 109–131 (1980).
4. M. D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in *Proceedings of the 1990 IJCNN International Joint Conference on Neural Networks* (IEEE, 1990), pp. 163–168.
5. V. Vapnik, *The Nature of Statistical Learning Theory* (Springer Science, New York, 2013).
6. X. H. Ma, *Research on Machine Learning Based Chinese Company Financial Risk Detect System* (Nanjing Univ., China, 2019).
7. Z. S. Ouyang and Y. Lai, "Systemic financial risk early warning of financial market in China using Attention-LSTM model," North Am. J. Econ. Finance **56**, 101383 (2021).
8. X. L. Ma, *Research on the Financial Crisis Prevention System of Agricultural Listed Companies in China* (Northwest Agricult. Univ., China, 2009).
9. J. X. Yu and S. F. Zheng, "Research on the application of Z-score financial early warning model in agricultural listed companies," Financ. Account. Commun.: Compreh. Part 2 **4**, 36–38 (2012).
10. P. Chen, *Research on Financial Risk Early Warning of Listed Companies in China's Agriculture and Forestry Industry* (Nanjing Forestry Univ., China, 2017).
11. X. Q. Sheng, *Research on the Optimal Methods of Our Country Agricultural Listed Company Financial Crisis Early Warning* (Shanghai Univ. Eng. Sci., Shanghai, 2016).
12. X. X. Chen and H. T. Guo, "Study on financial crisis warning of agricultural listed companies based on factor analysis and logistic regression model," J. Appl. Stat. Manage. **41**, 11–24 (2022).
13. G. J. Chen, *Research on Financial Crisis Warning of Agricultural Listed Companies Based on BP Neural Network Model* (Zhejiang A&F Univ., China, 2020).
14. X. Bian, X. Lv, and J. Tian, "Research on agricultural economic early warning based on genetic algorithm and SVM," J. Sensors **2022**, 3109468 (2022).
15. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res. **16**, 321–357 (2002).
16. K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," London, Edinburgh, Dublin Philos. Mag. J. Sci. **2**, 559–572 (1901).
17. L. I. Smith, A Tutorial on Principal Components Analysis. https://www.iro.umontreal.ca/ pift6080/H09/documents/ Accessed October 2, 2024.

18. J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (Univ. Michigan Press, Oxford, 1975).
19. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, Machine Learning* (Addion Wesley, Reading, MA, 1989).
20. D. E. Rumelhart and J. L. McClelland (PDP Res. Group), *Parallel Distributed Processing*, Vol. 1: *Explorations in the Microstructure of Cognition: Foundations* (MIT Press, Boston, 1986).
21. R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception* (Academic, New York, 1989), pp. 65–93.
22. X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," IEEE Trans. Neural Networks **8**, 694–713 (1997).
23. S. Ding, C. Su, and J. Yu, "An optimizing BP neural network algorithm based on genetic algorithm," Artif. Intell. Rev. **36**, 153–162 (2011).

C.2 The 2nd manuscript

CSEM-D-24-01161 - Submission Confirmation

发件人: Computational Economics (CSEM)<em@editorialmanager.com>
时　间: 2024年9月23日(星期一) 中午12:55
收件人: 高山流水<459030710@qq.com>

Dear Dr. Song,

Thank you for submitting your manuscript, "Hyperband-Optimized CNN-BiLSTM with Attention Mechanism for Corporate Financial Distress Prediction", to Computational Economics

The submission id is: CSEM-D-24-01161

Please refer to this number in any future correspondence.

During the review process, you can keep track of the status of your manuscript on the journal's website.

Your username is: yingyingsong565@gmail.com
If you forgot your password, you can click the 'Send Login Details' link on the EM Login page at https://www.editorialmanager.com/csem/

In case your manuscript is rejected for publication the form will be destroyed.

If your manuscript is accepted for publication in Computational Economics, you may elect to submit it to the Open Choice program. For information about the Open Choice program, please access the journal's webpage.

With kind regards,
Springer Editorial Office
Computational Economics

Now that your article will undergo the editorial and peer review process, it is the right time to think about publishing your article as open access. With open access your article will become freely available to anyone worldwide and you will easily comply with open access mandates. Springer's open access offering for this journal is called Open Choice (find more information on www.springer.com/openchoice). Once your article is accepted, you will be offered the option to publish through open access. So you might want to talk to your institution and funder now to see how payment could be organized; for an overview of available open access funding please go to www.springer.com/oafunding.
Although for now you don't have to do anything, we would like to let you know about your upcoming options.

This letter contains confidential information, is for your own use, and should not be forwarded to third parties.

C.3 The 3rd manuscript

Yingying Song <65010263002@msu.ac.th>

[Sustainability] Manuscript ID: sustainability-3527650 - Submission Received

1 message

Editorial Office <sustainability@mdpi.com> Thu, Feb 27, 2025 at 12:18 PM
Reply-To: sustainability@mdpi.com
To: Piyapatr Busababodhin <piyapatr.b@msu.ac.th>, Yingying Song <65010263002@msu.ac.th>
Cc: Monchaya Chiangpradit <monchaya.c@msu.ac.th>

Dear Professor Busababodhin,

Thank you very much for uploading the following manuscript to the MDPI submission system. One of our editors will be in touch with you soon.

Journal name: Sustainability
Manuscript ID: sustainability-3527650
Type of manuscript: Article
Title: Composite Triple Activation Function : Enhancing CNN-BiLSTM-AM for Sustainable Financial Risk Prediction in Manufacturing
Authors: Song Yingying, Monchaya Chiangpradit, Piyapatr Busababodhin *
Received: 27 Feb 2025
E-mails: 65010263002@msu.ac.th, monchaya.c@msu.ac.th, piyapatr.b@msu.ac.th
Economic and Business Aspects of Sustainability
https://www.mdpi.com/journal/sustainability/sections/management_aspects_of_sustainability

We encourage you to provide an Author Biography on this publication's webpage. Please click the following link to find the corresponding instructions and decide whether to accept our invitation:
https://susy.mdpi.com/user/manuscript/author_biography/dcfca314eaea6ce93b215d81f5234003

You can follow progress of your manuscript at the following link (login required):
https://susy.mdpi.com/user/manuscripts/review_info/dcfca314eaea6ce93b215d81f5234003

The following points were confirmed during submission:
1. Sustainability is an open access journal with publishing fees of 2400 CHF for an accepted paper (see https://www.mdpi.com/about/apc/ for details). This manuscript, if accepted, will be published under an open access Creative Commons CC BY license (https://creativecommons.org/licenses/by/4.0/), and I agree to pay the Article Processing Charges as described on the journal webpage (https://www.mdpi.com/journal/sustainability/apc). See https://www.mdpi.com/about/openaccess for more information about open access publishing.

Please note that you may be entitled to a discount if you have previously received a discount code, if your institute is participating in the MDPI Institutional Open Access Program (IOAP)(https://www.mdpi.com/about/ioap), or if a society you are a member of is part of our affiliation program(https://www.mdpi.com/societies_partnership). If you have been granted any other special discounts for your submission, please contact the Sustainability editorial office.

2. I understand that:

a. If previously published material is reproduced in my manuscript, I will provide proof that I have obtained the necessary copyright permission. (Please refer to the Rights & Permissions website: https://www.mdpi.com/authors/rights).

b. My manuscript is submitted on the understanding that it has not been published in or submitted to another peer-reviewed journal. Exceptions to this rule are papers containing material disclosed at conferences. I confirm that I will inform the journal editorial office if this is the case for my manuscript. I confirm that all authors are familiar with and agree with

submission of the contents of the manuscript. The journal editorial office reserves the right to contact all authors to confirm this in case of doubt. I will provide email addresses for all authors and an institutional e-mail address for at least one of the co-authors, and specify the name, address and e-mail for invoicing purposes.

If you have any questions, please do not hesitate to contact the Sustainability editorial office at sustainability@mdpi.com

Kind regards,
Sustainability Editorial Office
Grosspeteranlage 5, 4052 Basel, Switzerland
E-Mail: sustainability@mdpi.com
Tel. +41 61 683 77 34
Fax: +41 61 302 89 18

*** This is an automatically generated email ***

C.4 BioSat 2024 Conference

C.5 ICAS 2024 Conference

Abstracts: ICAS2024

Optimized Financial Risk Prediction in Enterprises Using Attention-Enhanced Deep Learning Models

Yingying Song, Monchaya Chiangpradit, Tossapol Phoophiwfa and Piyapatr Busababodhin*

Department of Mathematics, Faculty of Science, Mahasarakham University, Maha Sarakham 44150, Thailand
*Corresponding Author Email: piyapatr.b@msu.ac.th

Abstract

In the context of increasingly fierce global competition and continuous technological innovation transformation, the financial risks faced by enterprises are becoming more complex and variable. Traditional financial risk prediction methods show obvious limitations in dealing with these complex data. To address this challenge, this article proposed a deep learning model based on attention mechanism enhancement, which aimed to optimize financial risk prediction, based on 12 quarters of time series financial indicator data from listed companies T-5 to T-3. This method combines convolutional neural network (CNN), bidirectional long short-term memory network (BiLSTM), and attention mechanism, and constructed a CNN-BiLSTM-Attention comprehensive model. By comparing with five other models (CNN, BiLSTM, CNN-BiLSTM, CNN-Attention, BiLSTM-Attention), the research results show that the comprehensive model CNN-BiLSTM-Attention had the highest prediction accuracy, reaching 0.994. In addition, after introducing attention mechanism, the accuracy of CNN increased from 0.974 to 0.991, and the accuracy of BiLSTM increased from 0.985 to 0.989, further verifying the significant improvement of model performance by attention mechanism. This article provided enterprises with more efficient and accurate predictive tools for dealing with complex financial risks.

Keywords: Financial risk, Deep learning, Convolutional neural network, Bidirectional long short-term memory, Attention mechanism

Oral Presentation Schedule (Cont.)

Thursday 24 October 2024

TIME	Doi Sutep 1	Doi Sutep 2	Doi Nua
Session Chairperson	Akkaranan Pongsathornwiwat	Pramote Luenam	
15.00 - 16.45	**Topic:** Asymptotic Normality of Method of Moments Estimators for Birnbaum-Saunders Distribution in Flood Risk Assessment: A Case Study of Northeastern Thailand **Authors:** *Tossapol Phoophiwfa, Sujitta Suraphee, Andrei Volodin and Piyapatr Busababodhin (page 16)*	**Topic:** Feature Scaling and Pre-processing Techniques for Machine Learning Analysis of Durian Data in Eastern Thailand **Authors:** *Pattharaporn Thongnun, Jiratchaya Chomjinda and Janjira Piladaeng (page 22)*	**Special Session:** Longitudinal and panel data analysis in regression framework การวิเคราะห์ข้อมูลระยะยาวและพาเนลในกรอบการวิเคราะห์การถดถอย **Asst.Prof.Dr.Arnond Sakworawich**
	Topic: The Generalized Extreme Value Distribution for Inter-Amount Time Data in South Korea Using the Reciprocal Transformation Method **Authors:** *Thanawan Prahadchai and Sanghoo Yoon (page 17)*	**Topic:** Optimized Financial Risk Prediction in Enterprises Using Attention-Enhanced Deep Learning Models **Authors:** *Yingying Song, Monchaya Chiangpradit, Tossapol Phoophiwfa and Piyapatr Busababodhin (page 23)*	
	Topic: A New Mixed Moving Average - Extended Exponentially Weighted Moving Average Control Chart **Authors:** *Khanittha Talordphop and Saowanit Sukparungsee (page 18)*	**Topic:** A Comparative Study of Early Fusion and Multimodal Siamese Neural Network in Food Classification **Authors:** *Kanokporn Sintarasirikulchai, Akarin Phaibulpanich and Seksan Kiatsupaibul (page 24)*	
	Topic: Determining Minimum Initial Capital of Discrete-time Surplus Model in Non-Life Insurance using Simulation Approach **Authors:** *Adisak Moumeesri, Kannigar Hirunkasi and Phimradarat Isarakool (page 19)*	**Topic:** Embedding Topic Models for Soft Customer Segmentation **Authors:** *Jutiyakhun Srisoparp and Donlapark Ponnoprat (page 25)*	
	Topic: A Study on Factors Influencing Residential Satisfaction in Multi-Ethnic Urban Communities in Guangxi: A Hybrid Approach Based on SEM and ANN **Authors:** *Zai Xiaona, Sujitta Suraphee and Piyapatr Busababodhin (page 20)*	**Topic:** Toward Classifying Imbalanced Financial Datasets **Authors:** *Pornthera Aimrod, Praifa Kosasirisin, Anamai Na-Udom and Jaratsri Rungrattanaubol (page 26)*	
	Topic: Economic Order Quantity Model for Perishable under Imperfect Items and Time-Dependent Demand **Authors:** *Wirongrong Singsakul, Pimraphat Phuchamchot and Tammarat Kleebmek (page 21)*	**Topic:** Machine Learning Algorithms for Predicting ESG Scores: Evidence from the Stock Exchange of Thailand **Authors:** *Charwat Promlum and Wanyok Atisattapong (page 27)*	

พหุน ปณ ฑิโต ชีเว

REFERENCES

พหุน ปณฺ ฑิโต ชีเว

BIOGRAPHY

NAME	Song Yingying
DATE OF BIRTH	1986.7.29
PLACE OF BIRTH	Hubei
ADDRESS	Poly Tongji Mansion, No. 5 Tongji West Road, Chancheng District, Foshan City, Guangdong Province
POSITION	Foshan
PLACE OF WORK	Guangzhou Institute of Science and Technology
EDUCATION	2005 High School in Yicheng Second Senior High School, China 2009 Bachelor of Science in Information Management and Information Systems, South China University of Technology, China 2012 Master of Management in Enterprise Management, South China University of Technology, China 2025 Doctor of Philosophy in Statistical Management Science,Mahasarakham University,Thailand