Automatic Parameter-Optimized XGBoost for Risk Early Warning Algorithm Application in Securities

*Yiyi Zhang**

Wuhan Technical University, China
20180021@wtc.edu.cn

Abstract

The stability of the securities market is crucial to a nation's economy, and accurately predicting stock price crashes helps regulators implement protective measures in advance. Existing research has demonstrated that machine learning methods can accurately predict securities risks. Among them, the XGBoost algorithm, due to its characteristics of integrated learning, can combine the prediction results of multiple gradient boosting decision trees, thereby reducing the probability of false positives of the model. Therefore, this paper proposes a securities risk prediction model that integrates XGBoost with an evolutionary algorithm to enhance the accuracy and stability of stock crash prediction. To address the varying adaptability of different financial indicators in the model, an evolutionary algorithm is introduced to optimize XGBoost's parameter configuration. Additionally, the classification task is transformed into a regression task, combined with a dynamic threshold-setting mechanism to mitigate data imbalance issues. Experimental results demonstrate that the proposed method outperforms baseline models across multiple datasets and maintains strong stability under different training-test set ratios, achieving an average performance improvement of 15.52% to 25.79% compared to state-of-the-art (SOTA) methods.

Keywords: Securities risk early warning, Machine learning, Evolution algorithm, Automatic parameter configuration, XGBoost

1 Introduction

The securities market, an essential component of a nation's economy, plays a crucial role in predicting stock price crashes for market regulators [1]. Accurate forecasting helps regulators formulate effective protective policies in advance, mitigating the losses caused by stock market crashes and ensuring economic stability [2]. Understanding how systemic risk emerges and designing policies to curb such risks is an urgent task [3].

However, evaluating securities risk involves designing multiple features, among which financial indicators have been proven to significantly impact security prices [4-5]. Different financial indicators may contain information of varying importance; therefore, machine learning methods have been reasonably demonstrated to be suitable for securities risk prediction tasks, as they can integrate information from multiple aspects to establish a mapping from input features to outcomes [6-7]. Among various machine learning methods, XGBoost has been widely applied in high-dimensional financial data analysis due to its strong feature selection capabilities and ability to handle nonlinear relationships [8-10]. Based on gradient boosting decision trees (GBDT), XGBoost continuously optimizes model weights through iterative training, effectively capturing complex relationships among financial indicators and improving the accuracy and robustness of securities risk predictions.

However, for financial indicators across different domains, their adaptability within the XGBoost model may vary. Certain indicators might be more sensitive to specific model parameters, making it challenging to find a universal parameter configuration. Therefore, this paper attempts to leverage evolutionary algorithms to iteratively search for optimal parameters for subsets of financial indicators, effectively mitigating issues related to model generalization [11].

Additionally, existing machine learning methods often treat securities risk prediction as a binary classification task. In most available datasets, negative samples significantly outnumber positive samples, leading to class imbalance. To address this issue, this paper transforms the classification task into a regression task using an evolutionary algorithm to configure threshold parameters [12]. This parameter is then used to determine the model's prediction results, effectively alleviating the data imbalance problem. Specifically, the contributions of this paper are as follows:

- We propose a securities risk prediction model that integrates XGBoost with evolutionary algorithm-based parameter optimization, which outperforms other baseline methods on average.
- We experimentally determine the optimal configuration for this model, achieving the best performance when the crossover probability is set to 0.9 and the mutation probability to 0.01.
- By introducing a threshold-setting mechanism, we alleviate the issue of data imbalance.

The remainder of this paper is structured as follows: Section 2 reviews related work; Section 3 presents the proposed method; Section 4 describes the experimental

*Corresponding Author: Yiyi Zhang; Email: 20180021@wtc.edu.cn
DOI: https://doi.org/10.70003/160792642026052703012

setup; Section 5 discusses the results and analysis; and Section 6 concludes the paper and describes the future work.

2 Related Work

In this section, we will introduce the application of artificial intelligence technology in Securities Risk Early Warning and the relevant research on using evolutionary algorithms for parameter configuration.

2.1 Application of Artificial Intelligence Technology in Securities Risk Early Warning

Researchers usually regard Securities Risk Early Warning as a binary classification task and use methods such as machine learning and deep learning to predict it. Zhang et al. [13] utilized the Markov chain and the neural network-long short-term memory (LSTM) model to conduct risk identification on China's financial risk data. Koyuncugil et al. [14] utilized the Chi-squared Automatic Interaction Detection (CHAID) algorithm to develop an Early Warning System (EWS) model for financial risk detection based on data mining. They also verified this method using the financial data in 2007 of 7,853 small and medium-sized enterprises under the Central Bank of Turkey, demonstrating that it has relatively accurate predictive capabilities. Liu et al. [15] used the Rough Set Theory (RST) for feature selection, constructed a financial risk early-warning model based on the Back Propagation Neural Network (BPNN), and optimized it by training with historical data. Through cross-validation analysis and comparison with traditional methods, they alleviated the problem that listed companies currently face, namely, the difficulty of traditional methods in predicting financial risks timely and accurately due to the rapid market changes. Tong et al. [16] established a financial early warning system in big data by using the decision tree algorithm, which solved the problems existing in previous financial risk early warnings, such as numerous assumptions, limited data utilization, and the inability to track the fluctuation and change trends of financial indicators. Tan et al. [17] selected the daily frequency data of 20 indicators from China's money market, stock market, bond market, foreign exchange market, etc. They constructed the Index of China's Financial Stability (ICFS) by using the Dynamic Weighting Method based on the time-varying correlation coefficient, classified it through the Markov Regime Switching Model, put the basic indicators and categorical variables into the XGBoost model, and interpreted it through an interpretable framework, thus solving the problem of how to effectively provide early warnings of China's financial risks. Deng et al. [1] proposed a combination of XGBoost, NSGA-II, and SHAP. Using financial indicators as predictive variables, they classified samples with XGBoost, optimized hyperparameters with NSGA-II, and explained the outputs and calculated the importance of indicators with SHAP, thus solving the problem of the difficulty in effectively preventing the risk of individual stock price crashes in China's securities market. Wang et al. [2] proposed the XGBoost-LSTM-A model. First, they used XGBoost to select important features from the credit risk assessment system. Then, they employed the LSTM network to capture the dynamic nonlinear relationship between the selected indicators and the credit risk value in the time series and integrated the attention mechanism to screen out key information. In this way, they solved the problem of credit risk assessment for enterprises in the supply chain. Although the above methods predict securities risks by combining numerous features, they do not consider the problem of insufficient generalization ability of method parameters that may be brought about by different types of securities. There may be a situation where the parameter setting of a certain model performs well in one type of security but poorly in other types of securities. In this paper, by combining the evolutionary algorithm and setting a fitness function, the model can automatically configure the parameter configuration suitable for this type of security. At the same time, this binary classification task is also transformed into a probability task, so as to complete the prediction work more precisely.

2.2 Evolutionary Algorithms for Parameter Configuration

Using evolutionary algorithms to complete the hyperparameter configuration of methods, such as machine learning and the parameter setting of models, has been widely applied in the field of artificial intelligence. For instance, Zou et al. [18] used the genetic algorithm to adaptively adjust the hyperparameters and method parameters of XGBoost, enabling the model to have strong generalization ability in the tracking link recovery tasks of different requirement-code pairs. Xu et al. [19] utilized the improved NSGA-II based on reinforcement learning to determine a set of Pareto solutions for the configuration problem of an independent wind/PV/hydrogen production system, aiming to simultaneously minimize three objectives: the levelized cost of energy, the loss of power supply possibility, and the power abandonment rate. Nguyen et al. [20] combined the NSGA-II with the Artificial Neural Network (ANN) to optimize the process parameters for turning AISI 4340 alloy steel. Under the constraints of cutting parameters, they aimed to find the Pareto optimal solutions that simultaneously minimize the average surface roughness and cutting force, and they obtained the recommended value ranges for each parameter. Karaman et al. [21] integrated the Artificial Bee Colony (ABC) algorithm into the YOLO model to optimize the hyperparameters of the YOLO-based algorithms, aiming to enhance the performance of the real-time polyp detection system based on computer-aided diagnosis. In particular, this approach significantly im-proved the mean Average Precision (mAP) and the F1 value of the Scaled-YOLOv4 algorithm. Dontu et al. [22] utilized the Decisive Red Fox (DRF) algorithm to optimally select the key structures during the attack detection of the 5G-NIDD dataset, aiming to reduce the error rate of the classifier and improve the training speed. They also combined the DBRF classification

model with the Convolutional Neural Network (CNN) to classify the data and evaluate the performance of the model under different cyberattacks. Tsai et al. [23] utilized the Simulated Annealing (SA) algorithm to find out the appropriate number of neurons for each layer of the fully connected Deep Neural Network (DNN), aiming to improve the accuracy rate when solving the optimization problem of predicting the number of bus passengers. Krishna et al. [24] combined the Ant Colony Optimization (ACO) algorithm with the XGBoost algorithm to form ACXG, which is used for the timely detection and prediction of diabetes and Parkinson's disease. The above related works can prove that evolutionary algorithms have achieved good application results in many fields such as physics, healthcare, software engineering, transportation data, and network security. Therefore, we speculate that they can also play an important role in the securities risk early warning task.

3 Methodology

In this section, we will describe the method we proposed from three aspects: feature construction, risk early warning, and parameter optimization. The overall diagram of the method is shown in Figure 1.

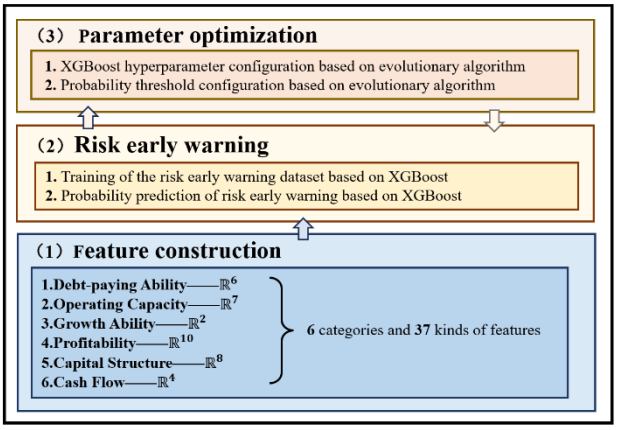

Figure 1. Overview of the proposed method

3.1 Feature Construction

We constructed 37 features in 6 categories for XGBoost training, covering debt-paying ability, operating capacity, growth ability, profitability, capital structure, and cash flow. The specific descriptions of the features are shown in Table 1. The debt-paying ability considers the company's capability to repay debts, including both short-term and long-term debts, to determine whether it has sufficient assets or cash flow to settle its obligations, which is crucial for assessing financial risks. The operating capacity examines the efficiency of the company's asset operation. For example, the turnover of accounts receivable and inventory reflects the conversion of assets into sales revenue and the control level of operating costs. The growth ability focuses on the company's development potential and trend, such as the expansion of asset scale and the degree of sustainable growth. The profitability reflects the company's level of profit acquisition, and the ability to generate profits after de-ducting costs is of vital importance. The capital structure considers the composition of debt and equity in the company's sources of funds, reflecting the financing strategy and the degree of financial risk tolerance. The cash flow focuses on the inflow and outflow of the company's cash. The efficiency of obtaining cash from core operating businesses and the ability of assets to generate cash affect the company's operation. A stable cash flow is the key to the company's survival and development.

3.2 Risk Early Warning by XGBoost

For the task of Securities Risk Early Warning, we use the XGBoost model for prediction. XGBoost can optimize the prediction performance through ensemble learning and regularization techniques. The overall overview of the method is shown in Figure 2. Specifically, let the training set be $D = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i represents the features described in Section 3.1, and $y_i \in \{0,1\}$ represents the risk label (1 indicates high-risk securities). The objective function of the model is defined as:

$$L(\phi) = \sum_{i=1}^{n} l\left(y_i, \widehat{y_i}\right) + \sum_{k=1}^{K} \Omega\left(f_k\right) \tag{1}$$

where $l(y_i, (\ \widehat{y_i}\ ))$ is the Logistic loss function:

$$l = -y_i ln\sigma\left(\widehat{y_i}\right) - \left(1 - y_i\right) \ln\left(1 - \sigma\left(\widehat{y_i}\right)\right) \tag{2}$$

is the sigmoid function, $\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i)$ is the cumulative predicted value, and K is the total number of trees. The regularization term $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$ controls the complexity of the model, where T is the number of leaf nodes, w is the leaf weight, and γ, λ are the penalty coefficients.

The tree structure optimization uses the second-order Taylor expansion to approximate the loss function. For the t-th iteration, the objective function is approximated as:

$$L^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right) \right] + \Omega\left(f_t\right) \tag{3}$$

where $g_i = \frac{\partial}{\partial \hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \frac{\partial^2}{\partial (\hat{y}^{(t-1)})^2} l(y_i, \hat{y}^{(t-1)})$ are the first-order and second-order gradients respectively. After mapping the samples to the leaf nodes $I_j = \{i|q(x_i) = j\}$, the objective function can be rewritten as:

$$L^{(t)} = \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \lambda T \tag{4}$$

The optimal leaf weight is:

$$w_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (5)$$

, and the corresponding split gain is:

$$G_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \quad (6)$$

where I_L and I_R are the sample sets of the left and right child nodes after splitting. Feature selection is completed by traversing all features and split points to maximize G_{split}.

In addition, to optimize the computational complexity, we adopt a sparse-aware algorithm to handle missing values and define the default direction as:

$$v^* = \arg max_{v \in \{L,R\}} G_{split}(v) \quad (7)$$

For the missing feature $x_{i,j}$, it is automatically assigned to the branch with the maximum gain. Parallel computing is achieved through feature-level parallelization. After sorting the features, they are split among multiple threads to calculate the split gain. In combination with the histogram algorithm, continuous features are discretized into b quantile points (b=256), reducing the computational complexity to O(mb).

Table 1. Feature details

Category	Feature name	Description
Debt-paying ability	Current Ratio	Measures a company's ability to pay short-term obligations using current assets.
	Quick Ratio	Measures short-term liquidity by excluding inventory from current assets.
	Debt to Asset Ratio	Shows the proportion of a company's assets financed by debt.
	Equity Multiplier	Indicates financial leverage by comparing total assets to shareholders' equity.
	Debt to Equity Ratio	Compares total debt to shareholders' equity.
	Long-Term Debt to Asset Ratio	Measures long-term debt as a percentage of total assets.
Operating capacity	Receivables Turnover Ratio	Measures how efficiently a company collects its receivables.
	Inventory Turnover Ratio	Shows how quickly a company sells and replaces inventory.
	Operating Cycle	The time taken to convert inventory into cash through sales.
	Current Assets Turnover Ratio	Evaluates how efficiently current assets generate sales.
	Fixed Assets Turnover Ratio	Measures how effectively fixed assets generate sales.
	Capital Intensity Rate	Indicates how much capital is needed to generate sales.
	Total Assets Turnover Ratio	Measures overall asset efficiency in generating sales.
Growth ability	Total Assets Growth Rate	Measures the growth rate of a company's total assets.
	Sustainable Growth Rate	The maximum growth rate is achievable without external equity financing.
Profitability	Return on Assets (ROA)	Measures profit generated per dollar of assets.
	Return on Total Assets (ROTA)	Similar to ROA, it evaluates profit relative to total assets.
	Return on Equity (ROE)	Measures profit generated per dollar of shareholders' equity.
	Gross Profit Margin	Shows profit after deducting the cost of goods sold (COGS).
	Operating Expense Ratio	Measures operating expenses as a percentage of sales.
	Operating Profit Margin	Evaluates profitability from core operations.
	Net Profit Margin	Measures net income as a percentage of sales.
	Expense to Sales Ratio	Compares total expenses to sales revenue.
	Administration Expense Ratio	Measures administrative costs relative to sales.
	Financial Expense Ratio	Shows financial costs (e.g., interest) as a percentage of sales.
Capital structure	Current Assets to Total Assets Ratio	Measures liquidity by comparing current assets to total assets.
	Cash to Assets Ratio	Shows cash holdings as a percentage of total assets.
	Working Capital to Total Assets Ratio	Evaluates short-term financial health by comparing working capital to total assets.
	Fixed Assets Ratio	Measures the proportion of fixed assets in total assets.
	Shareholder Equity Ratio	Shows the proportion of assets financed by shareholders' equity.
	Current Liability Ratio	Measures short-term debt as a percentage of total liabilities.
	Non-Current Liability Ratio	Measures long-term debt as a percentage of total liabilities.
	Operating Profit Percentage	Shows operating profit as a percentage of total revenue.
Cash flow	Operating Cash Flow to Sales Ratio	Measures cash generated from operations per dollar of sales.
	Net Operating Cash Flow to Sales Ratio	Similar to above but focuses on net operating cash flow.
	Cash Return on Total Assets	Evaluates how efficiently assets generate cash flow.
	Cash Operating Index	Compare operating cash flow to net income (earnings quality).

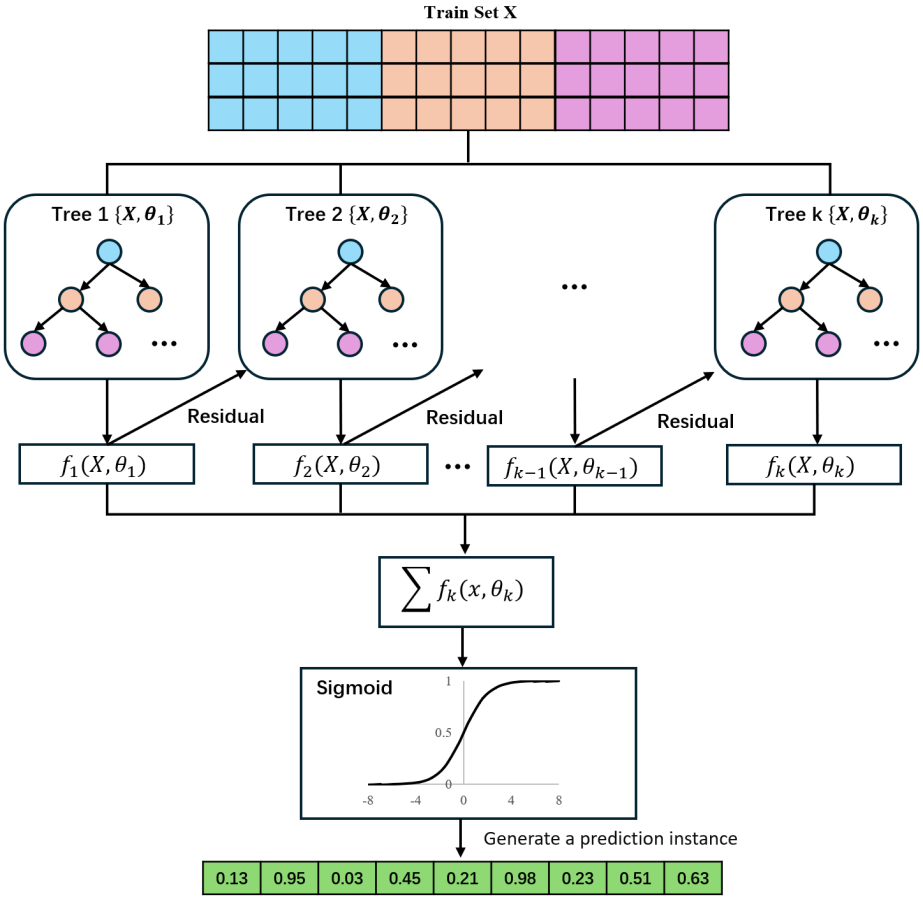

Figure 2. Overview of XGBoost

3.3 Parameter Optimization

The parameter optimization of the securities risk early warning model can be formalized as a high-dimensional mixed space search problem. As shown in Figure 2, the parameter vector is defined as $\theta = (\eta, d, w, s, \lambda, \theta_{th})$, where $\eta \in (0,1)$ is the learning rate, $d \in Z^+$ is the maximum depth of the tree, $w \in R^+$ is the sum of the sample weights of the minimum leaf nodes, $s \in (0,1)$ is the row sampling ratio, $\lambda \in R^+$ is the L2 regularization coefficient, and $\theta_{th} \in (0,1)$ is the risk determination threshold.

The optimization objective is defined as maximizing the F1 score and controlling the model complexity, and the fitness function is constructed as follows:

$$F(\theta) = \frac{1}{K}\sum_{k=1}^{K} F1^{(k)}(\theta) - \alpha \cdot \left(\frac{1}{n}\sum_{i=1}^{n} T_i + \frac{\lambda}{2}\|w\|_2^2\right) \quad (8)$$

where $F1^{(k)} = 2 \cdot \frac{P^{(k)} R^{(k)}}{P^{(k)} + R^{(k)}}$ is the F1 score of the k-th fold cross-validation (P is the precision rate and R is the recall rate), T_i is the number of leaf nodes of the i-th tree, w is the leaf weight vector, and α is the complexity penalty coefficient. A hybrid encoding evolutionary algorithm is adopted to solve this problem. Its core process includes population initialization, fitness evaluation, selection, crossover, mutation, and the elite retention mechanism.

In the population initialization stage, N individuals will be generated. The parameters of each individual θ_j follow a mixed distribution: the continuous parameters η, s, λ, θ_{th} are sampled from the uniform distributions U(0.01,0.3), U(0.6,1.0), U(0.1,10) and U(0.3,0.7) respectively. The discrete parameters d and w are randomly selected from the integer sets {3, 4, …, 10} and {1, 2, …, 10} respectively. A hybrid encoding strategy is adopted for gene expression, where continuous parameters are represented as floating-point numbers and discrete parameters are encoded as integers.

For the fitness evaluation, it is achieved through the rolling time window verification. Given a time window $W_t = [t - \Delta, t]$, the training set D_{train} contains the samples within W_t, and the validation set D_{val} consists of the samples of W_{t+1}. For an individual θ_j, an XGBoost model is trained, and the leaf weight $w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$ is calculated, where $g_i = \sigma((\widehat{y_i})) - y_i$ is the first-order gradient and $h_i = \sigma((\widehat{y_i}))(1 - \sigma(\widehat{y_i}))$ is the second-order gradient. The predicted value $\sigma(\widehat{y_i}) = \frac{1}{1 + e^{-(\widehat{y_i})}}$ is binarized by the

threshold θ_{th} as $\left(\widehat{y_i^{class}}\right) = I\left(\sigma\left(\left(\widehat{y_i}\right)\right) \geq \theta_{th}\right)$. Then, the F1 score and the regularization term are calculated.

For the selection operation, we adopt a tournament mechanism based on fitness ranking. Randomly select M individuals from the population to form a competition group. After ranking them according to the fitness $F(\theta_j)$, the top T individuals are retained and enter the mating pool. To maintain the diversity of the population, an adaptive selection probability is introduced:

$$P_{select}\left(\theta_j\right) = \frac{\exp\left(\frac{F\left(\theta_j\right)}{\tau}\right)}{\sum_{i=1}^{M}\exp\left(\frac{F\left(\theta_j\right)}{\tau}\right)} \quad (9)$$

where the temperature coefficient τ decays with the number of iterations t, following $\tau(t) = \tau_0 \cdot e^{-kt}$. In the initial stage, individuals with low fitness are allowed to survive with a certain probability, and in the later stage, the focus gradually shifts to individuals with high fitness.

For the crossover and mutation stages, we adopt a differentiated strategy to deal with continuous and discrete parameters. For continuous parameters, Simulated Binary Crossover (SBX) is used. For the parent individuals θ_p and θ_q, the i-th continuous parameter of the offspring individual θ_c satisfies:

$$\theta_{mut}\left(i\right) = \theta\left(i\right) + \delta \cdot \left(\mathrm{u}_{\mathrm{i}}^{(U)} - \mathrm{u}_{\mathrm{i}}^{(L)}\right) \quad (11)$$

where β follows a polynomial distribution, $\beta = (2u)^{(1/\eta c+1)(\text{if } u \leq 0.5)}$ or $\beta = (1/2(1-u))^{(1/\eta c+1)(\text{if } u > 0.5)}$, $u \sim U(0,1)$, and η_c controls the crossover intensity. For the discrete parameters d and w, uniform crossover is applied, and the parameter values of the parent individuals are exchanged with the probability p_{swap}. For the mutation operation, a polynomial perturbation is applied to the continuous parameters:

Where δ is generated from a polynomial distribution, $\delta = \min(\max(2u-1, -1), 1)$, $u \sim U(0,1)$, and $u_i^{(U)}$ and $u_i^{(L)}$ are the upper and lower bounds of the parameters. For the threshold parameter θ_{th}, directional mutation is adopted: if the recall rate R of the validation set is $R < R_{min}$, then the threshold is adjusted as $\theta_{th} := \theta_{th} - \gamma \cdot \Delta R (\Delta R = R_{min} - R$, γ is the learning rate) to reduce the risk of false negatives.

Finally, we will utilize the elite retention mechanism to preserve the top E individuals with the highest fitness in each generation of iteration and directly let them enter the next generation, so as to avoid the loss of high-quality genes. The convergence condition is set as the standard deviation of the population fitness $\sigma_F < \epsilon$ ($\epsilon = 1\times10^{-5}$) or reaching the maximum number of iterations G_{max}.

4 Experimental Setup

4.1 Datasets & Baselines

Based on the research of Tan et al. [17] and Deng et al. [1], we obtained relevant securities trading data from 2020 to 2025 from five databases, namely CSMAR, RESSET, WIND, the Central Depository & Clearing Co., Ltd. (CCDC), and the China Foreign Exchange Trade System (CCFETS), as the dataset for the experiments in this paper. In addition, we have also selected four methods, namely CHAID [14], RST-BPNN [15], XGBoost-NSGA-II [1], and XGBoost-Markov [17], as the baseline methods for this experiment.

4.2 Metrics

We will evaluate the method using Precision, Recall, and F1-score. Precision measures the proportion of true positive samples among all the samples predicted as positive. Its definition is:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (12)$$

Recall represents the proportion of true positive samples among all the actual positive samples that are correctly predicted as positive. Its definition is:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (13)$$

The F1-score is a weighted average of Precision and Recall, which is used to comprehensively evaluate the performance of the model. Its calculation formula is:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (14)$$

4.3 Hardware Configuration and Experimental Environment Setup

The hardware configuration used in this study is as follows: The graphics card is NVIDIA GeForce RTX 4070Ti Super with 16GB of video memory, which can accelerate data processing and machine learning model training. The processor is Intel Core i7-14700KF, and the 32GB of memory ensures the smooth operation of experimental programs.

The experimental environment is built based on the Python language. The XGBoost model is built with the help of the scikit-learn library, and the evolutionary algorithm is implemented through the deap library. Such hardware configuration and software environment lay the foundation for subsequent experiments.

4.4 Research Questions

In order to explore the optimal configuration of the method proposed in this paper, the effectiveness of the

strategy, the performance of the method, and the generalization ability of the method, we have put forward the following three research questions:

RQ1: What is the optimal configuration of the model?

RQ2: Can the performance of the model be better than that of the baseline methods?

RQ3: How about the generalization ability of the model?

To answer Research Question 1 (RQ1), we set up the following experiment. The data is divided into the training set, the validation set, and the test set in the ratio of 8:1:1. The experiment is carried out using the five datasets described in Section 4.1. We verified the applicability of the current mainstream evolutionary algorithms, including the Genetic Algorithm and the NSGA-II algorithm. In addition, for each algorithm, we also set up five configuration groups in sequence according to the research of Zou et al. [18]:

1. Without using the evolutionary algorithm.
2. Using the evolutionary algorithm with a crossover probability of 0.6 and a mutation probability of 0.1.
3. Using the evolutionary algorithm with a crossover probability of 0.9 and a mutation probability of 0.1.
4. Using the evolutionary algorithm with a crossover probability of 0.6 and a mutation probability of 0.01.
5. Using the evolutionary algorithm with a crossover probability of 0.9 and a mutation probability of 0.01.

To ensure the reliability of the experimental results, each group of experiments is repeated 50 times, and the average value of the results of these 50 experiments is taken as the final output. When evaluating the performance of the model, F1-score is selected as the evaluation metric. Through these metrics, the performance of the model under different configurations is measured to explore the optimal configuration of the model.

To answer the research question RQ2, we also divide the dataset in the ratio of 8:1:1. We conduct 50 repeated experiments on the model proposed in this paper and the baseline methods described in Section 4.1, and evaluate the performance of the models through Precision, Recall, and F1-score.

To answer the research question RQ3, we will evaluate the model proposed in this paper and the state-of-the-art (SOTA) methods according to a total of 9 division ratios of the training set and the test set, ranging from 9:1 to 1:9. The performance of the models will be evaluated solely through the F1-score.

5 Results

5.1 Answer RQ1: What Is the Optimal Configuration of The Model?

The experimental results are shown in Table 2. When comparing the use of evolutionary algorithms with the non-use of evolutionary algorithms, the average F1 score of the experimental group without evolutionary algorithms is 0.9968, while the average F1 score of the best configuration of the Genetic Algorithm is 0.9682. The average F1 score of the best configuration of the NSGA - II algorithm is 0.9551. This indicates that the average F1 score of the configuration groups using evolutionary algorithms is higher than that of the experimental group without evolutionary algorithms, suggesting that evolutionary algorithms greatly enhance the model's search ability through operations such as crossover and mutation.

Meanwhile, when further comparing the Genetic Algorithm and the NSGA - II algorithm, under the same configuration, the performance of the Genetic Algorithm is better. For example, in the configuration with a crossover probability of 0.9 and a mutation probability of 0.01, the average F1 score of the Genetic Algorithm is 0.9682, while that of the NSGA - II is 0.9551; when the crossover probability is 0.9 and the mutation probability is 0.1, the average F1 score of the Genetic Algorithm is 0.9622, and that of the NSGA - II is 0.9538, and so on. This is mainly because the Genetic Algorithm focuses on single-objective optimization and fully strengthens the search for the optimal solution through selection, crossover, and mutation operations. In contrast, as a multi-objective algorithm, NSGA - II needs to balance the diversity and convergence of the solution set through nondominated sorting and crowding distance in a single-objective task, which disperses the search pressure.

Finally, from the perspective of configuration, among the numerous configurations of the Genetic Algorithm, the configuration with a crossover probability of 0.9 and a mutation probability of 0.01 performs the most prominently. Its average F1 score is 0.9682, which is higher than other configurations. This shows that a high crossover probability promotes the full exchange of population genes and enhances the global exploration ability. A low mutation probability reduces the damage of random disturbances to excellent genes and improves the local development efficiency, thus effectively converging to the optimal solution while maintaining population diversity.

5.2 Answer RQ2: Can the Performance of The Model Be Better Than That of The Baseline Methods?

The experimental results are shown in Table 3. The proposed method in this paper outperforms the baseline methods in terms of Precision, Recall, and F1-score across all datasets, demonstrating stronger stability and predictive capability. For instance, on the CSMAR dataset, the Precision, Recall, and F1-score of the proposed method reach 0.9316, 0.9488, and 0.9401, respectively, while the relatively better-performing XGBoost-NSGA-II method achieves scores of 0.9122, 0.9247, and 0.9184 on the corresponding metrics. This indicates that the proposed method improves both precision and recall. This advantage is also validated on other datasets. Particularly on the CCFETS dataset, the F1-score of the proposed method reaches 0.9664, showing a significant improvement over XGBoost-NSGA-II's 0.9366, further proving the performance superiority of the model.

Among them, the machine learning-based methods are superior to other methods. We speculate that this is because machine learning methods can capture the complex

nonlinear relationships in the data, and their performance is inherently better than that of traditional statistical methods. For example, CHAID relies on rule-based data partitioning. Although RST-BPNN combines rough set theory and neural networks, it still lacks the adaptability of ensemble methods like XGBoost. As for XGBoost-NSGA-II and XGBoost-Markov, the advantage of the method proposed in this paper lies in the introduction of the Genetic Algorithm (GA) to achieve automatic parameter optimization, thus dynamically adapting to the feature differences of different datasets. Through iterative evolution, the optimal configuration is obtained, ensuring the robustness of the model on various datasets, which highlights the role of the Genetic Algorithm in dealing with data heterogeneity.

Table 2. Comparison of the F1 performance of the model under different configurations of evolutionary algorithms (The configuration is expressed as a-b, which means the crossover probability is a and the mutation probability is b.)

Algorithm	Configuration	CSMAR	RESSET	WIND	CCDC	CCFETS
-	Without evolution algorithm	0.8763	0.9015	0.8737	0.8873	0.8954
Genetic algorithm	0.6-0.1	0.9356	0.9531	0.9567	0.9501	0.9603
	0.9-0.1	0.9366	0.9538	0.9577	0.9512	0.9623
	0.6-0.01	0.9356	0.9531	0.9572	0.9505	0.9610
	0.9-0.01	0.9401	0.9577	0.9589	0.9525	0.9664
NSGA-II	0.6-0.1	0.9424	0.9601	0.9698	0.9669	0.9753
	0.9-0.1	0.9456	0.9634	0.9722	0.9701	0.9795
	0.6-0.01	0.9433	0.9621	0.9702	0.9686	0.9779
	0.9-0.01	0.9487	0.9653	0.9752	0.9732	0.9824

Table 3. Performance of Precision, Recall, and F1-score of the Method Proposed in this paper and the baselines on all datasets

Method	Metric	CSMAR	RESSET	WIND	CCDC	CCFETS
CHAID	Precision	0.7155	0.7298	0.7311	0.7289	0.7615
	Recall	0.7378	0.7439	0.7493	0.7355	0.7693
	F1	0.7265	0.7368	0.7401	0.7322	0.7654
RST-BPNN	Precision	0.8088	0.8154	0.8210	0.8133	0.8298
	Recall	0.8221	0.8338	0.8323	0.8197	0.8417
	F1	0.8154	0.8245	0.8266	0.8165	0.8357
XGBoost-NSGA-II	Precision	0.9122	0.9198	0.9214	0.9115	0.9281
	Recall	0.9247	0.9299	0.9355	0.9309	0.9453
	F1	0.9184	0.9248	0.9284	0.9211	0.9366
XGBoost-Markov	Precision	0.8911	0.8955	0.9084	0.9022	0.9141
	Recall	0.9113	0.9155	0.9233	0.9155	0.9227
	F1	0.9011	0.9054	0.9158	0.9088	0.9184
Ours	Precision	0.9316	0.9555	0.9585	0.9505	0.9655
	Recall	0.9488	0.9599	0.9593	0.9545	0.9673
	F1	0.9401	0.9577	0.9589	0.9525	0.9664

Table 4. The F1 performance of the method proposed in this paper and the SOTA method XGBoost-NGSA-II under different training set-test set ratios (Ratio = a:b indicates that the proportion of the training set is a/10 and the proportion of the test set is b/10)

Method	Ratio	CSMAR	RESSET	WIND	CCDC	CCFETS
XGBoost-NSGA-II	9:1	0.9541	0.9604	0.9645	0.9588	0.9647
	8:2	0.9184	0.9248	0.9284	0.9211	0.9366
	7:3	0.8354	0.8421	0.8359	0.8411	0.8255
	6:4	0.6987	0.6988	0.7184	0.6884	0.6845
	5:5	0.5484	0.5894	0.5448	0.5684	0.5487
	4:6	0.4554	0.4871	0.4894	0.4447	0.4879
	3:7	0.3878	0.3546	0.3841	0.3548	0.4084
	2:8	0.3108	0.3281	0.2818	0.3184	0.3548
	1:9	0.1958	0.2481	0.1894	0.2154	0.2849

	9:1	0.9877	0.9954	0.9964	0.9910	0.9996
	8:2	0.9401	0.9577	0.9589	0.9525	0.9664
	7:3	0.9084	0.9284	0.9269	0.9394	0.9184
	6:4	0.8784	0.8448	0.8619	0.8646	0.8456
Ours	5:5	0.8088	0.7818	0.7995	0.8078	0.7849
	4:6	0.7084	0.6897	0.6889	0.7108	0.6589
	3:7	0.5894	0.5464	0.5974	0.6054	0.5154
	2:8	0.4849	0.4561	0.4651	0.5058	0.4056
	1:9	0.2494	0.2881	0.3089	0.3118	0.2548

5.3 Answer RQ3: How About the Generalization Ability of The Model?

The experimental results are shown in Table 4. The method proposed in this paper demonstrates stable performance advantages under different training set-test set ratios. Compared with the SOTA method XGBoost-NSGA-II, our method has an average performance improvement of 15.52% to 25.79% on five datasets. Different from XGBoost-NSGA-II, which directly outputs classification results, we speculate that the probabilistic modeling, combined with the dynamic threshold strategy optimized by the genetic algorithm, enables the model to better adapt to changes in the data distribution. The parameter optimization framework of the standard genetic algorithm can more effectively find the optimal parameter combination compared with the multi-objective optimization of NSGA-II, thereby enhancing the generalization ability of the model. At the same time, as the proportion of training data decreases, our method shows a more robust performance retention ability. This advantage is particularly evident in situations of extreme data scarcity, indicating that probabilistic modeling can more effectively uncover statistical patterns in limited samples. Especially in extreme scenarios with class imbalance, the dynamic threshold mechanism provides a reliable guarantee for the model's performance.

6 Conclusion & Future Work

This paper proposes a securities risk prediction model that integrates XGBoost with evolutionary algorithm-based parameter optimization to address the problem of stock price crash prediction in the securities market. While traditional XGBoost models exhibit strong feature selection capabilities when handling high-dimensional financial data, the adaptability of different financial indicators within the model varies, making parameter optimization a critical challenge. To tackle this issue, this paper introduces an evolutionary algorithm to iteratively optimize and search for data configurations that adapt to different financial indicators, thereby enhancing the model's generalization ability. Additionally, to address the data imbalance problem, this study transforms the securities risk prediction task from a traditional binary classification problem into a regression task and incorporates a dynamic threshold-setting approach to improve the model's prediction stability.

Experimental results demonstrate that the proposed method outperforms baseline methods across multiple datasets. Furthermore, the method exhibits strong stability under different training-to-test set ratios, achieving an average performance improvement of 15.52% to 25.79% compared to SOTA methods. These findings indicate that by optimizing parameter configurations through evolutionary algorithms and employing a dynamic threshold strategy, the model can better adapt to varying data distributions, thereby improving prediction accuracy and generalization ability.

Furthermore, we believe this work can be further improved in the following aspects:

Enhancing Model Interpretability: While the proposed method achieves high predictive performance, its interpretability remains a challenge. Future work will explore techniques such as Shapley Additive Explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) to provide deeper insights into the contribution of financial indicators in risk prediction.

Exploring Alternative Evolutionary Algorithms: Although the genetic algorithm has shown effectiveness in parameter optimization, other evolutionary algorithms such as Particle Swarm Optimization (PSO) and Differential Evolution (DE) could be explored to further enhance optimization efficiency and convergence speed.

Incorporating Additional Market Factor: The current model mainly considers financial indicators, but external market factors such as macroeconomic indicators, news sentiment analysis, and investor behavior may further improve prediction accuracy. Future research will focus on integrating these additional features into the model.

Real-Time Adaptation and Deployment: Implementing the model in a real-time predictive environment would be valuable for market regulators and investors. Future work will investigate adaptive learning strategies that allow the model to update dynamically as new market data becomes available.

Addressing Extreme Market Conditions: Stock market crashes often occur under extreme conditions, where data patterns may significantly deviate from normal market trends. Future research will focus on improving the model's robustness under extreme scenarios by incorporating anomaly detection techniques and adversarial training strategies.

References

[1] S. Deng, Y. Zhu, S. Duan, Z. Fu, Z. Liu, Stock Price Crash Warning in the Chinese Security Market Using a Machine Learning-Based Method and Financial Indicators, *Systems*, Vol. 10, No. 4, Article No. 108, August, 2022. https://doi.org/10.3390/systems10040108

[2] D. Wang, J. Feng, W. Zou, H. Chen, Credit risk assessment and early warning of supply chain finance based on xgboost-lstm-a model, *Proceedings of the 2023 4th International Conference on Computer Science and Management Technology*, Xi'an, Shanxi, China, 2023, pp. 444-449. https://doi.org/10.1145/3644523.3644603

[3] F. S. Mishkin, Over the cliff: From the subprime to the global financial crisis, *Journal of Economic Perspectives*, Vol. 25, No. 1, pp. 49-70, Winter, 2011. https://doi.org/10.1257/jep.25.1.49

[4] G. A. Feltham, J. A. Ohlson, Valuation and clean surplus accounting for operating and financial activities, *Contemporary accounting research*, Vol. 11, No. 2, pp. 689-731, Spring, 1995. https://doi.org/10.1111/j.1911-3846.1995.tb00462.x

[5] M. Xu, X. Liu, The Correlation between Financial Indexes and Stock Prices in ChiNext, *Statistics and Application*, Vol. 7, No. 3, pp. 281-290, June, 2018. https://doi.org/10.12677/sa.2018.73033

[6] V. Diaz, W. E. Wong, Z. Chen, Enhancing Deception Detection with Exclusive Visual Features using Deep Learning, *International Journal of Performability Engineering*, Vol. 19, No. 8, pp. 547-558, August, 2023. https://doi.org/10.23940/ijpe.23.08.p7.547558

[7] S. Kaur, N. Sinha, P. Jain, S. Koli, A. Sharma, A. Lathwal, Enhanced Image Forgery Detection using a Hybrid Approach: Integration of ELA, CNN, and XGBoost, *International Journal of Performability Engineering*, Vol. 20, No. 6, pp. 367-378, June, 2024. https://doi.org/10.23940/ijpe.24.06.p4.367378

[8] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, California, USA, 2016, pp. 785-794. https://doi.org/10.1145/2939672.2939785

[9] C. Li, Z. Wang, R. Chen, M. Yang, intCV: Automatically Inferring Correlated Variables in Interrrupt-Driven Program, *2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security*, Chiang Mai, Thailand, 2023, pp. 562-568. https://doi.org/10.1109/QRS60937.2023.00061

[10] A. Ogundiran, H. Chi, J. Yan, R. Agada, Advancing Forensic Examination of Cyber Predator Communication Through Machine Learning, *2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion*, Cambridge, United Kingdom, 2024, pp. 464-473. https://doi.org/10.1109/QRS-C63300.2024.00065

[11] N. M. Aszemi, P. D. D. Dominic, Hyperparameter optimization in convolutional neural network using genetic algorithms, *International Journal of Advanced Computer Science and Applications*, Vol. 10, No. 6, pp. 269-278, July, 2019. https://dx.doi.org/10.14569/IJACSA.2019.0100638

[12] Z. Zou, B. Wang, Y. Deng, H. Wan, Z. An, Y. Cao, XWCoDe: XGBoost with Weighted Code Dependency for Requirements-to-Code Traceability Link Recovery, *2024 IEEE International Conference on Systems, Man, and Cybernetics*, Kuching, Malaysia, 2024, pp. 2670-2675. https://doi.org/10.1109/SMC54092.2024.10830945

[13] W. Zhang, Dynamic monitoring of financial security risks: A novel China financial risk index and an early warning system, *Economics Letters*, Vol. 234, Article No. 111445, January, 2024. https://doi.org/10.1016/j.econlet.2023.111445

[14] A. S. Koyuncugil, N. Ozgulbas, Financial early warning system model and data mining application for risk detection, *Expert systems with Applications*, Vol. 39, No. 6, pp. 6238-6253, May, 2012. https://doi.org/10.1016/j.eswa.2011.12.021

[15] T. Liu, L. Yang, Financial risk early warning model for listed companies using BP neural network and rough set theory, *IEEE Access*, Vol. 12, pp. 27456-27464, February, 2024. https://doi.org/10.1109/ACCESS.2024.3367228

[16] L. Tong, G. Tong, A novel financial risk early warning strategy based on decision tree algorithm, *Scientific Programming*, Vol. 2022, No. 1, Article No. 4648427, January, 2022. https://doi.org/10.1155/2022/4648427

[17] B. Tan, Z. Gan, Y. Wu, The measurement and early warning of daily financial stability index based on XGBoost and SHAP: Evidence from China, *Expert Systems with Applications*, Vol. 227, Article No. 120375, October, 2023. https://doi.org/10.1016/j.eswa.2023.120375

[18] Z. Zou, B. Wang, X. Hu, Y. Deng, H. Wan, H. Jin, Enhancing requirements-to-code traceability with GA-XWCoDe: Integrating XGBoost, Node2Vec, and genetic algorithms for improving model performance and stability, *Journal of King Saud University-Computer and Information Sciences*, Vol. 36, No. 8, Article No. 102197, October, 2024. https://doi.org/10.1016/j.jksuci.2024.102197

[19] C. Xu, Y. Ke, Y. Li, H. Chu, Y. Wu, Data-driven configuration optimization of an off-grid wind/PV/hydrogen system based on modified NSGA-II and CRITIC-TOPSIS, *Energy Conversion and Management*, Vol. 215, Article No. 112892, July, 2020. https://doi.org/10.1016/j.enconman.2020.112892

[20] A. T. Nguyen, V. H. Nguyen, T. T. Le, N. T. Nguyen, A hybridization of machine learning and NSGA-II for multi-objective optimization of surface roughness and cutting force in ANSI 4340 alloy steel turning, *Journal of Machine Engineering*, Vol. 23, No. 1, pp. 133-153, February, 2023. https://doi.org/10.36897/jme/160172

[21] A. Karaman, D. Karaboga, I. Pacal, B. Akay, A. Basturk, U. Nalbantoglu, S. Coskun, O. Sahin, Hyper-parameter optimization of deep learning architectures using artificial bee colony (ABC) algorithm for high performance real-time automatic colorectal cancer (CRC) polyp detection, *Applied Intelligence*, Vol. 53, No. 12, pp. 15603-15620, June, 2023. https://doi.org/10.1007/s10489-022-04299-1

[22] S. Dontu, S. R. Addula, P. K. Pareek, R. Vallabhaneni, M. H. Fallah, A Feature Selection based Decisive Red Fox Algorithm with Deep Learning for Protecting Cybersecurity Network, *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems*, Hassan, India, 2024, pp. 1-7. https://doi.org/10.1109/IACIS61494.2024.10721671

[23] C. W. Tsai, C. H. Hsia, S. J. Yang, S. J. Liu, Z. Y. Fang, Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing, *Applied soft computing*, Vol. 88, Article No. 106068, March, 2020. https://doi.org/10.1016/j.asoc.2020.106068

[24] A. Y. Krishna, K. R. Kiran, N. R. Sai, A. Sharma, S. P. Praveen, J. Pandey, Ant Colony Optimized XGBoost for Early Diabetes Detection: A Hybrid Approach in Machine Learning, *Journal of Intelligent Systems & Internet of Things*, Vol. 10, No. 2, pp. 76-89, October 2023. https://doi.org/10.54216/JISIoT.100207

Biography

Yiyi Zhang works at Wuhan Technical University, Wuhan, China. His research interests mainly include but are not limited to pattern recognition, Securities risk warning, and machine learning.