Adaptive Denoising and Evolutionary Ensemble Learning for Imbalanced Financial Data Classification

Lu Yang[1*], Dong Peng[2]
[1] Management School, Henan University of Urban Construction, Pingdingshan, 467036, China
[2] Pingdingshan Polytechnic College, Pingdingshan, 467001, China
E-mail: yanglu-yl@outlook.com
[*]Corresponding author

Keywords: ADT, oversampling technique, ensemble learning, evolutionary optimization, financial risk control

Received: October 10, 2025

In financial risk-control scenarios, complex data structures and high costs of misclassification impose greater demands on classification models. To enhance model discrimination under noise interference and class imbalance, an imbalanced financial data classification model is developed by integrating adaptive denoising technology and evolutionary ensemble learning. At the data level, the model performs noise-aware weighting and minority oversampling to achieve sample balance. At the model level, a multi-objective particle swarm optimization algorithm was employed to evolve and optimize the weights of multiple sub-classifiers, forming a self-adaptive ensemble structure. Experiments conducted on three public datasets, Lending Club, Credit Card Fraud Detection, and Give Me Some Credit, showed that the model achieved F1-scores of 92.1%, 90.8%, and 88.4%, respectively. The model maintained high classification accuracy and stability under different data distributions and operating conditions. Further simulation tests demonstrate consistent convergence and strong robustness, confirming the applicability and generalization capability of the model in complex financial data environments.

Povzetek: Članek predlaga napredni model za neuravnotežene finančne podatke, ki z adaptivnim razšumljanjem in evolucijskim ansamblom zanesljivo prepoznava redke rizične primere.

1 Introduction

The financial market is an important platform for capital flow, risk pricing, and wealth redistribution. In recent years, with the increasing coverage of mobile payments, online wealth management, and digital currencies, a large amount of user behavior, transaction records, and social public opinion data have synchronously flooded into the backend of financial institutions. These financial data exhibit complex characteristics such as large volume, disorderly structure, dense noise, and extreme classes [1-2]. These massive data have had a direct impact on traditional financial data risk control systems, which is mainly based on rule verification and statistical testing in the past [3]. How to capture hidden high-risk transactions in complex environments with high noise has become a common goal for regulatory authorities and enterprises in the current financial industry [4]. Song Y et al. built an ensemble deep learning financial risk intelligent monitoring and warning model to address the difficulties faced by traditional methods in handling diversified data in financial markets and adapting to the detection and warning needs of emerging risk types. This model could capture long-term dependencies and trends in financial data [5]. Elhoseny M et al. proposed a financial distress prediction model based on the Adaptive Whale Optimization Algorithm with Deep Learning. The model enhanced the multi-layer perceptron structure through an adaptive hyperparameter tuning mechanism to improve prediction accuracy for corporate financial distress. Experimental results on multiple financial datasets demonstrated that this approach significantly outperformed traditional machine learning models [6]. Chandola D et al. proposed a hybrid deep learning model combining Word2Vec and Long Short-Term Memory (LSTM) networks to predict the directional movement of stock prices based on financial time series and news text data. By integrating semantic and temporal representations, the model effectively captured market fluctuation patterns and provided forward-looking decision support for investors [7]. Cui Y et al. proposed a financial risk forecasting model integrating deep auto-encoder and reinforcement learning. The model combined feature extraction with distributed decision optimization, achieving efficient modeling and real-time prediction of financial risks while significantly outperforming traditional approaches in both accuracy and responsiveness [8].

Adaptive Denoising Technique (ADT) dynamically estimates and reduces the intensity of input noise during the training process, which can significantly improve the discriminative ability of downstream models while maintaining the semantic integrity of the original signal. ADT is widely used to process time series, image, and text data contaminated by noise [9]. In recent years, ADT has demonstrated outstanding performance in areas such as identity authentication, network security, and industrial monitoring, thanks to its robust modeling advantages for heterogeneous, high-dimensional, and high noise data. Padmapriya R et al. proposed an ADT that integrated the convolutional neural network and 3D filtering to improve the quality of image data. The proposed model consistently outperformed traditional and other cutting-edge denoising methods [10]. Wang D et al. proposed a novel self-supervised deep learning method for seismic reflection data noise attenuation based on a 2-neighborhood strategy to address the problem of requiring a large number of paired noise clean samples when training denoising models. This method took a U-shaped convolutional neural network as the main network to construct an end-to-end seismic data denoising self-learning process. The results showed that this method could quantitatively evaluate the performance of denoising algorithms without the need for labeled data, and the denoising effect was better than traditional methods [11]. Wang Y et al. introduced the denoising diffusion probability model into the field of drone communication network security for detecting intrusion behaviors in drone networks. The comparative experimental results on the dataset showed that the accuracy, recall, and F1-score of this model reached 97.9%, 99.3%, and 99.8%, respectively, which were significantly better than other models, proving that this model could effectively achieve network intrusion detection [12]. Mahmoudnezhad F et al. integrated denoising autoencoder and generative adversarial network to predict power load values and meteorological data. Compared with other traditional methods, this model could consistently maintain high prediction accuracy in experiments with real noise data, false data, and damaged data under attacks [13].

In summary, although existing studies employing adaptive denoising and ensemble learning models have achieved notable progress in financial anomaly detection, challenges remain in maintaining minority recall and overall model stability under highly noisy and imbalanced conditions. Therefore, this study proposes an adaptive denoising–evolutionary ensemble learning hybrid model designed to enhance robustness, accuracy, and real-time performance in complex financial risk detection tasks. The model integrates ADT, a perturbation-guided oversampling strategy, and evolutionary ensemble learning with multiple classifiers (EEL) to improve data quality perception, reconstruct minority distributions based on perturbation sensitivity, and optimize classifier weights through adaptive and complementary selection. Within a unified classification framework, the model aims to recognize financial fraud and default risks while maintaining strong generalization and convergence performance. The research goal is to enhance classification robustness under high noise and severe class imbalance, with success defined by sustaining high F1-score and recall across multiple evaluation scenarios, thereby demonstrating stability and practical applicability in financial risk control. The findings provide a new technical pathway for advancing intelligent and proactive financial defense systems.

2 Methods and materials

2.1 Construction of unbalanced financial data classification model based on adaptive denoising sampling technology

In financial risk control scenarios, common classification difficulties mainly focus on imbalanced distribution of financial data, high noise interference, and blurred boundaries. To improve the accuracy and generalization ability for identifying abnormal behaviors, a oversampling method based on ADT module is proposed to construct a highly robust classification model [14], as presented in Figure 1.

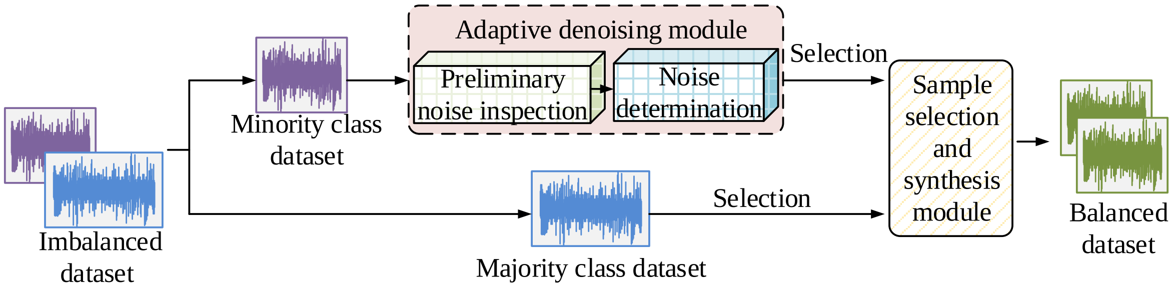


Figure 1: Adaptive denoising oversampling method framework

In Figure 1, the oversampling method based on ADT module mainly has three core modules, namely data preprocessing module, sample generation module, and classifier optimization module. The data preprocessing module first performs noise detection and cleaning on the original imbalanced financial dataset, identifies and removes noisy samples through density peak clustering algorithm, and provides a high-quality balanced data foundation for subsequent processing. The sample generation module adopts an improved oversampling algorithm, which uses a weighted minority class sample generation strategy to reasonably oversample minority class samples in the feature space, effectively alleviating the class imbalance in financial data. The classifier optimization module uses dynamic ensemble learning techniques to adaptively adjust the weight of the base classifier, further improving the recognition ability for minority class samples. In the sample weight scoring, the spatial similarity between samples is calculated based on the Euclidean distance, as presented in equation (1) [15].

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{ik}, x_{jk})^2} \qquad (1)$$

In equation (1), x_i and x_j represent sample vectors. d represents the feature dimension. The sample weight is defined as the weighted average of its neighborhood distance decay, as shown in equation (2).

$$w_i = \frac{1}{|N(x_i)|} \sum_{x_j \in N(x_i)} \exp\{-D(x_i, x_j)\} \qquad (2)$$

In equation (2), w_i represents the weight of sample x_i, and $N(x_i)$ represents its set of neighbors. The sample is generated using weighted interpolation method to generate new samples, and the weighted interpolation is shown in equation (3) [16].

$$x_{new} = x_i + \lambda(x_j - x_i), \lambda \in [0,1] \qquad (3)$$

In equation (3), x_{new} represents the synthesized sample generated. λ represents the difference factor, used to adjust the position of interpolated generated samples. To eliminate noise points, the study continues to introduce a sample credibility function, as shown in equation (4) [17].

$$\eta(x_{new}) = \begin{cases} 1, if \dfrac{|N_{maj}(x_{new})|}{|N(x_{new})|} < \theta \\ 0\text{，} \text{otherwise} \end{cases} \qquad (4)$$

In equation (4), x_{new} represents the sample generated through interpolation. $N(x_{new})$ represents all sample sets in the neighborhood of x_{new} . $N_{maj}(x_{new})$ represents the subset of samples belonging to the majority class in the neighborhood. θ represents the discrimination threshold. The parameter θ represents the discrimination threshold used to distinguish credible from non-credible samples, typically ranging between 0.5 and 0.8. A larger θ value enforces stricter noise filtering and enhances data purity but may cause some minority samples to be misclassified and discarded. A smaller θ value retains greater sample diversity but may introduce additional noise. η indicates whether the sample should be retained, with 1 indicating retention and 0 indicating discard. The resampled financial data is fed into the functional module of the the system for training and classification. Figure 2 presents the overall functional structure of the system.

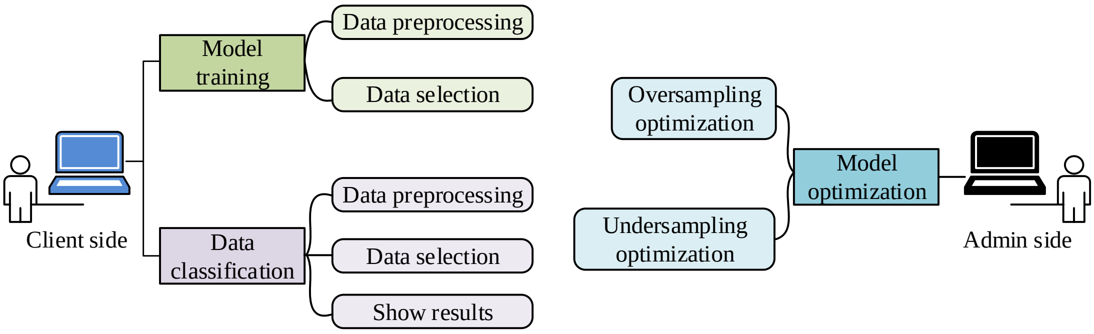


Figure 2: System function structure

In Figure 2, the system has two parts: the user side interaction layer and the backend management layer. The user end is divided into two core subsystems: model training and data classification, which are used to standardize, fill in missing values, and remove outliers from raw high noise and imbalanced financial data, providing a clean data foundation for subsequent processing. The backend management is mainly responsible for optimizing the model by oversampling minority class data and under-sampling majority class data. The advantages of multiple base classifiers are combined, and the adaptive weight adjustment mechanism is utilized to optimize model performance, further improving the accuracy of financial data classification. The specific process of the classification module is presented in Figure 3.

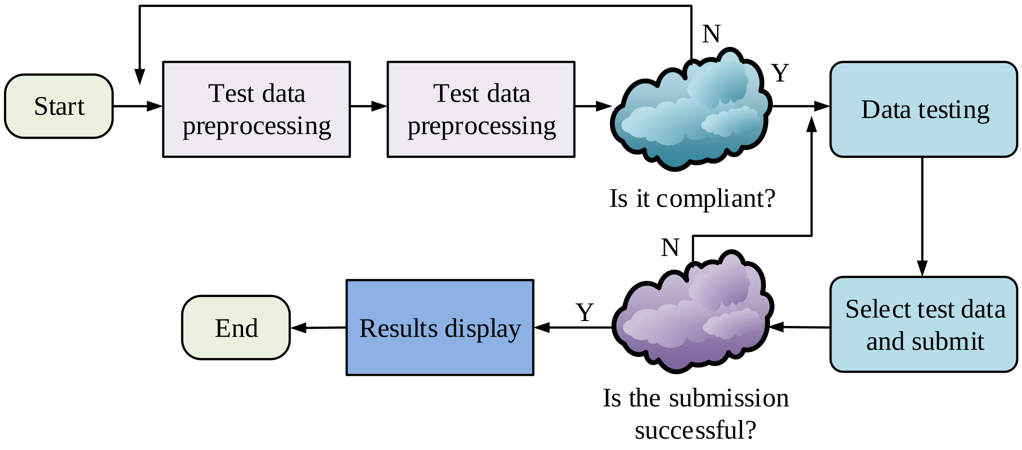


Figure 3: Classification modules process flow

In Figure 3, the classification process for imbalanced financial data adopts a branch judgment structure, which is divided into three core steps: data submission and verification, standard checking, and classification result output. The user first uploads the financial data to be analyzed on the test data preprocessing interface. The system conducts compliance checks on key elements such as data format and missing values. If the data does not meet the processing standards, the process will automatically redirect to the data selection interface, prompting the user to re-upload valid data. If the verification is successful, the system will directly perform the classification operation and output the predicted results on the result display interface. This process constructs a reliable data entry loop through a dual judgment mechanism of "data standardization" and "submission validity", effectively avoiding potential classification bias caused by imbalanced data samples. Meanwhile, it improves the stability of the overall process and user interaction experience, ensuring that the classification model has robust data support and processing capabilities in actual financial scenarios. The process avoids invalid operations through a mandatory verification mechanism, while optimizing user experience and achieving a complete closed-loop processing from data input to result output. The study introduces an attention mechanism to dynamically adjust the importance of different samples during this classification stage, as shown in equation (5).

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n}\exp(e_j)}, e_i = \tanh(Wx_i + b) \qquad (5)$$

In equation (5), α_i represents the final attention weight of sample x_i. e_i represents the average attention score of the sample. W represents a learnable weight matrix used for linear transformation of input features. b represents the bias. By introducing attention mechanisms, different weights can be dynamically assigned to each sample, making the model more focused on samples that have greater impacts on the results, and improving the robustness and classification performance of the final ensemble decision. Especially when dealing with financial scenarios with uneven data distribution and complex sample noise, more refined sample level differentiation and optimization strategies can be achieved. The classification result of the model is shown in equation (6).

$$\hat{y} = \sigma\{W_2 \cdot \mathrm{ReLU}(W_1 x + b_1) + b_2\} \qquad (6)$$

In equation (6), $\hat{y}$ represents the classification result. σ represents the Sigmoid function. W_1 and W_2 represent weights. During parameter optimization, the denoising coefficient and penalty factor are dynamically adjusted according to the sample noise intensity and classification residuals, forming a feedback-driven self-regulation mechanism. Conceptually, this mechanism aligns with adaptive control strategies in nonlinear systems, where parameters are continuously refined to counter input uncertainty and nonlinear disturbances. This feedback process allows the model to automatically balance denoising strength and penalty weighting during convergence, achieving a steady trade-off between robustness and stability, and enhancing its adaptability in complex financial data environments [18]. In summary, a classification model based on ADT and minority class weighted oversampling is constructed to address the imbalanced financial data. This model combines majority class denoising and minority class augmentation strategies, introduces interpolation to reconstruct sample distribution, and effectively alleviates the bias caused by imbalanced financial data samples.

2.2 Optimization of unbalanced financial data classification model based on multi-classifier ensemble learning

After constructing the data preprocessing and preliminary financial data classification model based on ADT and oversampling techniques, to further improve the generalization ability and stability in scenarios involving multi-class imbalanced financial data, EEL based on multiple classifiers is introduced [19]. The basic framework is presented in Figure 4.

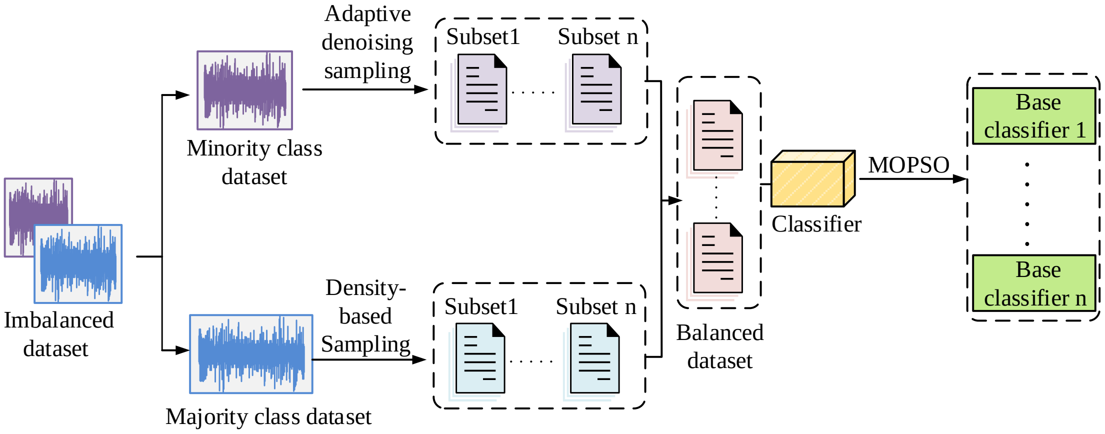


Figure 4: EEL framework

As shown in Figure 4, the model framework employs EEL to classify imbalanced financial datasets, following a multi-stage progressive structure. First, the training dataset is divided into majority and minority sample sets. The minority samples are augmented using the ADT to perform oversampling. Next, a density-based sample selection strategy is applied to majority samples to generate several majority-class subsets, which are then paired with the minority subset to form multiple balanced datasets. Each balanced dataset is used to train a corresponding base classifier independently. Finally, the Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is employed to partition regions and optimize the performance of all base classifiers. MOPSO simultaneously optimizes classification accuracy and classifier diversity within a multi-objective search space, dynamically adjusting the weights and structures of the sub-classifiers to enhance the robustness and generalization of the model while maintaining overall precision. The detailed training process of the model is illustrated in Figure 5.

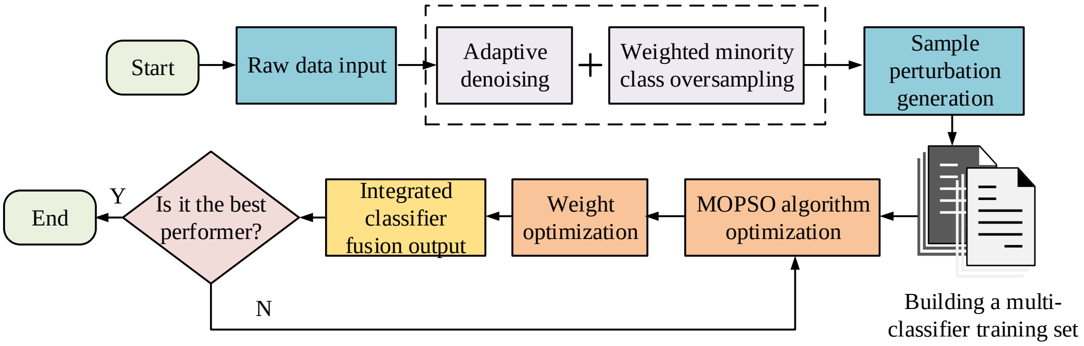

Start
Raw data input
Adaptive denoising
+
Weighted minority class oversampling
Sample perturbation generation
Building a multi-classifier training set
MOPSO algorithm optimization
Weight optimization
Integrated classifier fusion output
Is it the best performer?
Y
N
End

Figure 5: The model training flowchart

In Figure 5, the training process undergoes structural optimization and adjustment, further highlighting the iterative optimization mechanism of multi-classifier ensemble. The process begins with the input of raw data, which is preprocessed using ADT and majority class weighted few class oversampling techniques to generate perturbed samples and construct a multi-classifier training set. Subsequently, multiple candidate classification models are obtained through multi-objective particle swarm optimization algorithm optimization [20]. Next, the model performs weight optimization and classifier ensemble fusion output to improve overall performance. A performance evaluation module is added to assess the performance of the current ensemble model. If the ensemble model is the current optimal solution, the process terminates and the final classification result is output. If not, the process goes back to the sample perturbation stage and starts a new round of optimization iteration. In the sample partitioning stage, to avoid over-fitting, a perturbation function is defined to transform the input features, as shown in equation (7).

$$x_i' = x_i + \gamma \cdot \varepsilon_i \qquad (7)$$

In equation (7), γ represents the disturbance intensity parameter. ε_i represents a noise vector that follows a normal distribution. For each sub model, a confidence function based on sample voting scores is used to model individual learning performance. The function is shown in equation (8).

$$C_k(x_i) = \frac{1}{Z}\sum_{c=1}^{C} p_k(y=c|x_i) \cdot \log p_k(y=c|x_i) \qquad (8)$$

In equation (8), $C_k(x_i)$ represents the class confidence score of sample x_i. $p_k(y=c|x_i)$ represents the probability value of predicted class c. Z represents the normalization coefficient. C signifies the total number of classes. To ensure accuracy and diversity in model fusion, a correlation penalty function is introduced to constrain redundancy among sub models, as shown in equation (9).

$$\Omega(H)=\sum_{i=1}^{N}\sum_{k\neq 1}\{h_k(x_i)-h_l(x_i)\}^2 \qquad (9)$$

In equation (9), $h_k(x_i)$ represents the output result of the k-th sub model on sample x_i. $\Omega(H)$ represents the prediction redundancy of the overall model integration. By calculating the squared difference between outputs of different classifiers, the model quantifies inter-classifier diversity. When two classifiers produce similar predictions, their corresponding gradient updates are reduced, thereby lowering their relative influence during training. Conversely, classifiers that generate diverse outputs receive higher adaptive weights. This mechanism encourages diversity among classifiers while maintaining accuracy, effectively preventing ensemble redundancy and improving generalization. Next, a weighted fusion strategy is applied to the output results of all sub classifiers. The weighted fusion is shown in equation (10).

$$y_i^{fusion}=\sum_{k=1}^{K}\alpha_k\cdot h_k(x_i) \qquad (10)$$

In equation (10), α_k represents the weight of the k-th classifier, satisfying $\sum_k \alpha_k=1$. During the training process, a loss function is used as the training objective for backpropagation, as shown in equation (11)

$$\mathrm{L}=\frac{1}{N}\sum_{i=1}^{N}\mathrm{L}_{CE}(y_i,y_i^{fusion})+\beta\cdot\Omega(H) \qquad (11)$$

In equation (11), L_{CE} represents the cross-entropy loss function. β represents redundant penalty hyperparameters. In the optimization process, the main hyperparameters of the MOPSO algorithm include the learning rate η, inertia weight w, acceleration coefficients c_1 and c_2, and the maximum number of iterations T, where $\eta\in[0.01,0.1]$, $w\in[0.5,0.9]$, $c_1=c_2=2.0$, and $T=100$. A linearly decreasing inertia weight strategy is adopted to balance global and local exploration, maintaining strong global search capability in early iterations and enhancing local refinement in later stages. Experimental results show that the loss function converges smoothly after approximately 40 iterations with no significant oscillation. A dynamic learning rate decay schedule is further introduced to accelerate convergence while ensuring stable classification accuracy.

Ultimately, to optimize the migration adaptability of the ensemble model in different tasks or datasets, an integration function is taken as the final decision-making mechanism, as shown in equation (12).

$$y_i^{final}=\mathrm{softmax}\left\{\kappa_1\cdot y_i^{fusion}+\kappa_2\cdot\sum_{k=1}^{K}\phi_k(h_k)\right\} \qquad (12)$$

In equation (12), ϕ_k represents the sub model feature mapping function. κ_1 and κ_2 both represent adjustable fusion parameters. This equation combines fusion discrimination and sub model feature transformation results to achieve the final classification output. In summary, the final fusion classification model structure is shown in Figure 6.

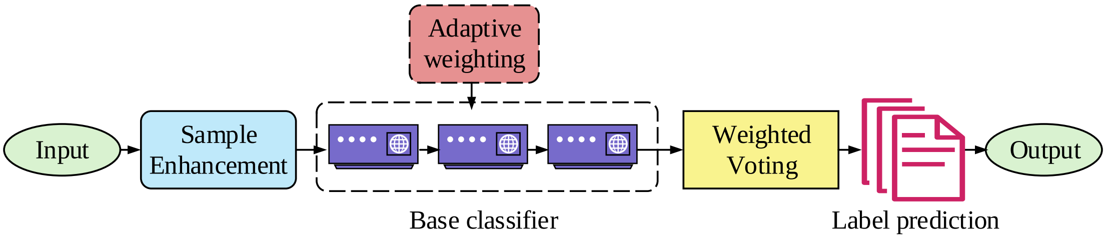


Figure 6: The final classification model structure

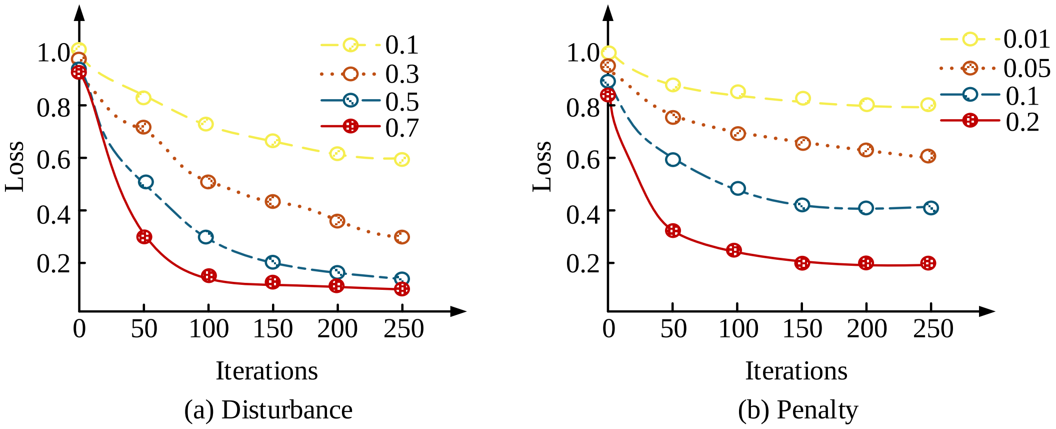
From Figure 6, the final classification model reflects the dual optimization mechanism of integrating multiple classifiers and EEL for imbalanced financial data. Multiple parallel sub classifier channels are set up in the model structure to independently learn from the samples that have been adaptively denoised and oversampled, improving the model's ability to recognize minority financial risk samples. Subsequently, the integration layer integrates the prediction results of each sub model through weighted fusion or voting mechanism to achieve more robust classification decisions. The ensemble strategy module introduced dynamically adjusts the weight distribution of each classifier through evolutionary algorithms, enabling the overall model to adapt to changes in data distribution in different financial scenarios, thereby improving the recognition accuracy for high-risk classes such as loan defaults and fraudulent behavior. The overall structure takes into account feature expression, multi-source fusion, and dynamic optimization, fully reflecting the systematic solution to imbalanced financial data classification.

3 Results

3.1 Parameter selection and ablation testing for classification models

This study sets up an experimental environment with Intel Xeon Gold 6338 CPU, NVIDIA A100 40GB GPU, 64GB memory configuration, and Ubuntu 20.04 LTS operating system. The model training is performed using the Adam optimizer with an initial learning rate of 0.001, a batch size of 64, and a maximum of 100 iterations. An early stopping strategy (patience = 10) is applied to prevent overfitting. The Lending Club Loan Default Dataset (LCD) and Credit Card Fraud Detection Dataset (CCFD) are taken as the data sources for model testing. LCD comes from a well-known lending platform in the United States, containing user loan application information and its final repayment status. It is a classic dataset in the fields of financial risk control and credit risk prediction. GMSC contains approximately 280,000 credit card consumption records and 30 anonymized features, suitable for anomaly detection and performance testing of classification models. To determine the optimal performance parameter configuration for model operation, the hyperparameter selection test is conducted on the disturbance intensity parameter γ and redundancy penalty parameter β. Figure 7 displays the test results.

(LCD:https://www.kaggle.com/datasets/wordsforthewise/lending-club
CCFD:https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)



(a) Disturbance

(b) Penalty

Figure 7: Hyperparameter test results

Figure 7 (a) displays the selection test results of the disturbance intensity parameter γ, and Figure 7 (b) displays the selection test results of the redundancy penalty parameter β. As shown in Figure 7 (a), with the gradual increase of disturbance parameters, the overall loss function during training showed a downward trend. When γ =0.7, the convergence speed of the model significantly accelerated at multiple iteration points, and the final loss value was the lowest. This indicates that higher disturbance intensity helps to enhance sample diversity and model robustness, and improve classification ability. When taking 0.1 and 0.3, the model training process tended to be stable but converged slowly, and was prone to getting stuck in local optima. This indicates that weak perturbations are not sufficient to stimulate the potential discriminative ability. In Figure 7 (b), with the increase of penalty parameters, the constraint ability of the model on outliers and classification errors gradually increased, and the training loss decreased accordingly. When β =0.2, the loss function reached its optimal level, and the model performed best in balancing classification accuracy and generalization ability. Therefore, considering the convergence efficiency and performance during the training process, the optimal hyperparameter combination was ultimately determined to be γ =0.7 and β =0.2, at which point the overall performance was optimal. The study continues to conduct ablation tests on two datasets, as displayed in Figure 8.

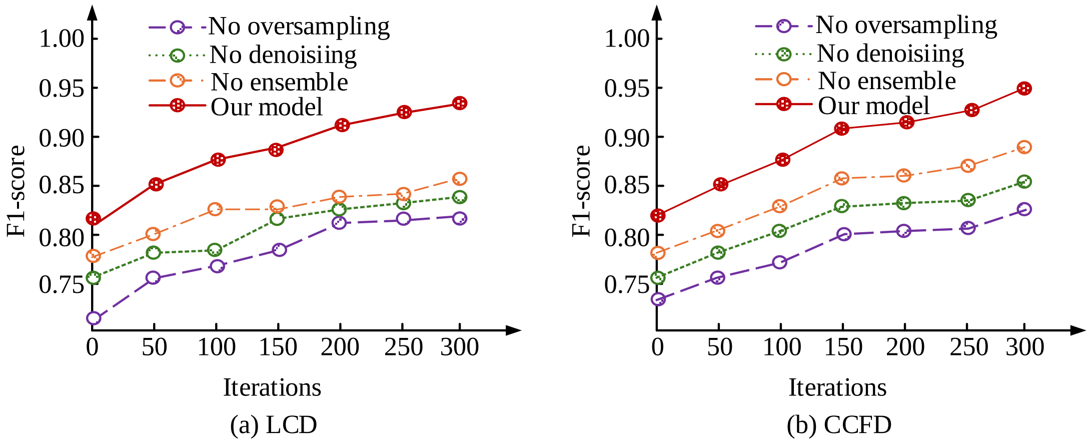


Figure 8: Ablation test results

Figure 8 (a) displays the ablation test results on the LCD dataset, and Figure 8 (b) displays the ablation test results on the CCFD dataset. In Figure 8 (a), the complete model maintained a high F1-score throughout each iteration, ultimately stabilizing at around 0.92. In contrast, the model without the ADT module showed a significant decrease in performance, with a final F1-score of approximately 0.86, indicating that the module effectively alleviated most types of noise interference. After removing the weighted oversampling module, the final F1-score was approximately 0.84. Removing the EEL module weakened the generalization ability to complex patterns, with an F1-score of only 0.87. As shown in Figure 8 (b), the complete model remained optimal on the CCFD dataset, with a stable F1-score of 0.89. To further validate the robustness and generalizability of the results, each model was repeatedly evaluated under multiple independent data splits. The mean, standard deviation, and significance test results are reported in Table 1.

Table 1: Statistical summary of ablation experiment results

Dataset	Method	Mean F1-score/%	Standard deviation	*p*(vs. Our model)
LCD	No denoising	86.0	0.009	0.012
	No oversampling	84.2	0.007	0.008
	No ensemble	87.3	0.009	0.015
	Our model	92.1	0.006	-
CCFD	No denoising	83.3	0.010	0.010
	No oversampling	81.4	0.009	0.009
	No ensemble	85.3	0.008	0.013
	Our model	90.8	0.005	-

As shown in Table 1, the complete model achieved the highest mean F1-scores and the lowest standard deviations on both the LCD and CCFD datasets. Moreover, the differences between the complete model and the ablated variants were statistically significant (p< 0.05), confirming the structural effectiveness and robustness of the model.

To further verify the applicability of the model to different types of imbalanced financial data, the Give Me Some Credit dataset (GMSC) is introduced as an additional testing source. The dataset has a minority class proportion of approximately 2.3%, indicating a significant imbalance, which is suitable for evaluating the stability and generalization capability of classification models under extreme class distribution conditions. The study further compares the performance of several baseline models and the complete model constructed in the study on multiple indicators, as displayed in Table 2.

Table 2: Test results for different indicators

Data set	Method	Accuracy/%	Precision/%	Recall/%	F1-score/%
LCD	No denoising	88.5	86.4	86.2	86.0
	No oversampling	87.2	84.6	83.9	84.2
	No ensemble	89.1	86.5	88.1	87.3

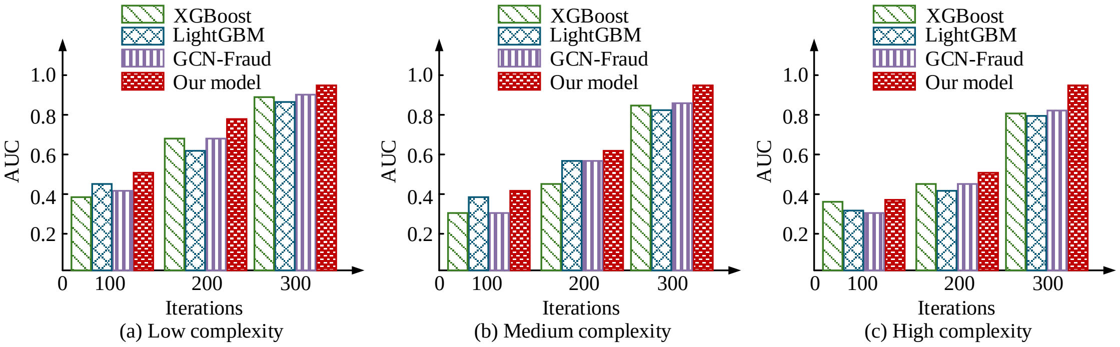
	Our model	94.6	93.2	92.4	92.1
CCFD	No denoising	86.9	85.1	82.3	83.3
	No oversampling	85.7	83.2	79.8	81.4
	No ensemble	88.2	86.0	83.9	85.3
	Our model	93.7	91.5	90.2	90.8
GMSC	No denoising	84.2	81.6	78.4	79.9
	No oversampling	82.8	80.2	77.1	78.6
	No ensemble	85.1	82.7	80.3	81.5
	Our model	91.2	89.4	87.6	88.4

As shown in Table 2, the proposed complete model demonstrated superior overall performance across all three financial datasets, including LCD, CCFD, and GMSC, compared with the ablation variants. On the LCD dataset, the model achieved an accuracy of 94.6%, a precision of 93.2%, a recall of 92.4%, and an F1-score of 92.1%. On the CCFD dataset, these values were 93.7%, 91.5%, 90.2%, and 90.8%, respectively. Even in the more severely imbalanced GMSC dataset, the model maintained an accuracy of 91.2% and an F1-score of 88.4%, indicating strong robustness and generalization capability. In summary, the complete model outperforms the reduced version in all indicators, verifying the effectiveness and synergy of the three core modules of ADT, oversampling, and EEL.

3.2 Classification model simulation testing

To further verify the adaptability and stability of the constructed model in practical application scenarios, a simulation testing environment based on financial risk data is designed. The simulation data used are sourced from the three real financial datasets. Based on the ratio of majority classes to minority classes in the dataset, the study sets three types of operating conditions: low complexity, medium complexity, and high complexity, to simulate financial business environments with different risk levels and data complexities. In the complexity analysis, the complete model consists of 10 base classifiers with approximately 1.1×10^5 parameters, requiring about 1.2 s per training epoch and 300 s in total. Three simulation conditions were designed by adjusting class imbalance ratios and noise levels: low complexity (1:5 ratio, 5% noise), medium complexity (1:10 ratio, 10% noise), and high complexity (1:20 ratio, 15% noise). These settings simulate progressively imbalanced and noisy financial risk scenarios to evaluate model adaptability and robustness in increasingly complex situations.

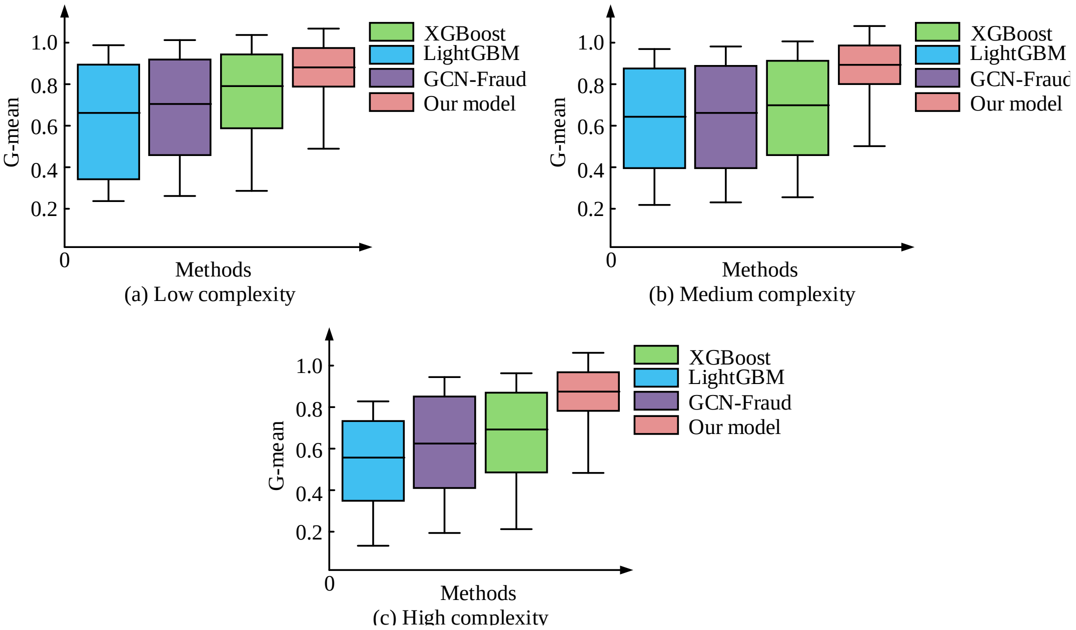
The study compares three commonly used models in this field, namely Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Graph Convolutional Network for Fraud Detection (GCN-Fraud), with the research model, taking Area Under Curve (AUC) as the testing metric. The results are shown in Figure 9.



(a) Low complexity
(b) Medium complexity
(c) High complexity

Figure 9: AUC test results

Figures 9 (a), 9 (b), and 9 (c) show the AUC test results of four models under three different complexities. From Figure 9 (a), in the low complexity test, the performance differences among the four models were relatively small, but the AUC value of the research model reached 0.94, which was better than that of XGBoost (0.91), LightGBM (0.89), and GCN-Fraud (0.90), demonstrating excellent fitting ability for simple sample structures. As shown in Figure 9 (b), in the medium complexity test, the difference further widened. The AUC of the research model increased to 0.95, while XGBoost, LightGBM, and GCN-Fraud were 0.89, 0.87, and 0.88, respectively. Similarly, in Figure 9 (c), the research model still maintained at 0.92, fully demonstrating its strong expression and classification ability under complex network structures and high-dimensional heterogeneous features. The study continues to take geometric mean as the testing indicator, and draws box plots of four models to compare the results. The results are shown in Figure 10.



(a) Low complexity

(b) Medium complexity

(c) High complexity

Figure 10: The box plot test results

Figures 10(a), 10(b), and 10(c) illustrate the box plot results of four models under three different levels of simulation complexity. Using the geometric mean as the evaluation metric, noticeable differences in stability and class balance can be observed among the models. The proposed integrated model maintained high median geometric means of 0.91, 0.93, and 0.90 under low, medium, and high complexity, respectively, with compact boxes and short whiskers, indicating strong robustness and balanced recognition capability under varying noise intensities and imbalance conditions. Although a slight decrease in convergence speed was observed under high complexity, the overall stability remained strong, reflecting the model's inherent adaptability and generalization strength across different levels of data complexity. The study continues to compare the test results of four models on True Positive Rate (TPR), False Positive Rate (FPR), prediction stability, and average operating time under three operating conditions, as displayed in Table 3.

Table 3: Test results under different conditions

Situation	Method	TPR/%	FPR%	Stability/	Running time/s
Low complexity	XGBoost	86.5	13.8	0.83	15.9
	LightGBM	85.2	14.1	0.81	16.4
	GCN-Fraud	88.8	12.9	0.87	20.4
	Our model	91.6	11.5	0.92	14.1
Medium complexity	XGBoost	87.4	15.1	0.80	16.7
	LightGBM	82.6	16.3	0.78	17.3
	GCN-Fraud	86.1	14.3	0.84	21.6
	Our model	89.2	12.3	0.89	14.6
High complexity	XGBoost	80.1	17.8	0.75	17.5
	LightGBM	78.3	19.5	0.73	18.1
	GCN-Fraud	82.5	16.7	0.81	22.8
	Our model	86.5	14.0	0.85	15.2

According to Table 3, the GCN-Fraud model outperformed the traditional XGBoost and LightGBM models in most operating conditions with the support of graph structure modeling capabilities. Its TPR reached 88.8% and 86.1% in low complexity and medium complexity operating conditions, respectively. However, the running time of the model was significantly high, averaging over 20 seconds. Although XGBoost and LightGBM had high operating efficiency, the FPR significantly increased under high complexity conditions, with a stability of only 0.73. In contrast, the proposed model maintained the highest TPR and stability among the three operating conditions, with the lowest FPR. The running time was also controlled within 15 seconds, demonstrating excellent performance balance and strong robustness.

Additionally, an error analysis was conducted to examine typical misclassified samples. The results indicate that under extreme imbalance and high noise, several high-confidence misclassifications occurred in overlapping feature regions, leading to localized prediction bias. This suggests that while the model maintains overall robustness, further improvement can be achieved by introducing dynamic feature re-weighting or adversarial perturbation strategies to reduce such errors. Overall, the proposed model achieves the best comprehensive performance in the simulation environment, with strong stability and scalability.

3.3 Related work comparison and summary

To highlight the differences and methodological improvements between this study and existing research, a comparative summary was conducted on dataset types, algorithm frameworks, and performance indicators reported in recent representative works. Most previous studies have focused on optimizing feature extraction or sampling strategies, achieving moderate improvements, but lacking adaptability under severe class imbalance and noisy financial data conditions. The overall comparison of these studies is shown in Table 4.

Table 4: Comparison of related works

Research	Dataset	Method	Main metric	Limitation
Liu et al.	Financial time series	GAN-based multi-classification	F1=0.86	Lack of adaptive noise-handling mechanism
Song et al.	Bank risk dataset	LSTM+Transformer	F1=0.89	Does not address class imbalance
Hesar H D et al	ECG dataset	Adaptive denoising filter	Accuracy=0.91	Not applicable to financial data
Mujahid M et al	Synthetic imbalanced data	Data oversampling+ML	F1=0.87	Limited ensemble model stability
Our model	LCD/CCFD/GMSC	ADT+EEL	F1=0.92	-

Compared with the existing works, the study introduces an adaptive denoising mechanism to dynamically suppress sample noise while combining evolutionary ensemble learning to optimize classifier weights. This integrated design realizes a synergistic effect between noise adaptation and classifier optimization, effectively addressing the lack of adaptive noise-handling and robustness in prior ensemble-based financial classification models.

4 Discussion

The experimental results indicate that the proposed model demonstrates strong adaptability and stability across multiple imbalanced financial datasets. Gradient boosting frameworks such as XGBoost and LightGBM perform efficiently in feature learning but tend to rely heavily on majority-class features when data complexity increases or noise interference intensifies. This fixed split criterion amplifies gradient propagation errors and weakens convergence stability, leading to declines in F1-score and TPR. The GCN-Fraud model enhances transaction-relationship modeling through graph convolution, but its node aggregation mechanism magnifies adjacency noise, causing noisy node information to spread through the network and resulting in higher FPR and blurred decision boundaries.

In contrast, the proposed model effectively mitigates these limitations through the synergistic mechanisms of ADT and EEL. ADT dynamically adjusts sample weights based on noise level and residuals, suppressing abnormal samples that distort boundary learning. Meanwhile, EEL uses multi-objective evolutionary optimization to continuously balance the weight distribution among sub-classifiers, maintaining steady convergence and classification robustness under different imbalance ratios and complexity levels. The AUC and TPR fluctuations remain within 3% under complex conditions, indicating strong self-regulation in noisy and nonlinear environments.

Misclassification analysis further reveals behavioral distinctions among models. XGBoost and LightGBM frequently exhibit “false absorption,” misclassifying minority samples near majority boundaries, while GCN-Fraud often suffers from “noise propagation,” causing neighboring nodes to be jointly misclassified. In contrast, the proposed model greatly reduces these two error modes, with misclassified samples mainly occurring in areas of feature overlap or blurred boundaries, without systematic bias. It exhibits stable and reliable differentiation in complex financial data scenarios.

However, the structural complexity of the model inevitably increases the opacity of its decision-making process. Future work may employ feature contribution visualization or decision-path analysis to better interpret internal mechanisms, thereby improving transparency and practical trustworthiness in financial risk detection.

5 Conclusion

A classification model that integrated ADT, oversampling strategy, and EEL structure was proposed to address the strong noise interference and extremely uneven class distribution in financial data classification. This model adjusted the intensity of sample perturbations and suppressed the false positive through perturbation parameters. Meanwhile, the EEL strategy based on multiple subsets achieved the optimal ensemble of classifier structures, effectively improving the robustness and generalization ability. The hyperparameter testing on two typical financial fraud datasets, LCD and CCFD, showed that when the disturbance intensity was set to 0.7 and the penalty coefficient was 0.2, the model obtained the optimal F1-score of 92.1%. In the ablation test, the research model achieved an accuracy of 94.6%, a precision of 93.2%, and a recall of 92.4%. The ablation test indicated that the loss of any module caused significant performance degradation. In the simulation test, the research model was compared with three classic models, XGBoost, LightGBM, GCN-Fraud. The results showed that the research model had a TPR of up to 91.6%, a FPR as low as 11.5%, and the strongest stability of 0.92 under the three types of conditions. The running time was controlled within 15 seconds, demonstrating strong adaptability and efficiency, especially suitable for real-time and high-frequency financial prediction scenarios that demand both stability and computational efficiency. The model can be deployed in lightweight batch-inference mode, ensuring consistent latency and memory consumption within practical operational limits.

Moreover, the model is designed with consideration for the computational characteristics of real-world financial systems, allowing flexible deployment across distributed or heterogeneous environments to support parallel processing and continuous optimization of multi-source data. Its self-regulating mechanism aligns with the concept of adaptive control in nonlinear dynamic systems, achieving parameter convergence and predictive stability under complex and evolving data conditions, and suggesting practical potential for real-world financial applications. However, the current model has not fully considered the more dynamic and systematic challenges such as changes in user behavior and cross-platform fraud collaborative detection. In the future, graph neural network architecture and federated learning framework can be combined to further expand its practicality in cross-platform risk control and privacy protection scenarios.

Data and code availability statement

The datasets used in this study are publicly available open-source datasets. The model implementation is developed in Python 3.10 with the PyTorch 2.0 framework. The code and experimental pipeline are archived in a private repository and can be made available upon reasonable request.

References

[1] Liu L, Pei Z, Chen P, Luo H, Gao Z, Feng K, Gan Z. An efficient gan-based multi-classification approach for financial time series volatility trend prediction. International Journal of Computational Intelligence Systems, 2023, 16(1): 40-41. https://doi.org/10.1007/s44196-023-00212-x

[2] Baydaş M, Yılmaz M, Jović Ž. A comprehensive MCDM assessment for economic data: success analysis of maximum normalization, CODAS, and fuzzy approaches. Financial Innovation, 2024, 10(1): 105-106. https://doi.org/10.1186/s40854-023-00588-x

[3] Bao W, Xiao J, Deng T, Bi S, Wang J. The challenges and opportunities of financial technology innovation to bank financing business and risk management. Financial Engineering and Risk Management, 2024, 7(2): 82-88. https://doi.org/10.23977/ferm.2024.070212

[4] Gao H, Kou G, Liang H. Machine learning in business and finance: a literature review and research opportunities[J]. Financial Innovation, 2024, 10(1): 86-87. https://doi.org/10.1186/s40854-024-00629-z

[5] Song Y, Du H, Piao T, Shi H. Research on financial risk intelligent monitoring and early warning model based on LSTM, transformer, and deep learning. Journal of Organizational and End User Computing (JOEUC), 2024, 36(1): 1-24. https://doi.org/10.4018/JOEUC.337607

[6] Elhoseny M, Metawa N, Sztano G. Deep learning-based model for financial distress prediction[J]. Annals of operations research, 2025, 345(2): 885-907. https://doi.org/10.1007/s10479-022-04766-5

[7] Chandola D, Mehta A, Singh S. Forecasting directional movement of stock prices using deep learning[J]. Annals of Data Science, 2023, 10(5): 1361-1378. https://doi.org/10.1007/s40745-022-00432-6

[8] Cui Y, Yao F. Integrating deep learning and reinforcement learning for enhanced financial risk forecasting in supply chain management[J]. Journal of the Knowledge Economy, 2024, 15(4): 20091-20110. https://doi.org/10.1007/s13132-024-01946-5

[9] Hesar H D, Hesar A D. Adaptive augmented cubature Kalman filter/smoother for ECG denoising. Biomedical Engineering Letters, 2024, 14(4): 689-705. https://doi.org/10.1007/s13534-024-00362-7

[10] Padmapriya R, Jeyasekar A. CA-EBM3D-NET: a convolutional neural network combined framework for denoising with weighted alpha parameter and adaptive filtering. International Journal of Information Technology, 2024, 16(8): 4855-4867. https://doi.org/10.1007/s41870-024-02160-x

[11] Wang D, Chen G, Chen J, Cheng Q. Seismic data denoising using a self-supervised deep learning network. Mathematical Geosciences, 2024, 56(3): 487-510. https://doi.org/10.1007/s11004-023-10089-3

[12] Wang Y, Ding J, He X, Wei Q, Yuan S, Zhang J. Intrusion detection method based on denoising diffusion probabilistic models for uav networks. Mobile Networks and Applications, 2024, 29(5): 1467-1476. https://doi.org/10.1007/s11036-023-02222-7

[13] Mahmoudnezhad F, Moradzadeh A, Mohammadi I B. Electric load forecasting under false data injection attacks via denoising deep learning and generative adversarial networks. IET Generation, Transmission & Distribution, 2024, 18(20): 3247-3263. https://doi.org/10.1049/gtd2.13273

[14] Mujahid M, Kına E, Rustam F,. Data oversampling and imbalanced datasets: An investigation of performance for machine learning and feature engineering. Journal of Big Data, 2024, 11(1): 87-88. https://doi.org/10.1186/s40537-024-00943-4

[15] Zhu X, Wu X, Wu B,. An improved fuzzy C-means clustering algorithm using Euclidean distance function. Journal of Intelligent & Fuzzy Systems, 2023, 44(6): 9847-9862. https://doi.org/10.3233/JIFS-223576

[16] Ohlert P L, Bach M, Breuer L. Accuracy assessment of inverse distance weighting interpolation of groundwater nitrate concentrations in Bavaria (Germany). Environmental Science and Pollution Research, 2023, 30(4): 9445-9455. https://doi.org/10.1007/s11356-022-22670-0

[17] Metiri F, Zeghdoudi H, Saadoun A. Some results on quadratic credibility premium using the balanced loss function. Arab Journal of Mathematical Sciences, 2023, 29(2): 191-203. https://doi.org/10.1108/AJMS-08-2021-0192

[18] Boulkroune A, Zouari F, Boubellouta A. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems[J]. Journal of Vibration and Control, 2025: 10775463251320258. https://doi.org/10.1177/10775463251320258

[19] Salih N, Ksantini M, Hussein N, Halima D B, Razzaq A, Ahmed S. Deep learning models and fusion classification technique for accurate diagnosis of retinopathy of prematurity in preterm newborn[J]. Baghdad Science Journal, 2024, 21(5): 21. https://doi.org/10.21123/bsj.2023.8747

[20] Hu L, Yang Y, Tang Z. FCAN-MOPSO: an improved fuzzy-based graph clustering algorithm for complex networks with multiobjective particle swarm optimization. IEEE Transactions on Fuzzy Systems, 2023, 31(10): 3470-3484. https://doi.org/10.1109/TFUZZ.2023.3259726