ITC 2/55

Information Technology and Control
Vol. 55 / No. 2/ 2026
pp. 805-824
DOI 10.5755/j01.itc.55.2.43595

Stock Price Trend Prediction and Securities Trading Optimization Based on CNN-GAA

Received 2025/11/13	Accepted after revision 2026/04/12

HOW TO CITE: Wang, J., Zhang, L. (2026) Stock Price Trend Prediction and Securities Trading Optimization Based on CNN-GAA. *Information Technology and Control*, 55(2), 805-824. https://doi.org/10.5755/j01.itc.55.2.43595

Stock Price Trend Prediction and Securities Trading Optimization Based on CNN-GAA

Jia Wang*

School of Economics and Management, The Open University of China, Beijing, 100039, China

Lei Zhang

Research and Development Department, Beijing Universal Xingxue Technology Development Co., Ltd., Beijing, 100191, China

Corresponding author: wangjiaxinxiang001@126.com

This study proposes an optimization model (SSCGL) that combines the stock price trend prediction algorithm (CNN-GAA-LSTM) with the securities trading algorithm (SE Net SMCM) aimed at improving the accuracy of stock price trend prediction. This model combines convolutional neural networks with gated attention aggregators to enhance feature representation, introduces long short-term memory networks for trend prediction, and uses a combination of compression and excitation networks and sequential Monte Carlo methods to optimize trading decisions. The innovation of this model lies in the deep integration of sequential Monte Carlo algorithm with Squeeze-and-Excitation Network to dynamically evaluate market trading signals. The experimental results show that the model exhibits strong predictive performance, with a trend prediction accuracy of 95.5%, an accuracy of 0.975, and a root mean square error of only 3.93%. In trading simulations, the model performed outstandingly with a Sharpe ratio of 1.812 and a Kalmar ratio of 3.871. Empirical evaluations showed that the annualized return rate was as high as 46.11%. These data fully demonstrate that the model can not only accurately capture market trends, but also improve trading efficiency by effectively controlling risks and reducing losses. Overall, this plan provides stable returns and reliable technical support for investment activities, helping financial institutions achieve deep integration with intelligent algorithm systems.

KEYWORDS: Stock price trend prediction; Securities trading optimization; CNN; Gated attention aggregator; SE-Net; SMCM; LSTM

1. Introduction

With the increasing complexity of data in global financial markets and the faster pace of trading [4]. As the core part of modern economy, studying the dynamic changes and volatility of stock price trends in the stock market has gradually become a global focus [9]. The current methods for predicting stock price trends have evolved from traditional regression models such as support vector machines to deep learning methods such as recurrent network neural models. However, existing methods still have certain limitations, such as the difficulty of grasping the feature correlation of multidimensional data with a single convolutional structure, and the potential for errors in capturing fluctuating trend signals at the same time, leading to a decrease in prediction accuracy [12]. Meanwhile, an effective securities trading strategy not only relies on accurate trend prediction, but also requires a combination of risk control and return optimization mechanisms to achieve stable trading under complex nonlinear data [1, 6]. Society can further optimize the process of securities trading, enhance risk control and return capabilities, and alleviate the subjectivity and data uncertainty issues of traditional methods. The local feature extraction structure of Convolutional Neural Network (CNN) makes the algorithm adept at capturing short-term time series features, improving the accuracy of data and signal recognition [27]. In addition, the dynamic weight allocation feature aggregation mechanism of the Gated Attention Aggregator (GAA) algorithm enables the algorithm to focus more on the processing of core information, thereby capturing deeper level feature connections and enhancing information processing capabilities [25]. Therefore, in order to improve the accuracy of predicting stock price trends and the efficiency of securities trading in dynamic environments, a composite model based on CNN-GAA is proposed by combining the advantages of CNN and GAA for the prediction of stock price trends and optimization of securities trading processes. It is expected to improve the accuracy of predicting stock price trends and optimizing securities trading, and provide high reliability technical support and trading decisions for society on a continuous basis. Compared with traditional methods, the existing work focuses on improving a single prediction module and integrating trend prediction and trading decisions into an end-to-end learning framework. The proposed model combines gated attention mechanism with state adaptive structure, enhancing real-time responsiveness to the market. The security of the model lies in quantifying uncertainty issues and controlling overfitting data, enabling real-time monitoring of position control and market pressure. The contribution of the research is mainly reflected in the following three aspects: firstly, proposing a stock price trend prediction algorithm based on CNN-GAA. The second is to construct an innovative SSCGL securities trading optimization model. The third is that the proposed model has demonstrated superior performance in predicting stock price trends and optimizing securities trading in comparative experiments, and can be applied in the fields of stock price trends and securities trading optimization. The structure of the study is arranged as follows:

The first section elaborates on the research background. The second section introduces the current research status in the field of stock price trend prediction and trading decision optimization, analyzes the application scenarios and limitations of convolutional neural networks (CNN) and generalized adaptive algorithms (GAA), and clarifies the research objectives. The third section provides a detailed explanation of the design concept of the stock price trend prediction algorithm and the construction process of the trading optimization model, ultimately forming a complete SSCGL model. The fourth section empirically tests the performance of stock price trend prediction and evaluates the optimization effect of securities trading, fully demonstrating the superiority of the proposed algorithm and model. The fifth section deeply analyzes the reasons for the excellent performance of the SSCGL model, and verifies the rationality and effectiveness of the model through comparison with relevant research in the field. The sixth section summarizes the overall framework and advantageous features of the model, pointing out the current limitations and future improvement directions.

2. Related Works

Predicting correct stock price trends can bring significant economic benefits to investors and businesses, while also serving as the foundation for automated trading. Liu and other scholars proposed a new learning framework based on excavator guides to make stock data representation clearer. This framework extracts features from high-frequency data, captures global trends, overcomes the influence of noise, and significantly improves the predictive performance of stock trends and understanding of financial markets [11]. Patel et al. proposed a prediction method that combines time convolution and linear models to improve the accuracy of predicting future stock prices. This method eliminates instability by driving data through random forest feature permutation, and then utilizes convolutional structures and linear models to analyze irregular trend components, thereby improving profits [14]. In order to improve the problem of high-frequency signal influence in securities trading, Yuan et al. proposed an online adaptive sequence learning framework. This framework learns to predict various high-frequency signal types of securities, and then uses an encoder to understand the market price curve, quantify market conditions, and improve the trading process [28]. Corazza [5] proposed an automated trading system based on simultaneously optimizing three types of parameters for optimizing financial transactions. This method optimizes financial transactions by only changing trading indicators and rules, without changing other parameters, and introducing a particle swarm optimization metaheuristic method to determine the optimal trading signal [5].

Due to the parameter sharing mechanism and local extraction characteristics of CNN, many scholars have conducted in-depth research on it. Nourbakhsh et al. proposed a CNN based method to improve the predictive performance of stock price trends. This method combines CNN and long short-term memory networks to predict fundamental factors such as a company's profitability. The experimental results show that the average absolute percentage error predicted by this method is 0.49 [13]. O. Researchers such as Rubasinghe proposed a hybrid convolutional neural network and long short-term memory model to predict monthly peak load. This model solves the problem of overfitting and improves prediction accuracy through sequence to sequence feature extraction [15]. Because GAA emphasizes the importance of core information in sequences and improves the reliability of algorithm output results, many scholars have studied it. W. Zhou et al. proposed a fusion strategy of embedded control gates combined with GAA to segment dynamic driving scenarios. This strategy uses the second-generation algorithm and encoder of mobile networks for semantic segmentation, and then utilizes attention residual learning strategy to restore image resolution and complete segmentation [29]. Li et al. proposed a gated transient fluctuation dual attention unit network to enhance the maintenance capability of rotating machinery. This method works collaboratively through three gating mechanisms, emphasizing target state information, adaptively adjusting the current state, and improving the predictive performance of the algorithm [10].

In summary, although existing research has made some progress in stock price prediction and securities trading, further research is needed on the prediction accuracy under dynamic conditions. Therefore, a CNN-GAA based model was proposed for stock price prediction and securities trading optimization, aiming to improve market trading efficiency.

3. Methods

3.1. Design of Stock Price Trend Prediction Algorithm Based on CNN-GAA

To enhance the precision of stock price trend forecasting and the optimization of trading strategies, this paper introduces an intelligent algorithm [26]. As a deep learning network model, CNN possesses powerful ability in detecting local regions and spatial hierarchies, with high precision and stability. Meanwhile, GAA computes probabilistic state attribution iteratively, identifies trend directions unambiguously, and uses its dynamic attractor finding mechanism to enhance market interpretation, making trend representation smarter and prediction more accurate [7]. Therefore, this study combines CNN and GAA to enhance strategy generalization and predictive capability and proposes a CNN-GAA algorithm for stock price trend prediction. GAA focuses dynamically on key points such as attractors and makes decisions to facilitate the prediction of trends in stock prices. The input of the CNN algorithm is converted into a feature map sequence by setting the stock price data of the opening, highest, lowest, and closing prices of 20 trading day windows, and then using a sliding convolution kernel to determine the overall trend based on the combination of feature map shapes and technical indicators such as trading volume. CNN's feature map position creation process is represented in Equation (1).

$$y_{i,j} = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_d-1} K_{m,n} \cdot I_{i+m,j+n} + b \,. \tag{1}$$

In Equation (1), I represents the input feature map, K represents the convolution kernel, b represents the bias term, k_h and k_d represent the width and height of the convolution kernel, and m and n represent internal indices of the kernel. The activation function used in CNN is shown in Equation (2).

$$f(x) = \max(0, x) \,. \tag{2}$$

In Equation (2), x represents the result of the convolution operation. The computation of the CNN pooling layer is shown in Equation (3).

$$MaxPooling(x)_{i,j} = \max_{p=1}^{n} \max_{q=1}^{n} x_{i+p-1,j+p-1} \,. \tag{3}$$

In Equation (3), n represents the pooling window size, p and q represent the indices of the pooling window, and $x_{i+p-1,j+p-1}$ represents each element in the pooling window of the input feature map x. The computation of the final fully connected layer is presented in Equation (4).

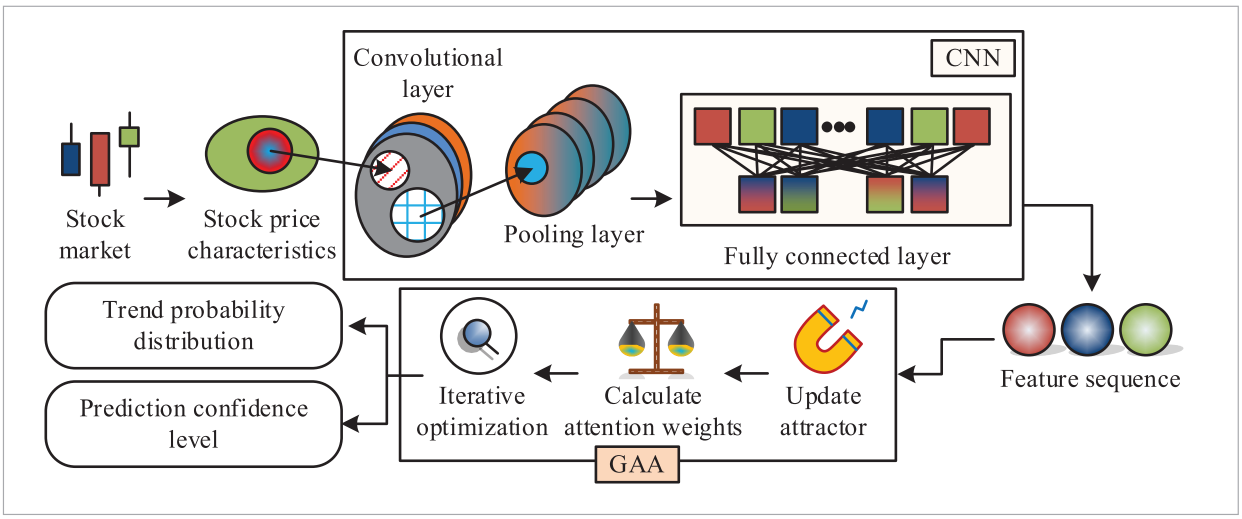
$$y = Wx + b \,. \tag{4}$$

In Equation (4), W is the weight matrix. Combining CNN and GAA enhances the algorithm's sensitivity to stock price changes, improving prediction accuracy and providing more reliable trend results. The workflow is shown in Figure 1.

As shown in Figure 1, CNN-GAA inputs historical stock price data and transforms it into multidimensional time series matrices, which are processed in the CNN module. The CNN module extracts local features from the time sequences. By sliding convolutional kernels of different widths along the time axis, new features are generated, processed through activation functions, and then input to the pooling layer. Pooling reduces feature dimensions while retaining key characteristics, which are then analyzed in the fully connected layer to produce refined feature maps containing local patterns.

Figure 1

CNN-GAA prediction flowchart.



The feature maps are processed in the GAA module using attractors and self-attention mechanisms to compute long-term dependencies and weighted sums. The GAA module uses a gated structure to control the output of the attention layer, iteratively optimizing and updating feature states to generate the predicted stock price trends. CNN-GAA applies the Sigmoid activation function to handle nonlinear data, facilitating gradient descent, as shown in Equation (5).

$$Sigmoid(x) = \frac{1}{1+e^{-x}}. \tag{5}$$

In Equation (5), x represents the input, and e represents the base of the natural logarithm. The algorithm also introduces the Gaussian Error Linear Unit activation function to achieve regularization, stabilizing the computation process, as shown in Equation (6) [18].

$$GELU(x) = 0.5x(1+\tanh(\sqrt{\frac{2}{\pi}}(x+0.044715x^3))). \tag{6}$$

Since CNN-GAA may face gradient vanishing issues in trend prediction, which can reduce accuracy, the LSTM algorithm, with its chain structure, allows information to flow continuously forward, effectively addressing gradient vanishing and enhancing attention to long-term dependencies [8]. The optimization process is shown in Figure 2.

As shown in Figure 2, the LSTM algorithm preprocesses historical stock price data and inputs it into the program for temporal dependency modeling in a two-layer stacked memory neural network. The network contains 128 hidden units in each layer and uses a dropout mechanism of 0.2 between layers to prevent overfitting. Information flows through three gates: the forget gate, input gate, and output gate. The forget gate controls which information is removed from the cell state, the input gate selects what new data to store, and the output gate determines the hidden state that represents short-term memory. The gates work collaboratively to update cell states, flowing vertically and horizontally, continuously updating feature data, and producing hidden state outputs that encode all historical information of the sequence [21]. The process of inputting features into memory cells in LSTM is shown in Equation (7).

$$i_t = \sigma(W_x \cdot [h_{t-1}, X_t] + b_i). \tag{7}$$

In Equation (7), W_x is the input gate weight matrix, and b_i is the bias term. The forget gate computation is shown in Equation (8).

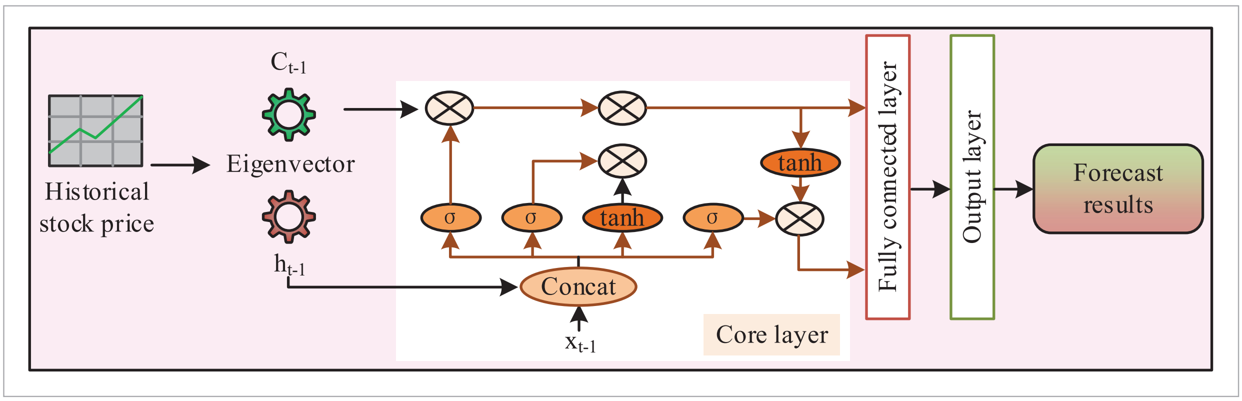
$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f). \tag{8}$$

In Equation (8), W_f represents the forget gate weight matrix, and b_f is the bias term. The candidate update value of the memory cell, combining input and forget gates, is shown in Equation (9).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t. \tag{9}$$

Figure 2

Optimization process of LSTM algorithm.



In Equation (9), $\tilde{C}_t$ represents the candidate value for the new memory cell, W_c represents the weight matrix for updating the memory cell. The updated memory cell computation is shown in Equation (10).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \,. \tag{10}$$

In Equation (10), C_t is the cell state, which contains the algorithm's memory. Leveraging LSTM's ability to handle long sequences, the prediction sensitivity of the algorithm is greatly enhanced. Therefore, LSTM is applied to optimize CNN-GAA, improving overall stock price prediction accuracy and data processing efficiency, forming the CNN-GAA-LSTM algorithm, abbreviated as CGL. The structure of CGL for predicting stock price trends is shown in Figure 3.

As shown in Figure 3, CGL extracts features from historical stock price data and inputs them into the CNN module for refinement. In CNN, features undergo convolution to capture patterns at different time scales. Nonlinear activation functions prevent overfitting, followed by pooling to generate high-level feature maps retaining main characteristics. The feature maps are analyzed by the self-attention mechanism in the GAA module. While retaining temporal structure, features are converted into vectors. Gated mechanisms enhance nonlinearity, suppress irrelevant factors, and weighted summation produces sequences input to the LSTM module. LSTM maps the data vectors, establishes long-term dependencies, and outputs data with dimensions matching the number of classes. The output layer transforms this into a probability distribution, selecting the highest probability as the final predicted trend. The process of extracting information from memory cells in CGL is shown in Equation (11).

$$\begin{cases} o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \\ h_t = o_t \cdot \tanh(C_t) \end{cases} . \tag{11}$$

In Equation (11), o_t represents the value of the output gate at t time, W_o represents the output gate weight matrix, b_o represents the bias term, $\tanh(C_t)$ represents the activated value of the memory cell tanh, and h_t represents the hidden state at t time.

3.2. Construction of Trend Prediction and Securities Trading Optimization Model

In securities trading, providing correct behavioral decisions is crucial. However, trading processes face issues such as signal fluctuations and information delays. In order to further improve the efficiency and effectiveness of the securities trading process, increase investment returns, control risks, and maintain market stability, a securities trading optimization model based on the CGL algorithm was proposed. The Squeeze-and-Excitation Networks (SE-Net) algorithm can improve the model's feature expression effect by enhancing adaptability and

Figure 3

Schematic diagram of the structure of the CGL algorithm.

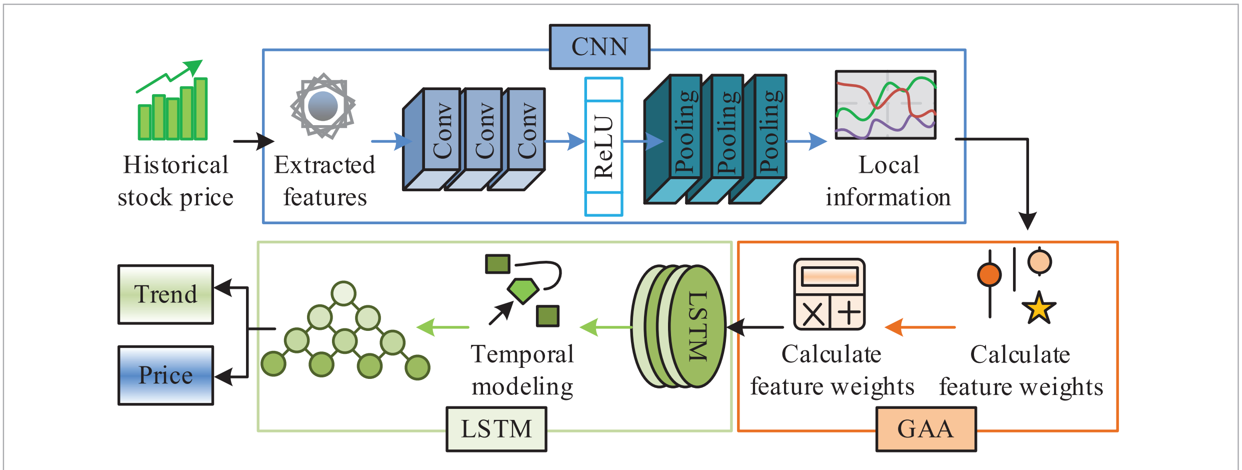
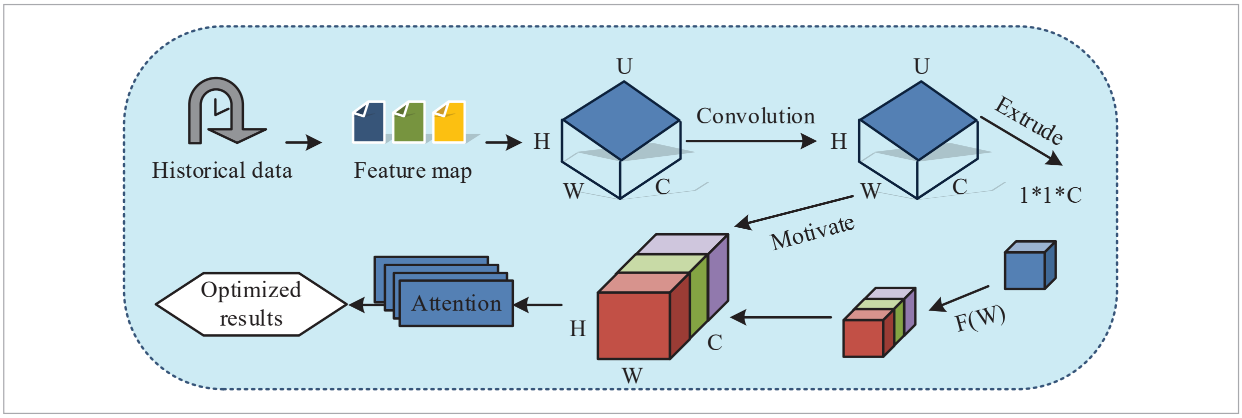


Figure 4

Transaction optimization process of SE-Net algorithm.



adjusting the activation values of feature channels in real time. The independent convolutional neural network structure of this algorithm can reduce the algorithm's computing resources and improve overall operating efficiency [20, 23]. In addition, the Sequential Monte Carlo Methods (SMCM) has the ability to process complex nonlinear data, capture the complex situation of multiple peaks, provide a more complete probability distribution, and quantify uncertainty information [30]. Therefore, this study combines SE-Net and SMCM to form SE-Net-SMCM, achieving dimensionality reduction and market risk control. The SE-Net algorithm process for securities optimization is shown in Figure 4.

As shown in Figure 4, SE-Net preprocesses historical financial data and extracts initial features. The initial features, as a set of feature channel numbers, are input into the compression stage of the algorithm. In the compression stage, convolution operations compress each channel's time series along the time dimension, generating global feature vectors that capture the overall distribution and importance of each channel over the time series. In the excitation stage, feature vectors are reduced in dimension through the first fully connected layer and then restored to higher dimension through the second fully connected layer, mapping the vectors to a unified space. Finally, through weight learning and summation, important feature channels are enhanced while irrelevant channels are suppressed, outputting the final trading decision and completing the optimization [2]. The research defines the trading objective as maximizing the accumulation of expected returns while controlling dynamic risks, and mathematically expressing it through a reward at time t (Rt) function composed of optimal current returns and risk penalty terms. The process of extracting and compressing channel information in SE-Net is shown in Equation (12).

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{w} u_c(i, j) . \tag{12}$$

In Equation (12), Z represents the global attention data for each channel feature, W and H represent the width and height of the feature map, c represents the number of channels, and u represents the result after convolution. The excitation network adaptively learns each channel and generates corresponding weight vectors through fully connected layers and activation functions, as shown in Equation (13).

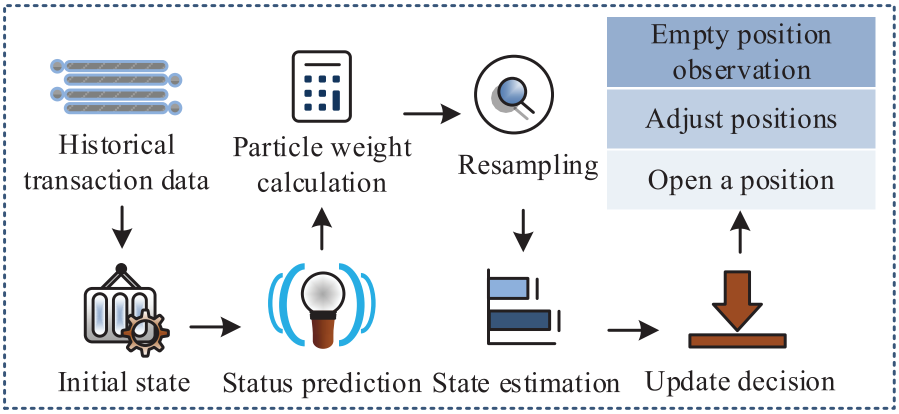
$$s = \sigma(W_2 \delta(W_1 Z)) . \tag{13}$$

In Equation (13), s represents the weight corresponding to each feature map, δ is the ReLU activation function, and W_2 and W_1 are parameters of two different fully connected layers. The SMCM simulates real-world data patterns by repeated sampling within a specific sample. The process of SMCM optimizing securities trading is shown in Figure 5.

As shown in Figure 5, the SMCM algorithm initializes historical financial data and establishes a particle swarm that includes hidden market states. Each

Figure 5

The process of optimizing securities trading using the SMCM.



particle represents a hypothesis about the complete market and shows the changes in market states over time. It is mainly improved by transforming deterministic judgments into probabilistic dynamic strategies. Logs calculate and update weights based on observations, further evaluating market information. According to the weight distribution, particles are sequentially resampled to form a new particle set, effectively preventing particle degeneration and displaying the posterior distribution of the current state. The algorithm estimates particle states, searches for the state with maximum likelihood, and outputs final trading decisions and signals. The importance weight computation in SMCM is shown in Equation (14).

$$w(x) = \frac{f(x)}{g(x)}. \tag{14}$$

In Equation (14), $w(x)$ represents importance weights, $g(x)$ represents the easily sampled distribution function, and $f(x)$ represents the density function. The diagnostic process for measuring effective sample size is shown in Equation (15).

$$ESS = \frac{(\sum w_i)^2}{\sum w_i^2}. \tag{15}$$

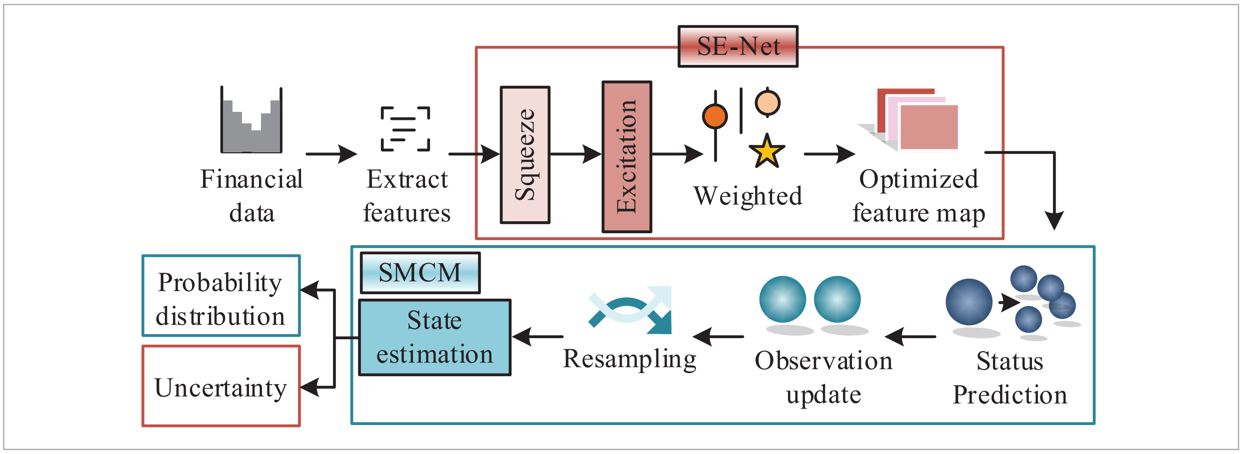
In Equation (15), ESS represents normalized weights, which approximate the target value. SE-Net suppresses background noise, focusing the program on target states, while SMCM adaptively tracks states and quantifies uncertainty. Combining the two produces complete trading probability distributions and controls market risks, providing reliable trading decisions [3]. The SE-Net-SMCM algorithm optimizing securities trading is shown in Figure 6.

As shown in Figure 6, SE-Net-SMCM captures initial features and uses them as input to SE-Net for inspection. Features go through the compression phase and pooling, where global information is used and new features that reflect global activation strength are produced. After the excitation phase and adaptive recalibration, a weight vector with a number of channels is generated, which signifies the relevance at an instant. The addition of the weight vector results in feature maps with the best outcomes, which are input to SMCM. The SMCM makes an estimate of state for the vectors, observes weights to determine them, and updates vector states. SMCM re-samples the set of particles in accordance with weights and gives the best state estimate of the market now. The algorithm calculates the mean of the set of particles, estimates uncertainty, and generates the final trading signal, concluding the securities trading optimization. Process for the enhancement of feature utilization in SE-Net-SMCM is captured through Equation (16).

$$X_c = s_c \cdot u_c. \tag{16}$$

Figure 6

Operation process of SE-Net-SMCM algorithm.

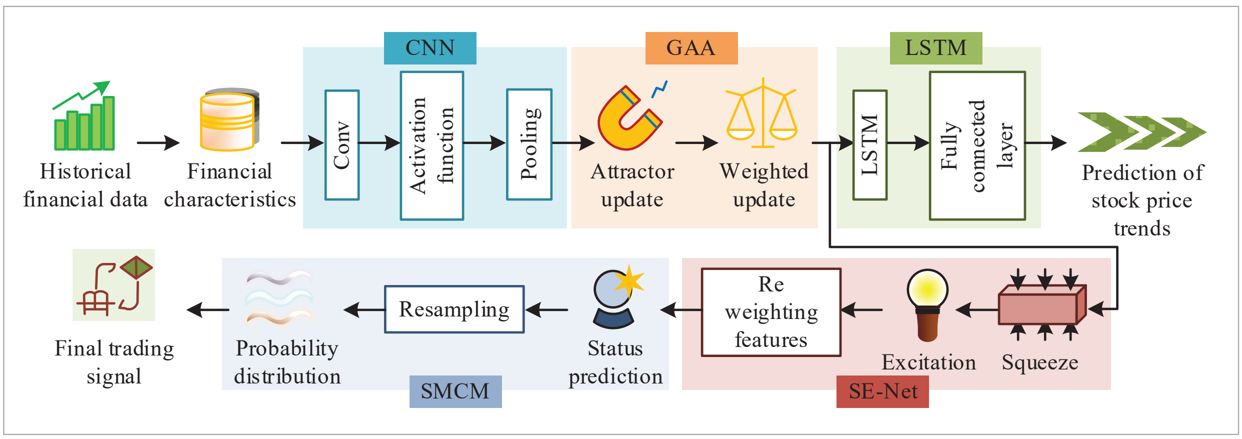


In Equation (16), S_c represents the weight vector, and X_c represents the final output feature map of the c-th channel. By integrating SE-Net-SMCM with the CGL algorithm, we developed the final securities trading optimization model named SE-Net-SMCM-CNN-GAA-LSTM, abbreviated as SSCGL model. This model accurately predicts stock price trends and optimizes securities trading. The application process of SSCGL is shown in Figure 7.

As shown in Figure 7, SSCGL inputs historical financial data and extracts initial features. The features first enter the CNN module, which extracts local patterns through sliding convolution kernels. After activation, features enter the pooling layer for analysis. The analyzed features enter the GAA module, where the gated structure and self-attention mechanism perform weighted summation, producing feature vectors containing the sequence's core data and condensed contextual information. The feature vectors are processed by two programs: stock price trend prediction and securities trading optimization. In trend prediction, vectors enter the LSTM module to construct long-term dependencies, recognize complex data patterns, and output trend probabilities. In securities trading optimization, vectors pass through the channel attention mechanism for global average pooling and weight learning, enhancing important features and suppressing irrelevant ones. Weighted features then enter the reinforcement learning module, where the dynamic strategy network evalu-

Figure 7

Workflow of the SSCGL model.



ates long-term returns for trading actions, enabling intelligent buy/sell decisions. During training, a multi-task loss function simultaneously optimizes prediction accuracy and trading returns, improving overall performance. Finally, weighted features enter the SMCM module, where state prediction and resampling quantify uncertainty and output final trades, providing higher-confidence trading decisions. Among them, the reinforcement learning module introduced in the study is a deep reinforcement learning framework based on policy gradient, which maximizes the cumulative return of securities trading as the reward function. Through the action space of buying, holding, and selling, as well as the state space of stock price characteristics and market state characteristics, the optimization of trading decisions is achieved, as shown in Equation (17).

(17)

In Equation (17), R_t represents the reward function, r_t represents the transient time benefit, λ represents the transaction cost

4. Results

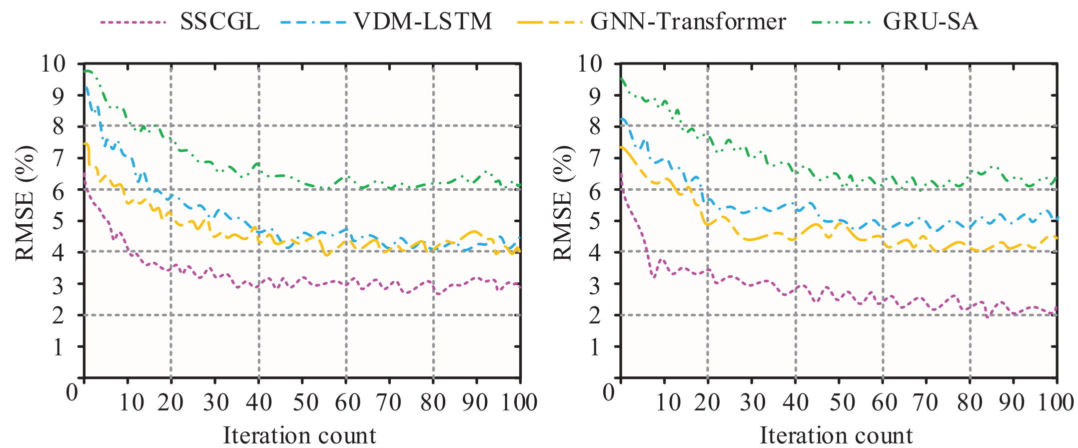
4.1. Performance Validation of Stock Price Trend Prediction for SSCGL Model

To evaluate the prediction performance of the SSCGL model for stock price trend directions, the study compared it with three models: the VDM-LSTM model combining Variational Dropout Mechanism (VDM) and LSTM, the GNN-Transformer model integrating Graph Neural Network (GNN) and Transformer, and the GRU-SA model incorporating Self Attention Mechanism (SA) to improve the Gated Recurrent Unit (GRU) algorithm. The experimental setup was configured as follows: the computer ran Ubuntu 16.04 LTS operating system, equipped with an AMD Ryzen 7 6800H processor and NVIDIA GeForce RTX 3050 graphics card, along with 24GB RAM. The deep learning framework employed PyTorch 1.9.0, with Python 3.7.6 as the programming language. The training parameters included 100 iterations and an initial learning rate of 0.001. The experimental datasets adopted daily closing prices of stock indices from China and the United States, specifically the Shanghai Stock Exchange Composite Index (SSE) and the Standard & Poor's 500 Index (S&P 500). The stock price trend prediction study adopts a binary classification task, with the core objective being to predict whether stock prices will rise or fall the following day. Regression methods are employed to model price volatility amplitude. Evaluation metrics for classification tasks include confusion matrices, accuracy, recall, and F1 score, while root mean square error and coefficient of determination serve as auxiliary indicators for regression performance. The integration of these dual metrics provides a comprehensive assessment of trend prediction effectiveness. The conversion between classification and regression tasks is achieved through thresholding processing, using the next-day price return rate obtained from regression fitting as an intermediate variable. When the return rate exceeds 0, it is labeled as the "upward" category; when it is ≤ 0, it is labeled as the "downward" category. This process transforms continuous values into discrete categories, ensuring consistency between the two types of tasks. The research defines the direction of stock price trend as the short-term trend of the next day, which can represent the upward or downward direction of the closing price of a single trading day relative to the closing price of the previous trading day. The prediction period is set to one trading day, that is, based on the historical characteristics set at a certain time and before, the stock price upward or downward trend of the next time set at a certain time is predicted. The study conducted experimental evaluations of prediction accuracy for the SSCGL model against VDM-LSTM model, GNN-Transformer model, and GRU-SA model on both the SSE dataset and S&P 500 dataset, using Root Mean Square Error (RMSE) as the performance metric. The experimental results are presented in Figure 8.

In Figure 8(a), the RMSE of prediction accuracy for the SSCGL model on the SSE dataset after 100 iterations was 3.93%. The GRU-SA model had the highest RMSE value for prediction accuracy, which increased by 2.26% compared to the proposed model, with an RMSE of 6.19%. In Figure 8(b), the RMSE for the proposed model's prediction accuracy experiment on the S&P 500 dataset was 2.98%, which was 3.43% lower than the GRU-SA model and also lower than the other two models. From the above results,

Figure 8

Comparison of RMSE results of prediction accuracy.



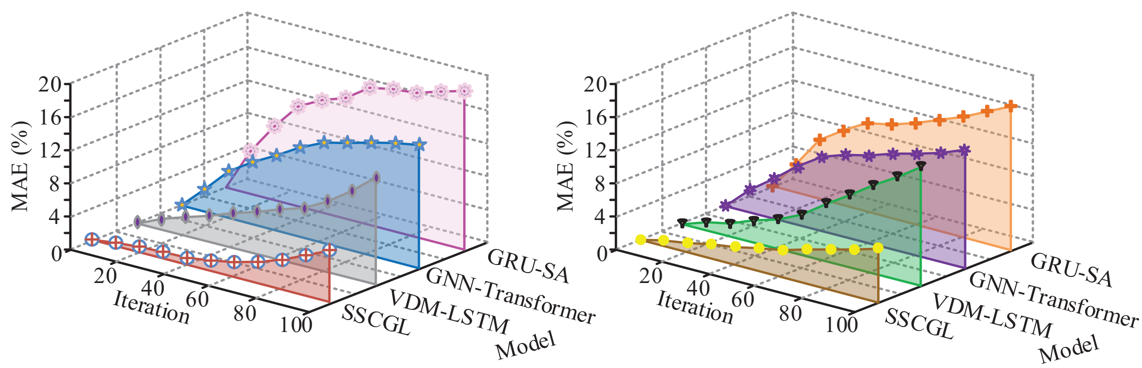
(a) SSE (b) S&P 500

the proposed model was able to control errors, suppress the influence of noise factors such as background interference, and predict stock price trends more accurately. It possessed excellent prediction performance and could provide more reliable result data. To further explore the advantages of the proposed model in predicting stock price trend directions, the study tested it with the three comparison models on the SSE dataset and S&P 500 dataset, using cumulative mean absolute error of prediction accuracy as the analysis indicator. The test results are shown in Figure 9.

As shown in Figure 9(a), the cumulative mean absolute error of the proposed model on the SSE dataset after 100 iterations of prediction accuracy testing was 6.07%, significantly lower than the other three models, decreasing by 7.17%, 9.84%, and 12.68% respectively compared to the other three models. In Figure 9(b), the cumulative mean absolute error of the proposed model on the S&P 500 dataset was 6.45%, also superior to the comparison models, decreasing by 13.43% compared to the GRU-SA model. From the above results, it could be seen that the proposed model demonstrated advantages in cumu-

Figure 9

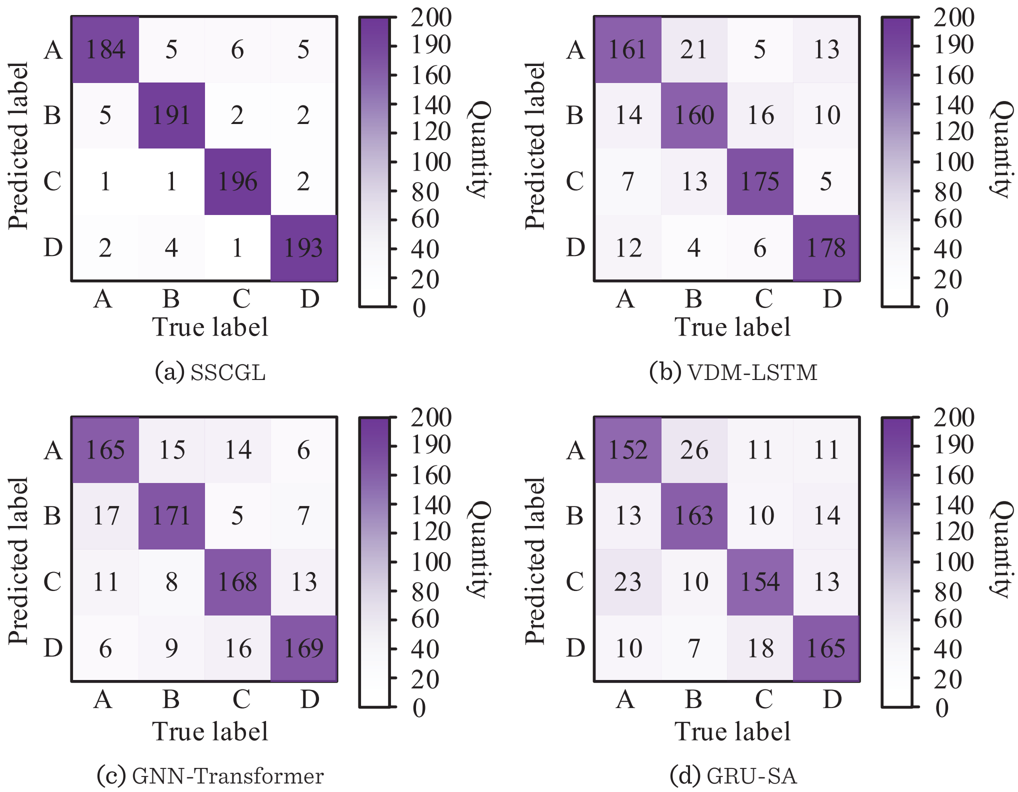
Comparison of cumulative mean absolute error.



(a) SSE (b) S&P 500

Figure 10

Confusion matrix results of prediction accuracy.



lative mean absolute error testing on both datasets, proving that this model possessed excellent generalization performance, could adapt to financial data from different stock markets, perform more accurate data analysis, and provide more reliable support for stock price trend direction prediction. To verify that the proposed model possessed excellent stock price prediction performance, the study tested it with the above three comparison models on the SSE dataset, using prediction accuracy as the metric and displaying the data in confusion matrix format. The test results are shown in Figure 10.

In Figure 10(a), the prediction accuracy of the proposed model on the confusion matrix was 95.5%. In Figures 10(b), (c), and (d), the prediction accuracy of the VDM-LSTM model was 84.25%, while the prediction accuracies of the GNN-Transformer model and GRU-SA model decreased by 5.12% and 10% respectively compared to the proposed model. From the above data results, it could be seen that the stock price trend prediction performance of the proposed model was significantly superior to the other three comparison models. While controlling error occurrence, it achieved high-accuracy prediction, proving that this model could still maintain excellent stability and adaptability when facing complex market environments, could effectively capture nonlinear data features of market environments and identify key trend signals, thereby improving prediction performance accuracy. To demonstrate the superiority of the SSCGL model's stock price trend prediction performance, the study experimented with it and the three comparison models on the S&P 500 dataset and SSE dataset, analyzing model performance through prediction precision, F1 score, recall rate, and The Coefficient of Determination (R^2). The experimental results are shown in Table 1.

Table 1
Comparison of results of multiple comprehensive indicators.

Dataset	Model	Precision	F1 (%)	Recall (%)	R2
S&P 500	SSCGL	0.975	96.38	97.11	0.715
	VDM-LSTM	0.899	91.06	91.43	0.531
	GNN-Transformer	0.916	90.08	91.23	0.529
	GRU-SA	0.872	86.58	85.76	0.478
SSE	SSCGL	0.963	95.40	95.21	0.709
	VDM-LSTM	0.874	86.35	85.49	0.508
	GNN-Transformer	0.869	86.08	85.25	0.506
	GRU-SA	0.838	82.95	83.47	0.413

In Table 1, the precision of the proposed model on both the S&P 500 and SSE datasets reached above 0.95. Its F1 score increased by 9.84% and 12.45% respectively compared to the GRU-SA model, the recall rate was also above 95%, and the R^2 coefficients reached 0.715 and 0.709, significantly higher than the comparison models. The experimental results demonstrated that the SSCGL model could maintain excellent prediction capability in different market environments, achieving an effective balance between high precision and high recall rate. This also proved that the proposed model, benefiting from the attention mechanism, could dynamically focus on key feature dimensions, effectively enhancing the model's ability to identify stock price data trend changes and its robustness and generalization performance against external factors.

4.2. Performance Evaluation of Securities Trading Optimization for SSCGL Model

After verifying the performance of the SSCGL model in predicting stock price trend directions, the study further analyzed the securities trading optimization performance of the SSCGL model. The study compared it with the Level 2 Limit Order Book (L2VLOB) model, the Volume-Weighted Average Price (VWAP) model, and the Black-Litterman Model (BLM) respectively. The experimental environment was set up as follows: the computer version was Microsoft Windows 10, the operating system was Ubuntu 16.04 LTS, the processor was Intel E5-2678, the CPU model was r7-6800H, the GPU model was GeForce RTX 3050, the graphics card model was NVIDIA GTX 1080Ti, the running memory was 64GB, the deep learning framework used was Pytorch 1.9.0, the programming language was Python 3.7.6. The experimental datasets adopted stock data from the AK stock data sharing library and Yahoo Finance as datasets, as well as a self-made dataset. The self-made dataset consisted of Midea Group stock data, with the time period set from January 2024 to December 2025. The study experimented with the SSCGL model, L2VLOB model, VWAP model, and BLM model on the Yahoo Finance dataset and AK stock data sharing library dataset, using the Sharpe ratio as the indicator to judge optimization performance. The experimental results are shown in Figure 11.

As shown in Figure 11(a), the Sharpe ratio of the SSCGL model on the Yahoo Finance dataset after 100 iterations reached 1.812, which was 0.133 higher than the L2VLOB model. The VWAP model and BLM model decreased by 0.501 and 0.623 respectively compared to the proposed model. In Figure 11(b), the Sharpe ratio of the proposed model on the AK stock data sharing library dataset reached 1.894, which was superior to the L2VLOB model and VWAP model by 0.240 and 0.478 respectively. From the above data results, it could be seen that the proposed model demonstrated better risk-adjusted return capability on both datasets and exhibited strong stability when facing different market environments, proving that this model could obtain considerable return income through lower risk exposure and was a robust model with excellent risk

Figure 11

Comparison of experimental results of Sharpe ratio.

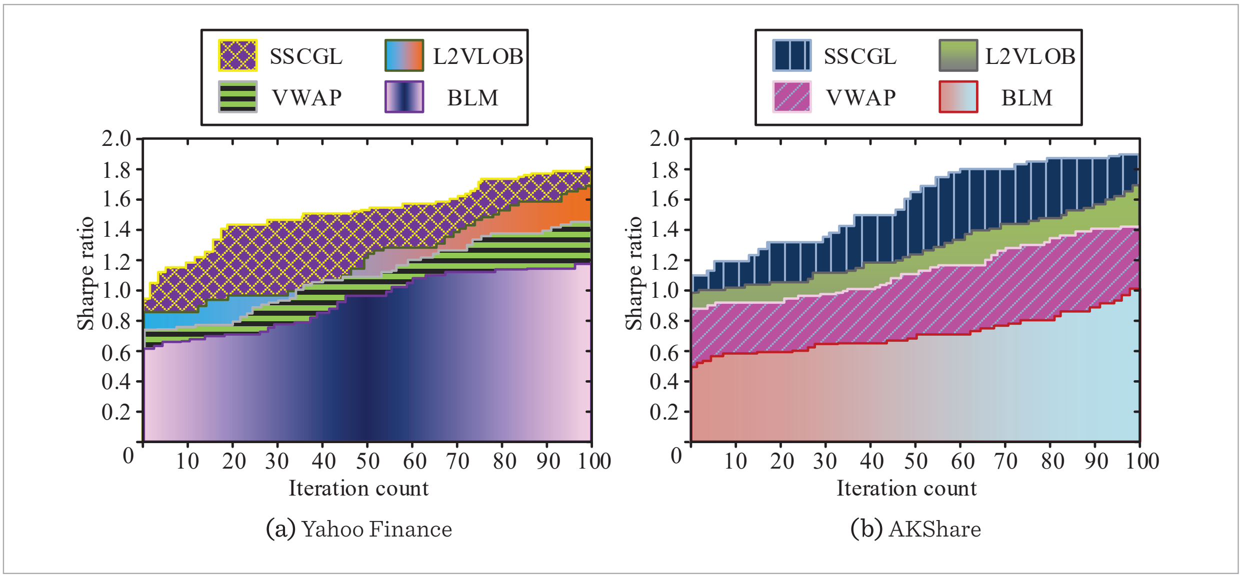
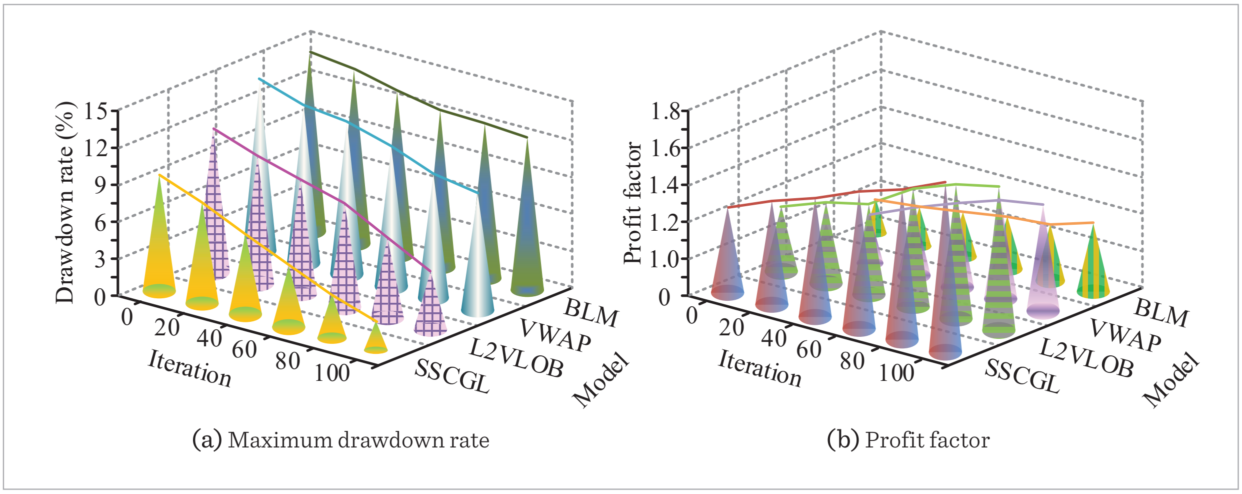


Figure 12

Comparison of maximum drawdown rate and profit factor results.

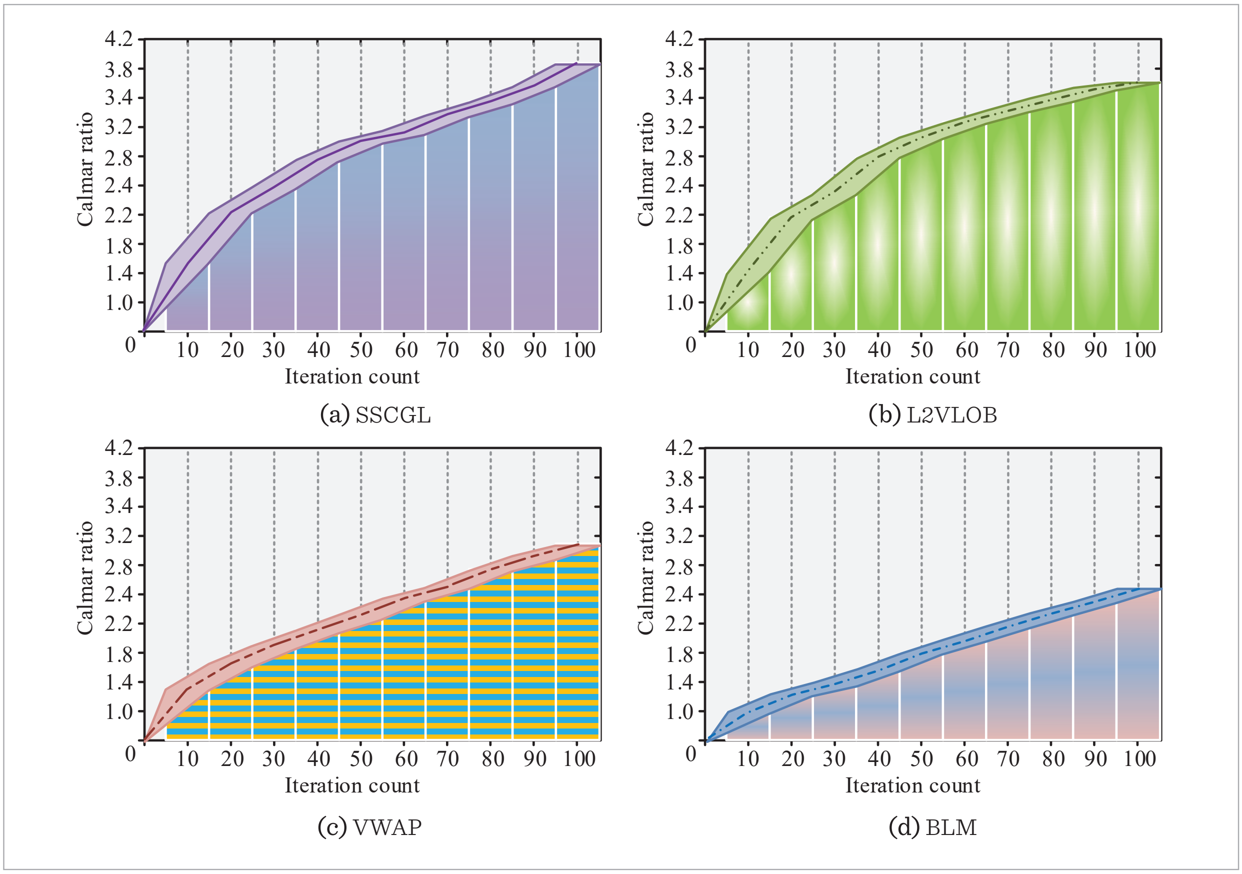


control capability and profitability. To comprehensively assess the optimization performance of the proposed model in securities trading, the study tested it with the three comparison models on the AK stock data sharing library dataset, using maximum drawdown rate and profit factor as judgment indicators. The test results are shown in Figure 12.

As shown in Figure 12(a), the maximum drawdown rate indicator of the proposed model after 100 iterations reached 2.51%, which decreased by 1.99% compared to the L2VLOB model. In Figure 12(b), the profit factor of the SSCGL model after 100 iterations was 1.738. Among the four models, the BLM model had the lowest profit factor of only 1.193, and SSCGL improved by 0.545 compared to BLM. From the above results, the proposed model was significantly superior to the comparison models in maximum drawdown control, representing that the model had stronger capability in controlling risk and preserving capital. Additionally, a higher profit

Figure 13

Comparison of Calmar ratio test results.



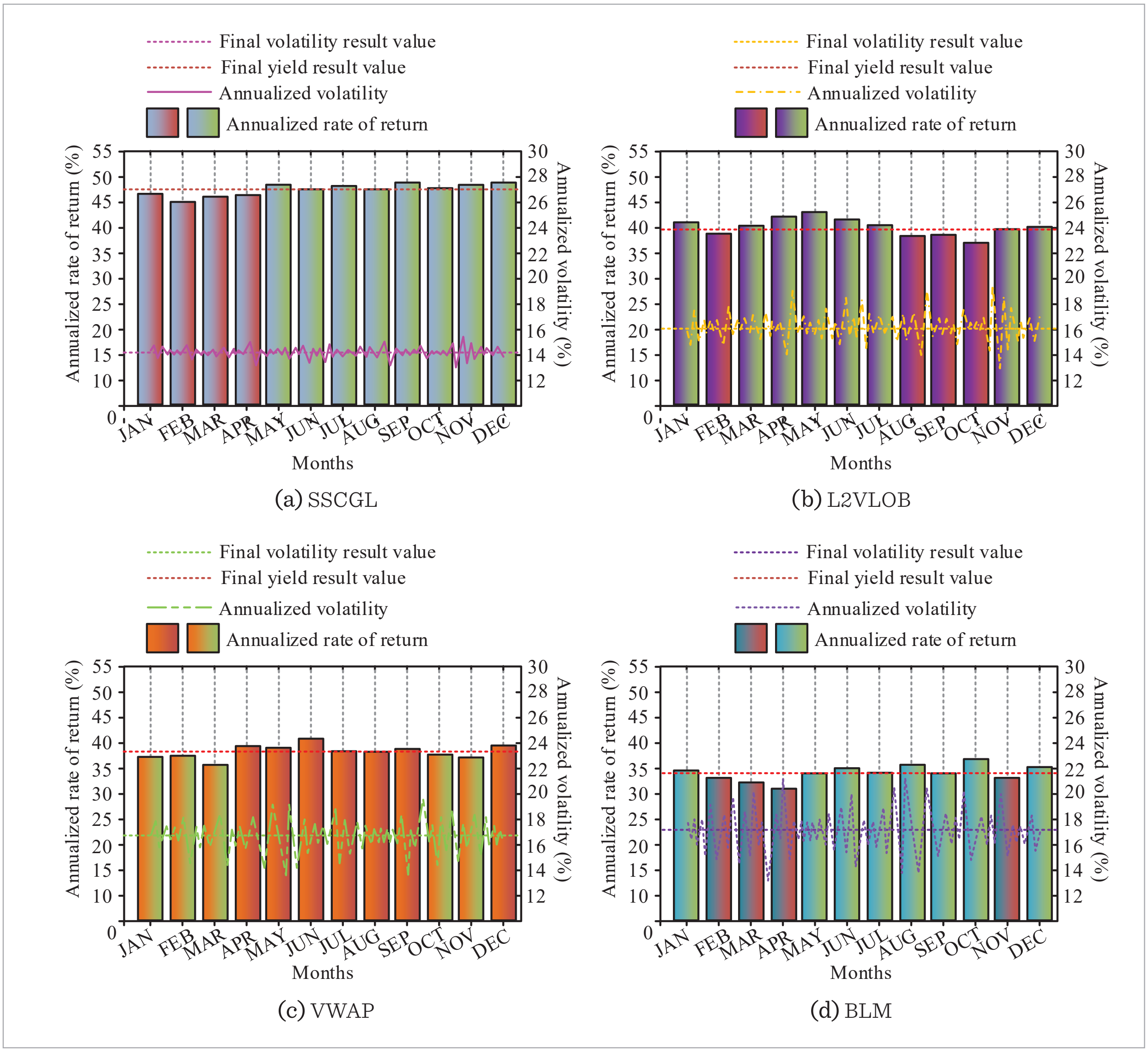
factor proved that the proposed model had stronger profitability in the securities trading process, reflecting that this model possessed excellent risk control capability and capital drawdown management capability, with more superior optimization performance. To further verify the advantages of the proposed model in optimizing the securities trading process, the study tested it with the above three comparison models on the Yahoo Finance dataset, using the Calmar ratio as the analysis indicator. The test results are shown in Figure 13.

In Figure 13(a), the Calmar ratio of the proposed model on the Yahoo Finance dataset after 100 iterations reached 3.871, significantly higher than the other three comparison models. In Figures 13(b)-(d), the Calmar ratios of the L2VLOB, BLM, and VWAP models were 3.625, 3.151, and 2.492 respectively. The L2VLOB model performed best, but its Calmar ratio still decreased by 0.246 compared to the proposed model. From the above results, when facing drawdown situations, the proposed model could still rapidly restore returns to previous highs, proving that this model's optimization of the securities trading process was more excellent, enabling it to achieve outstanding risk-adjusted return performance, provide enterprises with higher-confidence trading signals and decisions, and effectively improve capital utilization efficiency. To verify the securities trading optimization performance of the proposed model, the study experimented with it and the L2VLOB model, VWAP model, and BLM model on the self-made dataset, analyzing the optimization performance of each model through two indicators: annualized volatility and annualized return rate. The results are shown in Figure 14.

As shown in Figure 14(a), the annualized return rate obtained by the proposed model tested on the self-made dataset was 46.11%, the annualized volatility

Figure 14

Comparison of annualized volatility and annualized rate of return.



was 14.23%, and the fluctuation degree of the volatility curve was relatively small. In Figures 14(b), (c), and (d), the L2VLOB model and VWAP model decreased by 6.2% and 7.37% respectively relative to the SSCGL model. The BLM model had the worst performance in annualized return rate and annualized volatility, decreasing by 11.59% and increasing by 9.66% compared to the proposed model, and the fluctuation degree of the volatility curve was relatively large. From the above results, it could be seen that the optimization decisions provided by the proposed model under controllable risk could help enterprises obtain returns as high as possible. The low level of volatility meant that the model's prediction performance was good, proving that the model's optimization performance was more superior, significantly improving profitability efficiency, while enhancing the model's applicability in different securities trading markets, which also meant the stability and reliability of the model's optimization performance. To validate the independent contributions of each core module in the model, an ablation

experiment was designed. Using SSCGL as the base model, components such as CNN, GAA, and LSTM were sequentially removed, and the model's prediction accuracy and RMSE were tested on a self-generated dataset. The results are shown in Table 2.

Table 2
Experimental results of model ablation.

Model Variant	Prediction Accuracy (%)	RMSE (%)
Full SSCGL	95.51	3.93
w/o CNN	87.34	6.58
w/o GAA	85.96	7.12
w/o LSTM	84.68	7.54
w/o SE Net	90.29	5.31
w/o SMCM	91.82	4.95

As shown in Table 2, removing any single module resulted in a decline across all evaluation metrics, with the complete SSCGL model achieving results of 95.51% and 3.93% respectively. The most significant performance degradation occurred when GAA or LSTM was removed, indicating that attention mechanisms and long-term dependency mechanisms play a pivotal role in trend prediction. While the impact of removing SENet and SMCM was relatively minor, their contributions to risk control and uncertainty quantification demonstrate the complementary nature of these two modules. The outstanding performance of the SSCGL model stems from effective integration of its components, while also validating the independent validity of each module.

5. Discussion

The SSCGL model proposed in the study demonstrated superior performance in comparative experiments on stock price trend direction prediction performance. In the tests on RMSE and cumulative mean absolute error indicators of prediction accuracy on the SSE dataset, the RMSE of the proposed model was 3.93% and the cumulative mean absolute error was 6.45%, which decreased by 2.26% and 7.17% respectively compared to the GRU-SA model. This was related to the advantages of the SSCGL model in performing feature extraction and establishing long-term dependencies. Through the gating mechanism, it suppressed the influence of noise such as background interference, enhanced important feature expression, and thus achieved the purpose of controlling errors. This was also related to the multimodal network algorithm structure combining LSTM proposed by T. Xiang et al. for monitoring hypertension. Through the memory gate and forget gate of LSTM, the model avoided errors created by irrelevant factors and performed more refined information selection [24]. In terms of prediction accuracy, both the prediction accuracy and precision of the proposed model were above 95%, significantly superior to the other three comparison models. This was because the CNN algorithm could efficiently extract local features, helping the model learn the basic patterns of data and improve data analysis capability. This was similar in principle to the sparse image target detection algorithm structure integrating CNN proposed by P. Sun et al. for improving the accuracy of target detection tasks. Both utilized the convolutional structure of CNN to extract data features, strengthened the capability of capturing information, and achieved improvement in data processing accuracy [16].

In the experiments verifying the optimization performance of the securities trading process, the SSCGL model also demonstrated its excellent optimization capability. In the comparative tests of Sharpe ratio and Calmar ratio indicators, the Sharpe ratio of the proposed model reached 1.812 and the Calmar ratio reached 3.871, improving by 0.133 and 0.246 respectively relative to the L2VLOB model. This was related to the channel attention mechanism in the proposed model. Through dynamic adjustment of weight allocation, the model focused on the parts with the greatest information content. Through excellent prediction capability, it enhanced risk control capability, provided optimized trading decision recommendations, and thus improved investment returns. This result was consistent with the squeeze-and-excitation densely connected convolutional neural network structure proposed by N. V et al. for improving pepper yield. Both enhanced the model's capability to identify and analyze important features through attention mechanisms, thereby achieving the purpose of optimizing trading decision effects [19]. In terms of maximum drawdown rate and profit factor, the maximum drawdown rate of the proposed model was 2.51% and the profit factor was 1.738, and the performance of the other three comparison models was inferior to this model. This was because the collaborative effect of the multi-layer gating structure and attention mechanism of the proposed model effectively suppressed the noise interference impact caused by market fluctuations, accurately captured trading signal changes, and thus completed the process of optimizing securities trading. This was related to the novel spatiotemporal graph attention gated recurrent transformer network proposed by the team led by D. Wu for predicting traffic flow. Through the GAA mechanism dynamically adjusting weight allocation in different market environments, the model adaptively adjusted strategy parameters when facing different risks and drawdown situations, achieving more stable returns [22]. In terms of annualized volatility and annualized return rate on the self-made dataset, the proposed model achieved a balanced state of controlling volatility at a relatively smooth level and returns at a stable level. This was because the SMCM structure of this model dynamically tracked trading signals according to market changes, captured quantifiable uncertainty probability distributions, and provided optimized trading decisions with high confidence for the securities trading process. This mechanism utilized the same principle as the conditional variational autoencoder framework proposed by W. Sun et al. for industrial sensor sampling problems. Through the SMCM structure quantifying uncertainty distribution to assess market risk, the model maintained robust returns even when facing volatility situations [17].

The contributions of the study were mainly reflected in the following three aspects: first, proposing a stock price trend direction prediction algorithm based on CNN-GAA structure; second, constructing an innovative SSCGL securities trading optimization model; third, the proposed model could be successfully applied to predicting stock price trend directions and optimizing securities trading procedures, both demonstrating superior performance. These contributions provided new methods for society to conduct securities trading, could serve as decision support tools for enterprises to make investments, and promoted the joint development of financial information and intelligent algorithms. The proposed model outperforms the comparative methods in terms of processing efficiency and accuracy, and ensures the reliability and security of the data through a multi-level guarantee mechanism.

6. Summary

To more accurately respond to the complex and changing securities market environment, control risk and cost management, and improve long-term effective and stable value-added returns, the study proposed a stock price trend prediction and securities trading optimization model based on CNN-GAA algorithm. This model first extracted local features of stock price time series through CNN, then combined GAA to dynamically weight important features. The processed data by the LSTM mechanism captured long-term dependencies for the stock price trend prediction process. At the same time, the SE-Net-SMCM model comprehensively analyzed and quantified the market uncertainty and provided reliable trading decisions to complete the optimization of the securities trading process. The experimental results show that this model had a good performance of generalization and prediction accuracy in experiments around stock price trend prediction. Although these two datasets represent distinct market environments, the study employed a time-segmented approach to minimize data leakage risks. By implementing non-comprehensive generalization capability evaluation and carefully designing training/test periods, overlapping samples between datasets were eliminated. Additionally, rolling forward validation was applied, where the model was retrained using historical data at fixed intervals to simulate real-world trading conditions. This methodology ensures all reported results represent genuine out-of-sample outcomes. In securities trading optimization tests, this model also had a strong capability of risk control and stable return capability. The processed data from the model could calculate and provide more reliable trading decisions and had more superior optimization performance. The proposed model had also strong adaptability under different performance validations, but it needed to improve the robustness of the model further when it faced sudden situation market environments. Future research is expected to undertake a comprehensive investigation in this field, enhance the model's capability of identifying complex financial data, improve the efficiency of the model's analysis and promote the intelligent development of the enterprises.

References

1. Al-Khasawneh, M. A., Raza, A., Khan, S. U. R., Khan, Z. Stock Market Trend Prediction Using Deep Learning Approach. Computational Economics, 2025, 66(1), 453-484. https://doi.org/10.1007/s10614-024-10714-1
2. Badar, T., Särkkä, S., Zhao, Z., Visala, A. Rao-Blackwellized Particle Filter Using Noise Adaptive Kalman Filter for Fully Mixing State-Space Models. IEEE Transactions on Aerospace and Electronic Systems, 2024, 60(5), 6972-6982. https://doi.org/10.1109/TAES.2024.3409644
3. Bi, Y., Pan, Y., Li, G. A Mixed-Feed Identification Method for Sizer Crusher Based on Squeeze-and-Excitation Residual Network. IEEE Sensors Journal, 2024, 24(13), 21706-21718. https://doi.org/10.1109/JSEN.2024.3400519
4. Chinthamu, N., Karukuri, M. Data Science and Applications. Journal of Data Science and Intelligent Systems, 2024, 1(1), 83-91. https://doi.org/10.47852/bonviewJDSIS3202837
5. Corazza, M., Pizzi, C., Marchioni, A. A Financial Trading System with Optimized Indicator Setting, Trading Rule Definition, and Signal Aggregation Through Particle Swarm Optimization. Computational Management Science, 2024, 21(1), 26-56. https://doi.org/10.1007/s10287-024-00506-1
6. Heydarpour, M., Ghanbari, H., Mohammadi, E., Shavvalpour, S. Robust Portfolio Optimization Using LSTM-Based Stock and Cryptocurrency Price Prediction: An Application of Algorithmic Trading Strategies. Iranian Journal of Accounting, Auditing and Finance, 2025, 9(3), 151-169. DOI: 10.22067/IJAAF.2025.46039.1466
7. Jiang, K., Wang, Q., An, Z., Wang, Z., Zhang, C., Lin, C.-W. Mutual Retinex: Combining Transformer and CNN for Image Enhancement. IEEE Transactions on Emerging Topics in Computational Intelligence, 2024, 8(3), 2240-2252. https://doi.org/10.1109/TETCI.2024.3369321
8. Keshun, Y., Puzhou, W., Yingkui, G. Toward Efficient and Interpretative Rolling Bearing Fault Diagnosis via Quadratic Neural Network with Bi-LSTM. IEEE Internet of Things Journal, 2024, 11(13), 23002-23019. https://doi.org/10.1109/JIOT.2024.3377731
9. Li, J., Chen, W., Liu, Y., Yang, J., Zeng, D., Zhou, Z. Neural Ordinary Differential Equation Networks for Fintech Applications Using Internet of Things. IEEE Internet of Things Journal, 2024, 11(12), 21763-21772. https://doi.org/10.1109/JIOT.2024.3376748
10. Li, S., Zhang, C., Liu, L., Zhang, X. Gated Transient Fluctuation Dual Attention Unit Network for Long-Term Remaining Useful Life Prediction of Rotating Machinery Using IIoT. IEEE Internet of Things Journal, 2024, 11(10), 18593-18604. https://doi.org/10.1109/JIOT.2024.3363837
11. Liu, Y., Xu, C., Hou, M., Liu, W., Bian, J., Liu, Q. Digger-Guider: High-Frequency Factor Extraction for Stock Trend Prediction. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(12), 7973-7985. https://doi.org/10.1109/TKDE.2024.3424475
12. Ma, X., Zheng, F., Tang, D. Identifying the Head-and-Shoulders Pattern Using Financial Key Points and Its Application in Consumer Electronic Stocks. IEEE Transactions on Consumer Electronics, 2024, 70(2), 4945-4954. https://doi.org/10.1109/TCE.2023.3339140
13. Nourbakhsh, Z., Habibi, N. Combining LSTM and CNN Methods and Fundamental Analysis for Stock Price Trend Prediction. Multimedia Tools and Applications, 2023, 82(12), 17769-17799. https://doi.org/10.1007/s11042-022-13963-0
14. Patel, M., Jariwala, K., Chattopadhyay, C. A Hybrid Relational Approach Toward Stock Price Prediction and Profitability. IEEE Transactions on Artificial Intelligence, 2024, 5(11), 5844-5854. https://doi.org/10.1109/TAI.2024.3408129
15. Rubasinghe, O., Zhang, X., Chau, T. K., Chow, Y. H., Fernando, T., Iu, H. H.-C. A Novel Sequence-to-Sequence Data Modelling Based CNN-LSTM Algorithm for Three Years Ahead Monthly Peak Load Forecasting. IEEE Transactions on Power Systems, 2024, 39(1), 1932-1947. https://doi.org/10.1109/TPWRS.2023.3271325
16. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W. Sparse R-CNN: An End-to-End Framework for Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(12), 15650-15664. https://doi.org/10.1109/TPAMI.2023.3292030
17. Sun, W., Xiong, W., Chen, H., Chiplunkar, R., Huang, B. A Novel CVAE-Based Sequential Monte Carlo Framework for Dynamic Soft Sensor Applications. IEEE

Transactions on Industrial Informatics, 2024, 20(3), 3789-3800. https://doi.org/10.1109/TII.2023.3299611

18. Tao, H. Smoke Recognition in Satellite Imagery via an Attention Pyramid Network with Bidirectional Multilevel Multigranularity Feature Aggregation and Gated Fusion. IEEE Internet of Things Journal, 2024, 11(8), 14047-14057. https://doi.org/10.1109/JIOT.2023.3339476
19. V., N., G., Y., B., N. N., R., M., P., P. Empirical Analysis of Squeeze and Excitation-Based Densely Connected CNN for Chili Leaf Disease Identification. IEEE Transactions on Artificial Intelligence, 2024, 5(4), 1681-1692. https://doi.org/10.1109/TAI.2024.3364126
20. Wang, J. BGcsSENet: Bidirectional GRU with Spatial and Channel Squeeze-Excitation Network for Bundle Branch Block Detection. IEEE Transactions on Human-Machine Systems, 2023, 53(2), 449-457. https://doi.org/10.1109/THMS.2023.3244938
21. Wei, Y., Jang-Jaccard, J., Xu, W., Sabrina, F., Camtepe, S., Boulic, M. LSTM-Autoencoder-Based Anomaly Detection for Indoor Air Quality Time-Series Data. IEEE Sensors Journal, 2023, 23(4), 3787-3800. https://doi.org/10.1109/JSEN.2022.3230361
22. Wu, D., Peng, K., Wang, S., Leung, V. C. M. Spatial-Temporal Graph Attention Gated Recurrent Transformer Network for Traffic Flow Forecasting. IEEE Internet of Things Journal, 2024, 11(8), 14267-14281. https://doi.org/10.1109/JIOT.2023.3340182
23. Wu, J., Abolghasemi, V., Anisi, M. H., Dar, U., Ivanov, A., Newenham, C. Strawberry Disease Detection Through an Advanced Squeeze-and-Excitation Deep Learning Model. IEEE Transactions on AgriFood Electronics, 2024, 2(2), 259-267. https://doi.org/10.1109/TAFE.2024.3412285
24. Xiang, T., Jin, Y., Liu, Z., Clifton, L., Clifton, D., Zhang, Y. Dynamic Beat-to-Beat Measurements of Blood Pressure Using Multimodal Physiological Signals and a Hybrid CNN-LSTM Model. IEEE Journal of Biomedical and Health Informatics, 2025, 29(8), 5438-5451. https://doi.org/10.1109/JBHI.2025.3548771
25. Xiao, F., Liu, R., Zhu, Y., Zhang, J., Chen, S. A Dense Multicross Self-Attention and Adaptive Gated Perceptual Unit Method for Few-Shot Semantic Segmentation. IEEE Transactions on Artificial Intelligence, 2024, 5(6), 2493-2504. https://doi.org/10.1109/TAI.2024.3369553
26. Ye, W., Yang, J., Chen, P. Short-Term Stock Price Trend Prediction with Imaging High Frequency Limit Order Book Data. International Journal of Forecasting, 2024, 40(3), 1189-1205. https://doi.org/10.1016/j.ijforecast.2023.10.008
27. Yi, M.-K., Lee, W.-K., Hwang, S. O. A Human Activity Recognition Method Based on Lightweight Feature Extraction Combined with Pruned and Quantized CNN for Wearable Device. IEEE Transactions on Consumer Electronics, 2023, 69(3), 657-670. https://doi.org/10.1109/TCE.2023.3266506
28. Yuan, Z., Liu, J., Zhou, H., Zhang, D., Liu, H., Zhu, N. LEVER: Online Adaptive Sequence Learning Framework for High-Frequency Trading. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(11), 6547-6559. https://doi.org/10.1109/TKDE.2023.3336185
29. Zhou, W., Lv, Y., Lei, J., Yu, L. Embedded Control Gate Fusion and Attention Residual Learning for RGB-Thermal Urban Scene Parsing. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(5), 4794-4803. https://doi.org/10.1109/TITS.2023.3242651
30. Zhou, X., Liang, L., Zhang, J., Wen, C.-K., Jin, S. Gradient-Based Markov Chain Monte Carlo for MIMO Detection. IEEE Transactions on Wireless Communications, 2024, 23(7), 7566-7581. https://doi.org/10.1109/TWC.2023.3342618