Stock Price Prediction Model Based on Feature Engineering and Optimized Transformer: An Empirical Study on Improving the Precision of Investment Decision-making through Time Series Analysis

Yingjie Cheng

School of Software, Xinjiang University, Xinjiang, China

chengyingjie04@qq.com

Abstract. Stock price prediction has always been a topic of great concern in the financial field, and it has important practical significance for investment decision-making and market supervision.The CEEMDAN-Lasso-BO-Transformer model constructed in this paper obtains features through Complementary Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), screens key features using the Least absolute shrinkage and selection operator (Lasso), builds the model with Transformer, and tunes hyperparameters through Bayesian Optimization (BO), thus finally completing feature engineering and model optimization.Experiments show that the CEEMDAN-Lasso-BO-Transformer model constructed this time has a fitting degree as high as 98.7%, and its indicators such as MAE, MAPE, and RMSE are the best. It successfully predicts the trend of SSE stock prices from April to June 2025. Simulated trading shows that the Sharpe ratio of this model reaches 0.75, with the most stable profit, which can provide more accurate decision-making support for investors.

Keywords: Stock Price Prediction; CEEMDAN; Lasso; BO; Transformer Model.

1. Introduction

1.1 Research Background

As a "barometer" of economic operation, the fluctuations of stock prices are the result of the intertwined effects of multiple factors such as the macro-economy, the globalization pattern, and technological revolutions. Macroeconomic indicators directly affect enterprises' profit expectations and market liquidity [1]. Globalization has intensified the linkage of international markets and profoundly changed the performance of the stock market [2]. Technological revolutions have reshaped the pricing logic of the stock market and promoted structural adjustments in the market. Under multiple challenges such as the intensification of aging and technological disruptions, traditional methods struggle to capture complex dependencies, while the combination of machine-learning tools with high-frequency data analysis has become a key strategy for dealing with complex market environments and capturing structural opportunities.

1.2 Research Significance

The study of stock prices can help us understand market efficiency and reflect the market's expectations of a company's financial situation and the macro-economic environment. For investors, it is helpful for making more accurate investment decisions, avoiding risks, and achieving asset diversification. In addition, it can provide a reference for policymakers, as well as for market efficiency evaluation and risk management. Meanwhile, it reveals the promoting effect of investors' sentiment [3] and social psychology on prices.

1.3 Research Status

1.3.1 Research Status of Modal Decomposition

Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) are commonly-used methods for decomposing non-linear and non-stationary signals. However, both have many deficiencies, especially in noise handling.

The CEEMDAN [1] method improves the noise removal mechanism by gradually eliminating the influence of noise, effectively overcoming the noise interference in EMD and EEMD [4], and significantly improving the accuracy and robustness of decomposition. It has achieved remarkable results in fields such as financial market forecasting. For example, Diaa S. Metwally et al. [5] used CEEMDAN to propose the CEEMDAN-SVR model, which performed excellently in indicators such as RMSE and MAE, outperforming other traditional hybrid models.

1.3.2 Research Status of Feature Engineering

In the practical applications of machine learning and data mining, feature selection [6] can significantly improve the accuracy of models, reduce overfitting, accelerate the training process, and enhance the interpretability of models. Among them, Lasso selects features through L1 regularization, reduces overfitting, and can significantly improve the interpretability of models [7].

1.3.3 Research Status of Transformer

Based on the self-attention mechanism, Transformer can efficiently capture the dependencies between different positions in a sequence, overcoming the limitations of traditional recurrent neural networks and LSTM models when dealing with long sequences. In recent years, it has increasingly demonstrated powerful advantages in time-series forecasting. For example, Hao Jianlong [8], Liu Zhibin, etc. proposed a stock trend prediction framework based on an improved Transformer dynamic hypergraph convolutional neural network. Experimental results show that this model demonstrates significant advantages in prediction performance compared with existing advanced models.

1.3.4 Research Status of Bayesian Optimization

As an efficient global optimization method, Bayesian optimization [9] is widely used in deep learning. Since it can efficiently explore the hyperparameter space and is applicable to high-dimensional complex problems, it has become the core method for modern hyperparameter optimization.

1.4 Research Content

In this study, by using the data of the Shanghai Composite 50, through CEEMDAN decomposition [10], Lasso feature screening, Transformer modeling and BO tuning, a prediction model is constructed to verify its superiority in stock price prediction. The aim is to provide accurate and reliable stock market price prediction through various methods, so as to help investors make more accurate decisions.

2. Theoretical Basis of the Model

2.1 CEEMDAN

CEEMDAN is a signal decomposition method based on the Fourier transform, used to handle adaptive non-linear and non-stationary signals [11]. It analyzes non-linear data by introducing EMD and decomposes complex signals into multiple Intrinsic Mode Function (IMF) signals and a residual signal. Suppose the data sequence is expressed as $s(t)$, the general steps of EMD are as follows [12]:

(1) Calculate the extreme values of the original sequence $s(t)$;

(2) Use the local minimum and maximum values as interpolation points for curve interpolation to obtain the required upper and lower envelope curves, expressed as $S_{\max}(t)$ and $S_{\min}(t)$;

(3) Calculate the average value of the two envelope curves, denoted as $f_1(t)$:

$$f_1(t)=\frac{S_{\max}(t)+S_{\min}(t)}{2} \tag{1}$$

(4) Calculate $m_1(t)=s(t)-f_1(t)$. If $m_1(t)$ belongs to the first-order IMF of $m_1(t)$, then proceed to step (5); otherwise, move to the next step.

(5) Consider $m_1(t)$ as a new sequence, and repeat steps 1 to 4 until the Standard Deviation (SD) drops below a specified threshold. At this point, $m_1(t)$ represents the first-order IMF of $s(t)$. The calculation of SD is as follows:

$$SD=\sum_{t=0}^{T}\frac{\left|m_{k-1}(t)-m_k(t)\right|^2}{m_{k-1}^2(t)} \tag{2}$$

(6) After obtaining the first-order IMF, the remaining data is:

$$c_1(t)=s(t)-IMF(1) \tag{3}$$

(7) By considering the data $c_1(t)$ as the original sequence and repeating this process, successively derive $IMF(2),...,IMF(n)$ and the residual component $r(t)$ according to steps 1 to 6. Finally, the restored signal is:

$$s(t)=\sum_{i=1}^{n} IMF(i)+r(t) \tag{4}$$

Although EMD has achieved remarkable results in signal processing, it still has the problem of mixed modes in practical applications. To address this issue, CEEMDAN emerged. It not only reduces the impact of mode mixing but also overcomes the problem that Gaussian white noise cannot be completely eliminated [13].

The implementation process of the CEEMDAN algorithm in this paper is as follows [14]. Let E_m represent the m-th IMF component after performing EMD decomposition on the sequence, ε_m represent the noise coefficient added to the input sequence by CEEMDAN at the m-th stage, and $c_m(t)$ be the m-th IMF component generated by CEEMDAN:

(1) Add Gaussian white noise N times to the original signal $s(t)$ to construct a total of N pre-processed sequences $s_n(t)$, where $n=1,2,...,N$.

$$s_n(t)=s(t)+\varepsilon_0\delta_n(t) \tag{5}$$

(2) Perform EMD decomposition on all pre-processed sequences $s_n(t)$ to obtain the first IMF component $c_1^n(t)$. Take the average value as the first IMF component $c_1(t)$ obtained by CEEMDAN decomposition, and at the same time, obtain the first residual sequence $r_1(t)$, as shown below respectively.

$$\begin{gathered} c_1(t)=\frac{1}{N}\sum_{n=1}^{N}c_1^n(t) \\ r_1(t)=s(t)-c_1(t) \end{gathered} \tag{6}$$

(3) Similarly, add Gaussian white noise to the residual sequence $r_1(t)$ to construct N new sequences $r_1(t)+\varepsilon_1E_1(\delta_n(t))$. After performing EMD decomposition on these N sequences,

calculate the average value to obtain the second IMF component $c_2(t)$, and then find the difference to get $r_2(t)$. By analogy, the m-th residual is shown in the following formula:

$$r_m(t) = r_{m-1}(t) - c_m(t) \tag{7}$$

(4) Perform N times of EMD decomposition on $r_m(t) + \varepsilon_m E_m(\delta_n(t))$, and then the $(m+1)$- th IMF sequence after CEEMDAN decomposition can be obtained, as shown below:

$$c_{m+1}(t) = \frac{1}{N}\sum_{n=1}^{N} E_1\left(r_m(t) + \varepsilon_m E_m(\delta_n(t))\right) \tag{8}$$

(5) Repeat the above steps until the decomposition stops. The final residual sequence is shown in the following formula:

$$R(t) = s(t) - \sum_{m=1}^{M} c_m(t) \tag{9}$$

Finally, the expression of the signal sequence $s(t)$ after CEEMDAN decomposition can be summarized as follows.

$$s(t) = R(t) + \sum_{m=1}^{M} c_m(t) \tag{10}$$

The algorithm flow chart of CEEMDAN is shown in Figure 1.

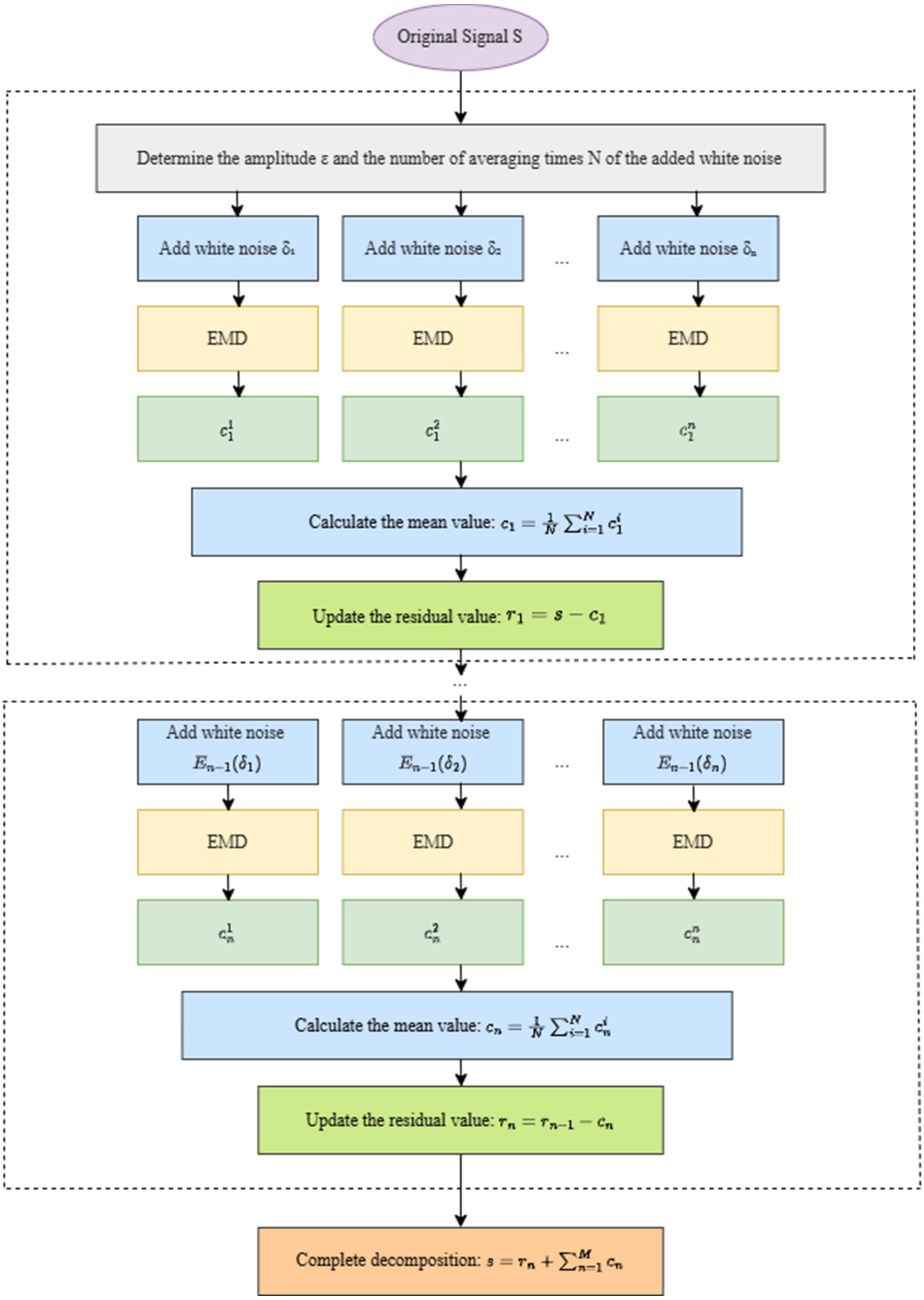

Figure 1 Algorithm Flow Chart of CEEMDAN

2.2 Lasso

Linear regression has long been widely used as a simple and effective prediction method in many fields such as economics and physics. However, with the development of data, traditional linear regression can no longer adapt and show good application results, often resulting in overfitting. To address this issue, this study employs Lasso regression analysis, which has unique advantages in variable selection and model construction [15], and can obtain a sparse linear model in a high-dimensional variable space. By introducing an L1 regularization term, it promotes the sparsification of certain coefficients of the model, thus achieving automatic feature selection. The linear regression model is defined as [16]:

$$y_i = a_i + \sum_{j=1}^{p} \beta_j x_{ij} \tag{11}$$

where x_{ij} is the predictor variable and y_i is its corresponding response variable. We assume that each observation is independent and the predictor variables are standardized. Then the Lasso regression is described as:

$$\hat{\beta}_{lasso} = \arg\min \left\{ \left\| Y - X\beta \right\|^2 + \lambda \sum_j \left| \beta_j \right| \right\} \tag{12}$$

The first half of the right-hand side of the equation is a least-squares term, and the second half is an added penalty term. Here, λ is a non-negative regularization parameter, and λ is a set hyperparameter. As λ increases, Lasso shrinks the coefficients to 0 and finally selects an appropriate λ . The steps of the Lasso algorithm in this study are as follows:

Input: IMF features after CEEMDAN modal decomposition and original features.

Output: Relatively important features (original features + IMF) selected after Lasso regression analysis.

Step 1: Data preprocessing to ensure that the input data meets the standardization steps.

Step 2: Use the Lasso training model to calculate the solution path: a) Select an appropriate regularization parameter and conduct model training; b) After training, assign corresponding coefficients to each feature value to complete the feature value evaluation; c) Rank the importance of features according to the absolute value of the coefficients;

2.3 Transformer

The Transformer model was initially designed for natural language processing tasks and marks an important turning point in the field of natural language processing [17]. In addition, due to its unique architecture, Transformer has also achieved remarkable results in image classification, object recognition, and other fields. In this study, the Transformer model will be used and improved for time-series prediction. Time-series prediction relies on the unique self-attention mechanism of Transformer, which enables the model to be more efficient and flexible when processing sequential data [18]. Below, taking a single encoder as an example, the self-attention mechanism and application of the Transformer model in this paper will be mainly introduced.

The self-attention mechanism is the core advantage of Transformer. The difference from the attention mechanism lies in that both the query and the key of the self-attention mechanism come from the same set of elements. They complete attention aggregation among each other, capture the relationships between different time points in the sequence, thereby establishing global dependencies. By expanding the receptive field, more context information is obtained. The processing process of a single encoder is as follows. Given the input $x_D^{(l)}$, the self-attention mechanism receives three inputs: query, key, and value, as shown in the following formula:

$$\begin{Bmatrix} Q^{(l)} = x_D^{(l)} W_Q^{(l)} \\ K^{(l)} = x_D^{(l)} W_K^{(l)} \\ V^{(l)} = x_D^{(l)} W_V^{(l)} \end{Bmatrix} \tag{13}$$

where $W_Q^{(l)}, W_K^{(l)}, W_V^{(l)}$ are their respective weight matrices. Then, calculate their scaled dot-product to obtain the attention scores. The formula is as follows:

$$Attention^{(l)}\left(Q^{(l)}, K^{(l)}, V^{(l)}\right) = soft\max\left(\frac{Q^{(l)}\left(K^{(l)}\right)^T}{\sqrt{d}}\right)V^{(l)} \quad (14)$$

where $soft\max(\ )$ is used for the normalization of the dot-product calculation, and the scaling ratio $\sqrt{d}$ is used to prevent $soft\max(\ )$ from saturating when the gradient is extremely small. l represents the layer index in the Transformer model. The architecture diagram of the self-attention mechanism is shown in Figure 2:

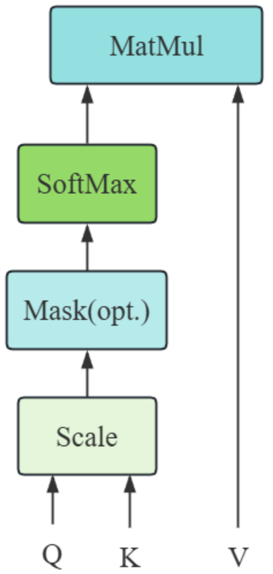

Figure 2 Architecture Diagram of Self-attention Mechanism

The calculation formula for multi-head attention is as follows [19].:

$$\left.\begin{cases} head_j^{(l)} = Attention^{(l)}\left(Q_j^{(l)}, K_j^{(l)}, V_j^{(l)}\right) \\ MultiHead\ \ Attention^{(l)}\left(Q^{(l)}, K^{(l)}, V^{(l)}\right) = Concat\left(head_1^{(l)}, \ldots, head_H^{(l)}\right) \end{cases}\right\} \quad (15)$$

The architecture diagram is shown in Figure 3.:

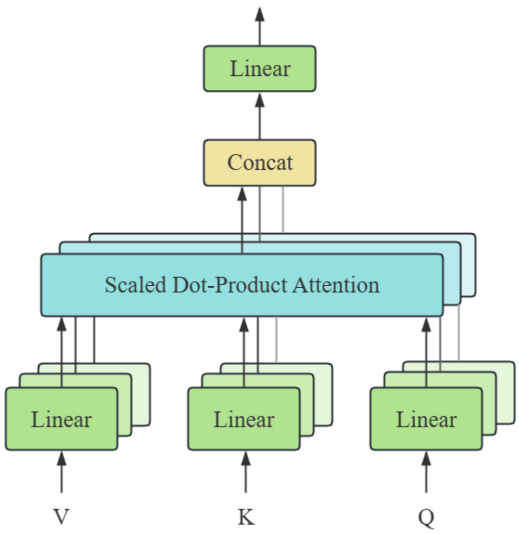

Figure 3 Architecture Diagram of Multi-head Attention

After passing through the encoder, the output of $x_D^{(l)}$ is expressed by the following formula:

$$\begin{cases} x_e^{(l)} = LayerNorm\left(x_D^{(l)} + Multi\ Head\ Attention^{(l)}\left(x_D^{(l)}\right)\right) \\ x_e^{(l)} = LayerNorm\left(x_e^{(l)} + FFN\left(x_e^{(l)}\right)\right) \\ FFN\left(x_e^{(l)}\right) = \max\left(0, x_e^{(l)}W_1 + b_1\right)W_2 + b_2 \end{cases} \tag{16}$$

Finally, we use a flatten layer and a linear layer to project $x_e^{(l)}$, and then combine them as the final multi-variable time-series prediction result. The formula is as follows:

$$\begin{cases} y^{(l)} = W_2\left(flatten\left(x_e^{(l)}\right)\right) + b_2 \\ Y = concatenate\left(y^{(1)},\ldots,y^{(k)}\right) \end{cases} \tag{17}$$

where $y^{(i)} \in R^{M\times T}$, M represents the number of variables, T represents the prediction step size, and Y represents the prediction result.

2.4 Bayesian Optimization

Hyperparameter optimization is a crucial step in the development of machine-learning models. In recent years, Bayesian optimization has gained importance as an effective hyperparameter optimization method due to its ability to efficiently explore the hyperparameter space [20]. The Bayesian optimization algorithm is based on Bayes' theorem. It can update its estimate of the objective function when new data is continuously observed, and adjust the selection of the next hyperparameter accordingly. It can obtain a global optimal solution at a relatively low cost and complete hyperparameter optimization [21].

In this study, we will use Bayesian optimization to optimize the Transformer model, including the number of heads in the multi-head attention mechanism, the number of layers in the Transformer encoder, the learning rate of the Adam optimizer, etc. The steps are as follows:

(1) Set the objective function $f(\theta)$;

(2) Select a prior distribution θ for the hyperparameter $p(\theta)$;

(3) Randomly select n points $\{\theta_1, \theta_2, \ldots, \theta_n\}$ in the search space, evaluate these points, and use the results as the initial dataset $D = \left\{\left(\theta_i, f\left(\theta_i\right)\right)\right\}_{i=1}^{n}$;

(4) Define the mean function $m(\theta)$ and the covariance function $k(\theta, \theta')$ to construct a Gaussian process model;

(5) Construct the acquisition function $a(\theta)$, with the formula as follows:

$$a(\theta) = E\left[f(\theta) - f\left(\theta_{best}\right) \mid \theta\right] \tag{18}$$

where θ_{best} is the current optimal hyperparameter.

(6) Use the acquisition function to select the next evaluation point θ_{new}, evaluate the objective function at the selected point, and add the result to the dataset, $D_{new} = D \cup \left\{\left(\theta_{new}, f\left(\theta_{new}\right)\right)\right\}$, then update the Gaussian process model with the new data.

(7) Repeat steps 5-6 until the stopping condition is met, that is, the improvement of the objective function is no longer significant.

(8) Select the hyperparameter that optimizes the objective function from all the evaluated points, with the formula as follows:

$$\theta^* = \arg\min_{\theta \in D} f(\theta) \tag{19}$$

2.5 Chapter Summary

This chapter covers the algorithms used in the research and their improvements. The algorithms selected for stock prediction are: CEEMDAN algorithm, Lasso regression analysis algorithm, Transformer model, and BO algorithm. The self-attention mechanism of the Transformer makes it suitable for time-series prediction. This chapter details the principles and calculation steps of each method, and the comparison of the actual prediction results of the Transformer optimized by BO will be discussed in subsequent chapters.

3. Introduction to Data, Evaluation Indicators, and Feature Engineering

3.1 Data Introduction

The stock price prediction dataset used in this paper is sourced from the Shanghai Composite 50, which includes information such as opening prices, closing prices, highest prices, lowest prices, and trading volumes. The specific data structure is shown in Table 1. To ensure data integrity, missing values were first imputed, and the data was sorted. Eventually, a complete stock dataset was constructed. This dataset covers stock prices from December 1990 to March 2025, containing a total of 8361 records. This paper aims to predict the stock prices in May 2025, helping investors make more accurate investment decisions amidst the influence of multiple factors such as policy games, technological progress, and geopolitics.

Table 1 Data Definition

Stock Indicator	Meaning
Opening Price (Open)	The price of the first stock transaction after the market opens on a trading day
Closing Price (Close)	The price of the last stock transaction on a trading day
Highest Price (High)	The highest price at which stocks are traded on a trading day
Lowest Price (Low)	The lowest price at which stocks are traded on a trading day
Trading Volume (Volume)	The total number of individual stock buy-sell transactions in the stock trading market within a unit of time

3.2 Feature Engineering

In the field of stock price prediction, the features related to a single trading day often have limitations. If these features are used directly, it is highly likely that the model will have a sub-optimal fitting effect. To effectively solve this problem, feature engineering constructs a feature system highly compatible with stock price prediction through careful design, improving the prediction accuracy of the model. To ensure that different features have the same scale, we normalized the data, scaling the feature values to a unified range between 0 and 1, thus avoiding the adverse effects on model training caused by some features having too large or too small value ranges. The calculation formula is:

$$X_i = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)} \tag{20}$$

where X_i represents the i-th sample value in the original dataset.

In the stock price prediction task, to improve the prediction accuracy, it is crucial to retain high-value indicators and eliminate low-correlation indicators. We innovatively adopt a method combining CEEMDAN and LASSO. With the help of CEEMDAN, the stock price series is effectively decomposed to explore intrinsic features, and then LASSO is used to screen and model the indicators, accurately capturing key factors to achieve more accurate stock price prediction. First, we use the sliding window method to construct input-output samples. The time window size is set to 10, that is, each time we use the feature data of the past 10 time steps to predict the closing price of the next time step. By continuously sliding this window through the entire dataset, a series of input-output pairs are generated. Then CEEMDAN is used to decompose the closing price, decomposing it into 10 Intrinsic Mode Functions. The decomposition result is shown in Figure 4 below.

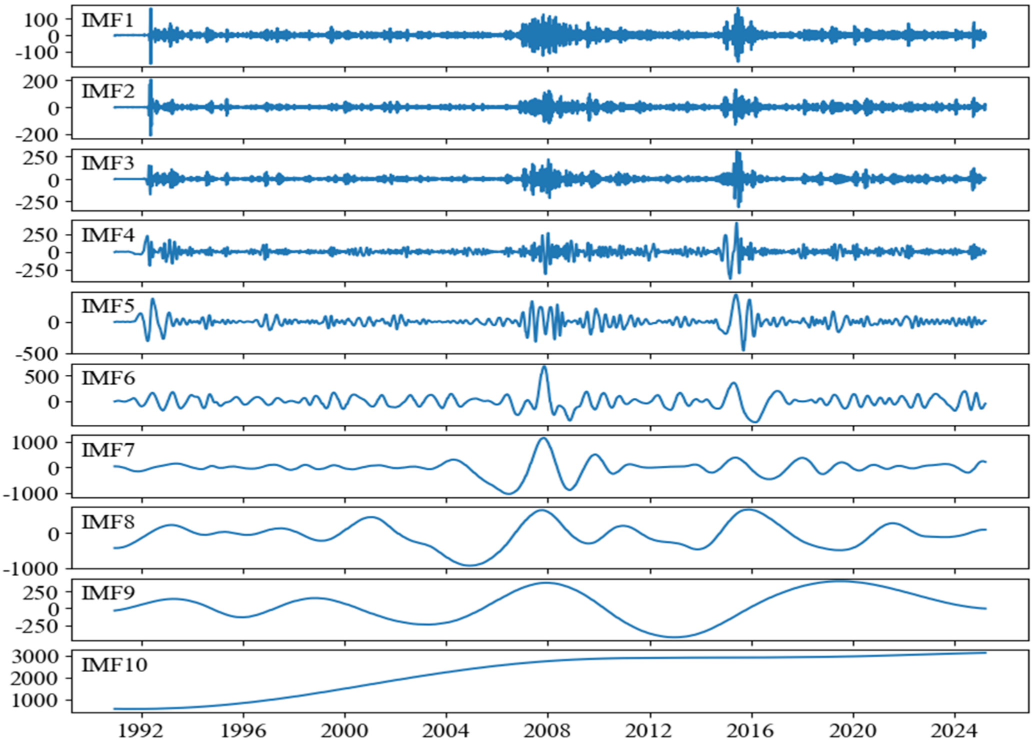

Figure 4 Decomposition Results of CEEMDAN

Figure 4 shows 10 IMF components, labeled IMF1-IMF10 from top to bottom. Each IMF component is presented in the form of a time series, where the horizontal axis represents time and the vertical axis represents the values of the corresponding IMF component at different time points. Among these IMF components, the high-frequency IMFs (IMF1-IMF4) fluctuate frequently and violently, representing the high-frequency detailed components in the signal, and mainly reflecting the short-term violent fluctuations of stock prices. The high-frequency IMFs contain rapid changes and noise during fluctuations, which helps traders capture the instantaneous changes in the market and identify short-term buying and selling signals or trend turning points. In contrast, the low-frequency IMFs (IMF7-IMF10) fluctuate more slowly and smoothly, reflecting the long-term trend of stock market prices. The low-frequency IMFs help analyze the long-term direction of the stock market by shaping the overall contour of the market during the signal reconstruction process.

Then, we organize the IMF components obtained from the CEEMDAN decomposition together with the original financial market data such as opening prices, lowest prices, and highest prices into a feature matrix. Each row represents a sample at a time point, and each column corresponds to a feature. This feature matrix is fed into the LASSO model as input. Through training on the input feature matrix, the LASSO model will output the coefficients corresponding to each feature. The absolute values of these coefficients reflect the importance of each feature in the model. The importance of each feature is shown in Figure 5 below.

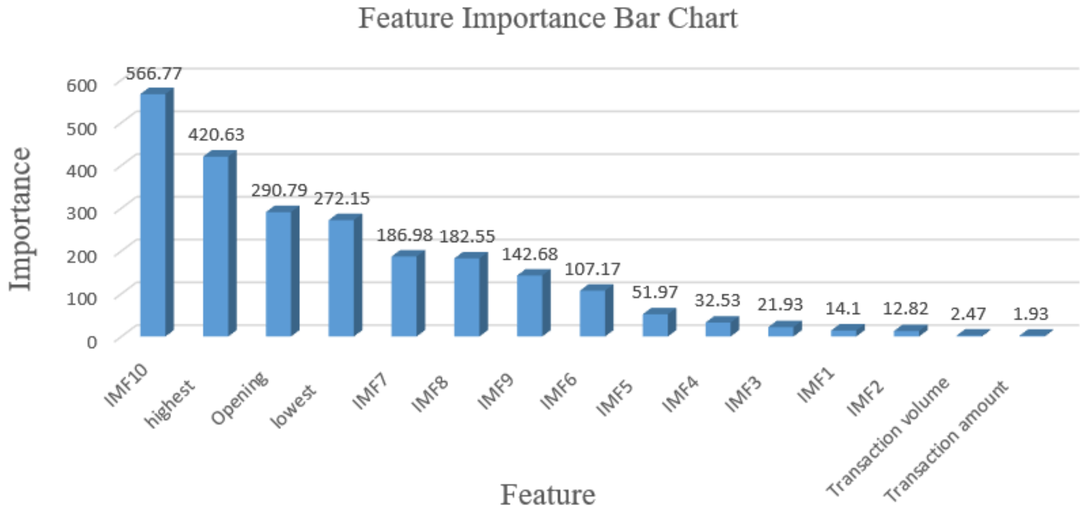

Figure 5 Bar Chart of Feature Importance

As can be seen from Figure 5, the importance values of different features vary significantly. The importance value of "IMF10" is the highest, reaching 566.77, while that of "turnover" is the lowest, at 1.94. Through calculation, the median of feature importance is 107.18. We screened out the features whose importance values are greater than the median. Finally, "opening price", "lowest price", "highest price", and "IMF7", "IMF8", "IMF9", "IMF10" were selected as key features. These key features will be used for subsequent model training and analysis.

In terms of data division, we divided the dataset according to a ratio of 8:2. This 8:2 division can effectively prevent the model from overly relying on the noise in the training data and reduce the risk of overfitting. The first 80% of the data, consisting of 7100 data points from 1990 to 2019, was used for model training and optimization verification. The remaining 20% of the data, from 2020 to March 2025, was used for model testing and validation, as shown in Table 2.

Table 2 Data Characteristics for Model Building

Dataset	Data Time Period	Number of Samples
Training Data	1990-December 31, 2019	7100
Testing Data	2020-March 19, 2025	1261

To visually present the changing characteristics of the closing price in different time periods, we drew the following line chart for closing price analysis, as shown in Figure 6.

Figure 6 Changing Characteristics of Closing Prices

In Figure 6, the blue line represents the training data, and the orange line represents the testing data. The vertical dashed line in the figure divides the data into training data and testing data. During the period of the training data, the closing price as a whole showed a trend of slow upward fluctuation at first. Around 2008 and 2016, the price rose sharply and then dropped significantly and continued to fluctuate. The testing data followed the later-stage trend of the training data, fluctuating within a relatively stable range without the extreme large-scale rises and falls seen in the training data.

3.3 Model Performance Evaluation Indicators

To accurately evaluate the prediction performance of the model for stock prices, we introduced reasonable evaluation indicators. With the help of these indicators, we can well judge the performance level of the model. The descriptions and calculation formulas of the evaluation indicators used in the experiment are as follows:

MSE: This method is used to evaluate the fitting degree of the model. It calculates the average of the squared differences between the predicted values and the true values. The smaller the MSE, the better the model fits.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{21}$$

MAE: It evaluates the accuracy of the model by calculating the average of the absolute errors between the predicted values and the true values. The smaller the MAE, the better the model fits.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{22}$$

RMSE: RMSE is the square root of MSE and has the same unit as the original data, making it more interpretable. The smaller the RMSE, the better the model fits.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{23}$$

MAPE: MAPE measures the percentage of the prediction error relative to the actual value, enabling comparison of data of different scales. The smaller the MAPE, the better the model fits.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{24}$$

where y_i refers to the actual stock price at the i-th time point, $\hat{y}_i$ refers to the predicted stock price at the i-th time point, and n is the total number of stock price data points.

4. Stock Financial Analysis Based on CEEMDAN-Lasso-BO-Transformer Model

In this paper, CEEMDAN, Lasso, BO, and Transformer are combined to establish a stock financial analysis and prediction model based on the CEEMDAN-Lasso-BO-Transformer model. The CEEMDAN decomposition technique is used to obtain features, Lasso regression analysis is used to screen features, which are then applied to the Transformer model for training. BO is used to optimize the parameters of the Transformer model to further fit and predict stocks. At the same time, a ablation experiment was carried out, systematically removing or replacing different components of the model to explore the specific roles of each algorithm in the model and the prediction performance of each model. In addition, LSTM was introduced to fit and predict stocks, and its results were compared with those of the model established in this study.

4.1 Empirical Analysis Based on CEEMDAN-Lasso-BO-Transformer Model

4.1.1 Basic Statistical Analysis

Calculate basic indicators such as the mean and standard deviation for the prediction results output by each prediction model, and draw a basic statistical analysis table (Table 3). By analyzing the data in Table 3, we can intuitively understand the central tendency and degree of dispersion of the prediction results of each model.

Table 3 Basic Statistical Analysis Table

Indicator	Mean	Standard Deviation	Minimum	Median	Maximum
Actual Prices	3139.27	271.34	2464.36	3137.10	3715.37
LSTM	3226.30	310.34	2465.91	3222.49	3932.08
Transformer	3211.04	291.59	2515.33	3194.05	3814.69
BO-Transformer	3199.67	281.66	2499.10	3201.78	3854.35
CEEMDAN-BO-Transformer	3156.60	270.76	2437.37	3156.18	3721.98
CEEMDAN-Lasso-BO-Transformer	3112.27	270.79	2433.09	3113.11	3691.67

In this paper, the Transformer model and the LSTM model in deep learning are introduced. Through a comparative experiment, one of them is selected as the basic prediction model for this study. As can be seen from Table 3, the mean values of both models are generally higher than the mean value of the actual prices. The mean value of the Transformer model is lower than that of the LSTM model, indicating that the prediction results of the former are closer to the actual values. In terms of standard deviation, the standard deviation of the Transformer model is 291.59, which is significantly lower than that of the LSTM model, indicating that the Transformer model has better stability.

Considering the above factors, this paper selects the Transformer model as the basic prediction model, and this will be further confirmed in the subsequent analysis of this paper. Among the models that improve the Transformer model, the mean values of the BO-Transformer, CEEMDAN-BO-Transformer, and CEEMDAN-Lasso-BO-Transformer models show a downward trend. The mean value of the CEEMDAN-Lasso-BO-Transformer model is closest to the mean value of the actual prices, indicating that this model performs better in predicting the overall level of stock prices. In terms of standard deviation, the CEEMDAN-Lasso-BO-Transformer model maintains a relatively low value among all the prediction models, with a value of only 270.79, showing small fluctuations and better stability.

4.1.2 Trend and Error Analysis

Analyzing the stock price trend is of great guiding significance for investors to make investment decisions. This study will compare the LSTM, Transformer, and three improved Transformer models. By combining the evaluation index table of each model (Table 4) and the analysis chart of the actual price trend and prediction errors of each model (Figure 7), the performance of the models will be evaluated in depth.

Table 4 Evaluation Index Table of Each Model

Model	MAE	MSE	RMSE	MAPE
LSTM	89.580	10800.093	103.924	2.78%
Transformer	78.729	9340.338	96.645	2.49%
BO-Transformer	60.723	4880.830	69.863	1.93%
CEEMDAN-BO-Transformer	32.386	1939.156	44.036	1.04%
CEEMDAN-Lasso-BO-Transformer	28.588	978.359	31.279	0.91%

Judging from the evaluation indicators in Table 4, the fitting degree of the LSTM model is relatively low, only 85.3%. Its MAE, MAPE, and RMSE values are relatively large. The MAE, MAPE, and RMSE of the Transformer model are 78.729, 2.49%, and 96.645 respectively, which are 12.11%, 10.43%, and 7.00% lower than those of the LSTM model respectively, showing obvious advantages. Therefore, the Transformer model is finally selected as the basic prediction model.

The BO-Transformer model introduces BO optimization based on the Transformer model. Compared with the Transformer model, its MAE, MAPE, and RMSE have improved by 22.87%, 22.49%, and 27.71% respectively.

Continuing with the ablation experiment, the CEEMDAN decomposition algorithm is introduced. Compared with the BO-Transformer model, the MAE, MAPE, and RMSE of this model have improved by 46.66%, 46.11%, and 36.97% respectively.

Then the Lasso regression algorithm is introduced to construct the CEEMDAN-Lasso-BO-Transformer model. Its MAE, MAPE, and RMSE are 28.588, 0.91%, and 31.279 respectively, which are 11.72%, 12.20%, and 28.97% higher than those of the CEEMDAN-BO-Transformer model respectively.

In order to show the overall trend of the evaluation indicators, a 3D bar chart of the evaluation indicators of each model is drawn. Through the 3D bar chart, the differences of different models in multiple evaluation indicators can be visually compared, as shown in Figure 7.

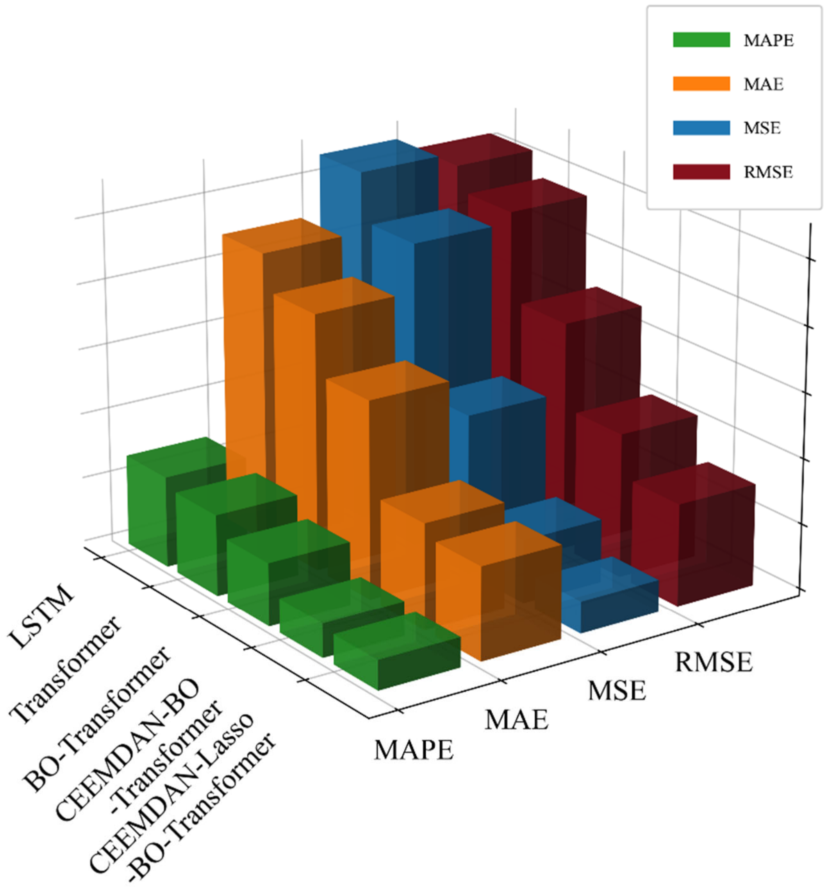

Figure 7 3D Bar Chart of Evaluation Indicators for Each Model

It can be clearly observed from Figure 7 that in terms of key evaluation indicators such as MAE, MSE, RMSE, and MAPE, the values of these indicators show a decreasing trend from the LSTM model to the CEEMDAN-Lasso-BO-Transformer model, fully demonstrating the superiority of the CEEMDAN-Lasso-BO-Transformer model in stock price prediction.

In order to better observe the fitting effect between the prediction results of each model and the actual price, a line chart of the actual price trend of the Shanghai Composite 50, the fitting results of each model, and a local enlarged analysis is drawn, as shown in Figure 8. The prediction accuracy of the model is evaluated from both the overall trend and local details.

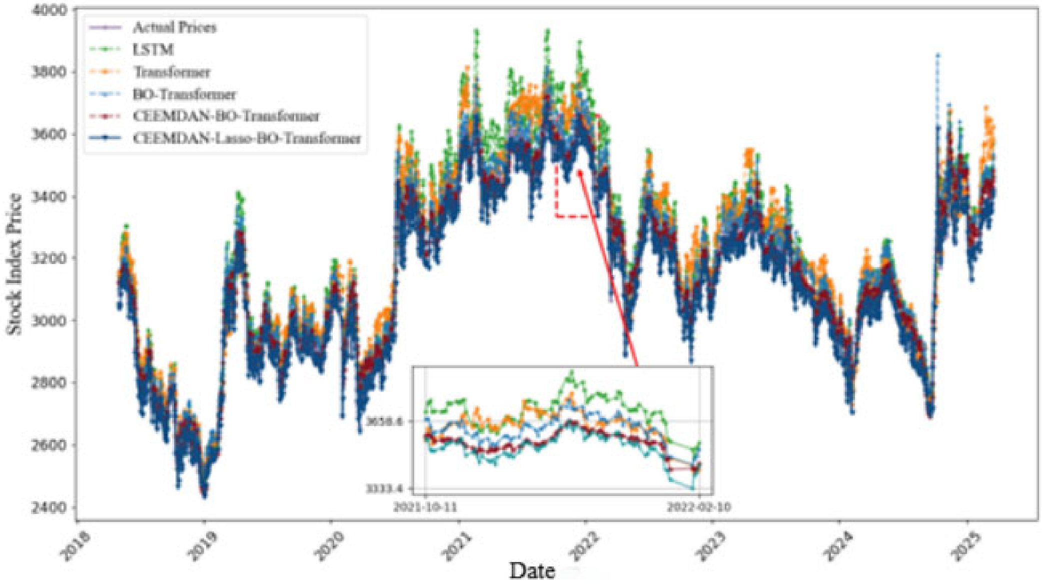

Figure 8 Actual Price Trend of Shanghai Composite 50, Fitting Results of Each Model and Local Enlarged Analysis

As can be seen from Figure 8, the CEEMDAN-Lasso-BO-Transformer model performs most prominently with a higher degree of fitting. The trend and error chart of the prediction results and actual prices of the CEEMDAN-Lasso-BO-Transformer model is drawn separately, as shown in Figure 9.

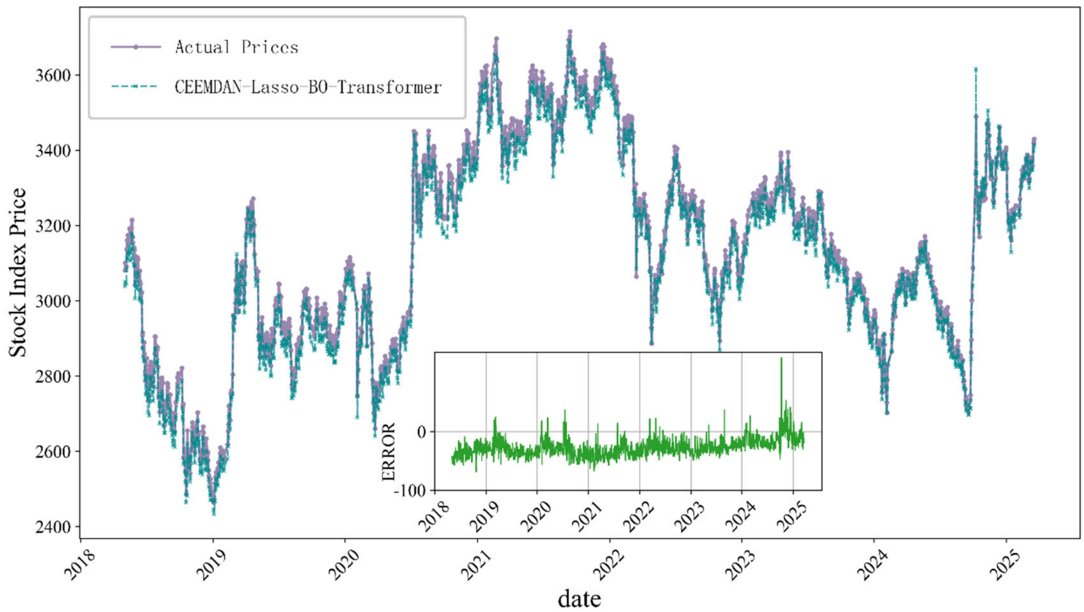

Figure 9 Actual Price Trend and Prediction Error Analysis of CEEMDAN-Lasso-BO-Transformer Model

As can be seen from Figure 9, the curve of the actual price almost completely coincides with that of the prediction results of the CEEMDAN-Lasso-BO-Transformer model in terms of trend, indicating that the model has a good fitting effect. In the error analysis, the error line fluctuates near the 0 value with small fluctuations, further indicating that the model has high prediction accuracy.

Through the above-mentioned ablation experiment verification, the prediction model based on CEEMDAN-Lasso-BO-Transformer established in this study can more accurately reflect price changes in both the upward and downward phases of the market, verifying the superiority and accuracy of this model in predicting stock price time series.

4.1.3 Model Optimization Analysis

This study records the hyperparameters after Bayesian optimization and other related information. In the BO-optimized model parameter table (Table 5), the Value value is used to measure the quality of the model. The smaller the Value value, the better the model performs.

Table 5 Optimized Hyperparameters of Each Model and Corresponding Value Table

Model	BO-Transformer	CEEMDAN-BO-Transformer	CEEMDAN-Lasso-BO-Transformer
d_model	32	64	128
nhead	2	2	2
num_layers	1	2	2
lr	0.0000143	0.0006528	0.0000506
Value	0.0000941	0.0003290	0.0000725

It can be seen from Table 5 that the Value value of the CEEMDAN-Lasso-BO-Transformer model is the smallest, only 0.0000725, which is far smaller than that of the other two improved models. This indicates that its comprehensive performance after model optimization is the best. Combined with the analysis results in the previous text, it further verifies the superiority of the CEEMDAN-Lasso-BO-Transformer model among numerous models. Therefore, the CEEMDAN-Lasso-BO-Transformer model is finally selected in this study.

4.2 Model Result Analysis

Through the above experiments, the stability and superior performance of the CEEMDAN-Lasso-BO-Transformer model constructed in this paper in stock prediction have been verified. In this section, the CEEMDAN-Lasso-BO-Transformer model is used to predict the stock prices in April, May and June 2025, and the prediction results are shown in Table 6.

Table 6 Predictions of Stock Prices in the Next Three Months by the CEEMDAN-Lasso-BO-Transformer Model

Time	April	May	June
Year 2025	3266.14	3243.19	3267.30

As can be seen from Table 4, the stock price trend of the Shanghai Composite Index in the next three months is fluctuating. For short-term investors, they can buy at a low price in May and sell at a high price in June. However, the stock market is complex and changeable, and corresponding countermeasures should be taken in advance.

4.3 Simulated Trading

In the previous text, the excellent prediction effect and stability of the constructed CEEMDAN-Lasso-BO-Transformer model in predicting the stock price sequence have been verified. However, a good prediction accuracy of the model in stock price prediction does not necessarily mean that good returns can be achieved or risks can be reduced in actual investment transactions. Therefore, a simulated trading experiment is carried out in this section to verify the effect of the model strategy in actual applications. Two methods are designed for this simulated trading: one is the holding and selling strategy; the other is the strategy based on the CEEMDAN-Lasso-BO-Transformer model.

The Sharpe ratio is selected as the evaluation index of the trading strategy in the simulated trading experiment. The larger the Sharpe ratio is, the higher the return of the strategy is. A simulated trading experiment is carried out on the Shanghai Composite Index test set, and the results are shown in Table 7 as follows.

Table 7 Simulated Trading Results under Different Trading Strategies

Stock	Investment Strategy	Sharpe Ratio
Shanghai Composite Index	CEEMDAN-Lasso- BO-Transformer	0.75
	BO-Transformer	0.57

CEEMDAN-BO-Transformer	0.63
Transformer	0.51
LSTM	0.42
Holding-and-Selling Strategy	-0.03

As can be seen from the data in the table, the CEEMDAN-Lasso-BO-Transformer model has the highest Sharpe ratio of 0.75. This fully verifies that the CEEMDAN-Lasso-BO-Transformer model, as an investment strategy, can achieve relatively high and stable returns in stock market investment.

5. Conclusions and Prospects

5.1 Conclusions

This paper started with the selection of a basic prediction model. Through experimental verification, the Transformer model was ultimately chosen as the basic model. To further enhance the prediction performance of the Transformer model, the CEEMDAN and Lasso algorithms, which have obvious advantages in processing feature data, were introduced into the Transformer model, and the BO algorithm was added for hyperparameter tuning. Finally, the CEEMDAN-Lasso-BO-Transformer model was constructed. Through the integration of multiple techniques, this model has demonstrated excellent performance in stock price prediction. It has the highest Sharpe ratio in simulated trading experiments, enabling it to bring relatively high and stable returns, and can provide accurate decision-making support for investors.

5.2 Research Prospects

The research work of this thesis is based on the CEEMDAN-Lasso-BO-Transformer model. Using this model to predict the stock prices of the Shanghai Stock Exchange has confirmed the feasibility and effectiveness of this method. In the future, multi-market tests will be expanded, and more factors such as sentiment indicators will be incorporated to improve the practicality and performance of the model.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to the school for providing valuable resources and a research platform. This has enabled us to search for and study relevant statistical modeling knowledge, offering strong support for our research.

Secondly, I am especially thankful to the teachers at the school. Thank you for your meticulous guidance and assistance during the competition and research process. The teachers not only promptly provided us with competition information and relevant learning resources but also offered clear directions and valuable feedback.

Finally, I would like to deeply thank my teammates. Throughout this research process, my teammates have always maintained a high level of teamwork spirit, working together with one heart and full cooperation. Whether it was in the initial stage of choosing the thesis topic, when we had joint discussions and pooled our ideas, or during the in-depth study of relevant knowledge and literature, we took the initiative to divide tasks and support each other. During the nearly one-month process of thesis writing, my teammates dedicated themselves whole-heartedly. When facing difficulties in the research, we encouraged each other and overcame them together to ensure that every step could proceed smoothly.

References

[1] Zhang Q, Zhang Y, Bao F, et al. Graph-based stock prediction with multisource information and relational data fusion[J]. Information Sciences, 2025, 690: 121561.

[2] Das N, Sadhukhan B, Ghosh C, et al. Utilizing Ensemble Learning and Dimension Reduction in Predicting Stock Prices: A Transparent Methodology with Insights from Explainable AI[J]. SN Computer Science, 2025, 6(1): 1-21.

[3] Dong Z, Zhou Y. A novel hybrid model for financial forecasting based on CEEMDAN-SE and ARIMA-CNN-LSTM[J]. Mathematics, 2024, 12(16): 2434.

[4] Zhang Y, Peng Y, Song Y. Realized Volatility Forecasting for Stocks and Futures Indices with Rolling CEEMDAN and Machine Learning Models[J]. Computational Economics, 2024: 1-54.

[5] Metwally D S, Ali M, Alghamdi S M, et al. A novel hybrid model to forecast the stock price based on CEEMDAN and support vector regression[J]. Journal of Radiation Research and Applied Sciences, 2025, 18(2): 101385.

[6] Zhou Z. A note on sharp oracle bounds for Slope and Lasso[J]. Communications in Statistics-Theory and Methods, 2025, 54(4): 949-967.

[7] Yang G, Li Y, Liu X. Asymmetry and determinants of financial connectivity in G20: Evidence from a quantile-based and lasso regression analysis[J]. The North American Journal of Economics and Finance, 2025: 102379.

[8] Hao Jianlong, Liu Zhibin, Zhang Chen, et al. Research on Stock Trend Prediction Method Based on Improved Transformer and Hypergraph Model [J] Journal of Intelligent Systems, 2024, 19(5): 1126-1135.

[9] Solís-Martín D, Galán-Páez J, Borrego-Díaz J. CONELPABO: composite networks learning via parallel Bayesian optimization to predict remaining useful life in predictive maintenance[J]. Neural Computing and Applications, 2025: 1-19.

[10] Chen X, Yang F, Sun Q, et al. Research on stock prediction based on CED-PSO-StockNet time series model[J]. Scientific Reports, 2024, 14(1): 27462.

[11] Cao J, Li Z, Li J. Financial time series forecasting model based on CEEMDAN and LSTM[J]. Physica A: Statistical mechanics and its applications, 2019, 519: 127-139.

[12] Yin Y, Liu Y, Fan Y. Research on Vegetable Commodity Price Prediction with Improved Lstm Based on Emd Decomposition and iCHOA Optimization[C]. IEEE, 2023: 153-157.

[13] Ho R, Hung K. A comparative investigation of mode mixing in EEG decomposition using EMD, EEMD and M-EMD[C]. IEEE, 2020: 203-210.

[14] Ulina M, Purba R, Halim A. Foreign exchange prediction using CEEMDAN and improved FA-LSTM[C]. IEEE, 2020: 1-6.

[15] Chen J, Zhao Z, Zheng Y, et al. Study on the effect of occupational exposure on hypertension of steelworkers based on Lasso-Logistic regression model[J]. Public Health, 2025, 239: 15-21.

[16] Fan L, Chen S, Li Q, et al. Variable selection and model prediction based on lasso, adaptive lasso and elastic net[C]. IEEE, 2015, 1: 579-583.

[17] Ghosh N, Santoni D, Saha I, et al. A Review on the Applications of Transformer-based language models for Nucleotide Sequence Analysis[J]. Computational and Structural Biotechnology Journal, 2025.

[18] Valle J, Bruno O M. Forecasting chaotic time series: Comparative performance of LSTM-based and Transformer-based neural network[J]. Chaos, Solitons & Fractals, 2025, 192: 116034.

[19] Koresh E, Gross R D, Meir Y, et al. Unified CNNs and transformers underlying learning mechanism reveals multi-head attention modus vivendi[J]. arxiv preprint arxiv:2501.12900, 2025.

[20] Yan-guang Z, Yi-fan Z. Robust temporal constraint optimization based on Bayesian optimization algorithm[C]. IEEE, 2010: 186-189.

[21] Lai Y. Application and Effectiveness Evaluation of Bayesian Optimization Algorithm in Hyperparameter Tuning of Machine Learning Models[C]. IEEE, 2024: 351-355.