Highlights

Enhancing financial time series forecasting with hybrid Deep Learning: CEEMDAN-Informer-LSTM model

Jiang-Cheng Li, Li-Ping Sun, Xiao Wu, Chen Tao

- A novel hybrid prediction model based on deep learning is proposed.
- The optimal frequency components after noise reduction by CEEMDAN are selected for prediction using Informer and LSTM models.
- The proposed model is compared with various benchmark models in terms of predictive performance.
- Robustness tests are carried out on the predictive performance for various trends.
- The proposed CEEMDAN-Informer-LSTM model demonstrates superior predictive capabilities.

Enhancing financial time series forecasting with hybrid Deep Learning: CEEMDAN-Informer-LSTM model

Jiang-Cheng Li[a,*], Li-Ping Sun[a], Xiao Wu[a], Chen Tao[a]

[a] *School of Finance, Yunnan University of Finance and Economics, Kunming, 650221, Yunnan, P. R. of China*
[b] *School of Finance, Yunnan University of Finance and Economics, Kunming, 650221, Yunnan, P. R. of China*

Abstract

Financial time series forecasting is fraught with challenges due to its inherent noise and uncertainty, which can significantly bias model prediction outcomes. To enhance predictive accuracy, this study introduces a novel hybrid deep learning forecasting model based on Complete Ensemble Empirical Mode Decomposition of Adaptive Noise (CEEMDAN), Informer, LSTM, and optimal model selection: the CEEMDAN-Informer-LSTM model, which demonstrates a significant advantage in handling large-scale nonlinear data. The model begins with the CEEMDAN algorithm for signal preprocessing, effectively reducing the impact of noise. Subsequently, leveraging the results of model selection, Informer and LSTM are employed to predict the high-frequency and low-frequency data components derived from the decomposition, respectively. The forecasts from these models are then integrated to yield an optimized predictive outcome. In comparisons with eight benchmark models and through robustness tests across various forecast horizons, our model consistently exhibits superior predictive performance. This research not only provides a new methodology for financial time series forecasting but also offers a practical tool for investors and policymakers, assisting them in gaining a deeper understanding of and effectively navigating the complexities of financial markets.

* Corresponding author:Jiang-Cheng Li

Email addresses: lijiangch@ynufe.edu.cn; lijiangch@163.com (Jiang-Cheng Li), 2417117797@qq.com (Li-Ping Sun), 1643554454@qq.com (Xiao Wu), taoch1128@163.com (Chen Tao)

Preprint submitted to Elsevier *July 29, 2024*

Keywords: Time series forecasting, Machine learning, Deep learning, Hybrid prediction model, CEEMDAN, Informer, LSTM, BP, RNN, Stock index price

1. Introduction

Financial time series forecasting is crucial in the fields of financial asset pricing and risk management. On the one hand, accurate stock index predictions help regulatory agencies formulate financial policies more scientifically, better cope with market turmoil and fluctuations, and enhance the scientific and effectiveness of regulatory decisions. On the other hand, investors typically prefer financial products that can offer higher investment returns. Precise stock price predictions allow investors to better determine the ideal timing for buying and selling, thereby maximizing profits and reducing potential risks.

Among these, stock prices are one of the most critical concerns for investors, and stock price time series forecasting is a problem that many scholars are committed to solving and a skill that practitioners must learn. With the development of economic levels, the scale of the financial market continues to expand, leading to increased uncertainty in the stock market. The fluctuation of this uncertainty profoundly affects the volatility of the stock market, and stock market volatility, an indicator that serves as a guide in measuring risk, is often used as a key variable in many scholars' studies in both the fields of asset pricing and investment [1]. Such changes in the size and volatility of the stock market have made stock forecasting dramatically more difficult, in addition to the combined effects of exogenous market factors (political, economic, cultural, etc.) and endogenous company factors (company management structure, company size, company capital structure, etc.), resulting in strong volatility in the stock price series [2]. According to the different prediction methods, the existing methods can be categorized into three kinds, namely, traditional stock price prediction methods, machine learning prediction methods, and hybrid model prediction methods.

The classical methods of traditional stock price time series and forecasting analysis are primarily based on the integrated autoregressive moving average (ARIMA) model and its various versions. Many scholars typically apply a variety of traditional econometric models in traditional stock price forecasting methods [3]. The ARIMA model has been theoretically perfected by many scholars, and its practical application has been widely used in power, climate, medical, petrochemical, financial, and other directions. widely used in the direction of electricity, climate, medical, petrochemical, finance, etc., and has made significant contributions to the research of many industries. ARIMA can effectively process and forecast linear time series data and has universality in dealing with many situations [4, 5, 3, 6]. However, in practical applications, financial time series data tend to be complex and volatile, leading to stock prices that are characterized by high levels of noise and non-stationary nonlinearity[7]. Although some scholars have extended the model, for example, Dong et al. [8] has improved the method's ability to handle nonlinear time series by introducing a data window selection process prior to applying the ARIMA model. While this enhancement has somewhat increased the effectiveness of the ARIMA model, the prediction accuracy remains suboptimal. With the need to improve forecasting accuracy, most scholars in traditional measurement will increase the length of the time series, but this also leads to the increasing difficulty of traditional econometric model forecasting. Low and Sakk [9] in the study of ARIMA model and LSTM model, each demonstrates distinct advantages and disadvantages.

Academics are becoming more aware of how crucial it is to use machine learning approaches to address the issues that traditional forecasting methods have in predicting stock price series. This change in focus is being made in an attempt to get beyond the drawbacks of conventional techniques and to leverage machine learning's potential for more precise forecasting. Machine learning can capture the nonlinear patterns in time series data and, through training, can predict new data. Due to these characteristics, this predictive approach is now favored by many scholars, with various techniques such as KNN, SVM, BNN, LSTM, and Transformer. The LSTM is adept at capturing time series information in stock price prediction. By learning and memorizing data, LSTM effectively captures the long-term dependencies of stock prices. This characteristic makes LSTM able to get better performance in the increasingly complex stock price prediction nowadays[10, 11, 12]. Zhang et al. [13]improved the LSTM model in order to obtain better prediction results and proposed a CNN-BiLSTM-Attention model by combining CNN and BiLSTM. Empirical studies have found that the prediction accuracy of the LSTM model alone is inferior to that of the proposed new model. This study suggests that combining different models may yield more effective results. Lu et al. [2] Based on the full consideration of the correlation between the stock price data and the trend of the amount of change between the data, CNN and GRU are combined together, while the attention mechanism in behavioral finance is introduced into the model, and finally a model with stronger learning ability and better prediction fit is obtained, i.e., the CNN-Attention-GRU-Attention model. With the rise of machine learning methods, the Google team joined this research field and introduced a new neural network model in 2017 called the Transformer. The key feature of this model is its reliance entirely on the attention mechanism, eliminating the need for recursive and convolutional processes. Since its introduction, the Transformer model has become one of the most popular neural network models, following the LSTM model [14]. In a study comparing the Transformer model and the LSTM model, Greff et al. [15]found that the Transformer outperforms the LSTM in practical applications, excelling in both prediction accuracy and computational efficiency. However, there are still some problems with the Transformer model, which has high data requirements, poor interpretability, and limited learning ability for longer time series, so the Transformer model needs to be selected and optimized according to the task requirements and data characteristics. To address the challenges of using the Transformer model for long-term series prediction, Zhou et al. [16] developed the Informer model. This new model, which builds on the core principles of the Transformer, significantly enhances computational efficiency and fitting accuracy for long-term time series forecasting. Informer model Although it solves the problem of long series forecasting to a certain extent, as longer and longer time series bring more and more pressure to the model forecasting, this pressure will limit the research on time series forecasting.

Based on the above problems, many scholars have started to combine the respective advantages and disadvantages of traditional econometrics and machine learning in order to construct hybrid forecasting models to improve stock price forecasting abilities. In time series studies, it has been observed that as the length of the series increases, stock prices exhibit highly unstable fluctuations, leading to reduced model accuracy in predicting stock prices. In order to solve this problem, many scholars in the academic world have tried to split the data with different frequencies and then introduce the split data into the model, and the results show that this practice has improved the data analysis ability of the model. After conducting a thorough research and evaluation of the EMD model, Xuan et al. [17] suggested that EMD can alleviate the nonlinearity problem in stock series to a certain extent. This finding contributes significantly to the enhancement of scholarly understanding of stock series. Furthermore, Ali et al. [18] established that models such as EMD, CEEMD, and CEEMDAN are adept at decomposing high-volatility data into more manageable frequency components, thereby showcasing their reliability and utility in the analysis of data frequency. In recent years, the algorithms CEEMD and CEEMDAN have garnered greater attention within the domain of time series forecasting. Xian et al. [19] integrated the ICA and EEMD models to tackle the challenges posed by the nonlinearity and non-smoothness inherent in time series data. Yang et al. [20] improved the performance of CEEMDAN by employing Hermite interpolation and the preliminary calculation of residuals. This advanced approach, known as Improved CEEMDAN (ICEEMDAN), performs a quadratic decomposition of the high-complexity components, thereby making them more manageable and resulting in enhanced forecasting outcomes. Following the advent of methodologies like EMD, Jin et al. [21] combined EMD with investor sentiment analysis, thereby achieving improved predictive accuracy by methodically dissecting complex stock price series in a step-by-step manner. In the research conducted by Jothimani and Yadav [22], the author utilized a frequency decomposition algorithm to evaluate the predictive potential of the hybrid model. Their findings indicated that superior prediction results could be attained by incorporating frequency decomposition techniques into other machine learning models. In the use of hybrid models, numerous scholars have embarked on the integration of machine learning with frequency decomposition models, encompassing LSTM, Informer, and various other derivatives [23, 24]. The research conducted by Wang et al. [25] illustrated that time series data could be efficiently decomposed using hybrid models that integrated machine learning techniques, including LSTM, Informer, and frequency decomposition algorithms. This finding suggests that non-stationary time series pose greater challenges for model analysis as their volatility increases. The superiority of hybrid models lies in their ability to amalgamate the complementary attributes of various models, thereby enhancing forecast precision. This demonstrates how modeling becomes more difficult for non-stationary time series with increasing volatility. To improve model prediction accuracy, hybrid models can be used to combine the best features of each model. Rezaei et al. [26]proposed a new hybrid prediction algorithm, namely CEEMD-CNN-LSTM and EMD-CNN-LSTM, using the LSTM model and frequency decomposition method, which can mine the features of the time series to a greater extent compared to a single model, thus improving the analytical prediction ability of the model. In the study by Zhu et al. [27], it was found that CEEMDAN outperforms the EMD decomposition algorithm. They developed the CEEMDAN-SC-LSTM model by integrating the SG filter and LSTM, significantly enhancing the accuracy of stock index price predictions. In their study on stock prediction, Ren et al. [28] embedded the EF (Encoder Forest) into the Informer model to mitigate noise-related issues in longer stock price series. Concurrently, they decomposed the stock data into high and low frequency signals. The findings indicate that this hybrid approach exhibits greater reliability in terms of prediction accuracy and model robustness. In recent years, numerous scholars have proposed various hybrid models, underscoring their superior prediction accuracy, robust generalization capabilities, practicality, and suitability for addressing stock prediction challenges.

Based on the research conducted, this paper aims to organize the existing methods for price prediction, as demonstrated in Table 1. Upon reviewing the aforementioned research, it becomes evident that many scholars currently employ deep learning techniques for stock price prediction. Still, there are a few issues that need to be resolved. Firstly, most of the intrinsic modal functions (IMFs) decomposed by the CEEMDAN signal are directly utilized for subsequence prediction, disregarding the distinct frequencies present in each IMF. Consequently, employing the same parameters for prediction may result in a failure to capture the unique signal characteristics of each IMF effectively. Secondly, certain IMFs may exhibit rapidly changing trends, while others may display relatively smooth patterns. By utilizing only one model for prediction, the model might excessively focus on certain IMFs while neglecting other features during the learning process, which can potentially impact the overall performance of the model. Lastly, the emergence of the Informer model has successfully addressed the challenge of predicting long time series. However, the existing literature underutilizes Informer models, thus failing to fully explore their potential advantages in deep learning applications. Combined with the above problems, it has been observed that the combination of the CEEMDAN decomposition algorithm and LSTM has been extensively researched. Furthermore, the emergence of the Informer model, as highlighted in Zhou et al. [16] and Ren et al. [28], has shown improvements in prediction accuracy. Consequently, this paper proposes a novel hybrid model that integrates LSTM, Informer, and CEEMDAN decomposition methods for stock price prediction. Building upon this framework, we adopt the "decomposition-prediction-reconstruction" approach, wherein the stock price is decomposed into different sub-sequences using the CEEMDAN decomposition method. Subsequently, each sub-sequence is individually predicted using the designated model. Finally, the prediction results are reconstructed to generate the final prediction outcome.

In this paper, we propose a novel deep learning hybrid model called CEEMDAN-Informer-LSTM. The primary objectives of this research include the following contributions:

- The main focus of this research lies in the integration of the CEEMDAN, Informer, and LSTM models. This hybrid model aims to address the limitations of single models in terms of predictive performance and the challenges associated with capturing features from nonlinear data. By leveraging the strengths of each deep learning model, our proposed approach effectively captures signal features and enhances the overall predictability of the model.

- In this study, in order to enhance the accuracy and credibility of the predictions, the parameters for each run were carefully selected, and the mean value of 50 results was computed.

- The comparative analysis in this study involves the utilization of RNN, BP, Informer, CEEMDAN-BP, and CEEMDAN-LSTM models. The validity of these models is verified by forecasting data for different time horizons, including the next 5 days, 21 days, and 120 days. The proposed model holds potential for application in financial time series forecasting, providing valuable insights to investors for making informed decisions in the stock market.

The structure of this paper is as follows: Section 2 reviews prior research on using deep learning to forecast stock prices; Section 3 details the methodologies applied in our study; Section 4 presents the experimental results and analysis; and finally, Section 5 provides conclusions and suggestions for future research directions.

Article	Models	Types
Fan et al. [4]	ARIMA	Conventional forecasting models
Liu et al. [5]	SARIMA	Conventional forecasting models
Dong et al. [8]	ARIMA	Conventional forecasting models
Xu et al. [6]	ARIMA, GARCH, OLS	Conventional forecasting models
Xian et al. [19]	EMD-ICA	Conventional forecasting models
Ren et al. [10],Swathi et al. [11]	LSTM	Machine Learning Predictive Models
Wu et al. [12]	LSTM	Machine Learning Predictive Models
Zhang et al. [13]	CNN-BiLSTM	Machine Learning Predictive Models
Lu et al. [2]	CNN-GRU	Machine Learning Predictive Models
Greff et al. [15]	Transformer	Machine Learning Predictive Models
Zhou et al. [16]	Informer	Machine Learning Predictive Models
Xuan et al. [17]	EMD-LSTM-CSI	Hybrid Predictive Models
Ali et al. [18]	CEEMD-RF-KRR	Hybrid Predictive Models
Jin et al. [21]	EMD-LSTM	Hybrid Predictive Models
Jothimani and Yadav [22]	CEEMDAN-SVR	Hybrid Predictive Models
Yan et al. [24]	CEEMD-PCA-LSTM	Hybrid Predictive Models
Rezaei et al. [26]	CEEMD-CNN-LSTM	Hybrid Predictive Models
Chen et al. [23]	CEEMD-SE-ICA-PSO-LSTM	Hybrid Predictive Models
Zhu et al. [27]	CEEMDAN-SC-LSTM	Hybrid Predictive Models
Ren et al. [28]	CEEMDAN-Informer-EF	Hybrid Predictive Models

Conventional forecasting models: conventional techniques for predicting stock prices.
Machine Learning Predictive Models: machine learning stock price prediction methods.
Hybrid Predictive Models: hybrid models combine machine learning and other traditional stock price prediction methods.

Table 1: Models for predicting stock prices

2. Related methods

2.1. CEEMDAN Moedl

The empirical modal decomposition (EMD)-based CEEMDAN model, also known as adaptive noise-complete ensemble empirical modal decomposition, is a signal processing technique primarily utilized for the study of nonsmooth and nonlinear data. It adds adaptive noise to improve handle signal noise, taking cues from EMD. The following is the algorithmic principle:

(1) Let $x(t)$ be the input raw signal, and let M be the number of iterations.

(2) For each iteration (m = 1, 2,..., M), generate a random noise signal. The noise generation method can be a uniform distribution, a Gaussian distribution, etc., depending on the implementation of CEEMDAN.

$$y^m(t) = x(t) + n^m(t) \tag{1}$$

(3) It will produce a sequence of intrinsic modal functions (IMFs) and a residual term after m iterations of EMD decomposition:

$$y^m(t) = \sum_{i=1}^{N} c_i^m(t) + r^m(t) \tag{2}$$

$c_i^m(t)$ represents the i-th IMF, $r^m(t)$ denotes the residual term.

(4) For every iteration, subtract the corresponding signal noise from each IMF $c_i^m(t)$.

$$h_i^m(t) = c_i^m(t) - n^m(t) \tag{3}$$

Subtract the corresponding signal noise from each IMF $c_i^m(t)$for each iteration.

(5) Repeat steps (2) through (4) until the predefined number of iterations (M) is reached. Calculate the residual term and the average of each IMF over the course of all iterations.

$$h_i(t) = \frac{1}{M}\sum_{m=1}^{M} h_i^m(t) r(t) = \frac{1}{M}\sum_{m=1}^{M} r^m(t) \tag{4}$$

(6) Finally, the CEEMDAN model is obtained by:

$$CEEMDAN = \sum_{i=1}^{N} h_i(t) + r(t) \tag{5}$$

2.2. LSTM Model

The core concept of LSTM revolves around the cell state and the "gate" mechanisms. The cell state serves as a pathway for information transfer, enabling data to be transmitted throughout the sequence. You can think of it as the network's "memory." The memory gate, controlled by a signal, determines whether the information should be retained.

(1) The role of the forgetting gate is mainly to decide which old information needs to be removed from the cell state and keep the cell state valid. Based on the filtering results, the forgetting gate retains useful information in the cell state and removes useless information, thus optimizing the content of the cell state to keep it useful for the current task.

$$f_t = \delta(W_f \cdot [h_{t-1}, x_t + b_f]) \tag{6}$$

where h_{t-1} denotes the output value at the previous moment, δ is the activation function of the nervous system, and f_t is the forgetting information.

(2) Input gates play a crucial role in determining what new information should be incorporated into the cell state to update long-term memory. Utilizing an activation function (typically sigmoid) to evaluate incoming data to determine whether to incorporate it into the cell state.

$$i_t = \delta(W_i \cdot [h_{t-1}, x_t] + b_i]) \tag{7}$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_c]) \tag{8}$$

where i_t represents the information to be stored in the cell state, tanh denotes the activation function, and $\widetilde{C}_t$ represents the new information.

(3) The main role of the output gate is to determine what information is extracted from the cell state as the output of the current time step.

$$O_t = f_t * C_{t-1} + i_t * \widetilde{O_t} h_t = O_t * C_t \tag{9}$$

where h_t and O_t respectively represent the current time-step output value and the current time-step cell state.

2.3. Informer Model

Informer is a deep learning model for time series prediction that was presented at AAAI 2021. This model effectively solves the computational complexity and accuracy problems in long time series prediction by combining the self-attention mechanism and the probabilistic sparse attention mechanism. Informer has demonstrated excellent performance on several public datasets and has become the deep learning field for time series. The informer has the following unique features:

(1) The ProbSparse self-attention mechanism

The difference between the distribution $p(k_j|q_i)$ of a query and the uniform distribution $q(k_j|q_i)$ can be measured by the KL dispersion Eq. 10. By removing the constant, the sparsity metric of the ith query can be defined as Eq. 11.

$$KL(q \parallel p) = \ln \sum_{l=1}^{L_K} exp(\frac{q_i k_l^T}{\sqrt{d}}) - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}} - lnL_K \tag{10}$$

$$M(q_i, K) = \ln \sum_{j=1}^{L_k} exp(\frac{q_i k_i^T}{\sqrt{d}}) - \frac{1}{L_k} \sum_{j=1}^{L_K} \frac{q_i k_i^T}{\sqrt{d}} \tag{11}$$

$\ln \sum_{j=1}^{L_k} exp(\frac{q_i k_i^T}{\sqrt{d}})$ is the LSE operation; $\frac{1}{L_k} \sum_{j=1}^{L_K} \frac{q_i k_i^T}{sqrtd}$ is the arithmetic mean. For all queries, a number of queries with $M(q_i, K)$ ranked in the top u are selected as $\hat{Q}$, where $u = clnL_Q$ and c is a constant sampling factor, then the process of ProbSparse Self-attention is denoted as:

$$A(Q, K, V) = soft\max(\frac{\hat{Q}K^T}{\sqrt{d}})V \tag{12}$$

(2) Self-Attention Distillation Technique

To address the problem of inclusion redundancy in the algorithm, Informer significantly reduces the input time dimension by introducing a self-attention distillation operation to extract the main self-attention.

$$X_{j+1}^t = MaxPool(ELU(Convld([X_j^t]_{AB}))) \tag{13}$$

The equation $[\cdot]_{AB}$ denotes Multi-head ProbSparse Self-attention and other necessary operations (including Add, LayerNorm, FFN, etc.), and $Convld$ denotes the execution of a one-dimensional convolution in the temporal dimension.

(3) Generative decoder

Informer proposes a generative reasoning process to increase the speed of reasoning, specifically, the The input to the decoder is:

$$X_{feedde}^t = Concat(X_{token}^t, X_0^t) \in \mathbb{R}^{(L_{token}+L_y)\times d_{model}} \tag{14}$$

where 0^t represents a placeholder (predicted value); $X_{token}^t \in \mathbb{R}^{(L_{token}+L_y)\times d_{model}}$ denotes the starting “token,” which is a straightforward and effective setup.

3. Proposed methodology

The CSI 300 index is examined in this study, primarily based on historical stock index closing prices. The deep learning approach is utilized to forecast historical stock closing prices, overcoming the drawbacks of the single stock prediction method and further enhancing stock prediction accuracy. The hybrid model presented in this research incorporates a deep learning hybrid model of CEEMDAN, Informer, and LSTM, which is based on the previously proposed theory. Fig. 1 illustrates the CEEMDAN-Informer-LSTM deep learning hybrid model framework, which is primarily made up of three modules:

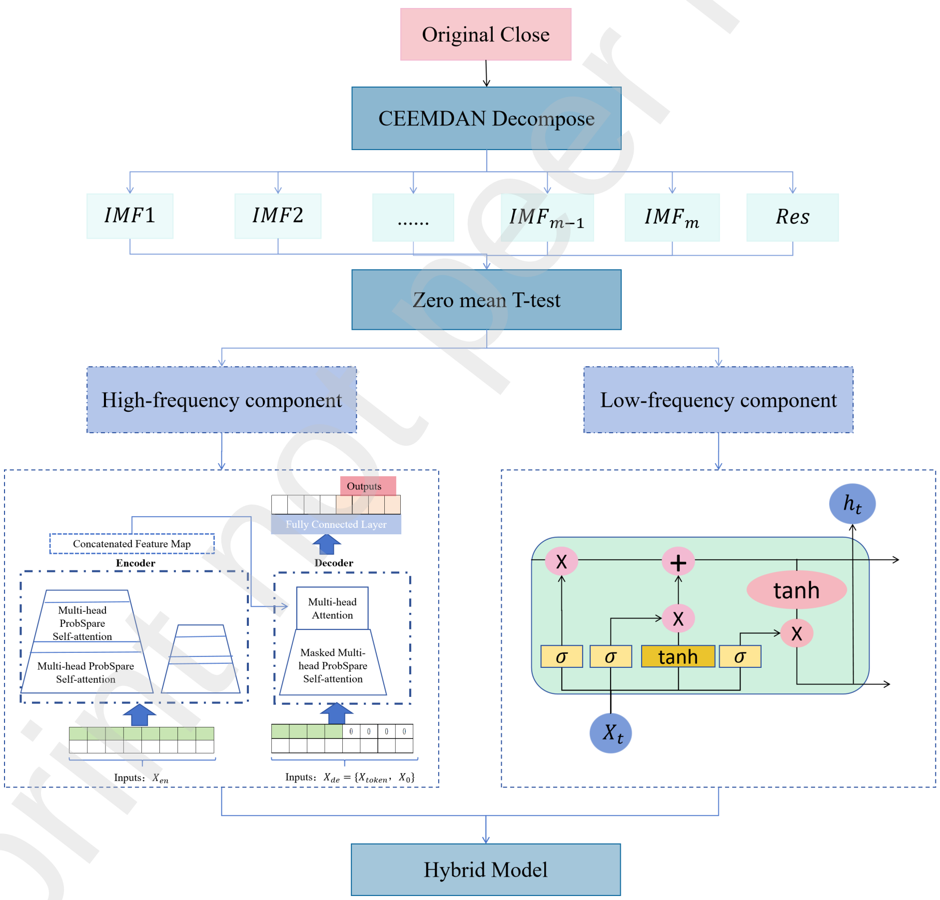


Figure 1: Hybrid Model

The first module is the data preprocessing part using CEEM-

DAN decomposition. The closing price is preprocessed by CEEMDAN to obtain the IMF and residual term. Considering the similar complexity and correlation of IMFs after decomposition, the IMFs and residual terms are categorized into high frequency and low frequency, and the high frequency is defined as High-IMF (H-IMF) and the low frequency as Low-IMF (L-IMF). Using the idea of fine-to-coarse reconstruction proposed by Zhang et al. [29], i.e., the zero-mean T-test, the IMFs obtained after decomposition are analyzed using the zero-mean T-test. method, and the IMF components obtained after decomposition are reconstructed. The key to the zero-mean t-test is to find the IMF components that switch from high to low frequencies. The fundamental principle is that IMF components extracted through CEEMDAN decomposition are organized based on their frequency, with higher frequencies listed first. According to the T-test results, the initial component showing significant structural changes is labeled as IMF_i. Because of the independence and orthogonality between each IMF component, the IMF of each part can be distinguished, i.e., H-IMF and L-IMF can be classified.The components before IMF_i are the high-frequency parts, and those before IMF_i and after are the low-frequency parts. A one-sample t-test is done for each IMF component, where the T-statistic is calculated as:

$$t = \frac{\bar{X} - \mu_0}{\sqrt{\frac{S}{n-1}}} \tag{15}$$

The original hypothesis H0: The average of the IMF component is 0; alternative hypothesis H1: the mean of the IMF component is not 0. The original hypothesis is supported if the p-value is less than the given confidence interval. Analytical steps for hypothesis testing:

(1) Null hypothesis H_0: The average of the IMF component is 0.; Alternative hypothesis H_1: The average of the IMF component is not 0.

(2) Perform a one-sample T-test for each IMF component.

(3) Calculate and test the selected statistics to obtain the P-value.

(4) Determine the level of significance α. If $p < \alpha$, reject the original hypothesis. Conversely, accept the original hypothesis. Find the first component IMF_i whose mean is significantly non-zero based on the

p-value judgment, then IMF_i and the subsequent components are L-IMF.

The second module is Informer and LSTM models to predict H-IMF and L-IMF, respectively. After preprocessing with CEEMDAN in the previous step, the divided H-IMF and L-IMF are predicted using Informer and LSTM, respectively. It has been observed that Informer is more effective in predicting H-IMF, while LSTM performs better in predicting L-IMF. Consequently, this paper utilizes these two models for their respective predictions.

(1) Zero-mean normalization: after decomposition, it is necessary to preprocess the data. This article uses zero-mean normalization to scale the data for the decomposed eigencomponents, and the formula for normalization is Eq. 16:

$$x^* = \frac{x - \mu}{\delta} \tag{16}$$

where σ represents the standard deviation, δ represents the average of the input data with the corresponding feature, and x and x^* represent the data prior to and following normalization under the feature, respectively. Following the prediction, it can be reduced using the following reduction formula to the value of the original data space:

$$\widehat{x} = x^* \times \delta + \mu \tag{17}$$

where $\widehat{x}$ represents the output of the prediction model.

(2) This study mainly utilizes Informer and LSTM models as the main deep learning hybrid models. Detailed specifications of their main parameters are presented in Table 2. The results predicted by the model are random in nature, on the one hand, due to the Informer and LSTM model in the training of the "shuffle" operation, that is, each time from a random location to select the batch (batch) as an input to train the model and repeat many times. On the other hand, the initial parameters of the random assignment characteristics can lead to the prediction of the deep learning neural network. The random assignment of initial parameters, on the other hand, can lead to fluctuations in the prediction results. To enhance the robustness of the prediction results, this study will repeat the process 50 times, and the final outcomes will be averaged across these 50 predictions.

The third module aggregates and sums the H-IMF and L-IMF predicted by Informer and LSTM, respectively, to obtain our final

prediction.

The specific computational procedures for the three modules can be described as follows:

- Module 1 **Data preprocessing using CEEMDAN decomposition**:
  - Step 1 **CEEMDAN decomposition preprocessing**:
    * Preprocess the closing price using CEEMDAN decomposition to obtain IMFs and residuals (Res).
  - Step 2 **Separate H-IMF and L-IMF**:
    * The IMF and Res results obtained after decomposition are divided into H-IMF and L-IMF.
    * Utilize the zero-mean T-test method to reconstruct the IMF components obtained after CEEMDAN decomposition, identify the first component where a significant structural change occurs, denoted as IMF_i, then IMF_i and subsequent components are classified as L-IMF and the previous ones as H-IMF.

  Informer and LSTM models to predict H-IMF and L-IMF:
  - Step 3 **Zero-mean normalization**:
    * Normalize the segmented feature components and scale the data.
    * After prediction, restore the predicted values to the original data space.
  - Step 4 **Model Prediction and Parameter Setting**:
    * Prediction of segmented H-IMF and L-IMF using Informer and LSTM models.
    * Set the main parameters as shown.
    * To enhance the result's robustness, repeat the process 50 times and take the average of the 50 predicted values.
- Module 3 **Summarize Predictions**:
  - Step 5 Sum the H-IMF and L-IMF predicted by Informer and LSTM, respectively, to obtain the final prediction value.

The procedure for calculation algorithm mentioned above can be written as Algorithm1.

Hyper parameters	**Explanation**	**Value**
Informer		
Features	Specify the type and method of forecasting	S
Target	Specify the forecast target column	close
Seq_len	Sequence length of the input encoder	96
Label_len	Enter the sequence length of the decoder	48
Pred_len	Length of the predicted sequence	1
D_model	Specify the features of the transform's hidden layer	256
Enc_in	Input encoder data dimensions	1
Dec_in	Dimension of the input decoder data	1
D_out	Specify the predicted value	1
E_layers	Number of layers in the encoder	2
D_layers	Number of layers in the decoder	1
Attn	Methodology for calculating the attention mechanism	prob
Batch_size	Number of samples used in each training of the model	32
LSTM		
Look_back	Step size	20
Pred_len	Predicted sequence length	1
Batch_size	Number of samples used in each training of the model	32
Param input_size	Input feature dimension	1
Param	Dimension of output quantity	1
Param cell_size	Dimension of cell state and hidden state vectors in a cell	4
Epochs	Number of times a complete traversal of the training dataset was completed	100
Droupt		0.1

Table 2: Hyper parameters

Algorithm 1: Pseudocode algorithm for the proposed method.

```
Input   : Closing price sequence
Output: Final prediction value
Result: Predicted value

Function CEEMDAN decomposition preprocessing(Closing price sequence):
    Perform preprocessing on the closing price using CEEMDAN
      decomposition to obtain IMFs and Residue (Res) components;
    H-IMF and L-IMF division of IMF and Res;
    Apply the zero-mean T-test method for reconstruction to identify the
      first IMF component where a significant structural change occurs,
      denoted as IMF_i, then IMF_i and subsequent components are
      classified as L-IMF and the previous ones as H-IMF;
Function Data normalization processing(Partitioned IMF components):
    Zero-mean normalization of the divided IMF components scales the
      data;
    Upon prediction completion, revert the predicted values to the
      original data space values;
Function Model prediction(Standardized IMF components):
    Prediction of segmented H-IMF and L-IMF using Informer and
      LSTM models;
    Set the main parameters;
    To enhance the robustness of the prediction results, this study will
      repeat the process 50 times, and the final outcomes will be averaged
      across these 50 predictions.;
Function Summarize prediction results(Predictions from the Informer and LSTM models):
    The H-IMF predicted by Informer and the L-IMF predicted by LSTM
      are combined and summed to derive the final prediction value.;

CEEMDAN decomposition preprocessing(Closing price sequence);
Data normalization processing(Partitioned IMF components);
Model prediction(Normalized feature components);
Summarize prediction results(Informer and LSTM prediction results);
```

4. Experimental Results and Analysis

4.1. Datasets

This study adopts the CSI 300 index as the research object for two reasons: the first reason is that the CSI 300 index includes 300 large-scale and liquid listed companies in Shanghai and Shenzhen, covering most of the industries in the market, which can reflect the overall trend of the A-share market in a more comprehensive way, and as a more mature and stable index, the historical data is more complete, which is easy to compare with other indices or assets. As a more established and stable index, historical data is more comprehensive, facilitating comparisons with other indices or assets and contributing to enhancing the accuracy and reliability of forecasts. The second reason is that the constituent stocks of the CSI 300 Index have high liquidity and are actively traded, with a large number of market participants and relatively abundant relevant information and data, which makes it easy to conduct analyses and forecasts.

The whole data uses trading data from January 4, 2010 to December 29, 2023, for a total of 3,401 samples. The data comes from the Cathay Pacific CSMAR database, and all models are run using Pycharm under Python 3.11. This article uses 2010-2022 data, totaling 3159 sample data, of which the first 90% is the training set and 10% is the validation set, using 2023 trading data as the test set, totaling 242 sample data.

Table 3 presents the analysis of the CSI 300 index's descriptive statistics. Among these findings, the Jarque-Bera statistics applied to closing prices reveal that the null hypothesis, which posits the absence of autocorrelation up to the 10th order, is rejected at the 1% significance level. This outcome implies the existence of serial correlation within the data. Furthermore, the initial hypothesis of normal distribution is rejected at the 1% significance level, signifying a deviation from normality in the series.

4.2. Loss function

When introducing a new prediction model, it's common to assess its performance by comparing its predictive accuracy with that of other prediction models. To gauge the quality of the model, this paper employs evaluation metrics, including mean absolute error (MAE), mean absolute percentage

	open	close	high	low	volume	turnover
max	5922.07	5807.72	5930.91	5747.66	6.86×10^6	9.49×10^7
min	2079.87	2086.97	2118.79	2023.17	2.19×10^5	2.12×10^6
mean	3490.93	3493.11	3519.25	3462.76	1.18×10^6	1.69×10^7
std	824.97	824.68	830.37	817.15	8.20×10^5	1.30×10^7
median	3458.83	3463.64	3486.82	3435.69	9.90×10^5	1.27×10^7
Skewness	0.2839***	0.2814***	0.2838***	0.2767***	2.5046***	1.7553***
Kurtosis	-0.5510***	-0.5537***	-0.5452***	-0.5696***	8.7795***	4.2270***
J-B	88.8300***	88.2867***	87.7479***	89.3331***	1.4426×10^4***	4.2770×10^3***
Q 10	3.3283×10^3***	3.3300×10^4***	3.3359×10^4***	3.3272×10^4***	2.5051×10^4***	2.7967×10^4*** ***

Note: Q(10) is the L-B statistic for up to 10th-order serial correlation. *, **, and *** denote rejection of the original hypothesis at the 10%, 5%, and 1% significance levels, respectively.

Table 3: descriptive statistics

error (MAPE), root mean square error (RMSE), and coefficient of determination (R^2). Their calculations are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{x}_i - x_i\right|, \tag{18}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{x}_i - x_i}{x_i}\right|, \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}, \tag{20}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}{\sum_{i=1}^{n}(\bar{x}_i - x_i)^2}. \tag{21}$$

4.3. CEEMDAN Decomposition

When decomposing the original time series using the CEEMDAN model, the closing prices are separated into eight IMFs and one RES. The following figure 2 plots the original closing price series as well as the decomposed IMFs and residual terms. It can be observed that the frequency of the IMF decreases progressively as the decomposition order increases within the same time frame. The frequency of change is different for each component and is significantly different from the other sequences.

The fluctuation trends of the decomposed IMFs become clearer and tend towards stability from top to bottom. Each IMF contains its own representative information. The IMFs with rapid fluctuations have shorter time intervals between adjacent peaks and troughs, primarily influenced by recent market supply and demand factors. In contrast, the IMFs with gentle fluctuations have longer time intervals between adjacent peaks and troughs, mainly influenced by economic conditions and official policies over some time. The residual series, reflecting the long-term trend of the stock market, aligns with the overall trend variations observed in the original time series data.

After decomposing the original stock index prices using CEEMDAN, considering the similar complexity and correlation among the decomposed IMFs, it is beneficial to classify IMFs of the same type. This not only helps to better understand the unique characteristics and frequencies of each IMF but also reduces the workload for individual IMF predictions. Therefore, the next step is to perform the null-mean hypothesis to divide the H-IMF and L-IMF.

4.4. Zero-mean T-test

The results of the zero mean t-tests for IMF1-IMF8 and RES are tabulated in Table 4. The one-sample t-test statistics for the IMF1 to IMF5 components are close to 0 and the p-values are all greater than the significance levels of 10%, 5%, and 1%, which means that the original hypothesis is supported; starting from IMF6 to the IMF8 component, the one-sample t-test statistics are significantly not 0 and the p-values are all less than the significance level of 10%, indicating the rejection of the original hypothesis. values are significantly non-zero and the p-values are all less than the significance level of 10%, which indicates the rejection of the original hypothesis, that is, IMF6 is the first IMF component whose mean value is significantly non-zero. Therefore, IMF1 to IMF5 components can be considered as H-IMF, and IMF6 to IMF8 as H-IMF. In this paper, RES is also categorized as L-IMF because the one-sample t-test statistic of Res is significantly not 0, the p-value is less than the significance level of 0.1, and its fluctuation is relatively smooth and roughly consistent with the original series, which is reflective of the overall trend. Categorizing Res as L-IMF for forecasting can improve the forecasting process. IMF for forecasting can improve the accuracy of forecasting.

4.5. Selection of models to build hybrid models

Most studies have used the same model to predict sub-sequences without recognizing the unique characteristics and frequencies of each IMF component, thus failing to establish models with more suitable parameters for each.

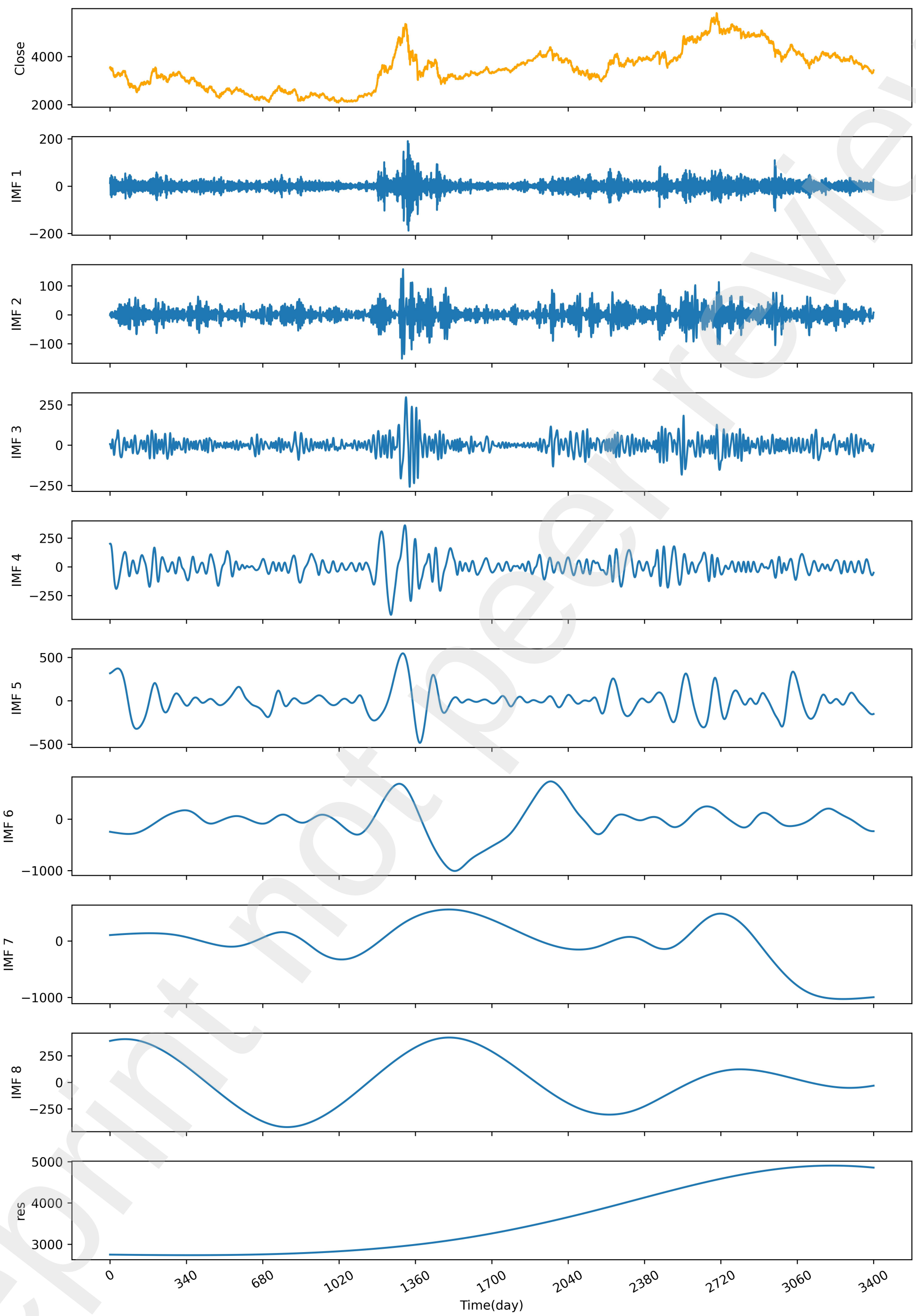


Figure 2: CEEMDAN model

IMF	Mean	Standard Deviation	Degrees of Freedom	T-value	P-value
IMF1	0.3652	25.0263	3400	0.8510	0.3948
IMF2	0.1472	27.1274	3400	0.3156	0.7517
IMF3	-0.5589	47.0719	3400	-0.6924	0.4887
IMF4	0.2127	82.9694	3400	0.1495	0.8811
IMF5	-3.5354	145.6004	3400	-1.4160	0.1569
IMF6	-37.5085	317.6531	3400	-6.8862	0.000***
IMF7	-37.2895	401.4903	3400	-5.4165	0.000***
IMF8	7.8139	244.0138	3400	1.8675	0.0619*
RES	3563.4661	811.9025	3400	255.9600	0.000***

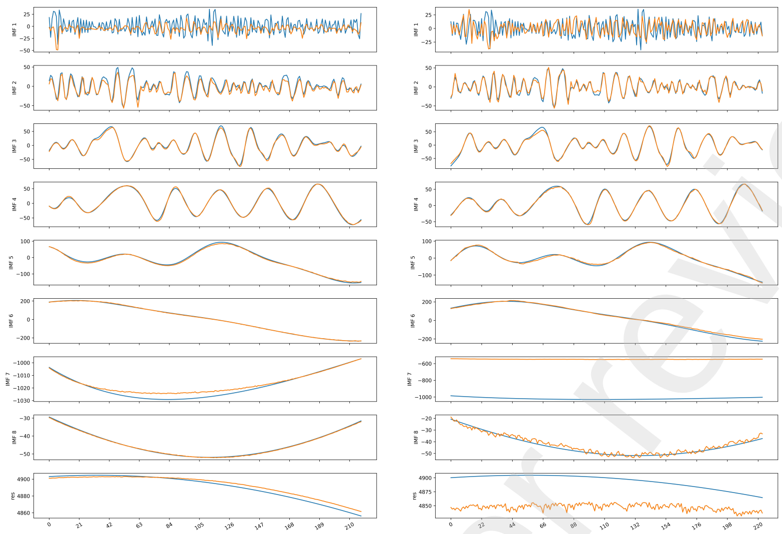
Note: *, **, and *** indicate rejection of the null hypothesis at the significance levels of 10%, 5%, and 1%, respectively.

Table 4: Zero mean T-test

Some components may exhibit rapid fluctuations, while others may have relatively stable trends. This variability might lead the model to excessively focus on certain components while neglecting others during the learning process, thereby impacting the overall model performance. Therefore, in this paper, LSTM and Informer models are employed to predict the feature components IMF1-IMF8 and RES, respectively. From Fig 3, as the frequency of decomposition decreases, the predictive accuracy of the model increases.

Table 5 The results show that the Rmse and MAE of Informer in predicting IMF1-IMF5 are significantly smaller than those of LSTM, and the Informer model is more advantageous in predicting H-IMF and unstable in predicting L-IMF, while LSTM is more advantageous and stable.

The information contained in the H-IMF is much more centralized, which leads to this part of the information being more difficult to predict, whereas the Informer model uses a one-time output result, which is much more conducive to a more centralized output of high-frequency information. LSTM excels at capturing prolonged dependencies within time series data. However, the low-frequency components extracted by CEEMDAN decomposition often contain information about long-term trends and periodicities in the time series. Combining the two enhances the ability to capture the persistent dynamic features of the data. Consequently, this study utilizes the Informer model for predicting the H-IMF and the LSTM model for forecasting the L-IMF, with the objective of improving the precision of stock index price predictions.



(a) CEEMDAN-LSTM (b) CEEMDAN-Informer

Figure 3: Comparison Chart of Predictive Performance of IMF Component Models

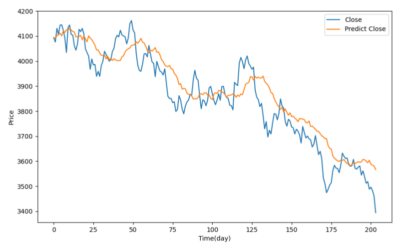
Informer			LSTM	
	MAE	RMSE	MAE	RMSE
IMF1	9.5621	12.7499	11.8074	14.8754
IMF2	3.8798	5.0405	5.8516	7.1732
IMF3	2.6596	3.7049	2.8463	3.4995
IMF4	1.6855	2.1698	1.8408	2.2660
IMF5	3.8069	4.5740	4.0754	4.7767
IMF6	8.5679	10.8869	1.7089	2.2293
IMF7	427.94	428.0769	2.0700	2.7291
IMF8	1.6526	1.9782	0.2022	0.2275
RES	27.2273	36.5503	2.8977	3.3292

Table 5: Performances of different Informer and LSTM models

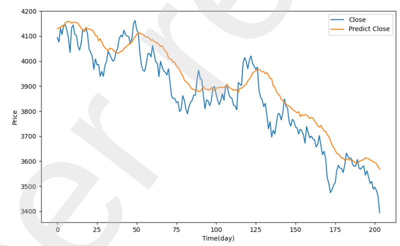
4.6. Model Predictive Performance

4.6.1. Single model prediction performance

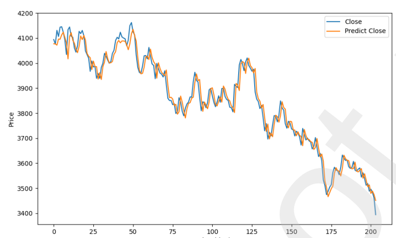
Since the sliding time window built into Informer is different from other methods, the result image obtained by Informer is different from other methods, and this paper demonstrates this by selecting the common parts to be plotted, and Fig. 4 plots the prediction performance results of the BP, RNN, LSTM, and Informer's single model. The results of the four evaluation indicators for the individual models are shown in Table 6, and it is found through comparison that Informer has the smallest MSE, RMSE, and MAPE of 27.0098, 34.0771, and 0.6945, respectively, and its R^2 is optimal at 0.9689. The single model prediction performance follows the order: RNN $<$ BP $<$ LSTM $<$ Informer.



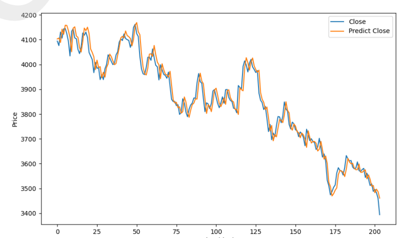
(a) BP Model



(b) RNN Model



(c) LSTM Model



(d) Informer Model

Figure 4: Comparison Chart of Predictive Performance of Single Models

	MAE	RMSE	MAPE	R2
Informer	27.0098	34.0771	0.6945	0.9689
LSTM	42.2656	52.9808	1.1169	0.9390
BP	65.9116	81.1685	1.7273	0.8686
RNN	72.8916	87.7741	1.8961	0.8463

Table 6: Table of Predictive Performance Results for Single Models

4.6.2. Hybrid model prediction performance

Figure 5 displays the plots of true values and predictions for the CEEMDAN-LSTM, CEEMDAN-BP, CEEMDAN-RNN, and CEEMDAN-Informer models. It can be observed that CEEMDAN-LSTM achieves the best prediction performance, with MAE, RMSE, MAPE, and R^2 values of 16.9274, 20.0831, 0.4034, and 0.9935, respectively. On the other hand, CEEMDAN-Informer exhibits the poorest prediction performance, mainly due to instability during the prediction of IMF components after CEEMDAN decomposition, which is consistent with the instability observed in the previous step when predicting low-frequency components with Informer. Therefore, not all models are suitable for CEEMDAN decomposition, and it is necessary to select appropriate algorithms based on the characteristics of the model.

Based on the previous experiments where Informer performed better on high-frequency components and LSTM performed better on low-frequency components, we propose the CEEMDAN-Informer-LSTM hybrid model. Combining the results from Table 7, it can be concluded that the proposed CEEMDAN-Informer-LSTM model achieves the best prediction performance, with MAE, RMSE, R2, and MAPE values of 11.6765, 15.6333, 0.9937, and 0.3004, respectively. In terms of the MAE loss function alone, compared to the single models Informer, LSTM, BP, and RNN, the CEEMDAN-Informer-LSTM model reduces the prediction errors by 56.77%, 72.37%, 82.28%, and 83.98%, respectively. Regarding the hybrid models CEEMDAN-Informer, CEEMDAN-LSTM, CEEMDAN-BP, and CEEMDAN-RNN, the reduction in prediction errors is 97.26%, 31.02%, 90.21%, and 83.17%, respectively. The CEEMDAN-Informer-LSTM model is separately depicted in Figure 6, showing a high degree of fit.

	MAE	RMSE	MAPE	R2
CEEMDAN-Informer	425.7388	430.4525	11.2076	-4.2353
CEEMDAN-LSTM	16.9274	20.0831	0.4034	0.9935
CEEMDAN-BP	119.2788	132.7667	3.1999	0.6484
CEEMDAN-RNN	69.3736	82.0239	1.8665	0.8658
CEEMDAN-Informer-LSTM	11.6765	15.6333	0.3004	0.9937

Table 7: Table of Predictive Performance Results for Hybrid Models

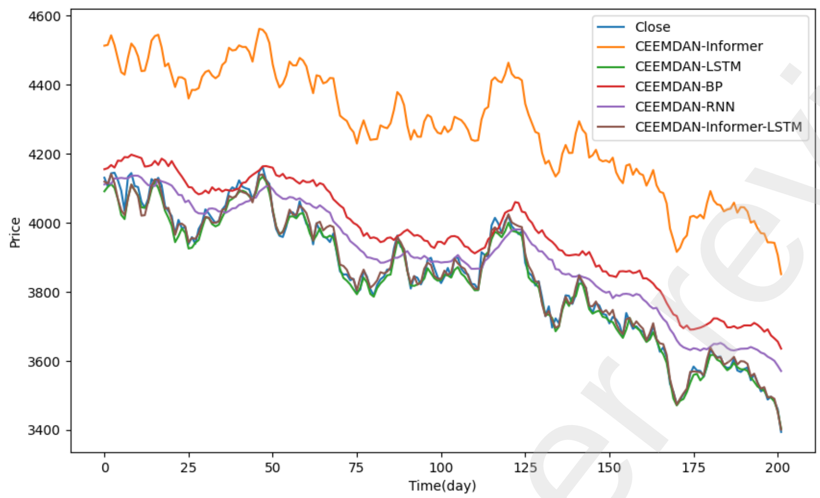


Figure 5: Comparison Chart of Predictive Performance for Hybrid Models

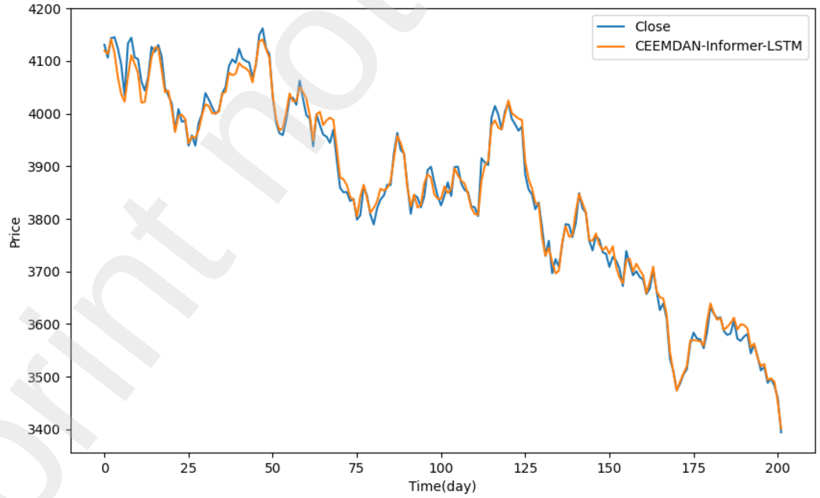


Figure 6: CEEMDAN-Informer-LSTM

4.7. Robustness test

Investors are interested in the models' medium- and long-term forecasting capabilities in addition to their short-term predicting ability. Therefore, the predictive accuracy of both the foundational and expanded models in forecasting future periods of 5 days, 21 days, and 120 days was further evaluated. The specific results are presented in Table 8. Among the single models, the Informer model demonstrates superior performance in forecasting future outcomes, particularly with regards to the 120-day forecast, where the MAE and RMSE are 273.1295 and 329.6211, respectively. CEEMDAN-Informer-LSTM hybrid model The evaluation metrics of (36.2807, 28.4072), (71.0481, 59.6087), and (94.4785, 83.5899) for 5, 21, and 120 days, respectively, are smaller than the other comparative models. It is evident that the proposed CEEMDAN-Informer-LSTM approach excels in forecasting the closing price of the CSI 300 stock index.

	5day		21day		120day	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Informer	77.3748	63.9467	167.5446	140.7500	329.6211	273.1295
LSTM	86.7430	70.3946	123.9958	103.2909	338.2312	323.7341
BP	99.4248	81.8987	145.8042	124.4356	389.7584	371.3363
RNN	108.4438	91.1005	161.5632	140.5236	419.6195	403.2629
CEEMDAN-Informer	407.9596	406.8264	479.9598	473.8614	828.6900	820.5524
CEEMDAN-LSTM	37.5011	30.1156	77.4444	63.6056	79.7334	61.4698
CEEMDAN-BP	154.5938	135.8024	212.3062	189.8113	468.8149	451.8859
CEEMDAN-RNN	105.6889	89.1731	160.4459	138.0351	404.9762	386.1863
CEEMDAN-Informer-LSTM	36.2807	28.4072	71.0481	59.6087	94.4785	83.5899

Table 8: Robustness Test of Various Models at Different Time Periods

5. Conclusion and Outlook

In this paper, the CEEMDAN-Informer-LSTM deep learning hybrid model is proposed to be applied in financial time series forecasting. And BP, RNN, LSTM, Informer, CEEMDAN-BP, CEEMDAN-RNN, CEEMDAN-LSTM, CEEMDAN-Informer models are selected for comparison. Four loss functions, MAE, RMSE, MAPE, and R^2, are used to validate the model performance, and it is found that the Informer model outperforms LSTM, BP, and RNN by comparing the results of the single models. The proposed CEEMDAN-Informer-LSTM model has the best prediction among the hybrid models and outperforms all the single-model predictions. Results. It is also concluded that not all models are suitable for CEEMDAN signal decomposition, and a suitable decomposition algorithm needs to be selected for the model based on its performance. In addition, this paper verifies the robustness of the hybrid model by comparing the results predicted for the next 5, 21, and 120 days.

Overall, the proposed CEEMDAN-Informer-LSTM deep learning hybrid model has the following advantages: First, the CEEMDAN signal decomposition algorithm is used to preprocess the data. The CEEMDAN algorithm can effectively decompose the original signal into IMFs. However, there are similar complexities and correlations among the IMFs obtained by decomposition. In this paper, we fully utilize the IMF similarity and correlation and use a zero-mean t-test on the obtained IMFs to classify the IMFs into H-IMFs and L-IMFs. This classification method can help with the next step of prediction using the deep learning model. Secondly, IMFs are predicted using Informer and LSTM models, respectively, and it is found that Informer is suitable for predicting H-IMFs and LSTM is suitable for predicting L-IMFs, and the selection of suitable parameters for the models for IMFs complements the application of Informer and LSTM models in stock market prediction. Thirdly, it provides decision-making references for investors. Accurate stock price forecasting enables investors to better determine the ideal timing for buying and selling, thus maximizing returns and reducing potential risks.

Regarding future work on price prediction, more factors affecting index prices, such as unforeseen events, socio-political environments, investor sentiment, etc., could be incorporated into the models to broaden the dimensionality of the input data. Deep learning is a rapidly evolving field, and while this study focused on representative and classical models, other deep learning models may yield better predictive results, presenting avenues for future research. Finally, deep learning theories can be applied not only to stock markets but also to derivative markets like futures and options or currency markets like bonds, offering opportunities for further study on application methods and effectiveness.

6. Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No.: 72163035), the Ministry of education of Humanities and Social Science project of China(Grant No.: 19YJAZH045), Applied Basic Research Foundation of Yunnan Province (Grant No.: 202401AT070276), Economic Complexity and Digital Fintech Innovation Team of Education Department of Yunnan Province.

References

M. Ghani, Q. Guo, F. Ma, T. Li, Forecasting pakistan stock market volatility: Evidence from economic variables and the uncertainty index, International Review of Economics & Finance 80 (2022) 1180–1189.

W. Lu, J. Li, J. Wang, L. Qin, A cnn-bilstm-am method for stock price prediction, Neural Computing and Applications 33 (2021) 4741–4753.

J. Wang, Q. Cui, X. Sun, M. He, Asian stock markets closing index forecast based on secondary decomposition, multi-factor analysis and attention-based lstm model, Engineering Applications of Artificial Intelligence 113 (2022) 104908.

D. Fan, H. Sun, J. Yao, K. Zhang, X. Yan, Z. Sun, Well production forecasting based on arima-lstm model considering manual operations, Energy 220 (2021) 119708.

X. Liu, Z. Lin, Z. Feng, Short-term offshore wind speed forecast by seasonal arima-a comparison against gru and lstm, Energy 227 (2021) 120492.

Z. Xu, M. Mohsin, K. Ullah, X. Ma, Using econometric and machine learning models to forecast crude oil prices: Insights from economic history, Resources Policy 83 (2023) 103614.

V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, G. K. Matsopoulos, A review of arima vs. machine learning approaches for time series forecasting in data driven networks, Future Internet 15 (2023) 255.

H. Dong, X. Guo, H. Reichgelt, R. Hu, Predictive power of arima models in forecasting equity returns: a sliding window method, Journal of Asset Management 21 (2020) 549–566.

P. R. Low, E. Sakk, Comparison between autoregressive integrated moving average and long short term memory models for stock price prediction, IAES International Journal of Artificial Intelligence (IJ-AI) 12 (2023) 1828–1835.

X. Ren, W. Xu, K. Duan, Fourier transform based lstm stock prediction model under oil shocks, Quantitative Finance and Economics 6 (2022) 342–358.

T. Swathi, N. Kasiviswanath, A. A. Rao, An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis, Applied Intelligence 52 (2022) 13675–13688.

S. Wu, Y. Liu, Z. Zou, T.-H. Weng, S_i_lstm: stock price prediction based on multiple data sources and sentiment analysis, Connection Science 34 (2022) 44–62.

J. Zhang, L. Ye, Y. Lai, Stock price prediction using cnn-bilstm-attention model, Mathematics 11 (2023) 1985.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, J. Schmidhuber, Lstm: A search space odyssey, IEEE transactions on neural networks and learning systems 28 (2016) 2222–2232.

H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 11106–11115.

Y. Xuan, Y. Yu, K. Wu, Prediction of short-term stock prices based on emd-lstm-csi neural network method, in: 2020 5th IEEE international conference on big data analytics (ICBDA), IEEE, 2020, pp. 135–139.

M. Ali, R. Prasad, Y. Xiang, Z. M. Yaseen, Complete ensemble empirical mode decomposition hybridized with random forest and kernel ridge regression model for monthly rainfall forecasts, Journal of Hydrology 584 (2020) 124647.

L. Xian, K. He, C. Wang, K. K. Lai, Factor analysis of financial time series using eemd-ica based approach, Sustainable Futures 2 (2020) 100003.

H. Yang, X. Yang, G. Li, Forecasting carbon price in china using a novel hybrid model based on secondary decomposition, multi-complexity and error correction, Journal of Cleaner Production 401 (2023) 136701.

Z. Jin, Y. Yang, Y. Liu, Stock closing price prediction based on sentiment analysis and lstm, Neural Computing and Applications 32 (2020) 9713–9729.

D. Jothimani, S. S. Yadav, Stock trading decisions using ensemble-based forecasting models: a study of the indian stock market, Journal of Banking and Financial Technology 3 (2019) 113–129.

Y. Chen, P. Zhao, Z. Zhang, J. Bai, Y. Guo, A stock price forecasting model integrating complementary ensemble empirical mode decomposition and independent component analysis, International Journal of Computational Intelligence Systems 15 (2022) 75.

B. Yan, M. Aasma, et al., A novel deep learning framework: Prediction and analysis of financial time series using ceemd and lstm, Expert systems with applications 159 (2020) 113609.

H. Wang, J. Wu, P. Zhang, Y. Chen, Learning shapelet patterns from network-based time series, IEEE transactions on industrial informatics 15 (2018) 3864–3876.

H. Rezaei, H. Faaljou, G. Mansourfar, Stock price prediction using deep learning and frequency decomposition, Expert Systems with Applications 169 (2021) 114332.

R. Zhu, G.-Y. Zhong, J.-C. Li, Forecasting price in a new hybrid neural network model with machine learning, Expert Systems with Applications 249 (2024) 123697.

S. Ren, X. Wang, X. Zhou, Y. Zhou, A novel hybrid model for stock price forecasting integrating encoder forest and informer, Expert Systems with Applications 234 (2023) 121080.

X. Zhang, K. K. Lai, S.-Y. Wang, A new approach for crude oil price analysis based on empirical mode decomposition, Energy economics 30 (2008) 905–918.