IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 January 2024, accepted 29 January 2024, date of publication 13 February 2024, date of current version 21 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3365558

RESEARCH ARTICLE

A Novel Hybrid Model of CEEMDAN-CNN-SAGU for Shanghai Copper Price Prediction

JINGYANG WANG, BOLIN DAI, TIANHU ZHANG, AND LIN QI
Hebei University of Science and Technology, Shijiazhuang, Hebei 050018, China
Corresponding author: Lin Qi (hebqilin@163.com)

This work was supported in part by the Defense Industrial Technology Development Program under Grant JCKYS2022DC10.

ABSTRACT As an essential metal, copper has the advantages of electrical conductivity and ductility, which is widely used in power transmission, electronics manufacturing and machining. The fluctuation of copper price has a great impact on the industry, especially on the development of the national economy, so predicting copper price has a great significance for economic development. However, traditional time series prediction models' prediction accuracy is low. Therefore, this paper proposes a Shanghai copper price forecasting model based on Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), Convolutional Neural Network (CNN), Self Attention Gated Unit (SAGU), named CEEMDAN-CNN-SAGU. CEEMDAN decomposes and reconstructs the Shanghai copper price data. In this paper, the zero-cross rate of the Intrinsic Mode Function (IMF) components is calculated, the IMF component with large noise is removed, and the remaining IMFs and Residual term (Res) are reorganized to obtain the high-frequency and low-frequency components. CNN performs convolution operations on the reconstructed components to extract time series features. SAGU is a new time series data prediction model proposed in this paper. SAGU includes two gated units (a forgetting gate and an input gate), and two data processing modules (Self Attention (SA) and Transition (Tra)). The SA module is responsible for processing the input data, redistributing the weight coefficients of different data, and improving the model's attention to important information. The Tra module improves the low limit of the forgetting gate's output so that the gated unit can keep the cell state from the previous moment to the current moment as much as possible. This paper uses Shanghai copper trading data from Shanghai Futures Exchange as experimental data. The comparison experiment with the other eleven prediction models shows that the CEEMDAN-CNN-SAGU model outperforms other models in all evaluation indexes.

INDEX TERMS SAGU, CEEMDAN, CNN, Shanghai copper.

I. INTRODUCTION

As a vital metal, copper plays an essential role in production and daily life, particularly in construction and electricity [1]. Fluctuations of copper price can significantly impact the production and operations of numerous manufacturing industries, ultimately influencing economic development [2], [3]. Forecasting the Shanghai copper futures' price is paramount to mitigating the risks associated with copper price fluctuations.

Shanghai copper futures price data belongs to the category of time series data, characterized by nonlinearity. The decomposition of time series data can improve the capture of trends of different frequencies and patterns and enhance the flexibility and performance of prediction models. Huang et al. introduced the Empirical Mode Decomposition (EMD), which adaptively decomposes data into IMFs based on local data characteristics [4]. However, EMD's effectiveness can be hindered by noise interference, leading to significant errors [5]. In response to these limitations, Ensemble Empirical Mode Decomposition (EEMD) was proposed, which involves the addition of white noise and its even distribution across the entire time-frequency space [6]. Nevertheless,

The associate editor coordinating the review of this manuscript and approving it for publication was Usama Mir.

EEMD may introduce refactoring errors due to the increasing white noise. To improve decomposition efficiency while maintaining adaptability, CEEMDAN was introduced [7]. CEEMDAN's adaptive decomposition produces IMF component sequences with varying time feature scales. These IMFs exhibit regularity, reducing the model's prediction difficulties [8]. Subsequently, CNN are applied to perform one-dimensional convolution operations on the decomposed CEEMDAN components. This step extracts time series features from the input data, enhancing the model's learning capabilities [9].

The SA mechanism, as an integral part of the model, facilitates data-related interactions across different locations within a data series by applying weighting coefficients. These coefficients reflect the significance of different data points [10]. Within the SA module in SAGU, input data undergoes calculation and weight coefficient redistribution, enabling the model to discern the data with the most substantial impact on prediction. Notably, the input data for the forgetting gate and the input gate in SAGU include the previous time's cell state, the previous time's hidden state, and the SA module's output data at the current time. Using knowledge from past time steps, the forgetting gate and input gate process the current data, thereby enhancing the model's learning capability [11]. The Tra data processing module in SAGU adjusts the minimum threshold for the forgetting gate's output, striving to maintain the previous time's cell state as much as possible in the current time step.

This paper presents several notable contributions and innovations:

(1) This paper proposes a new gated unit, SAGU, which includes a forgetting gate, an input gate, an SA module, and a Tra module. SAGU makes full use of past learning experiences to process the input data at the current moment to improve the model's prediction accuracy, and has outperformed LSTM and GRU in predicting Shanghai copper price.

(2) This paper introduces an innovative restructuring method to recombine the IMFs and Res generated by CEEMDAN decomposition. Based on the zero-crossing rate and comparative experiments, the noisy IMF component is removed. The remaining IMFs and Res are merged into high-frequency and low-frequency components. The influence of noisy data on model prediction is effectively reduced, thus reducing the learning difficulty of the model.

(3) A new hybrid model of CEEMDAN-CNN-SAGU is proposed to predict Shanghai copper price. The experimental results show that the prediction performance of the hybrid model is better than that of the eleven comparison models, such as LSTM, CNN-SAGU, and CEEMDAN-CNN-LSTM.

II. RELATED WORK

Bathla et al. used LSTM to predict stock prices. The experimental results indicated that LSTM outperformed traditional statistical forecasting methods [12]. The three gated units in LSTM could capture the dependency between time series data and improve LSTM's learning ability [13]. Vo et al. used Bi-LSTM to predict the Brent crude oil's price with high accuracy [14]. The model consisted of two modules, the first module used a Bi-LSTM network to save basic information about input features, and the second used a fully connected layer to predict oil prices. Bi-LSTM processes forward and backward input data to better capture feature relationships in time series [15], [16]. The prediction effect of this model is good, but there are a lot of parameters in the model that need to be trained, and the calculation is large.

Kim et al. proposed the SAM-LSTM model to predict the cryptocurrency price. The experimental findings demonstrated that the model achieved a good prediction accuracy [17]. SAM enhances the model's focus on key information, which is relevant to the self attention mechanism we studied. The self attention increases the model's attention to important information [18].

Guo et al. proposed the CEEMDAN-LSTM model to forecast the annual precipitation of Zhengzhou City, which had better prediction performance than other models [19]. CEEMDAN's effectiveness in mitigating modal confusion and reducing residual noise align with our utilization of CEEMDAN in our research. CEEMDAN solves the problem of modal confusion in EEMD decomposition and improves the problem of residual noise to a certain extent [20]. Kala et al. used the CEEMDAN-LSTM model to predict monthly rainfall across India. Compared with the LSTM, the effectiveness of adding the CEEMDAN module was verified [21]. After CEEMDAN decomposes the original sequence, the sequence obtained after decomposition has certain regularity and smoothness, which is conducive to LSTM prediction.

Li et al. proposed the CEEMDAN-SE-LSTM for power forecasting. CEEMDAN decomposed the power data, reconstructed it into two component sequences based on sample entropy analysis, and then predicted respectively. The power data was decomposed into data of different frequencies and then predicted, which greatly improves the model's prediction accuracy [22]. Wu et al. proposed a CEEMDAN-FE-LSTM to predict the Air Quality Index (AQI), which achieved an excellent fitting effect. The original AQI data series fluctuates greatly and contains noise. The actual data sequence was decomposed into several relatively stable subsequences by CEEMDAN, and then the subsequences were predicted by LSTM. Finally, the prediction result of AQI was obtained by adding the predicted value of the subsequence [23]. Their approach of decomposing the noisy AQI data into stable subsequences using CEEMDAN is akin to our method for handling noisy data. Unlike previous studies, in our research, CEEMDAN-CNN-SAGU uses CEEMDAN to decompose the data and further optimizes the data by removing the IMF component, which contains more noise. This step plays a key role in reducing noise interference and improving the model's performance and prediction accuracy and is unique to our study.

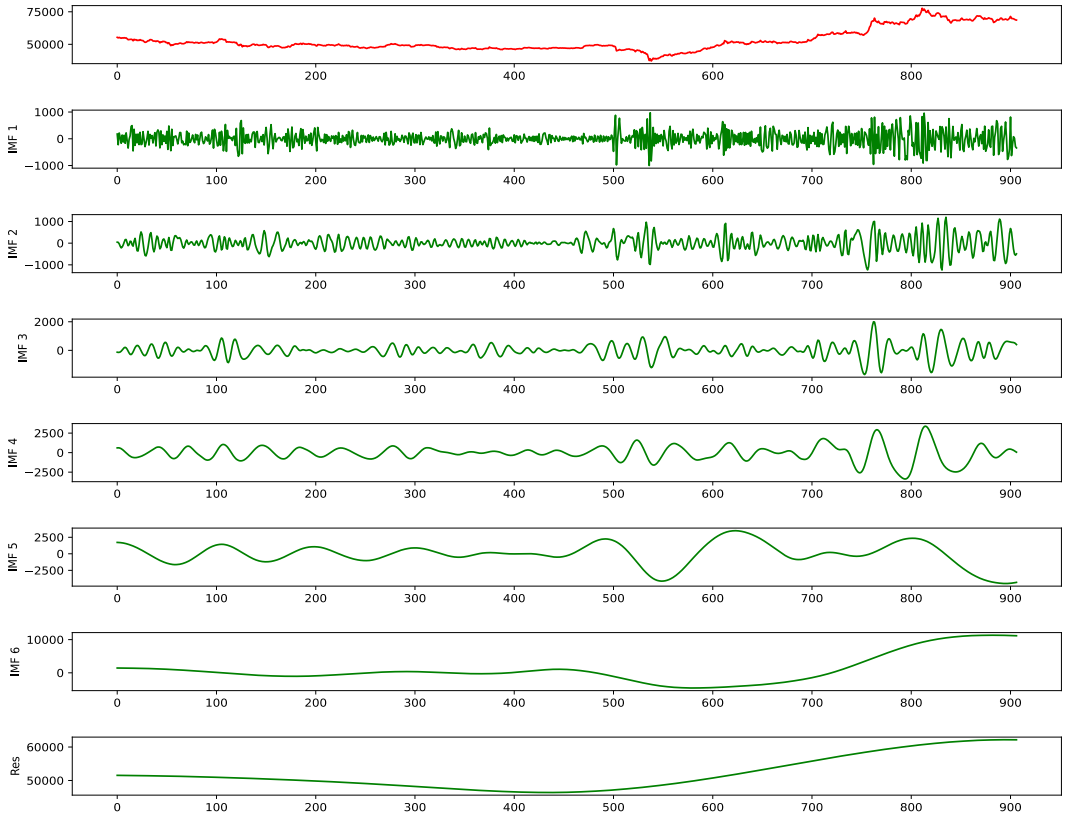


FIGURE 1. Decomposition results of the training set.

III. MODELS

A. CEEMDAN

CEEMDAN makes further improvements for EEMD. Adaptive white noise is added to each decomposition to smooth the interference pulse, making the decomposition process completer and more efficient, and reducing the reconstruction error. The specific decomposition process of CEEMDAN is as follows:

1) By adding Gaussian white noise with the mean of 0 to the original signal $x(t)$, the preprocessing sequence $x_i(t), (i = 1, 2, \ldots\ldots, k)$ is obtained.

$$x_i(t) = x(t) + \varepsilon\delta_i(t) \tag{1}$$

where ε is the weight coefficient of adding Gaussian white noise; $\delta_i(t)$ is the white Gaussian noise added at the i time of processing.

2) The EMD algorithm is used to decompose $x_i(t)$ sequence for k times, and k decomposition results are obtained, namely $IMF_1^i(t), (i = 1, 2, \ldots\ldots, k)$. The first IMF component $I_1(t)$ is obtained by calculating the mean using the formula (2).

$$I_1(t) = \frac{1}{k} \cdot \sum_{i=1}^{k} IMF_1^i(t) \tag{2}$$

$$r_1(t) = x(t) - I_1(t) \tag{3}$$

where $r_1(t)$ is the residual component obtained after the first decomposition.

3) After adding Gaussian white noise to the residual sequence of j-th stage, the EMD decomposition is continued.

$$I_j(t) = \frac{1}{k} \cdot \sum_{i=1}^{k} E_1(r_{j-1}(t) + \varepsilon_{j-1} \cdot E_{j-1}(\delta_i(t))) \tag{4}$$

$$r_j(t) = r_{j-1}(t) - I_j(t) \tag{5}$$

where $I_j(t)$ is the j-th IMF component obtained by CEEMDAN decomposition; $r_{j-1}(t)$ is the residual component obtained at the j-1 stage; ε_{j-1} is the noise factor added by CEEMDAN to the residual component of the j-1 stage. E_{j-1} is the j-1 IMF component obtained after the EMD decomposition of the sequence.

4) Repeat the above steps. If $r_j(t)$ has become a monotone function or constant, or if the amplitude falls below a given threshold and IMF cannot be extracted further, the decomposition process ends. Otherwise, $r_j(t)$ is regarded as the sequence $x(t)$ to be decomposed, and the iterative process of steps 1 to 3 is re-performed.

The Shanghai copper price data is decomposed, and the decomposition results of the training set, verification set, and test set are shown in Figures 1-3. The Shanghai copper price data in the training set is decomposed into six IMFs and one Res. The horizontal axis represents the time number of the sequence in days. The vertical axis of the topmost chart in Figure 1 shows the range of Shanghai copper price. The remaining vertical axis represents the value range from IMF1 to Res.

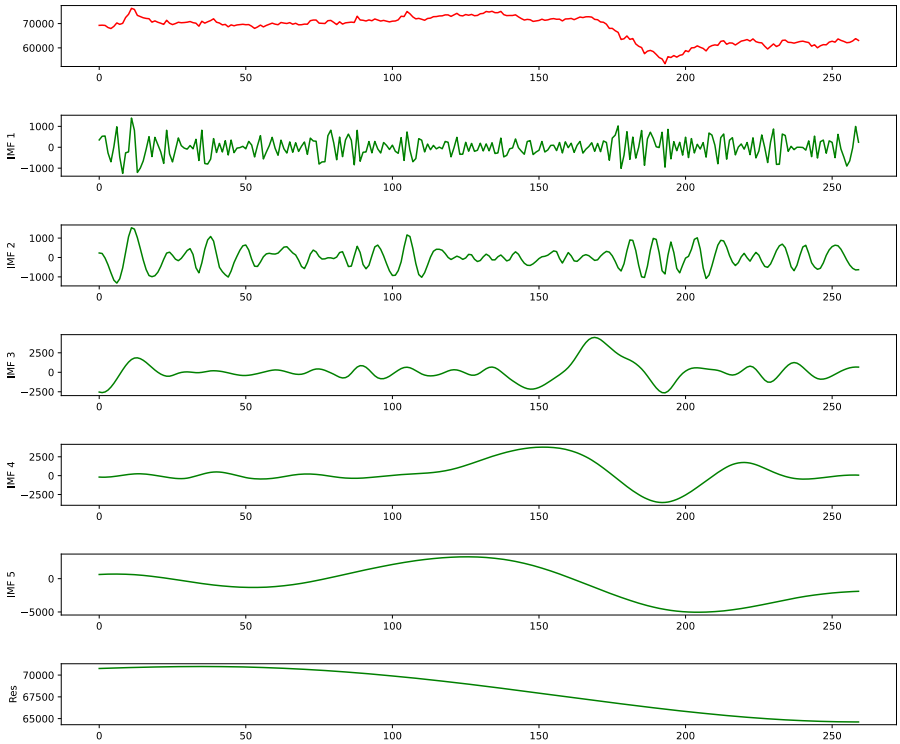


FIGURE 2. Decomposition results of the validation set.

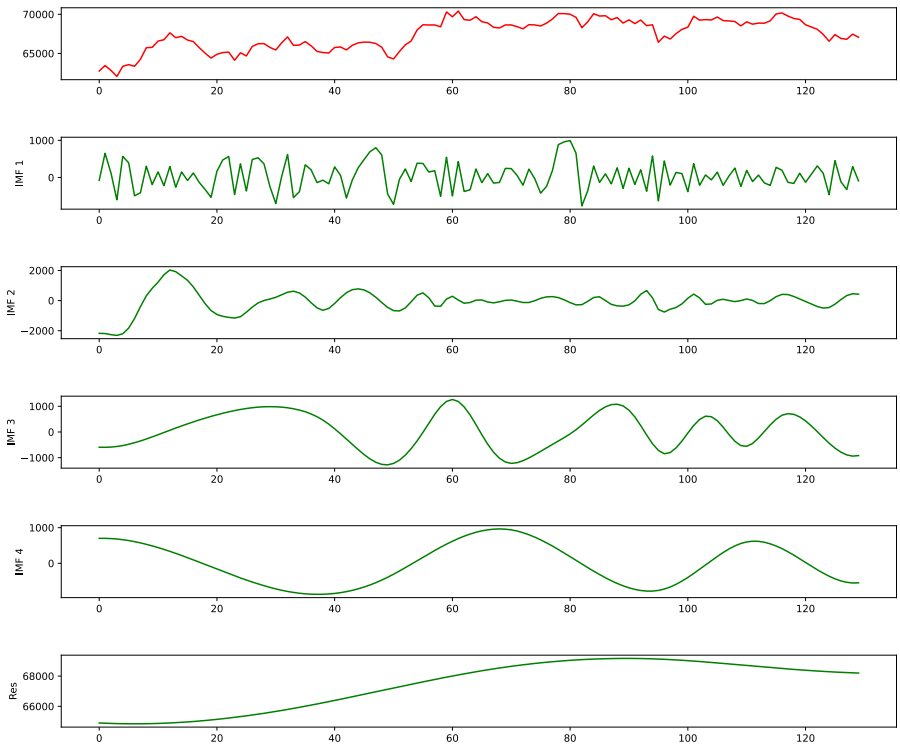


FIGURE 3. Decomposition results of the test set.

B. SAGU

Through the in-depth study of LSTM, GRU and SA, this paper introduces a novel gated unit named SAGU. The gated unit consists of an SA module, an input gate, a forgetting gate, and a Tra module. The SA calculates the input data x_t and assigns higher weight to the input data that has a

greater impact on the prediction result. The input and the forgetting gates' input data include the previous moment's hidden state output h_{t-1}, the output data a_t of the SA module of the current moment, and the previous moment's cell state c_{t-1}. The input gate is responsible for processing the current moment's input data and selectively passing the input data to the current moment's cell state. The forgetting gate is responsible for selectively forgetting the previous moment's cell state, preserving the remaining previous moment's cell state to the cell state of the current moment. The input data of the input gate and the forgetting gate include the cell state of the previous moment, which is beneficial for the current moment's cell to use the previous moment's information to process the current moment's input data, which is conducive to the learning of the gated unit and enhances the model's predictive accuracy. The forgetting gate's output value range is [0,1]. After the calculation of the Tra module, the range of the output values is changed to [0.24,1]. The structure diagram of SAGU is shown in Figure 4.

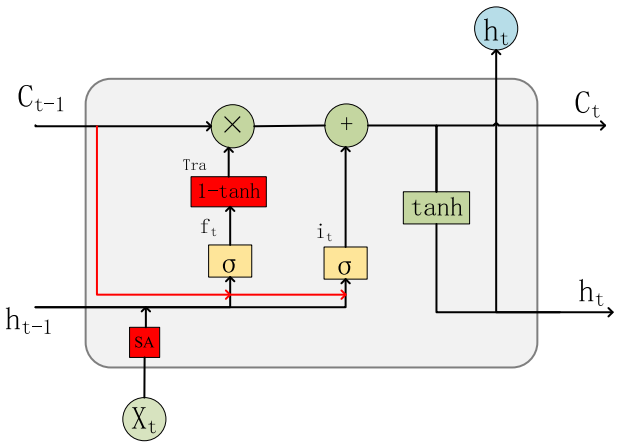


FIGURE 4. SAGU structure diagram.

The following describes the formula for calculating the SA module.

$$q^i = w^q \cdot a^i \tag{6}$$
$$k^i = w^k \cdot a^i \tag{7}$$
$$v^i = w^v \cdot a^i \tag{8}$$

where q^i is the Query vector with the i-th eigenvalue, k^i, v^i is its Key vector and Value vector separately. w^q, w^k, and w^v are the corresponding weight coefficient vectors. a^i is the i-th eigenvector.

$$a_{ij} = \frac{q^i \cdot k^j}{\sqrt{d}} \tag{9}$$

a_{ij} is the similarity between the i-th eigenvector and the j-th eigenvector. d is the dimension of the i-th eigenvalue Key vector.

a'_{ij} is the weight coefficient between the i-th eigenvector and the j-th eigenvector.

$$a'_{ij} = \frac{\exp(a_{ij})}{\sum_{j=0}^{n} \exp(a_{ij})} \tag{10}$$
$$b^i = \sum_{i=0}^{n} a'_{ij} \cdot v^i \tag{11}$$

b^i is the output of the SA.

The SA calculates the input data x_t and assigns higher weights to the input data that has a more significant influence on the prediction result, allowing the model to learn which data is important.

The forgetting gate in SAGU controls how much of the previous moment's cell state c_{t-1} is kept in the current moment's cell state c_t. The forgetting gate processes the previous moment's hidden state h_{t-1}, the previous moment's cell state c_{t-1}, and the output data a_t of the SA module of the current moment through the sigmoid function. The sigmoid function's output is constrained within the range of 0 to 1. When the output value is 0, the previous moment's cell state is completely discarded. When the output value is 1, the previous moment's cell state is completely preserved to the current moment's cell state. The forgetting gate's calculation formula is shown in formula 12.

$$f_t = \sigma(w_{fh} \cdot h_{t-1} + w_{fa} \cdot a_t + w_{fc} \cdot c_{t-1} + b_f) \tag{12}$$

where f_t represents the forgetting gate's output for the current moment.

The Tra module in SAGU mainly changes the forgetting gate's output value and then changes the proportion of retaining the cell state from the previous moment to the current moment. The output value of the forgetting gate is between [0,1]. When the input data is in [0,1], the tanh function's output value range is in [0,0.76], and 1-tanh function's output value range is in [0.24,1]. After the Tra module is used, the forgetting gate's value range changes to [0.24,1], and the previous moment's cell state is retained as much as possible. The current moment's gated unit can leverage the knowledge acquired from the previous moment to effectively process the current moment's input data, thereby enhancing the model's learning capability. The computational expression for the Tra data processing module is depicted in formula 13.

$$Tra = 1 - \tanh(f_t) \tag{13}$$

where Tra is the output value of Tra module.

The input gate controls how much of the current time's input data is retained in the current time's cell state. The input gate processes the previous moment's hidden state, the previous moment's cell state and the data processed by the SA module of the current moment through the sigmoid function. When the sigmoid's output value is 0, the current moment's input data is completely discarded. When the sigmoid's output value is 1, the input data of the current moment is completely retained in the current moment's cell state. Formula 14 represents the computational formula for

the input gate.

$$i_t = \sigma(w_{ih} \cdot h_{t-1} + w_{ia} \cdot a_t + w_{ic} \cdot c_{t-1} + b_f) \quad (14)$$

where i_t represents the input gate's output for the current moment.

The current moment's cell state c_t is obtained by multiplying the output of the Tra module with the previous moment's cell state, and then adding the output value of the input gate at the current moment. The current moment's cell state is calculated by the formula 15.

$$c_t = Tra \cdot c_{t-1} + i_t \quad (15)$$

The current moment's hidden state h_t is the current moment's output value of SAGU, and the calculation formula is shown in formula 16.

$$h_t = \tanh(c_t) \quad (16)$$

C. CEEMDAN-CNN-SAGU

The CNN-SAGU prediction model's structure is shown in Figure 5.

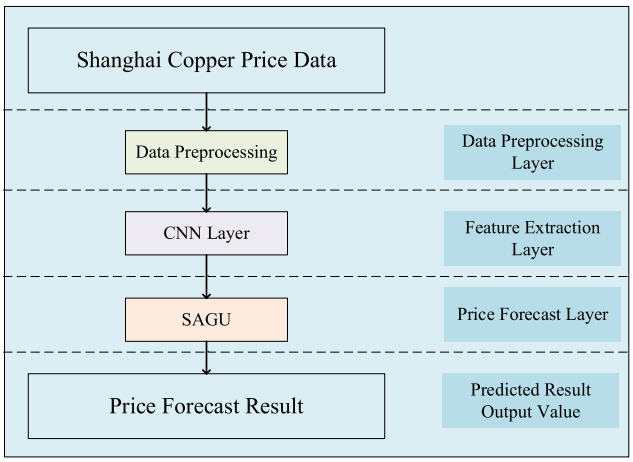


FIGURE 5. Structure of CNN-SAGU prediction model.

The corresponding weight file is obtained after the CEEMDAN-CNN-SAGU model is trained according to the training set data. The model loads weight file to predict future Shanghai copper price. The prediction formula is shown in formula 17.

$$Y = f_{SAGU}(f_{CNN}(f_{CEEMDAN}(X, W_{CEEMDAN}), W_{CNN}), W_{SAGU}) \quad (17)$$

where Y is the predicted output; f_{SAGU} is the SAGU model; f_{CNN} is the CNN model; $f_{CEEMDAN}$ is the CEEMDAN decomposition and reconstruction process; $W_{CEEMDAN}$, W_{CNN}, and W_{SAGU} are the parameters of the corresponding models.

The CEEMDAN-CNN-SAGU prediction model's structure is shown in Figure 6.

CEEMDAN layer: CEEMDAN decomposes Shanghai copper price data into IMFs with different frequencies and amplitudes to reduce noise in the original data.

IMFs recombination layer: The IMF component with large noise is removed according to the zero-crossing rate, and the remaining IMFs and Res are reorganized into high-frequency and low-frequency components.

Data preprocessing layer: Normalize the data, scale the data to the [0,1], and eliminate the impact of data value distribution differences on model learning.

Feature extraction layer: CNN carries out convolution operations on high-frequency and low-frequency components, respectively, to capture data features in different components.

Price forecast layer: SAGU captures time dependencies in different component sequences and makes predictions based on learned data law.

Predicted value plus layer: The SAGU's forecast results for low-frequency and high-frequency components are added to get the forecast result of Shanghai copper price.

Predicted result output value: The predicted result of Shanghai copper price is outputted by reverse normalization.

IV. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENT

The operating system of the server used in this experiment is Windows 11, the CPU is 12th Gen Intel Core i7-12700H 2.30GHz. The development tool used in this experiment is PyCharm 2020.1.3, the programming language is Python3.7.0, and the software base platform is Anaconda 4.5.11.

B. DATA ACQUISITION

The experimental data selected in this experiment is the Shanghai copper trading data from January 2, 2018, to April 28, 2023, with 1297 pieces of data. The experimental data is obtained from the Tushare data service platform website through the third-party data interface. Daily trading data includes the Shanghai copper futures' opening price, the highest price, the lowest price, the closing price, the settlement price, the price change, the volume, and the day's revenue. Shanghai copper price fluctuates between 37,200 yuan and 77,720 yuan, which is volatile and difficult to predict. Table 1 shows some Shanghai copper price data.

In table 1, trade-date indicates the trade time; open stands for Shanghai copper's opening price on the same day; high stands for the highest price of the day; low stands for the lowest price of the day; close stands for the closing price of Shanghai copper of the day; settle stands for the settlement price of the day; change stands for the rise or fall; vol stands for Shanghai copper's volume of the day; oi stands for the day's revenue.

To select the impact factors that affect the price of Shanghai copper, this paper chooses the grey relation analysis method to calculate the correlation between different impact factors

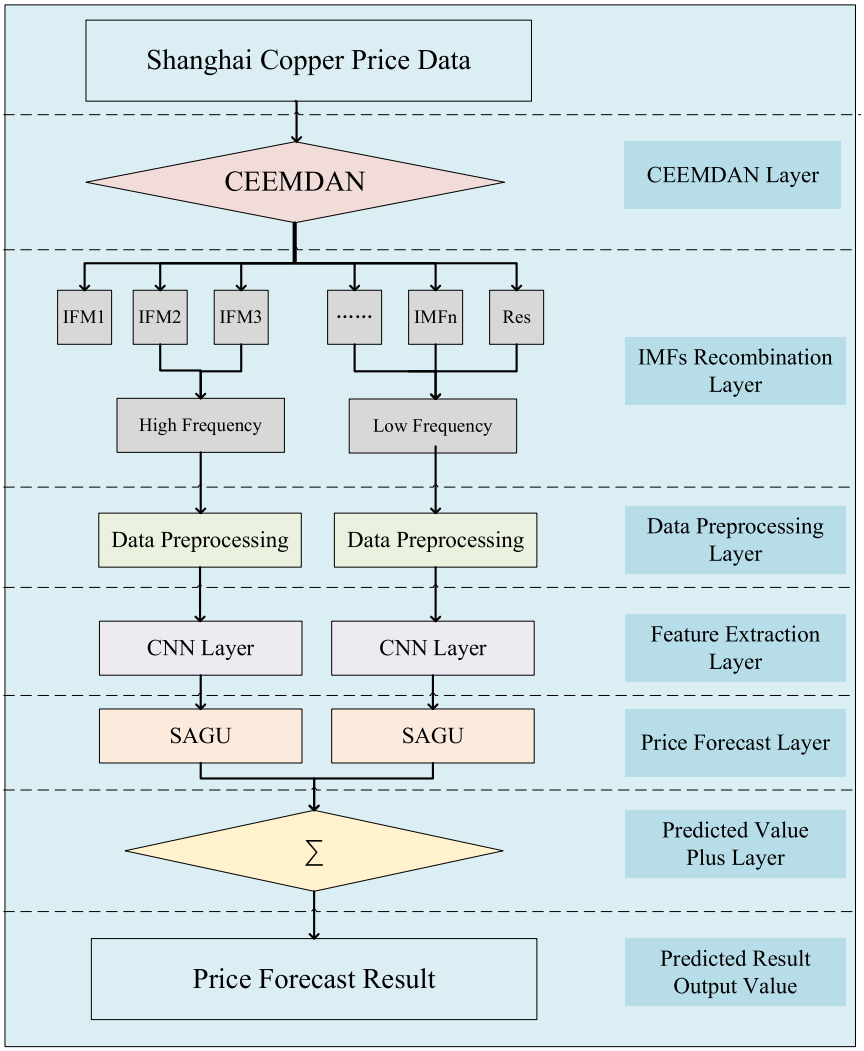


FIGURE 6. Structure of CEEMDAN-CNN-SAGU prediction model.

TABLE 1. Shanghai copper price data table.

Trade-date	open	high	low	close	settle	change	vol	oi
2018/1/2	55540	55760	54960	55390	55270	-470	8946600	21984800
2018/1/3	55250	55350	54810	54970	55050	-220	9797600	21848000
2018/1/4	54870	55420	54680	55320	55020	-30	13586000	22644400
2018/1/5	55460	55480	54800	55010	55070	50	11817000	22509000
2018/1/8	54710	54840	54420	54690	54600	-470	15826000	22620600

TABLE 2. Correlation coefficient of influence factors.

Influence factor	Correlation value
SPX500	0.648656
JPN225	0.748827
SHCI	0.714816
HK	0.612309
UK100	0.547445

and Shanghai copper price [24]. The calculated correlation coefficient evaluates the correlation between different influencing factors and the Shanghai copper price. A higher correlation coefficient indicates a higher correlation between them. The relationship between various influencing factors and Shanghai copper price is calculated by selecting several

TABLE 3. Impact factor data table.

trade_date	SPX500	JPN225	SHCI	HK	UK100
2018/1/2	2708.9	23166.5	3348.3	30533	7692.2
2018/1/3	2723.9	23638.5	3369.1	30684	7691.2
2018/1/4	2741.9	23802.5	3385.7	30744	7712.7
2018/1/5	2732.9	23720.5	3391.7	30714	7721.5
2018/1/8	2749.6	23841.5	3409.7	30966	7730.3

influencing factors. Table 2 presents the correlation coefficient between various impact factors and the Shanghai copper price. The final selected impact factors are SPX500, JPN225, SHCI, HK, and UK100.

SPX500 represents the Standard & Poor's 500 Index; JPN225 represents Nikkei 225 Index; SHCI represents Shanghai Financial Index; HK represents Hong Kong Hang Seng Index; UK100 represents FTSE 100 Index.

Table 3 shows the data of the selected impact factors in this paper.

C. DATA PREPROCESSING

Because the value range of different data varies greatly, to avoid the excessive influence of some feature data on model training. This paper uses the method of mean normalization to scale all data to [0,1], which makes the weight update more balanced and beneficial to the model's training [25]. The first 908 pieces of data are selected as the training dataset, 258 pieces of data between 909-1167 are selected as the verification dataset, and the remaining 130 pieces of data are selected as the test dataset.

D. 3D Construction of Data

The dataset can be represented by $S = \{X_1, X_2, \ldots, X_n\}$, where n is the number of pieces of data. X_n is a d-dimensional vector representing the price data of Shanghai copper data on a certain day. Therefore, the dimension of this dataset is (n, D). The time series sample set generated by this dataset is three-dimensional data with dimensions (n, T, D). Where n represents the time slices' number, T represents the time step in each sample, and D represents the data items' number contained in a Shanghai copper price data. Assuming that there is n items in the experimental data, 3D data is constructed according to the setting of step size 1 and sequence length 3. Data from the first to the third forms the first layer (Y_1 layer), then data from items 2 to 4 forms the second layer (Y_2 layer), and so on. A total of $n-2$ layers $(Y_1, Y_2, \ldots, Y_{n-2})$ structure, each layer contains 3 pieces of data, that is, to complete the construction of three-dimensional data. The prediction model is characterized by the first two data of each side, and the third copper price data is the target of prediction evaluation. The time series construction process is shown in Figure 7.

E. MODEL PARAMETERS

The model's prediction performance is significantly affected by the selection of hyperparameters. This experiment considers the selection of several hyperparameters, among which

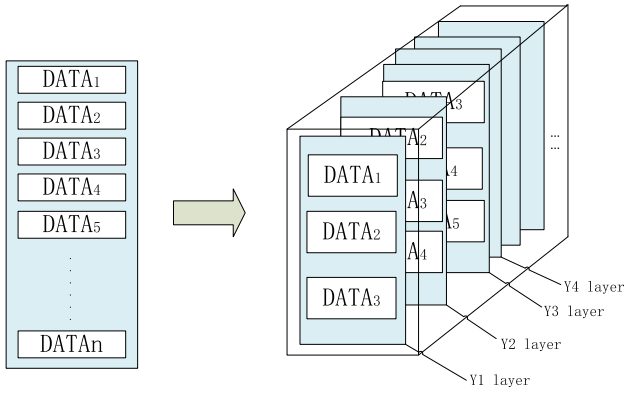


FIGURE 7. Time sequence construction process.

specifically introduce three major hyperparameters: the candidate values of the 'units' parameter are {128, 160, 192, 224, 256}; Candidate values for the 'filters' parameter are {16, 24, 32, 40, 48, 56, 64}; Candidate values for the 'epsilon' parameter are {0.001, 0.0005, 0.0001}. To determine the optimal hyperparameter combinations of different models, this experiment uses a grid search method, which helps to comprehensively consider various hyperparameter combinations while maintaining the fairness of the experiment. Three evaluation indexes are used to measure the predictive performance of models, namely, Mean Absolute Error (MAE), Explained Variance Score (EVS), R Squared (R^2). These evaluation indexes are helpful in evaluating the model comprehensively. Through the grid search method, the optimal hyperparameters of different models are shown in Table 4.

F. EXPERIMENTAL RESULTS

To properly reduce the complexity of the CEEMDAN-CNN-SAGU model, IMFs and Res components obtained from CEEMDAN decomposition are reorganized into high-frequency and low-frequency components. In order to test the contained noise in each IMF component, zero-crossing analysis is performed for the IMFs and the Res. The results of the zero-crossing rate table of the training set, verification set, and test set are shown in Table 5-7.

To reduce the complexity of the CEEMDAN-CNN-SAGU prediction model and avoid overfitting, the six IMF components and one Res generated after the decomposition of the training set are reorganized. The zero-crossing rate table shows that the IMF1 component contains large noise, so the IMF1 component is deleted. To verify whether the IMF1

TABLE 4. Table of model hyperparameters.

Model	Parameters
SVR	kernel='linear'
MLP	units=256, activation='tanh'
LSTM	units=256, activation='tanh'
GRU	units=256, activation='tanh'
SAGU	units=192, activation='tanh'
CNN-LSTM	filters=16, kernel_size=3, units=224, activation='tanh'
CNN-GRU	filters=64, kernel_size=3, units=256, activation='tanh'
CNN-SAGU	filters=16, kernel_size=3, units=192, activation='tanh'
CEEMDAN-LSTM	units=192, activation='tanh'
CEEMDAN-CNN-LSTM	trials=10, epsilon=0.0001, filters=64, kernel_size=3, units=192, activation='tanh'
CEEMDAN-CNN-GRU	trials=10, epsilon=0.0001, filters=24, kernel_size=3, units=192, activation='tanh'
CEEMDAN-CNN-SAGU	trials=10, epsilon=0.0001, filters=16, kernel_size=3, units=256, activation='tanh'

TABLE 5. Zero-Crossing rate table of the training set.

IMF	Zero-crossing rate value
IMF1	0.6395
IMF2	0.2602
IMF3	0.1213
IMF4	0.0529
IMF5	0.0221
IMF6	0.0066
Res	0.0000

TABLE 6. Zero-Crossing rate table of the verification set.

IMF	Zero-crossing rate value
IMF1	0.6385
IMF2	0.2462
IMF3	0.1078
IMF4	0.0462
IMF5	0.0154
Res	0.0000

TABLE 7. Zero-Crossing rate table of the test set.

IMF	Zero-crossing rate value
IMF1	0.6384
IMF2	0.2308
IMF3	0.0769
IMF4	0.0462
Res	0.0000

component is removed, a comparative experiment is conducted in this paper. The comparative experimental results of whether to remove the IMF1 component are shown in Table 8.

TABLE 8. Results of IMF1 comparison experiment.

Model	MAE	EVS	R^2
IMF1+IMF2+IMF3	481.802065	0.944126	0.902735
IMF2+IMF3	392.726425	0.944500	0.934822

Comparative experiments verify the effectiveness of removing IMF1. To verify the optimal combination of high-frequency components, comparative experiments are conducted on different IMF component combinations, and the experiment results of different high-frequency combinations are shown in Table 9.

TABLE 9. Experiment results of different high-frequency combinations.

Combination mode	MAE	EVS	R^2
IMF2+IMF3	392.726425	0.944500	0.934822
IMF2+IMF3+IMF4	343.596236	0.948649	0.948639
IMF2+IMF3+IMF4+IMF5	387.001283	0.943493	0.943273
IMF2+IMF3+IMF4+IMF5+IMF6	378.331730	0.950819	0.937789

As shown in Table 9, the best combination of high-frequency component is IMF2+IMF3+IMF4. Removing the noisy IMF1 component reduces the prediction difficulty. The high-frequency component used in this experiment is IFM2+IMF3+IMF4, and the low-frequency component is IMF5+IMF6+Res. To verify the CEEMDAN-CNN-SAGU's validity, eleven other models are selected for comparative tests, and MAE, EVS, R^2 evaluation indexes measure different models's prediction performance. The high-frequency component of CEEMDAN-LSTM, CEEMDAN-CNN-LSTM, CEEMDAN-CNN-GRU and CEEMDAN-CNN-SAGU prediction models is IMF2+IMF3+IMF4, and the low-frequency component is IMF5+IMF6+Res. The reorganization method of the verification and test sets is the same as that of the training set. IMF1 is removed both in verification and test sets. The high-frequency of the verification set is IMF2+IMF3, and the low-frequency is IMF4+IMF5+Res. The high-frequency of the test set is IMF2+IMF3, and the low-frequency is IMF4+Res. The different models' prediction results for Shanghai copper price are shown in Table 10. From the prediction results, the CEEMDAN-CNN-SAGU model has the best performance. Figures 8-10 illustrate the comparisons of MAE, EVS, and R^2, respectively.

TABLE 10. Shanghai copper price prediction results table.

Model	MAE	EVS	R^2
SVR	848.322742	0.865843	0.694947
MLP	787.794752	0.867059	0.734340
LSTM	542.178334	0.858809	0.847631
GRU	532.713951	0.855450	0.855420
SAGU	527.852301	0.863351	0.862098
CNN-LSTM	537.106791	0.870462	0.870379
CNN-GRU	516.008089	0.878500	0.871186
CNN-SAGU	521.707738	0.877017	0.878643
CEEMDAN-LSTM	483.313401	0.930716	0.901221
CEEMDAN-CNN-LSTM	424.183074	0.929954	0.927177
CEEMDAN-CNN-GRU	402.967575	0.939554	0.930846
CEEMDAN-CNN-SAGU	343.596236	0.948649	0.948639

The prediction effect of Multilayer Perceptron (MLP) and Support Vector Regression (SVR) on Shanghai copper price is poor, and the fitting degree only reaches 0.734340 and 0.694947, respectively. The fitting degree of LSTM, GRU, and SAGU reaches 0.847631, 0.855420, and 0.862098, respectively. The SA module in SAGU processes the input data, redistributes the different data's weight, and assigns a

larger weight coefficient to the input data that has a more significant influence on the prediction result. By reassigning weight coefficients, the model can learn which data is important and improve the model's learning ability. The input data of the forgetting gate and the input gate in SAGU includes the previous moment's cell state, the output data of the SA module at the current moment, and the previous moment's hidden state. The two gated units improve the model's learning ability by learning from the previous moment's experience to calculate the current moment's input data. The output value of the forgetting gate is in [0,1], and the Tra module increases the output range of this value to [0.24,1], which preserves the previous moment's cell state to the current moment to a large extent so that the model can fully learn the previous moment's experience to calculate the current moment's input.

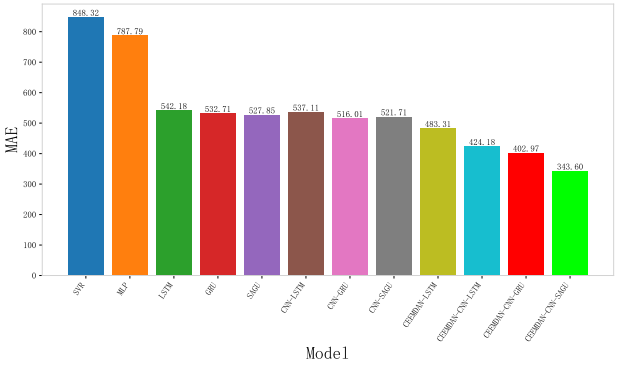


FIGURE 8. MAE comparison.

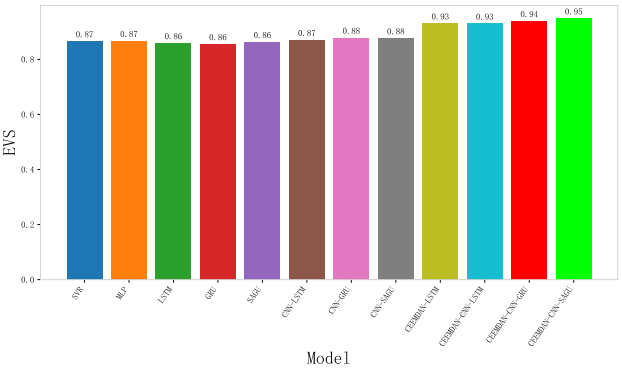


FIGURE 9. EVS comparison.

CNN uses the convolution and pooling layers to extract local features from Shanghai copper price data. CNN extracts local features from Shanghai copper trading data, and LSTM, GRU, or SAGU captures the time series rules in Shanghai copper trading data. It is shown in Table 10 that the CNN-combined model's prediction performance is better than that of the single model.

The Shanghai copper price data is non-linear and non-stationary. CEEMDAN can decompose the Shanghai copper price data into IMFs and a Res. CEEMDAN can decompose data on different time scales, reducing the model's prediction difficulty. According to the zero-crossing rate analysis, the IMF1 component contains large noise, which has a bad effect on the model prediction of Shanghai copper price. The IMF1 component containing more noise is removed, and the remaining components are reorganized into high-frequency and low-frequency components, which reduces the model prediction's difficulty. Table 10 shows that adding CEEMDAN improves the model's prediction accuracy.

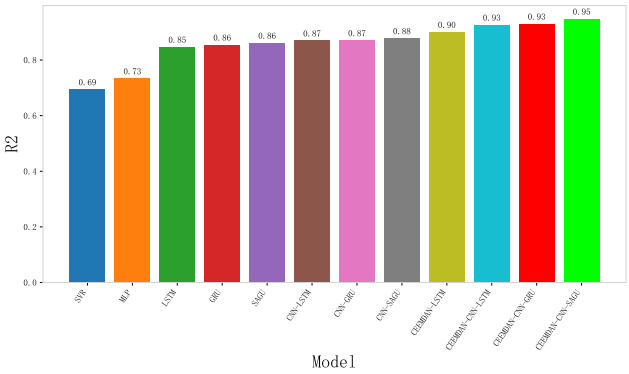


FIGURE 10. R^2 comparison.

To verify the overfitting and validity of the model, the trained CEEMDAN-CNN-SAGU model is used to forecast the Shanghai copper price in May 2023. The prediction result is shown in the table 11.

TABLE 11. Prediction results table.

Model	MAE	EVS	R^2
CEEMDAN-CNN-SAGU	316.860939	0.958109	0.955284

G. MODEL GENERALIZATION

To verify the CEEMDAN-CNN-SAGU's generalization, the trading data of Shanghai silver is selected as experimental data for the prediction experiment. According to the prediction results of Shanghai silver price, the CEEMDAN-CNN-SAGU is better than other models. Table 12 shows the experimental results for predicting the Shanghai silver price.

TABLE 12. Shanghai silver price forecast experiment results table.

Model	MAE	EVS	R^2
SVR	78.731266	0.895610	0.873152
MLP	72.682464	0.895824	0.895761
LSTM	63.478273	0.912231	0.907323
GRU	62.174301	0.916776	0.910578
SAGU	61.114584	0.921826	0.914559
CNN-LSTM	59.901978	0.916792	0.914278
CNN-GRU	61.013967	0.916638	0.916068
CNN-SAGU	58.822319	0.924201	0.921455
CEEMDAN-CNN-LSTM	46.694931	0.963068	0.955483
CEEMDAN-CNN-GRU	49.638210	0.958739	0.957732
CEEMDAN-CNN-SAGU	41.682213	0.971330	0.972100

V. DISCUSSION

Compared with other models, the CEEMDAN-CNN-SAGU proposed in this paper performs best in Shanghai copper

price prediction. The CEEMDAN module decomposes the Shanghai copper price data into IMFs and a Res. According to the zero-crossing rate, the IMF component with high noise is removed, and the remaining IMFs and Res are combined to obtain high-frequency and low-frequency components. CNN calculates the recombination components to extract the data rules in different frequency components. The SAGU calculates the convolution data, mines the time series rules in the data, makes predictions. The final Shanghai copper price prediction result is obtained by adding the high-frequency and low-frequency components. The reasons for CEEMDAN-CNN-SAGU's high prediction accuracy for the Shanghai copper price forecast are as follows:

(1) The SA module in SAGU processes the input data, redistributes the weight coefficients of different input data, and assigns larger weight coefficients to the input data that have a greater impact on the prediction results so that the model can learn which input data are important data, thus improving the model's learning ability. The input data of the forgetting gate and the input gate in SAGU includes the previous moment's cell state, the previous moment's hidden state output, and the output of the SA module at the current moment. SAGU improves the learning ability of the model by fully learning from the previous moment's experience to process the current moment's input data. The Tra module in SAGU changes the forgetting gate's output value, elevates the output range of [0,1] to [0.24,1], and preserves the cell state of the previous moment to the current moment to a large extent.

(2) The addition of the CEEMDAN module can decompose Shanghai copper data into multiple IMFs and a Res. According to the zero-crossing rate, the IMF component containing large noise is removed. Then the remaining IMFs and Res are recombined to obtain the high-frequency and low-frequency components. By recombining the IMFs and Res, the difficulty of predicting the Shanghai copper price is reduced.

(3) CNN can effectively learn the temporal correlation in time series data through the sliding window operation of the convolution kernel.

VI. CONCLUSION

Due to the high nonlinearity and non-stationarity of Shanghai copper price data, this paper constructs a hybrid model based on CEEMDAN-CNN-SAGU to forecast Shanghai copper price. CEEMDAN decomposes and reorganizes the Shanghai copper price data, removes the IMF component with larger noise, and reorganizes the remaining IMFs and Res to obtain high-frequency and low-frequency components with less noise, which reduces the difficulty of the model's prediction of Shanghai copper price. Through CNN convolution, the time series features of Shanghai copper price data are effectively extracted. The SA module in SAGU redistributes the weight coefficients of different input data, which is beneficial for the model to learn which input data greatly influences the prediction results. SAGU keeps the learning experience of the previous moment as much as possible to process the current moment's input data, which improves the model's learning ability. In the comparison experiment, the CEEMDAN-CNN-SAGU model has the best performance. The prediction of Shanghai silver shows that CEEMDAN-CNN-SAGU has good generalization.

However, the computational complexity of CEEMDAN is relatively high, especially when processing large-scale time series data. The need for multiple decomposition and reconstruction operations, it will lead to high computing resource consumption. In future work, we should consider integrating different types of time series models, make full use of the advantages of different models, respectively predict different sequences obtained after CEEMDAN decomposition, and then add different models' prediction results to get the final prediction results. Thus, the composite model's prediction performance will be improved.

REFERENCES

[1] X. Jin, S. Zhou, K. Yin, and M. Li, "Relationships between copper futures markets from the perspective of jump diffusion," *Mathematics*, vol. 9, no. 18, p. 2268, Sep. 2021, doi: 10.3390/math9182268.

[2] L. Zhang, Z. Cai, J. Yang, Y. Chen, and Z. Yuan, "Quantification and spatial characterization of in-use copper stocks in Shanghai," *Resour. Conservation Recycling*, vol. 93, pp. 134–143, Dec. 2014, doi: 10.1016/j.resconrec.2014.10.010.

[3] M. Vochozka, E. Kalinova, P. Gao, and L. Smolikova, "Development of copper price from July 1959 and predicted development till the end of year 2022," *Acta. Montanistica Slovaca*, vol. 26, no. 2, pp. 262–280, Sep. 2021, doi: 10.46544/AMS.v26i2.07.

[4] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. Lond. Ser. Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998, doi: 10.1098/rspa.1998.0193.

[5] H. V. Young, Y.-C. Lin, and Y.-H. Wang, "On the memory cost of EMD algorithm," *IEEE Access*, vol. 10, pp. 114242–114251, 2022, doi: 10.1109/ACCESS.2022.3218417.

[6] J. Zhao, G. Nie, and Y. Wen, "Monthly precipitation prediction in luoyang city based on EEMD-LSTM-ARIMA model," *Water Sci. Technol.*, vol. 87, no. 1, pp. 318–335, Jan. 2023, doi: 10.2166/wst.2022.425.

[7] B. Gao, X. Huang, J. Shi, Y. Tai, and J. Zhang, "Hourly forecasting of solar irradiance based on CEEMDAN and multi-strategy CNN-LSTM neural networks," *Renew. Energy*, vol. 162, pp. 1665–1683, Dec. 2020.

[8] X. Wang, W. Liu, Y. Wang, and G. Yang, "A hybrid NOx emission prediction model based on CEEMDAN and AM-LSTM," *Fuel*, vol. 310, Feb. 2022, Art. no. 122486, doi: 10.1016/j.fuel.2021.122486.

[9] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4741–4753, Nov. 2020, doi: 10.1007/s00521-020-05532-z.

[10] Y. Wei, D. Wu, and J. Terpenny, "Bearing remaining useful life prediction using self-adaptive graph convolutional networks with self-attention mechanism," *Mech. Syst. Signal Process.*, vol. 188, Apr. 2023, Art. no. 110010, doi: 10.1016/j.ymssp.2022.110010.

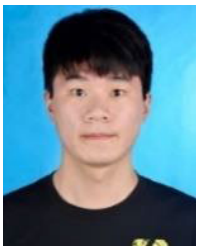
[11] H. Wang, B. Dai, X. Li, N. Yu, and J. Wang, "A novel hybrid model of CNN-SA-NGU for silver closing price prediction," *Processes*, vol. 11, no. 3, p. 862, Mar. 2023, doi: 10.3390/pr11030862.

[12] G. Bathla, R. Rani, and H. Aggarwal, "Stocks of year 2020: Prediction of high variations in stock prices using LSTM," *Multimedia Tools Appl.*, vol. 82, no. 7, pp. 9727–9743, Mar. 2023, doi: 10.1007/s11042-022-12390-5.

[13] W. Ban and L. Shen, "PM2.5 prediction based on the CEEMDAN algorithm and a machine learning hybrid model," *Sustainability*, vol. 14, no. 23, p. 16128, Dec. 2022, doi: 10.3390/su142316128.

[14] A. H. Vo, T. Nguyen, and T. Le, "Brent oil price prediction using bi-LSTM network," *Intell. Autom. Soft Comput.*, vol. 26, no. 4, pp. 1307–1317, 2020, doi: 10.32604/iasc.2020.013189.
[15] X. Hu, T. Liu, X. Hao, and C. Lin, "Attention-based conv-LSTM and bi-LSTM networks for large-scale traffic speed prediction," *J. Supercomput.*, vol. 78, no. 10, pp. 12686–12709, Mar. 2022, doi: 10.1007/s11227-022-04386-7.
[16] T. Le, M. T. Vo, B. Vo, E. Hwang, S. Rho, and S. W. Baik, "Improving electric energy consumption prediction using CNN and bi-LSTM," *Appl. Sci.*, vol. 9, no. 20, p. 4237, Oct. 2019, doi: 10.3390/app9204237.
[17] G. Kim, D.-H. Shin, J. G. Choi, and S. Lim, "A deep learning-based cryptocurrency price prediction model that uses on-chain data," *IEEE Access*, vol. 10, pp. 56232–56248, 2022, doi: 10.1109/ACCESS.2022.3177888.
[18] Z. Zhang, H.-N. Dai, J. Zhou, S. K. Mondal, M. M. García, and H. Wang, "Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115378, doi: 10.1016/j.eswa.2021.115378.
[19] S. Guo, Y. Wen, X. Zhang, G. Zhu, and J. Huang, "Research on precipitation prediction based on a complete ensemble empirical mode decomposition with adaptive noise–long short-term memory coupled model," *Water Supply*, vol. 22, no. 12, pp. 9061–9072, Dec. 2022, doi: 10.2166/ws.2022.412.
[20] Y. Hu, Y. Ouyang, Z. Wang, H. Yu, and L. Liu, "Vibration signal denoising method based on CEEMDAN and its application in brake disc unbalance detection," *Mech. Syst. Signal Process.*, vol. 187, Mar. 2023, Art. no. 109972, doi: 10.1016/j.ymssp.2022.109972.
[21] A. Kala, S. Ganesh Vaidyanathan, and P. Sharon Femi, "CEEMDAN hybridized with LSTM model for forecasting monthly rainfall," *J. Intell. Fuzzy Syst.*, vol. 43, no. 3, pp. 2609–2617, Jul. 2022, doi: 10.3233/jifs-213064.
[22] K. Li, W. Huang, G. Hu, and J. Li, "Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network," *Energy Buildings*, vol. 279, Jan. 2023, Art. no. 112666, doi: 10.1016/j.enbuild.2022.112666.
[23] Z. Wu, W. Zhao, and Y. Lv, "An ensemble LSTM-based AQI forecasting model with decomposition-reconstruction technique via CEEMDAN and fuzzy entropy," *Air Qual. Atmos. Health*, vol. 15, no. 12, pp. 2299–2311, Sep. 2022, doi: 10.1007/s11869-022-01252-6.
[24] G. Sun, X. Guan, X. Yi, and Z. Zhou, "Grey relational analysis between hesitant fuzzy sets with applications to pattern recognition," *Expert. Syst. Appl.*, vol. 92, pp. 521–532, Sep. 2018, doi: 10.1016/j.eswa.2017.09.048.
[25] X. Liu, Y. Zhang, and Q. Zhang, "Comparison of EEMD-ARIMA, EEMD-BP and EEMD-SVM algorithms for predicting the hourly urban water consumption," *J. Hydroinformatics*, vol. 24, no. 3, pp. 535–558, Mar. 2022, doi: 10.2166/hydro.2022.146.

JINGYANG WANG received the B.Eng. degree in computer software from Lanzhou University, China, in 1995, and the M.Sc. degree in software engineering from the Beijing University of Technology, China, in 2007. He is currently a Professor with the Hebei University of Science and Technology, Shijiazhuang, Hebei, China. His research interests include machine learning, deep learning, natural language processing, and big data processing.



BOLIN DAI is currently pursuing the master's degree with the Hebei University of Science and Technology. His research interests include machine learning and deep learning.

TIANHU ZHANG is currently pursuing the master's degree with the Hebei University of Science and Technology. His research interests include machine learning and deep learning.

LIN QI received the B.Eng. degree in computer science and application from the Hebei University of Technology, China, in 1999, and the master's degree in measurement technology and instruments from the Hebei University of Science and Technology, Shijiazhuang, Hebei, China, in 2007. He is currently a Lecturer with the Hebei University of Science and Technology. His research interests include machine learning, deep learning, and big data processing.