North American Journal of Economics and Finance 57 (2021) 101421

Contents lists available at ScienceDirect

North American Journal of Economics and Finance

journal homepage: www.elsevier.com/locate/najef

Forecasting stock index price using the CEEMDAN-LSTM model

Yu Lin [a], Yan Yan [a,*], Jiali Xu [a,*], Ying Liao [a,*], Feng Ma [b,*]

[a] School of Business, Chengdu University of Technology, Chengdu, China
[b] School of Economics and Management, Southwest Jiaotong University, Chengdu, China

A R T I C L E I N F O

JEL classification:
C22
C53
C61
E37

Keywords:
Stock index price forecasting
Long short-term memory
CEEMDAN
Mixture models
MCS test

A B S T R A C T

This paper uses a mixture model that Long Short-Term Memory (LSTM) combines with Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) to forecast stock index price of Standard & Poor's 500 index (S&P500) and China Securities 300 Index (CSI300). CEEMDAN decomposes original data to obtain several IMFs and one residue. The LSTM forecasting model utilizes the decomposed data to obtain the prediction sequences. The prediction sequences are reconstructed to gain final prediction. The paper introduces contrast models such as Support Vector Machine (SVM), Backward Propagation (BP), Elman network, Wavelet Neural Networks (WAV) and their mixture models combined with the CEEMDAN. The MCS test is used as evaluation criterion and empirical results present that forecasting effects of CEEMDAN-LSTM is optimal in developed and emerging stock market.

1. Introduction

Forecasting stock index price is not only concerned by investors and many scholars, but also a focus of government supervision. The stock market is essentially a dynamic, nonlinear, non-stable, noisy and chaotic system (Abu-Mostafa and Atiya, 1996), which increases the difficulty of prediction. The scholars are committed to improving the predictive effect of the model, which is also the objective of this paper.

In the past few decades, methods of predicting financial time series have been mainly divided into statistics and artificial intelligence. Statistical approaches include auto regression (AR), moving average (MA), autoregressive moving average (ARMA) (Kocak, 2017), autoregressive integrated moving average (ARIMA) (Adebiyi, Adewumi, & Ayo, 2014), generalized autoregressive conditional heteroskedastic (GARCH) (Mantri, Gahan, & Nayak, 2010; Lin, Xiao, & Li, 2020), smooth transition autoregressive model (STAR), linear regression (LR), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) (Phichhang and Wang, 2009), support vector machines (SVM) (Huang, Nakamori, & Wang, 2005; Ballings, Dirk, Hespeels, & Gryp, 2015) and so on. According to the number of input variables, these methods can be divided into single variables input and multi-input variables. In the univariate analysis, only the financial original data are taken into account as the input, while the lagged time series and other types of data, such as technology, capital or inter-market indicators, are also taken into account in the multivariate analysis. Furthermore, most of these methods are based on linear, stable and normal distribution assumptions, and the efficiency and accuracy of the forecast results are poor (Chen and Hao, 2017).

Compared with statistical methods, artificial intelligence can handle nonlinear, dynamic and noisy financial time series. The

* Corresponding authors.
E-mail addresses: gayanyan2@163.com (Y. Yan), xu18936643469@163.com (J. Xu), liaoying_ly@outlook.com (Y. Liao), mafeng2016@swjtu.edu.cn (F. Ma).

https://doi.org/10.1016/j.najef.2021.101421
Received 28 February 2020; Received in revised form 11 January 2021; Accepted 16 March 2021
Available online 26 March 2021
1062-9408/

common techniques include Artificial Neural Network (ANN) (Kazem, Sharifi, Hussain, Saberi, & Hussain, 2013; Xi et al., 2014; Rather, Agarwal, & Sastry, 2015; Moghaddam, Moghaddam, & Esfandyari, 2016), Genetic Algorithm (GA) (Rahman, Sarker, & Essam, 2015), Hidden Markov Model (HMM) (Hassan, Nath, & Kirley, 2007), Decision Tree (DT) (Hu, Feng, Zhang, Ngai, & Liu, 2015), Rough Set Theory (Nair, Mohandas, & Sakthivel, 2010), Bayesian Analysis (BA) (Miao, Wang, & Xu, 2015), K-Nearest Neighbors (KNN) (Teixeira et al., 2010; Nayak, Mishra, & Rath, 2015), Particle Swarm Optimization (PSO) (Zhang and Shen, 2009), Deep Learning architectures i.e. Multilayer Perceptron (MLP) (Guresen, Kayakutlu, & Daim, 2011), Recurrent Neural Networks (RNN) (Rout, Dash, Dash, & Bisoi, 2015), Long Short-Term Memory (LSTM) (Baek and Kim, 2018; Cao, Li, & Li, 2019), Convolutional Neural Network (CNN) (Selvin, Vinayakumar, Gopalakrishman, Menon, & Soman, 2017), and so on. The models mentioned above can predict dynamic and nonlinear time series, while different model focuses on solving different problems. Deep Learning architectures have the ability to recognize and utilize the interactions and patterns that exist in time series by a self-learning process (Selvin et al., 2017). Among them, the LSTM model solves the long-term dependence problem in time series and improves the prediction accuracy.

In recent years, the LSTM model has obtained great concern for the rapid development of artificial intelligence and for its end-to-end modeling which could extract features automatically (Assaad, Boné, & Cardot, 2008). The LSTM is widely used in many fields of time series prediction, e.g., agriculture (Zhang, Zhu, Zhang, Ye, & Yang, 2018), automatically convert car images (Chen, He, & Fan, 2017), wind speed (Hu and Chen, 2018), power generation prediction (Han et al., 2019). LSTM has also been gradually applied to the field of financial time series prediction recently due to it can solve the long-term dependence (Fischer and Krauss, 2018; Li, Zhou, Yuan, Geng, & Cai, 2019; Wu, Wu, & Zhu, 2019). Therefore, LSTM is used to predict stock price index in this paper.

Nevertheless, financial time series have the characteristics of dynamic instability and long-term dependence (Wang, 2003). Furthermore, stock index price movements are the results of a variety of factors including macro-economy, corporate finances, investors' sentiments and so on (Long, Lu, & Cui, 2019). And all these factors may cause different extent of effects on the stock market. Some factors could cause short-term and medium-term influence between various stock markets, but the impact will decrease rapidly, while the other factors have great influence in the long-term (Zhou, Lin, & Li, 2018). In this paper, we use the daily closing price of CSI300 and S&P500 as the study object. According to the changes in market conditions and economic environment, there is noise in the original data of stock index price (Cheng and Wei, 2014). We often encounter the problem that how to improve the forecasting performance as possible using the data with noise while forecasting financial time series (Yu, Wang, & Lai, 2009). To solve the problem, this paper uses the empirical mode decomposition's (EMD) advanced versions to decompose original data.

EMD decomposes data with noise based on its own time-scale characteristics, and does not set any basis functions in advance, while has an obvious advantage in dealing with non-stationary and nonlinear data (Huang et al., 1998; Yu, Wang, & Lai, 2008). Based on these characteristics, it could be applied to any type of data decomposition i.e. earthquake (Wang, Zhang, Yu, & Zhang, 2012), fault detection (Zhang, Wang, Tang, Benbouzid, & Diallo, 2017), sea (Senthilkumar, Romolo, Fiamma, Arena, & Murali, 2015), medicine (Li, Zhou, Yuan, Geng, & Cai, 2013), font recognition (Yang, Yang, Qi, & Suen, 2006), economics (Kožić and Sever, 2014), and stock market (Wei, 2016; Wang and Wang, 2017). However, EMD remains the problem of mode mixing. To solve it, EMD's advanced versions have been proposed in succession, for example, ensemble empirical mode decomposition (EEMD) (Wu and Huang, 2009), complementary ensemble empirical mode decomposition (CEEMD) (Yeh, Shieh, & Huang, 2010) and complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) (Torres, Colominas, Schlotthauer, & Flandrin, 2011). Compared with other versions, CEEMDAN has the advantages of avoiding mode mixing and reducing the amount of noise in the modes (Colominas, Schlotthauer, & Torres, 2014). So CEEMDAN is adopted to preprocess the original data to improve the forecasting accuracy of LSTM model in this paper.

Therefore, a novel model CEEMDAN-LSTM is utilized for stock index price forecasting in this study. The paper uses MAPE, MSE, RMSE and MAE loss functions to evaluate prediction quality of the models, and the CEEMDAN-LSTM model is further verified superiority via MCS test. The purposes of this paper are to study whether the model can accurately predict stock index price and whether the forecasting accuracy has a significant difference in emerging and developed markets.

The remainder of this paper consists of the following sections: Section 2 elaborates the method adopted in this paper. Section 3 mainly introduces the concrete process of empirical research and analyzes the empirical results. The eventual Section 4 makes a conclusion.

2. Methodology

2.1. Empirical mode decomposition (EMD)

The EMD algorithm is proposed by Huang et al. (1998), which decomposes the complex original signal into a series of IMFs with different amplitude and a residual. Its essence is the transition from nonlinear and non-stationary signals to linear and stationary.

The decomposing steps of the EMD method are as follows: (1) Draw the upper and lower envelops according to the maximum and minimum of the original signal $S(t)$ by performing cubic spline interpolation. (2) Calculate the mean values of the upper and lower envelop, and draw the mean envelop $M(t)$. (3) The mean envelop is subtracted from the original signal to obtain the intermediate signal $h^1(t)$. (4) Judge whether the $h^1(t)$ satisfies the following two conditions: *i)* During the whole time range, the local extreme values points and zero crossing points must be equal, or at most one difference; *ii)* At any time point, the mean values of the upper and lower envelop must be zero. If so, the $h^1(t) = C_1(t) = IMF_1$; if not, think of $h^1(t)$ as a new signal to do 1) to 4). Repeat k times until $h^k(t)$ meets the definition of IMF, then extract the IMF with the highest frequency that is: $h^k(t) = C_1(t) = IMF_1$. (5) Subtract IMF_1 from the original signal $S(t)$ gets a new series $R(t) = S(t) - IMF_1$. Repeat above step (1) - (4) to give the IMF_2. (6) Repeat the above five steps, when $R(t)$ is constant quantity or becomes a monotone function, the EMD decomposition of the original signal ends.

Finally, n orthogonal IMFs are reconstructed by summation: $S(t) = \sum_i IMF_i(t) + R(t)$.

2.2. Complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN)

The Ensemble Empirical Mode Decomposition (EEMD) algorithm adds normally distributed white noise to the original signal based on the EMD algorithm, so that the signal is evenly distributed at the extreme point interval throughout the band, which reduces the mode mixing effect.

CEEMDAN algorithm adds finite adaptive white noise based on EEMD, and overcomes the problems of EEMD's incompleteness and reconstruction error after adding white noise. The specific implementation steps of the algorithm are as follows:

Step 1: Add white noise $v^i(t)$ with a standard normal distribution to the original signal $S(t)$. The signal for the ith is represented as $S^i(t) = S(t) + v^i(t), i = 1, 2, \cdots, I$. The EMD decomposed the experimental signal$S^i(t)$ to obtain IMF_1^i correspondingly, so $IMF_1 = \frac{1}{I}\sum_{i=1}^{I} IMF_1^i$, residual $r_1(t) = S(t) - IMF_1$.

Step 2: Add white noise $v^i(t)$ to the residual $r_1(t)$, perform i times experiment ($i = 1, 2, \cdots, I$), and each experiment adopts EMD to decompose $r_1^i(t) = x(t) + v^i(t)$ to obtain its first-order component IMF_1^i. $IMF_2 = \frac{1}{I}\sum_{i=1}^{I} IMF_1^i$, and residual $r_2(t) = S(t) - IMF_2$.

Step 3: Repeat the above decomposition process to obtain the IMF component that meets the conditions and the corresponding residuals. When the residual is monotony functions and cannot be decomposed by EMD, the program terminates. The original signal can be represented as $S(t) = \sum_{i=1}^{n} IMF_i + r_n(t)$.

2.3. Long short-term memory (LSTM)

LSTM is a neural network with memory ability, which is suitable for processing and predicting important events with relatively long interval and delay in time series. LSTM introduces gate mechanism on the structure of RNN to avoid the problem that effective historical information cannot be preserved for a long time.

LSTM consists of three gate structures which are shown in Fig. 1: forget gate, input gate and output gate. Each gate structure is a sigmoid layer, whose inputs are the output h_{t-1} of the previous network module and the input x_t of the current module.

The steps of LSTM are as follows:

(1) The "forget gate" determines which information will be discarded from the previous cell state C_{t-1}. If its output f_t is 1, it means "keep this information completely" and if the output f_t is 0, it will "drop this information completely".

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

Where W_f is weight matrix, b_f is bias vectors, σ refers to sigmoid function.

(2) This step primarily determines which new inputs need to be retained. The output i_t of "input gate" to decide which values need to be updated; Then, according to the output of the h_{t-1} and the input x_t, use the tanh layer (output values between -1 and 1) to produce a new candidate value that will be added to the cell state C_t.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

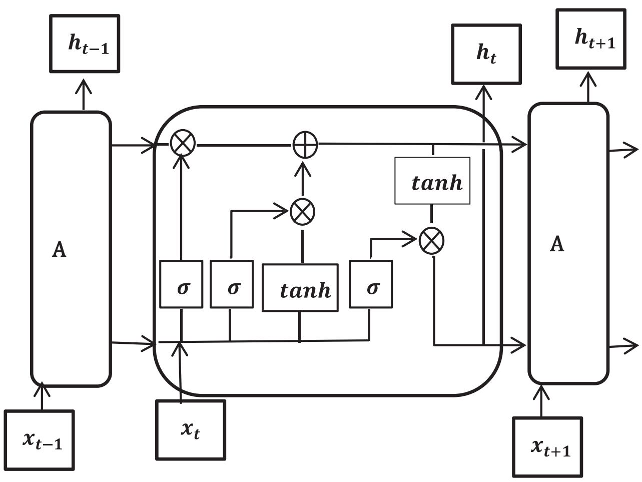


Fig. 1. The cell of long short-term memory neural network.

$$C_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

Where C_t represents the cell state update value, which is obtained by the input data x_t and cryptic nodes h_{t-1} through calculation; i_t is called an input gate, it is a vector where each element is within [0,1], calculated by x_t and h_{t-1} through the sigmoid activation function.

(3) Multiply the old state C_{t-1} by the output f_t of "forget gate" to forget the old information. Then we multiply the new candidate status C_t by the output i_t of "input gate" to add the new information we want.

$$C_t = f_t^* C_{t-1} + i_t^* C_t \tag{4}$$

Where f_t is forget gate, it means that these features of C_{t-1} are used to calculate C_t. f_t is a vector and each element of the vector is within [0,1].

(4) Use the current cell state to determine the output of the current network module. Firstly, using "output gate" to determine which parts of the cell state are output. Then use a *tanh* activation function on the current state of C_t to get the candidate value for the current output; multiply the value by the output o_t of the output gate to get the h_t of the current network module.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t^* tanh(C_t) \tag{6}$$

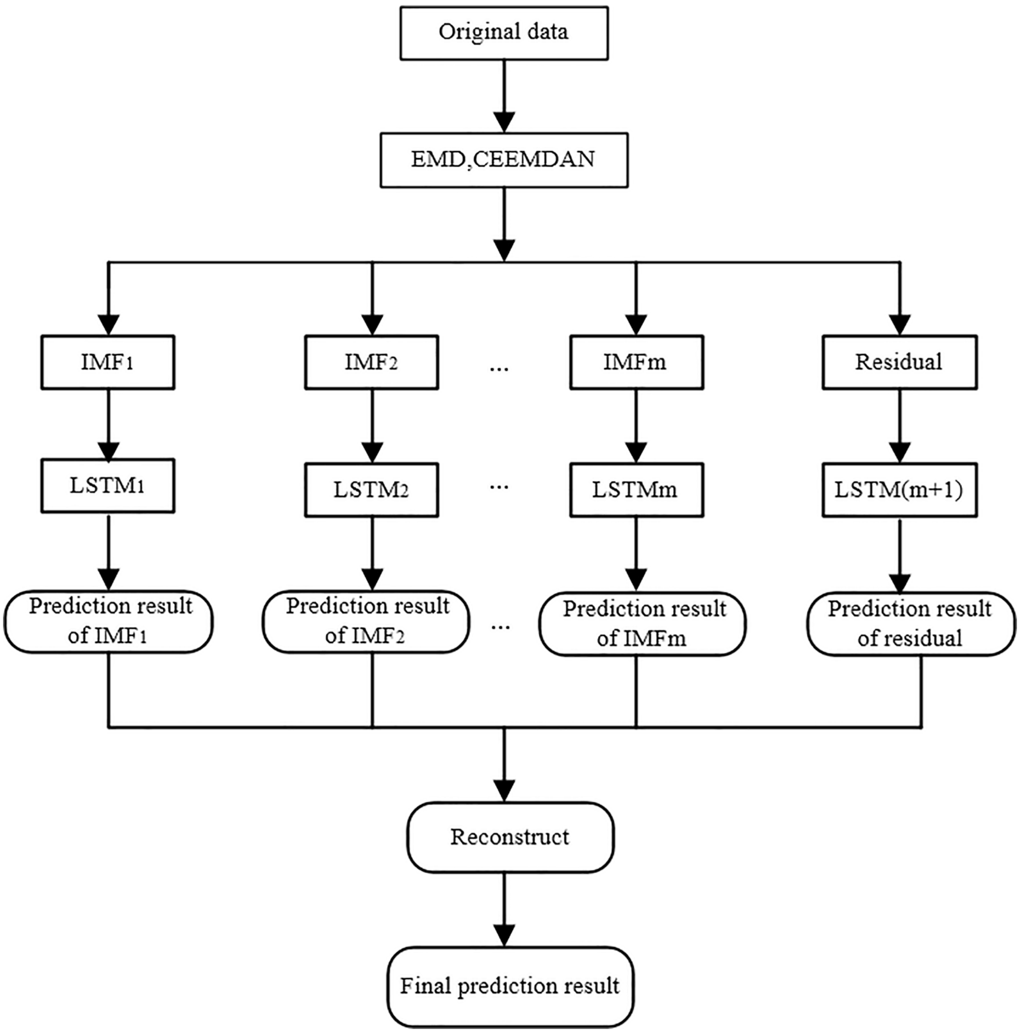


Fig. 2. Combined forecast flow chart based on CEEMDAN-LSTM.

2.4. The proposed CEEMDAN-LSTM forecasting model

In the LSTM model, the "ReLU" function is used as the activation function of the fully connected layer and the mean square error (MSE) is used as the loss function.

$$Loss = MSE = \frac{1}{N}\sum_{n=1}^{N}(d_n - y_n)^2 \tag{7}$$

Where N is the total predicted number of days, y_n is predictive value, and d_n is actual value.

Before using the LSTM model, the CEEMDAN method is used to decompose the original data to obtain several smooth sequences. The CEEMDAN-LSTM method is applied to forecast stock price index. The proposed CEEMDAN-LSTM model is generally composed of the following three main steps which are shown in Fig. 2.

Step 1: The original financial time series is decomposed into several IMFs sequences $C_i(t)(i = 1, 2, \cdots, M)$ and one residue $R_M(t)$ through EMD and CEEMDAN.

Step 2: The IMFs and the residue are used as the inputs to the LSTM model for training and receiving the predicted values respectively. The predicted results of the test set are $C_i(t)(i = 1, 2, \cdots, M)$ and $R_M(t)$.

Step 3: According to the following formula, the forecast results have been adjusted to achieve the final predicted result.

$$S_i(t) = \sum_{i=1}^{M} C_i(t) + R_M(t)(t = 1, 2, \cdots, L) \tag{8}$$

Where L is the length of the test series, and $S_i(t)$ is the final predictive series of the test set.

2.5. Evaluation criteria

So far, it is not clear which loss function is the most reasonable standard to measure the predictive effects. To accurately measure the forecasting performance, six different loss functions are used as the evaluation criteria of prediction accuracy of various time series models in the empirical research of this paper. The six loss functions are denoted as $L_i(i = 1, 2, \cdots, 6)$ respectively, where L_1 is Mean Squared Error (MSE) and L_2 is Mean Absolute Error (MAE), L_3 and L_4 are respectively MSE and MAE adjusted by heteroscedasticity. The specific definitions of each loss function are as follows:

$$L_1 : MSE = D^{-1}\sum_{d=H+1}^{H+D}(d_t - y_t)^2 \tag{9}$$

$$L_2 : MAE = D^{-1}\sum_{d=H+1}^{H+D}|d_t - y_t| \tag{10}$$

$$L_3 : HMSE = D^{-1}\sum_{d=H+1}^{H+D}\left(1 - \frac{y_t}{d_t}\right)^2 \tag{11}$$

$$L_4 : HMAE = D^{-1}\sum_{d=H+1}^{H+D}\left|1 - \frac{y_t}{d_t}\right| \tag{12}$$

$$L_5 : QLIKE = D^{-1}\sum_{d=H+1}^{H+D}\left(ln(y_t) + \frac{d_t}{y_t}\right) \tag{13}$$

$$L_6 : R^2LOG = \mathrm{D}^{-1}\sum_{d=H+1}^{H+D}\left(ln\left(\frac{d_t}{y_t}\right)\right)^2 \tag{14}$$

Where D is the total predicted number of days, $H+1$ is the predicted start date, $H+D$ is the predicted end date, y_t is predictive value, and d_t is actual value.

When we use different loss functions as evaluation criteria for predicting effects, the conclusions are often different. So, the question is which model is optimal. Hansen, Lunde, and Nason (2011) presented the model confidence set (MCS) test to solve this problem, and the basic setting of the MCS test is as follows:

(1) There are m_0 models to be tested, and the collection set of the m_0 models is the original set M^0. When the loss function is selected, let $L_{i,t}$ be the loss function value of the ith model in time t, so that we can calculate the relative loss function value of any two models in the M^0 set, which is called the relative performance variables in MCS, defined as follows:

$$d_{ij,t} = L_{i,t} - L_{j,t}, \mu_{ij} = E(d_{ij,t}) \tag{15}$$

(2) Define the set of superior objects as M^*, which needs to meet the conditions are:

$$M^* = \{ i \in M^0, \mu_{ij} \le 0, \forall j \in M^0 \} \tag{16}$$

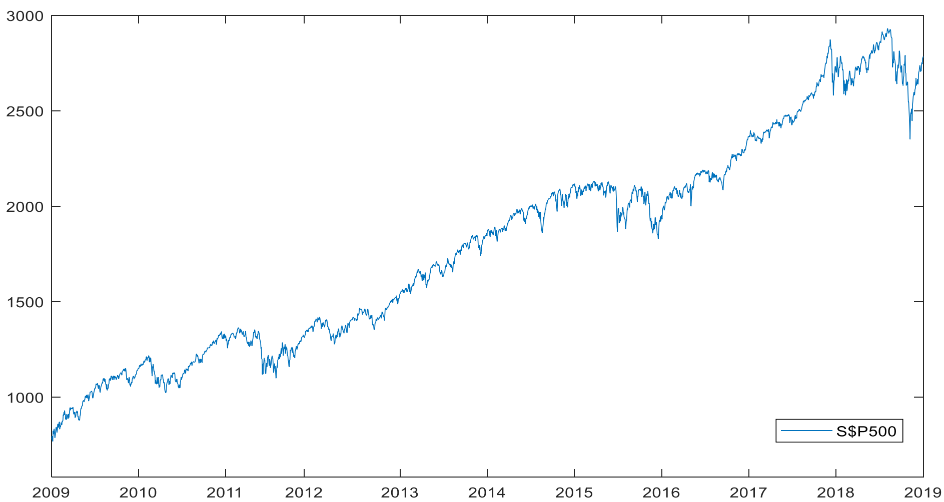
Where M^* is the result that the MCS seeks which means the performance of all models in M^* is the same or better than that in M^0.

The checkout procedure of MCS follows two basic laws that are the equivalence test (δ_M) and elimination rule (e_M). The equivalence test assumes that the models have similar forecasting performances on stock index price under the given significance level α. When the hypothesis of the equivalence test is rejected, we use the elimination rule to remove the model which is significantly inferior to the concentration of the model, and get $M \in M^0$. Finally, when the hypothesis of the equivalence test is no longer rejected, we get $\widehat{M}^*_{1-\alpha} = M$. $\widehat{M}^*_{1-\alpha}$ is not equivalent to M^* and represents a superior set of objects derived under the confidence level $1-\alpha$.

Where the original assumption of the equivalent test can be expressed as:

$$H_{0,M}: \mu_{ij} = 0, \forall i, j \in M \tag{17}$$

(a) CSI300



(b)S&P500

Fig. 3. The original series of stock daily closing price.

The alternative hypothesis is:

$$H_{A,M}: \mu_{ij} = 0, \exists i, j \in M \tag{18}$$

The statistics of MCS adopted in this paper is semi-quadratic statistic T_{SQ}, which is defined as:

$$T_{SQ} = max_{i,j \in M} \frac{\left(\overline{d}_{ij}\right)^2}{var\left(d_{ij}\right)} \tag{19}$$

$$\overline{d}_{ij} = \frac{1}{M} \sum_{d=H+1}^{H+D} d_{ij,t} \tag{20}$$

Where $\overline{d}_{ij}$ is the average value of the relative loss function of model i and j. If the p-value of statistic T_{SQ} is greater than the given significance level α, the null hypothesis is rejected.

3. Analysis of data

3.1. The statistical analysis of data

In this paper, the daily closing price of the CSI300 and S&P500 are selected as the original financial time series, which are both obtained from Wind database. The data of CSI300 are from November 10, 2008 to February 20, 2019 and the S&P500 is from March 17, 2009 to February 20, 2019 both with a total of 2500 observations, which eliminated the non-trading-time data in this paper. Furthermore, we select the top 90% of the data as training set to model training and the remaining 10% data are used as the test set to evaluate the model performance.

Fig. 3 is the daily closing price of S&P500 and CSI300 which shows the short-term instability and explosion. It is obvious that the CSI300 has a sharp fluctuated in 2005 and in other years has fluctuation with different amplitude by Fig. 3(a), and the S&P500 shows a trend of gradual increase by Fig. 3(b). Table 1 describes the statistical analysis of the original time series. The skewness value is greater than 0, which indicates the two stock indices are skewed to the right. The kurtosis values show the two stock indices are not comply with the normal distribution and have an obvious leptokurtosis. So, we use EMD and CEEMDAN models in this paper to decompose the original data.

3.2. Decomposed by EMD and CEEMDAN

Several IMFs and a residual are obtained by decomposing the original time series with EMD and CEEMDAN. The curve gets smoother, and the frequency gets lower from IMF1 to residual by Fig. 4.

The statistical characteristics of each IMF are described in Tables 2 and 3. Compared with the original sequence, the IMFs are more similar to the normal distribution, which could be found from the values of Skewness and Kurtosis in Tables 1–3. It is also clear that the IMFs decomposed by CEEMDAN are closer to normal distribution than EMD via Tables 2 and 3. Moreover, the t-value of the ADF test is less than the critical value of 5% significance level, which indicates that CEEMDAN can decompose the non-stable sequence into several independent stable sequences.

After decomposing, the decomposed data is normalized to obtain final predictive value. In this paper, the normalization method of linear function is adopted, and the normalization formula is:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{21}$$

Where X_{norm} is normalized data,X_{min} is the minimum value in the sample, and X_{max} is the maximum value in the sample. The original data are normalized to the range between [0,1] to achieve equal scaling of the original data. Finally, the forecasting value of the test set can be restored by the following formula:

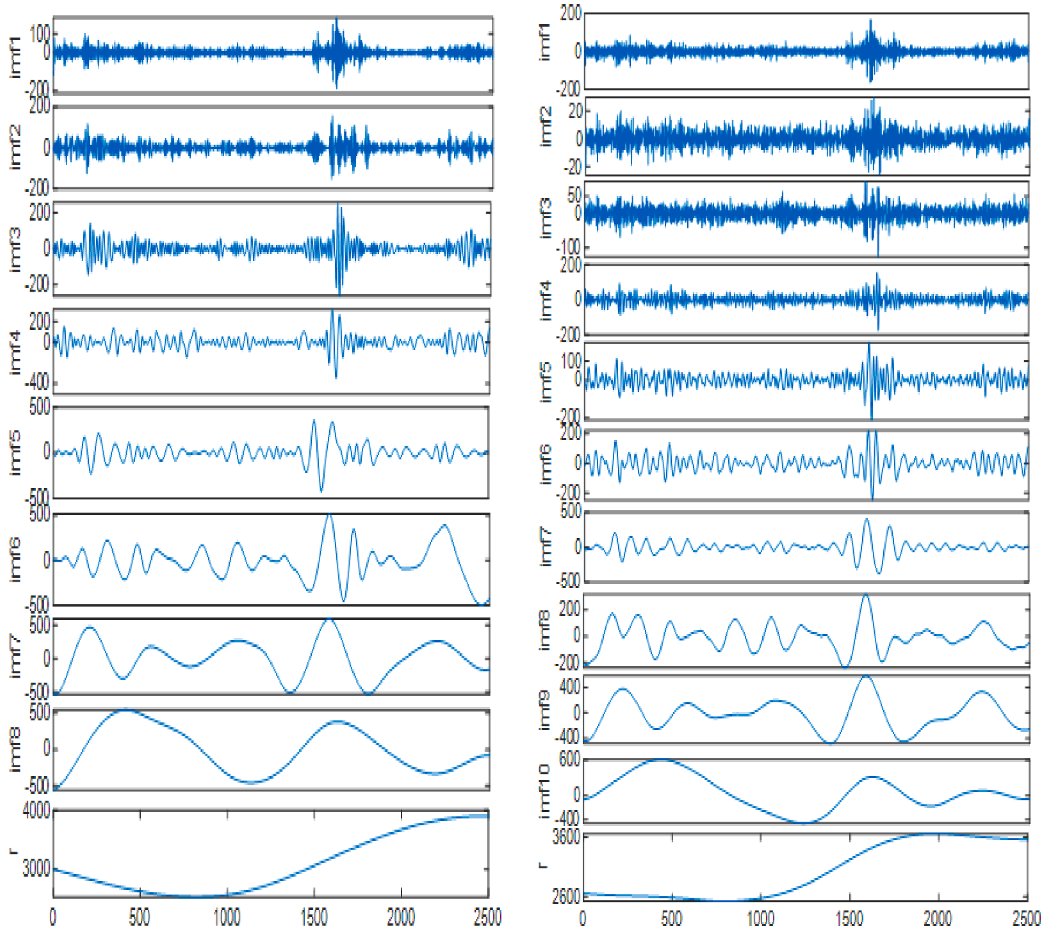
$$X_t = X_t^{'}(X_{max} - X_{min}) + X_{min} \tag{22}$$

$X_t^{'}$is the output value of the forecasting model.

Table 1
The statistical analysis of two stock index price.

Index	Count	Mean	Min	Max	Standard deviation	Skewness	Kurtosis
S&P500	2500	1797.0254	768.5400	2930.7500	559.7082	0.2006***	−1.0782***
CSI300	2500	3050.9328	1781.3600	5353.7500	627.6603	0.4213***	0.0434*

Note: Note: *, ** and *** represents the significance level at 10%, 5% and 1% respectively.



(a) Decomposed by EMD (b) Decomposed by CEEMDAN

Fig. 4. Decomposition results of CSI300 index.

Table 2
Frequency profile of each IMF decomposed by EMD (CSI300).

Component	Mean	Standard deviation	Skewness	Kurtosis	ADF
IMF1	0.4978	26.9436	−0.0097*	7.2007***	−46.0902**
IMF2	0.4510	28.8347	−0.0913***	3.0666***	−25.6302**
IMF3	−0.0134	41.8369	−0.1080***	6.7458***	−20.2931**
IMF4	1.2348	63.1550	−0.0592**	4.8330***	−15.6225**
IMF5	1.7000	89.9838	−0.0102*	4.6268***	−11.7143**
IMF6	−21.3874	187.3070	−0.0794**	0.6865***	−7.3162**
IMF7	1.6474	269.8499	−0.1526***	−0.5418***	−5.7452**
IMF8	−4.1296	313.2154	0.1687***	−1.2848***	−6.0403**

Note: *, ** and *** represents the significance level at 10%, 5% and 1% respectively.

3.3. *Training process*

After decomposition, each subsequence is divided into two parts: training set and test set, and then LSTM is used to train training set. In order to obtain the prime prediction effects of the subseries, the optimal hyper-parameters are selected through multiple trainings as shown in Tables 4 and 5. The “MiniBatchSize’’ represents the length of each sample series, and “epoch’’ is the times of training.

Table 3
Frequency profile of each IMF component decomposed by CEEMDAN (CSI300).

Component	Mean	Standard deviation	Skewness	Kurtosis	ADF
IMF1	0.1564	23.3486	−0.1016***	7.2261***	−12.0499**
IMF2	0.0430	5.8213	0.0048*	1.3591***	−29.7825**
IMF3	−0.0258	19.7446	−0.1024***	1.8198***	−26.6297**
IMF4	0.0267	27.1926	−0.0247**	3.0456***	−21.4082**
IMF5	0.1821	36.2834	0.1118***	4.1777***	−17.8594**
IMF6	−0.1954	53.3467	0.1366***	2.4183***	−14.7952**
IMF7	−1.8896	95.5356	0.0349**	4.5727***	−10.9017**
IMF8	−7.0787	99.8032	0.1865***	0.5104***	−6.9373**
IMF9	−21.0112	246.1680	0.1214***	−0.4770***	−5.5474**
IMF10	57.2477	289.4839	0.1475***	−0.6597***	−4.5034**

Note: *, ** and *** represents the significance level at 10%, 5% and 1% respectively.

Table 4
Hyper-parameters of EMD-LSTM.

	CSI300		S&P500	
	MiniBatchSize	Epoch	MiniBatchSize	Epoch
IMF1	3	320	2	250
IMF2	2	150	2	150
IMF3	2	150	2	150
IMF4	4	150	4	100
IMF5	4	150	5	250
IMF6	5	150	5	150
IMF7	4	150	4	150
IMF8	8	110	9	200
Residue	8	150	10	100

Table 5
Hyper-parameters of CEEMDAN-LSTM.

	CSI300		S&P500	
	MiniBatchSize	Epoch	MiniBatchSize	Epoch
IMF1	3	320	3	320
IMF2	3	150	3	300
IMF3	3	150	3	250
IMF4	2	150	2	150
IMF5	2	150	2	150
IMF6	2	150	2	150
IMF7	2	150	2	150
IMF8	2	110	2	110
IMF9	8	150	8	150
IMF10	7	150	9	100
Residue	8	150	10	100

3.4. Experimental result

Fig. 5 shows the forecasting effects of subseries for CSI300 index test set. The predicted values of the first several IMFs components visibly deviated from the original data owing to the high frequency of the components, and that of the latter IMF almost coincides with the original data.

Fig. 6 shows the predictive effects of the two indices, which demonstrates the predicted values of the two series are very close to the original value. Furthermore, we introduce Mean Absolute Percentage Error (MAPE), MAE, MSE and Root Mean Square Error (RMSE) as the evaluation criterion of forecasting performances for methods.

$$RMSE = \sqrt{D^{-1}\sum_{d=H+1}^{D+H}(d_t - y_t)^2} \tag{23}$$

$$MAPE = 100 \times D^{-1}\sum_{d=H+1}^{D+H}\left|\frac{d_t - y_t}{d_t}\right| \tag{24}$$

Where D is the total predicted number of days, $H+1$ is the predicted start date, $H+D$ is the predicted end date, y_t is predictive value

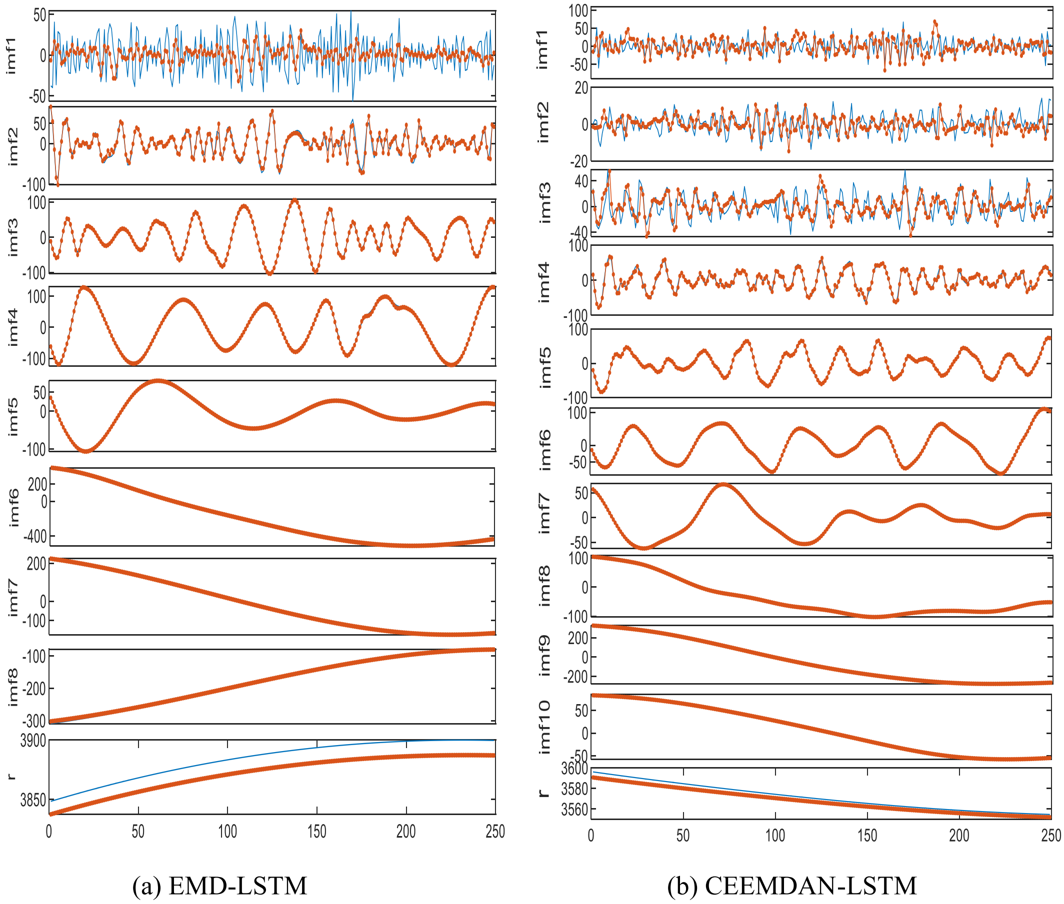


Fig. 5. Forecasting results of subseries for CSI300 index.

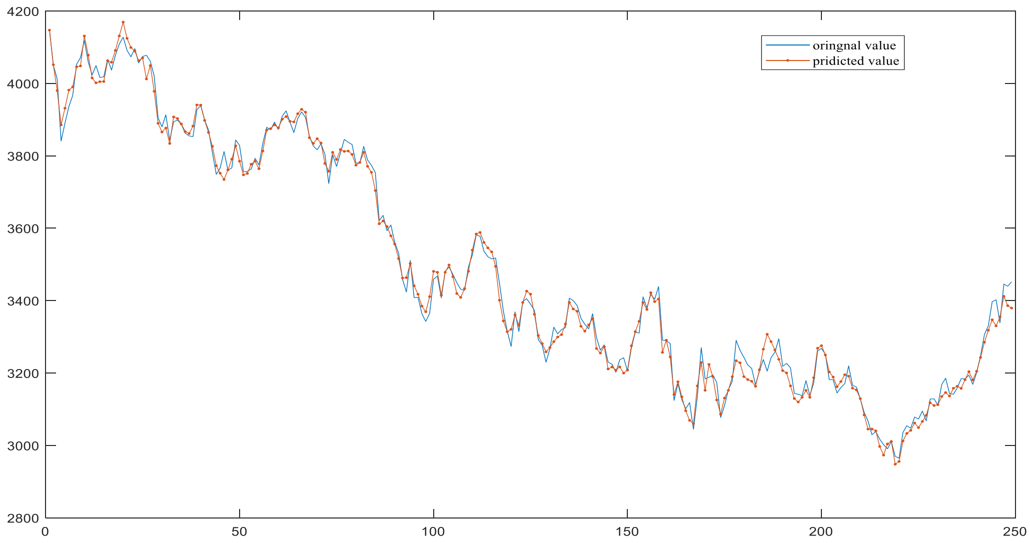
and d_t is true value. If the value of MAPE, MAE, MSE and RMSE is smaller, the predicted value is closer to the true value and predictive effect is better.

The forecasting error of EMD-LSTM and CEEMDAN-LSTM are reported in Table 6, and the values in bold indicates that the model is optimal under the judgment of the loss function. The decompose effect of CEEMDAN precedes EMD through the four loss-function values comparison. By comparing the bold value under the same function, the loss function value of CSI300 is not significantly different from that of S&P500, which indicates the forecasting effect of CEEMDAN-LSTM in developed market is similar to that of in emerging market.

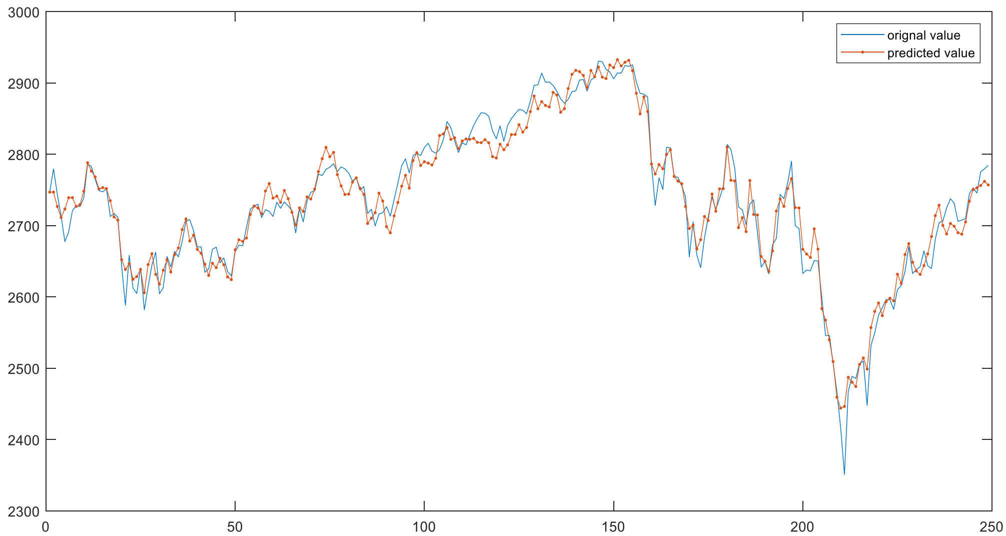
To evaluate prediction performance of CEEMDAN-LSTM model, four different models, including Support Vector Machine (SVM), Backward Propagation (BP), Elman network and Wavelet Neural Networks (WAV), are used for comparison in this paper. Meanwhile, these models combined with CEEMDAN are also considered to compare with CEEMDAN-LSTM.

The predictive effects of the mixture and single models are shown in Table 7, and the forecasting performance of all mixture methods are better than the corresponding single methods. And it also proves the CEEMDAN-LSTM precedes the other mixture models in predicting the CSI300 and S&P500 index price. Furthermore, CEEMDAN-LSTM is optimal. By calculating the difference between the loss function values of the single model and the mixture model, it can be found that the LSTM model is the most improved and easy to optimize. Additionally, Table 7 also describes the forecasting precision of CEEMDAN-LSTM in S&P500 is similar to that in CSI300 via comparing bold values under the same function, which testify the model has good stability.

To accurately evaluate the prediction performance of these methods, we adopt the MCS test and its result is reported in Table 8. The first column shows fifteen forecasting models, the first-row acts for six loss functions, and the values in the table represent p-values for MCS statistics under the corresponding model. The underlined means the p-value greater than 0.25 which is the threshold value we set that indicates the model survived under the loss function. The p-value is larger, the model accuracy is higher, and the boldface represents the best model in this group. As can be seen from Table 8, only the CEEMDAN-LSTM model passes the test under all loss function, CEEMDAN-SVM, CEEMDAN-WAV and EMD-SVM model pass the test under HMSE and QLIKE loss function, CEEMDAN-SVM and CEEMDAN-WAV model pass the test under MAE and HMAE loss function, CEEMDAN-SVM and EMD-SVM pass the test under MSE loss function. None of the remaining models passed the test of any loss function. The above analysis shows that the best model of prediction effect is CEEMDAN-LSTM.



(a) CSI300



(b) S&P500

Fig. 6. Forecasting results of two series by CEEMDAN-LSTM.

Table 6
Forecasting error of EMD-LSTM and CEEMDAN-LSTM.

Index	Model	MAE	MAPE	MSE	RMSE
S&P500	EMD-LSTM	30.2808	0.0111	1108.1015	33.2882
	CEEMDAN-LSTM	**16.3634**	**0.0060**	**436.0208**	**20.8811**
CSI300	EMD-LSTM	24.6992	0.0072	927.0769	30.4479
	CEEMDAN-LSTM	**18.9512**	**0.0055**	**615.0082**	**24.8008**

Note: Bold means the model is optimal under the loss function.

In this paper, it is necessary to take a robust test for CEEMDAN-LSTM. The robust test primarily divides the data of the test set into five subsequences, and then uses RMSE, MAE, MSE and MAPE loss functions to assess the forecasting performance of the model. Finally, compare the loss function values of each subsequence. The RMSE, MAE, MSE and MAPE loss functions value of each

Table 7
Comparison of prediction measures of different prediction models.

Index	Model	MAE	MAPE	MSE	RMSE
S&P500	LSTM	37.1056	0.0137	4554.9380	67.4903
	SVM	31.7137	0.0117	1386.4169	37.2346
	BP	26.2110	0.0096	1215.0920	34.8582
	Elman	29.4222	0.0107	1337.2046	36.5678
	WAV	35.2671	0.0130	2025.7638	45.0085
	CEEMDAN-LSTM	**16.3634**	**0.0060**	**436.0208**	**20.8811**
	CEEMDAN-SVM	16.8448	0.0062	467.2041	21.6149
	CEEMDAN-BP	21.7018	0.0079	732.4395	27.0636
	CEEMDAN-Elman	21.2412	0.0079	849.0972	29.1393
	CEEMDAN-WAV	18.4868	0.0067	603.9389	24.5752
CSI300	LSTM	44.3789	0.0127	3646.5100	60.3863
	SVM	37.4073	0.0107	2375.7305	48.7415
	BP	36.8710	0.0107	2319.6892	48.1632
	Elman	37.2899	0.0170	2327.1569	48.2406
	WAV	62.2222	0.0183	5500.6258	74.1662
	CEEMDAN-LSTM	**18.9512**	**0.0055**	**615.0082**	**24.8008**
	CEEMDAN-SVM	22.7804	0.0067	842.6585	29.0286
	CEEMDAN-BP	20.9586	0.0061	715.0465	26.7404
	CEEMDAN-Elman	36.8115	0.0106	2266.4362	47.6071
	CEEMDAN-WAV	28.7991	0.0085	1358.2591	36.8546

Note: Bold means the model is optimal under the loss function.

Table 8
MCS test results of different model prediction results (CSI300).

Loss function	MSE	MAE	HMSE	HMAE	QLIKE	R^2LOG
LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
EMD-LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
CEEMDAN-LSTM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
BP	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
EMD-BP	0.0045	0.0554	0.0089	0.0452	0.0081	0.0000
CEEMDAN-BP	0.0003	0.0002	0.0011	0.0002	0.0005	0.0000
Elman	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
EMD- Elman	0.0000	0.0000	0.0000	0.0000	0.0000	0.0926
CEEMDAN- Elman	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
WAV	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
EMD-WAV	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
CEEMDAN-WAV	0.1699	0.2891	0.3181	0.3340	0.2971	0.0000
SVM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
EMD-SVM	0.3246	0.2181	0.4014	0.2358	0.3255	0.0271
CEEMDAN-SVM	0.4971	0.6449	0.4949	0.6475	0.4421	0.0000

Note: The underlined means the p-value greater than 0.25 (the p-value is larger, the model accuracy is higher), the bold represents the best of all models.

subsequence with the developed and emerging stock market is demonstrated in Table 9. The forecasting performance of CEEMDAN-LSTM model in different subsequences from the same market is similar, and the forecasting performance is similar to the developed and emerging stock market. According to the analysis above, it can conclude that the CEEMDAN-LSTM model is robust.

Table 9
The values of RMSE loss function for each subsequence.

Index	Time	RMSE	MAE	MSE	MAPE
CSI300	2018/2/6–2018/4/24	26.4953	20.0819	702.0029	0.0051
	2018/4/25–2018/7/6	19.9658	16.1034	399.0742	0.0044
	2018/7/9–2018/9/14	19.8850	16.1923	397.1815	0.0048
	2018/9/17–2018/12/3	30.8755	24.0720	969.7949	0.0075
	2018/12/4–2019/2/20	25.0257	23.2425	938.1643	0.0073
S&P500	2018/2/23–2018/5/3	18.4732	13.5070	341.2596	0.0051
	2018/5/4–2018/7/16	17.9716	14.6970	322.9784	0.0054
	2018/7/17–2018/9/25	21.8726	18.7763	478.4123	0.0066
	2018/9/26–2018/12/6	20.2725	15.8617	410.9724	0.0058
	2018/12/7–2019/2/20	25.1071	19.0284	630.3684	0.0073

4. Conclusion

CEEMDAN-LSTM model is used to predict ten years of historical data of two indices namely S&P500 from developed stock markets and CSI300 from emerging stock markets in this paper. Firstly, the original data is decomposed by the CEEMDAN and EMD method. Then the decomposed data is predicted by the LSTM model to obtain forecasting value. Finally, the forecasting value is reconstructed to get final forecasting value. The paper adds SVM, BP, Elman and WAV models as comparative models, and uses MAE, MAPE, MSE and RMSE to evaluate the accuracy of single model and hybrid model, then concludes that the CEEMDAN-LSTM model is optimal. The advantages of CEEMDAN-LSTM are further verified by MCS test. Additionally, the experimental analysis demonstrates that forecasting accuracy of the CEEMDAN-LSTM model is similar in developed stock markets and emerging stock markets, and the method has a great robustness.

The accurate prediction of stock index price is very important for market participants, including buyers and sellers to maximize their returns. The model adopted in this paper has good prediction accuracy and robustness, which can effectively predict the trend of price changes, and effectively avoid investment losses. Meanwhile, it can accurately identify abnormal fluctuations in the market, and also play a certain reference role for regulators.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 71771032).

References

Abu-Mostafa, Y. S., & Atiya, A. F. (1996). Introduction to financial forecasting. *Applied Intelligence, 6*(3), 205–213. https://doi.org/10.1007/BF00126626

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics, 2014*, 1–7. https://doi.org/10.1155/2014/614342

Assaad, M., Boné, R., & Cardot, H. (2008). A new boosting algorithm for improved time-series forecasting with recurrent neural networks. *Information Fusion, 9*(1), 41–55. https://doi.org/10.1016/j.inffus.2006.10.009

Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications, 113*, 457–480. https://doi.org/10.1016/j.eswa.2018.07.019

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications, 42*(20), 7046–7056. https://doi.org/10.1016/j.eswa.2015.05.013

Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications, 519*, 127–139. https://doi.org/10.1016/j.physa:2018.11.061

Kocak, C. (2017). ARMA(p,q) type high order fuzzy time series forecast method based on fuzzy logic relations. *Applied Soft Computing, 58*, 92–103. https://doi.org/10.1016/j.asoc.2017.04.021

Chen, L., He, Y., & Fan, L. (2017). Let the robot tell: Describe car image with natural language via LSTM. *Pattern Recognition Letters, 98*, 75–82. https://doi.org/10.1016/j.patrec.2017.09.007

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications, 80*, 340–355. https://doi.org/10.1016/j.eswa.2017.02.044

Cheng, C.-H., & Wei, L.-Y. (2014). A novel time-series model based on empirical mode decomposition for forecasting TAIEX. *Economic Modelling, 36*, 136–141. https://doi.org/10.1016/j.econmod.2013.09.033

Colominas, M. A., Schlotthauer, G., & Torres, M. E. (2014). Improved complete ensemble EMD: A suitable tool for biomedical signal processing. *Biomedical Signal Processing and Control, 14*, 19–29. https://doi.org/10.1016/j.bspc.2014.06.009

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications, 38*(8), 10389–10397. https://doi.org/10.1016/j.eswa.2011.02.068

Han, S., Qiao, Y.-H., Yan, J., Liu, Y.-Q., Li, L.i., & Wang, Z. (2019). Mid-to-long term wind and photovoltaic power generation prediction based on copula function and long short term memory network. *Applied Energy, 239*, 181–191. https://doi.org/10.1016/j.apenergy.2019.01.193

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. *Econometrica, 79*(2), 453–497.

Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of HMM, ANN and GA for stock market forecasting. *Expert Systems with Applications, 33*(1), 171–180. https://doi.org/10.1016/j.eswa.2006.04.007

Hu, Y.-L., & Chen, L. (2018). A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and Differential Evolution algorithm. *Energy Conversion and Management, 173*, 123–142. https://doi.org/10.1016/j.enconman.2018.07.070

Hu, Y., Feng, B., Zhang, X., Ngai, E. W. T., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications, 42*(1), 212–222. https://doi.org/10.1016/j.eswa.2014.07.059

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., … Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceeding, 454*(1971), 903–995.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32*(10), 2513–2522. https://doi.org/10.1016/j.cor.2004.03.016

Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K. (2013). Support vector regression with chaos-bases firefly algorithm for stock market price forecasting. *Applied Soft Computing, 13*(2), 947–958.

Kožić, I., & Sever, I. (2014). Measuring business cycles: Empirical mode decomposition of economic time series. *Economics Letters, 123*(3), 287–290. https://doi.org/10.1016/j.econlet.2014.03.009

Li, S., Zhou, W., Yuan, Q.i., Geng, S., & Cai, D. (2013). Feature extraction and recognition of ictal EEG using EMD and SVM. *Computers in Biology and Medicine, 43*(7), 807–816. https://doi.org/10.1016/j.compbiomed.2013.04.002

Li, Y., Zhu, Z., Kong, D., Han, H., & Zhao, Y. (2019). EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowledge-Based Systems, 181*, 104785. https://doi.org/10.1016/j.knosys.2019.05.028

Lin, Y., Xiao, Y., & Li, F. (2020). Forecasting crude oil price volatility via a HM-EGARCH model. *Energy Economics, 87*, 104693. https://doi.org/10.1016/j.eneco.2020.104693

Long, W., Lu, Z., & Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems, 164*, 163–173. https://doi.org/10.1016/j.knosys.2018.10.034

Mantri, J. K., Gahan, P., & Nayak, B. B. (2010). Artificial neural networks-an application to stock market volatility. *International Journal of Engineering Science and Technology, 2*(5), 1451–1460.

Miao, J., Wang, P., & Xu, Z. (2015). A Bayesian dynamic stochastic general equilibrium model of stock market bubbles and business cycles: Stock market bubbles and business cycles. *Quantitative Economics, 6*(3), 599–635. https://doi.org/10.3982/QE505

Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science, 21*(41), 89–93. https://doi.org/10.1016/j.jefas.2016.07.002

Nair, B. B., Mohandas, V. P., & Sakthivel, N. R. (2010). A decision tree-rough set hybrid system for stock market trend prediction. International Journal of Computer Applications, 6(9), 1–6.

Nayak, R. K., Mishra, D., & Rath, A. K. (2015). A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices. *Applied Soft Computing, 35*, 670–680. https://doi.org/10.1016/j.asoc.2015.06.040

Phichhang, O., & Wang, H. S. (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science, 3*(12), 28.

Rahman, H. F., Sarker, R., & Essam, D. (2015). A genetic algorithm for permutation flow shop scheduling under make to stock production system. *Computers & Industrial Engineering, 90*, 12–24. https://doi.org/10.1016/j.cie.2015.08.006

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications, 42*(6), 3234–3241. https://doi.org/10.1016/j.eswa.2014.12.003

Rout, A. K., Dash, P. K., Dash, R., & Bisoi, R. (2015). Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach. *Journal of King Saud University - Computer and Information Sciences, 29*(4), 536–552. https://doi.org/10.1016/j.jksuci.2015.06.002

Selvin, S., Vinayakumar, R., Gopalakrishman, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In International Conference on Advances in Computing, Communications and Informatics, 1643–1647.

Senthilkumar, R., Romolo, A., Fiamma, V., Arena, F., & Murali, K. (2015). Analysis of wave groups in crossing seas using Hilbert Huang transformation. *Procedia Engineering, 116*, 1042–1049. https://doi.org/10.1016/j.proeng.2015.08.341

Teixeira, L. A., & Adriano Lorena Inácio de Oliveira, (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. Expert Systems with Applications, 37(10), 6885–6890.

Torres, M. E., Colominas, M. A., Schlotthauer, G., & Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. *Proceedings of IEEE International Conference on Acoustics*.

Wang, J., & Wang, J. (2017). Forecasting stochastic neural network based on financial empirical mode decomposition. *Neural Networks, 90*, 8–20. https://doi.org/10.1016/j.neunet.2017.03.004

Wang, T., Zhang, M., Yu, Q., & Zhang, H. (2012). Comparing the applications of EMD and EEMD on time–frequency analysis of seismic signal. *Journal of Applied Geophysics, 83*, 29–34. https://doi.org/10.1016/j.jappgeo.2012.05.002

Wei, L.-Y. (2016). A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting. *Applied Soft Computing, 42*, 368–376. https://doi.org/10.1016/j.asoc.2016.01.027

Wu, Y.-X., Wu, Q.-B., & Zhu, J.-Q. (2019). Improved EEMD-based crude oil price forecasting using LSTM networks. *Physica A: Statistical Mechanics and its Applications, 516*, 114–124. https://doi.org/10.1016/j.physa:2018.09.120

Wu, Z. H., & Huang, N. E. (2009). Ensemble empirical mode decomposition: A noise-assisted data analysis method. Advances in Adaptive Data Analysis, 01(01), 1–41.

Xi, L., Muzhou, H., Lee, M. H., Li, J., Wei, D., Hai, H., & Wu, Y. (2014). A new constructive neural network method for noise processing and its application on stock market prediction. *Applied Soft Computing, 15*, 57–66. https://doi.org/10.1016/j.asoc.2013.10.013

Yang, Z., Yang, L., Qi, D., & Suen, C. Y. (2006). An EMD-based recognition method for Chinese fonts and styles. *Pattern Recognition Letters, 27*(14), 1692–1701. https://doi.org/10.1016/j.patrec.2006.03.002

Yeh, J. R., Shieh, J. S., & Huang, N. E. (2010). Complementary ensemble empirical mode decomposition: a novel noise enhanced data analysis method. Advances in Adaptive Data Analysis, 2(2), 135–156.

Yu, L., Wang, S., & Lai, K. K. (2009). A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Applied Soft Computing, 9*(2), 563–574. https://doi.org/10.1016/j.asoc.2008.08.001

Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics, 30*(5), 2623–2635. https://doi.org/10.1016/j.eneco.2008.05.003

Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology, 561*, 918–929. https://doi.org/10.1016/j.jhydrol.2018.04.065

Zhang, M., Wang, T., Tang, T., Benbouzid, M., & Diallo, D. (2017). An imbalance fault detection method based on data normalization and EMD for marine current turbines. *ISA Transactions, 68*, 302–312. https://doi.org/10.1016/j.isatra.2017.02.011

Zhang, Y. Y., & Shen, W. (2009). Stock yield forecast based on LS-SVM in Bayesian inference. *International Conference on Future Computer & Communication IEEE*, 8–11.

Zhou, Z., Lin, L., & Li, S. (2018). International stock market contagion: A CEEMDAN wavelet analysis. *Economic Modelling, 72*, 333–352. https://doi.org/10.1016/j.econmod.2018.02.010