



# A novel general-purpose hybrid model for time series forecasting

Yun Yang<sup>1</sup> · ChongJun Fan<sup>1</sup> · HongLin Xiong<sup>1</sup>

Accepted: 17 April 2021 / Published online: 5 June 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Realizing the accurate prediction of data flow is an important and challenging problem in industrial automation. However, due to the diversity of data types, it is difficult for traditional time series prediction models to have good prediction effects on different types of data. To improve the versatility and accuracy of the model, this paper proposes a novel hybrid time-series prediction model based on recursive empirical mode decomposition (REMD) and long short-term memory (LSTM). In REMD-LSTM, we first propose a new REMD to overcome the marginal effects and mode confusion problems in traditional decomposition methods. Then use REMD to decompose the data stream into multiple intrinsic modal functions (IMF). After that, LSTM is used to predict each IMF subsequence separately and obtain the corresponding prediction results. Finally, the true prediction value of the input data is obtained by accumulating the prediction results of all IMF subsequences. The final experimental results show that the prediction accuracy of our proposed model is improved by more than 20% compared with the LSTM algorithm. In addition, the model has the highest prediction accuracy on all different types of data sets. This fully shows the model proposed in this paper has a greater advantage in prediction accuracy and versatility than the state-of-the-art models. The data used in the experiment can be downloaded from this website: <https://github.com/Yang-Yun726/REMD-LSTM>.

**Keywords** Data decomposition · REMD-LSTM · Time series prediction

## 1 Introduction

**Motivation** Time series prediction is an important technology to promote industrial intelligence. It plays a vital role in many wide-ranging applications, such as information system operation adjustment, Artificial intelligence for IT operations, stock investment, disease prediction, equipment abnormality warning, etc. Time series prediction helps to grasp the future development trend of data in real-time and make optimal decisions in advance. For example, if we can accurately predict the development trend of a certain disease and adjust the corresponding prevention plan in real-time based on the data, then we can better suppress or prevent the disease.

There are many different applications in the real world, and corresponding data streams will be generated accordingly. These different data streams often have different distribution characteristics and statistical information, and to predict these

data often means building different prediction models. To reduce resource consumption, it is necessary to build a more versatile predictive model. Empirical mode decomposition (EMD) uses mathematical methods to automatically decompose data into multiple more gentle Intrinsic Mode Functions (IMF), without considering the intrinsic information of the data. Then combined with the corresponding prediction model can get more accurate prediction results. This hybrid model can greatly reduce the versatility of the algorithm and greatly reduce the consumption of industrial resources. However, EMD itself has major flaws, so we consider improving EMD.

**Challenges** Time series data streams refer to a sequence of data generated in chronological order, in most real-world scenarios, it is a typical type of non-linear and non-stationary data [1, 2]. Therefore, if we want to obtain higher prediction accuracy, the following issues must be considered: (1) Since the data in most application scenarios has non-linear and unstable characteristics (such as stock prices and traffic flow data, etc.), this requires that the predictive model we build can be well applied to such data. (2) According to related research, time series data in real scenes is usually the result of a series of hidden features. Therefore, if we can decompose the time series data into multiple flat sub-sequences and then make

---

✉ ChongJun Fan  
fan.chongjun@163.com

<sup>1</sup> University of Shanghai for Science and Technology, Shanghai, China

predictions, it may improve the final prediction accuracy. (3) Due to the diverse types of time series data in real scenes, it is necessary to design a more general analysis and prediction model to be able to predict multiple time-series data streams. (4) The historical data of the time series data stream contains a lot of useful information. If the model can make full use of this information, it will help improve the prediction accuracy. (5) Streaming data is a kind of data with large amount of data and strong real-time performance. Among them, the dynamic data flow is not independent and distributed, which will cause the phenomenon of concept drift. This requires our model to have the function of using historical data.

In addition to the above difficulties, the EMD algorithm itself also has two flaws: 1) Marginal effect; 2) Mode mixing. These two flaws will cause the IMF components of the EMD decomposition to be inaccurate and information overlap.

**Proposed approach** To fully solve the above limitations and further improve the prediction accuracy of the time series data streams, this paper proposes a hybrid prediction model suitable for the characteristics of time series data stream, called REMD-LSTM. According to the architecture of the model, this paper divides REMD-LSTM into three parts: 1) REMD data decomposition; 2) LSTM data prediction; 3) Reconstruct prediction results.

In the data decomposition stage of REMD-LSTM, the traditional EMD algorithm has marginal effects and pattern confusion, which will reduce the accuracy of data decomposition. Therefore, we propose an improved decomposition algorithm, namely REMD. The model is divided into three steps. First, in each decomposition process, the last data of the data set is added as an extreme point to the extreme point set to weaken the marginal effect caused by the edge points. The second step is to add a new pair of positive white noise data and negative white noise data in each data decomposition process, and perform a double recursive operation on the EMD decomposition process to further reduce the marginal effect and eliminate the influence of mode confusion. Finally, the constructed REMD is used to decompose the data into multiple IMF (mode function) units. Subsequent comparative experiment results show that this improvement can greatly reduce marginal effects and mode confusion.

In the data prediction stage of REMD-LSTM, due to the unique network architecture of LSTM, it can make good use of historical data, which can greatly improve the overall prediction effect. Therefore, after using REMD to decompose the input data sequence into multiple IMF sequences, then use LSTM to predict each IMF and obtain the corresponding predicted value. Finally, in the result reconstruction stage of REMD-LSTM, the final prediction result is obtained by reconstructing multiple IMF prediction values.

**Contributions** To verify the prediction effect of REMD-LSTM, this paper selected four different types of real data in the experimental part. We then select five algorithms as comparison models. Finally, the experimental results verify the efficiency and accuracy of the REMD-LSTM model on the real data set. Overall, the main contributions of this article are as follows:

- The REMD algorithm proposed in this paper can solve the problem of marginal effect and mode confusion in EMD.
- Decomposing time series data into multiple components through REMD can reveal the specific influence of hidden variables in time series data to a certain extent.
- The algorithm proposed in this paper can predict data without any prior information and parameter assumptions (Such as the distribution of data and other statistical characteristics, etc.).
- The model proposed in this paper can be applied to different types of time series data streams.
- All experiments in this paper are carried out on the basis of real data sets. The experimental results fully show that the algorithm in this paper can not only solve the limitation problems mentioned above, but also has better detection accuracy than the current popular methods.

The structure of this paper is organized as follows. We briefly introduce the related work and the basic concepts involved in detail in the Section 2. In Section 3, we describe in detail the new model proposed in this article. In Section 4, we designed simulation experiments and gave specific prediction results. Finally, the conclusions are drawn.

## 2 Related work and preliminaries

### 2.1 Related research

Due to the crucial role of time series prediction in industrial intelligence, a lot of research has been done in academia. These algorithms can be roughly divided into the following three categories.

**Statistical model** In addition, there are many statistical learning methods used to predict time series data. For example, Wang et al. optimized the gray differential equation through the combination of background value optimization and initial term optimization, and then conducted prediction research on input data of unbiased exponential distribution [3]. Matamoros and others used the Autoregressive Integrated Moving Average (ARIMA) model to predict COVID-19 and obtained real-time prediction results of possible infections in various regions [4]. Amin and Wang used ARIMA to predict

electricity data and annual runoff data respectively, and got very good forecasting results [5, 6]. Besides, the autoregressive model (AR) [7, 8] and autoregressive conditional heteroskedasticity model (GARCH) [9, 10] are also widely used predictive models. However, this type of method is more suitable for static data sets, and when applied to dynamic data sets, the effect is often not satisfactory. In addition, other models built with a variety of statistical knowledge are also constantly being produced, such as Facebook's open-source Prophet model [11]. The Prophet algorithm can handle the situation of some outliers in the time series, and can also handle the situation of some missing values. It can also predict the future trend of the time series almost automatically. But for some time series that are cyclical or not very trending, Prophet may not be appropriate.

**Deep learning** In addition to the limitations of application scenarios, since most time-series data streams are non-linear and non-stationary data, so it is quite difficult to predict data fluctuation reliably and accurately. Moreover, since the time-series prediction in the data stream is not only related to the data at the current time but also to the data at an earlier time, the information carried by the data at an earlier time will be lost if only the data at the latest time is applied. Unlike traditional models, recurrent neural networks (RNN) establishes connections between hidden units, which makes this network enable to keep the memory of recent events [12]. RNN deals with the before-after associated data by the memory characteristics. It is very suitable for the prediction of time series. As an improved model of RNN, LSTM is widely used in natural language recognition, time series prediction, and other fields [13]. The information is selectively filtered through a 'gate' structure of LSTM, and more useful information is extracted from historical data in training. For example, on the basis of LSTM, Chang et al. combined wavelet transform (WT) and adaptive moment estimation (Adam) to construct a new hybrid model WT-Adam-LSTM to predict electricity prices, which can better improve the prediction accuracy [14]. Wang et al. combined LSTM, RNN and Time Correlation Modification (TCM) to construct a new prediction model to predict the power generation and load of Dianwang [15]. To improve the accuracy of system prediction under extreme conditions, Luo et al. designed a method based on dual-tree complex wavelet enhanced convolutional long and short-term memory neural network (DTCWT-CLSTM). This method can be effectively applied to the damage prediction of automobile suspension components under actual working conditions [16]. In addition, the LSTM model has many applications in other fields [17–19]. Although this type of prediction model based on deep learning can achieve high prediction accuracy, the overall time consumption is relatively large due to its own complex network architecture.

**Machine learning** For example, CAO uses support vector machine (SVM) to predict financial data [20], Qiu uses EMD to decompose the data first and then predicts [21]. To solve the defect that the back-propagation neural network is easy to fall into the local minimum, Yang combined with adaptive differential evolution to construct a new ADE-BPNN to improve the prediction accuracy of the model [22]. Zhang introduced Principal Component Analysis (PCA) on the basis of ADE-BPNN to reduce the dimensionality of the data, and constructed a new PCA-ADE-BPNN, so that the model can be better applied to high-dimensional data sets. [23]. In addition, WNN is also a widely used predictive model. For example, Yang proposed an improved genetic algorithm (GA) and wavelet neural network (WNN) based on a clustering search strategy to improve the convergence speed of WNN [24]. Based on WNN, Zhang et al. proposed a complex hybrid model IGA-WNN to improve the prediction accuracy of short-term traffic flow [25]. On the basis of Extreme Gradient Boosting (XGBoost), Zhou et al. combined with the feature selection method of association rules to predict urban fire accidents, which can effectively improve the form of public safety [26]. This type of method has strong versatility, requires less resources, and the overall prediction accuracy is relatively high. But the theoretical derivation of such models is often too complicated.

The time series prediction model based on EMD decomposition has attracted much attention [27–29]. For example, Agana used EMD-based deep belief network to predict port cargo throughput [30]. Wang uses a time series forecasting model based on EEMD decomposition to predict rainfall [31]. Zhang uses a time series forecasting model based on VMD decomposition to predict wind speed [32]. These methods can better solve problems in special scenarios, but cannot comprehensively solve the problem of marginal effects and mode confusion based on EMD, so these models do not have good generality [33, 34]. To this end, we propose a new data decomposition model REMD combined with LSTM to achieve accurate prediction of the data stream.

## 2.2 EMD

Empirical Mode Decomposition (EMD) is a typical algorithm for decomposing data according to its time scale characteristics, without any basis function being preset. The core of the EMD algorithm is empirical mode decomposition, which can decompose complex signals into a finite number of Intrinsic Mode Functions (IMF). Each IMF component contains local characteristic signals of different time scales of the original signal. Since the EMD decomposition model is based on the local characteristics of the time scale of the signal sequence, the algorithm has stronger adaptability [35–37].

For a given time series data  $X = \{x_1, x_2, \dots, x_n\}$ , the specific steps of data decomposition using EMD algorithm are:

- Find all extreme points of time series data  $X$ , including minimum and maximum.
- Use the cubic spline interpolation to fit the envelopes  $emax(t)$  and  $emin(t)$  of the upper and lower extreme points respectively, and find the average value  $m(t)$  of the upper and lower envelopes. Subtract it from  $X$ ,  
$$h(t) = X - m(t).$$
- According to the set condition criterion, judge whether  $h(t)$  is IMF;
- If  $h(t)$  is not an IMF, replace  $X$  with  $h(t)$ , and repeat steps (2), (3) until  $h(t)$  meets the judgment condition, then  $h(t)$  is the IMF that needs to be extracted;
- Every time an IMF sequence is obtained, it is deleted from the input data. Then repeat the above four steps until the last remaining part of the signal  $r_n$  is just a monotonic sequence or a constant sequence.

In this way, after decomposition by the EMD method, the original signal  $x(t)$  is decomposed into  $N$  different IMFs and the residue parts:

$$X = \sum_{i=1}^N C_i(t) + r_n(t) \tag{1}$$

Where  $rn(t)$  is the residue, representing the trend of the time series.

The purpose of EMD is to continuously extract the various scale components that make up the original signal from high frequency to low frequency, and the order of the characteristic mode functions obtained by decomposition is arranged in order of frequency from high to low.

### 2.3 LSTM

Recurrent neural network (RNN) is currently a popular neural network model for time series prediction. But RNN has the problems of gradient disappearance and gradient explosion, and the LSTM proposed by Graves [3] successfully solves these two problems. The LSTM unit is composed of a storage unit, and the stored information is updated by three special units: input gate, forget gate, and output gate [20, 38]. The structure of the LSTM cell is shown in Fig. 1.

The specific calculation steps of LSTM are as follows,

- Calculation the forget gate: The hidden state  $h_{t-1}$  of the previous sequence and the data  $x_t$  of this current sequence, through an activation function, usually sigmoid, get the output  $f_i$  of the forget gate. Since the output  $f_i$  of sigmoid is between  $[0,1]$ , the output  $f_i$  represents the

probability of forgetting the hidden cell state of the previous layer. Then the forget gate output is as follows,

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{2}$$

Where  $W_f$ ,  $U_f$  and  $b_f$  are coefficients and bias of the linear relationship above.  $\sigma$  is the sigmoid function.

- Calculate the input gate: The input gate consists of two parts, the first part use  $h_{t-1}$  and  $x_t$  to calculate which information to update through the sigmoid function, expressed by  $i_t$ . The second part use  $h_{t-1}$  and  $x_t$  to obtain new candidate cell information  $a_t$  through the  $\tanh$  function. The role of the two is to update the cell state. The mathematical expression is

$$i_y = \sigma(W_i h_{t-1}) + U_i x_t + b_i; a_t = \tanh(W_a x_t + b_a).$$

Where  $W_i$ ,  $U_i$ ,  $b_i$ ,  $W_a$ ,  $U_a$  and  $b_a$  are coefficients and bias of the linear relationship above.

- Cell state update: The results of the previous forget gate and input gate will act on the cell state  $C_t$ . Next, let's calculate the cell update, that is from the cell state  $C_{t-1}$  to  $C_t$

$$C_t = C_{t-1} * f_t + i_t * a_t.$$

- Calculation output gate: The update of the hidden state  $h_t$  consists of two parts. The first part is  $O_t$ , which is obtained from the hidden state  $h_{t-1}$  of the previous sequence and the data  $x_t$  of this current sequence through the activation function sigmoid, to control the output of the current cell, The second part consists of hidden state  $C_t$  and  $\tanh$  activation function, as follows

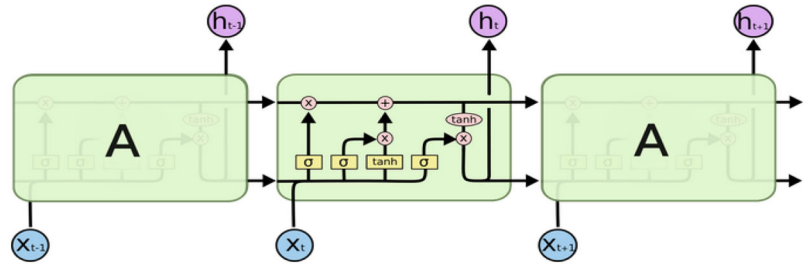
$$O_t = \sigma(W_o * h_{t-1} + U_o * x_t + b_o); h_t = O_t * \tanh(C_t)$$

LSTM replaces the hidden nodes in the RNN chain structure with specific memory units and realizes the ability of time-series information retention and long-term memory.

### 3 Proposed method

To solve the three problems raised in the introduction, we divide REMD-LSTM into two parts: 1) use REMD to decompose the input data; 2) use LSTM to predict and get the final result. Algorithm 1 details the REMD-LSTM.

**Fig. 1** LSTM memory cell structure



- 1) In REMD, set the threshold of the number of decompositions, and then decompose the input data into a fixed number of IMF subsets.
- 2) Each IMF subset uses a separate LSTM function to predict and get the prediction result.
- 3) Combine the prediction results of all IMF subsets to obtain the final prediction result of the input data set.

**Algorithm 1** REMD-LSTM.

**Input:** data stream  $X = \{x_1, x_2, \dots, x_i, \dots\}$ ,  $x_i \in R$ , epoch  $I$ , number of iterations  $K$ , error parameters  $\sigma$ , cycle index  $N$ , Number of decompositions  $m$ , White noise data  $W$ .

**Output:** Predicted result  $O$ .

```

1:  $X \leftarrow x_1, x_2, \dots, x_i, \dots, x_i \in R$  // Input data
2:  $O \leftarrow \{\}$  // Output data
3: for each  $i \in N$  do
4:    $H_i^+ \leftarrow x + W_i^+$ ;  $H_i^- \leftarrow x + W_i^-$ 
5:   for  $j \in I$  do
6:      $H_{ij}^+ \leftarrow REMD(H_i^+)$ ;  $H_{ij}^- \leftarrow REMD(H_i^-)$ 
7:   end for
8: end for
9: for  $j \in m$  do
10:   $O \leftarrow \sum(LSTM(IMF_j))$ 
11: end for

```

To further elaborate the REMD-LSTM model, we divide it into two parts: 1) REMD model; 2) REMD-LSTM combined model. The following algorithm 1 gives the pseudo-code of the REMD-LSTM model.

**3.1 REMD model**

The EMD algorithm introduced in Section 2.2 has two major problems:

- Marginal effect: Because the endpoints of the sequence are often not extreme points, the upper and lower envelope functions constructed will diverge at the end. This divergence will continue to iterate inward with the continuous decomposition process, which will affect the entire decomposition accuracy.
- Mode mixing: It is caused by incomplete data decomposition that signals of disparate scales and frequencies are

not completely separated, and signals of multiple scales and frequencies are often mixed in one IMF. When multiple modes are mixed in several IMFs, the physical meaning of the EMD algorithm is lost.

To fully solve the above two problems, we propose a new decomposition model REMD. The specific decomposition steps of REMD are as follows.

- Given input data  $X = \{x_1, x_2, \dots, x_n\}$ , add white noise sequences  $w_i^+, w_i^-, i = 1, 2, \dots, m$  with a mean value of 0 and opposite numbers to  $X$  respectively. Then construct new data as follows:

$$H_i^+ = X + w_i^+; H_i^- = X + w_i^-.$$

where  $i$  is the number of white noise added.

- This step mainly solves the problem of marginal effects. First, find all the extreme points of  $H_i^+$  and  $H_i^-$ , then add the marginal points of each sequence to the extreme point set as maximum or minimum points. Figure 2 shows an exploded view of this improved EMD and EMD.

It can be seen from Fig. 2 that the data at the end of each IMF component after the improvement is smoother than the IMF component of EMD, which shows that the improvement helps to solve the marginal effect of the EMD algorithm.

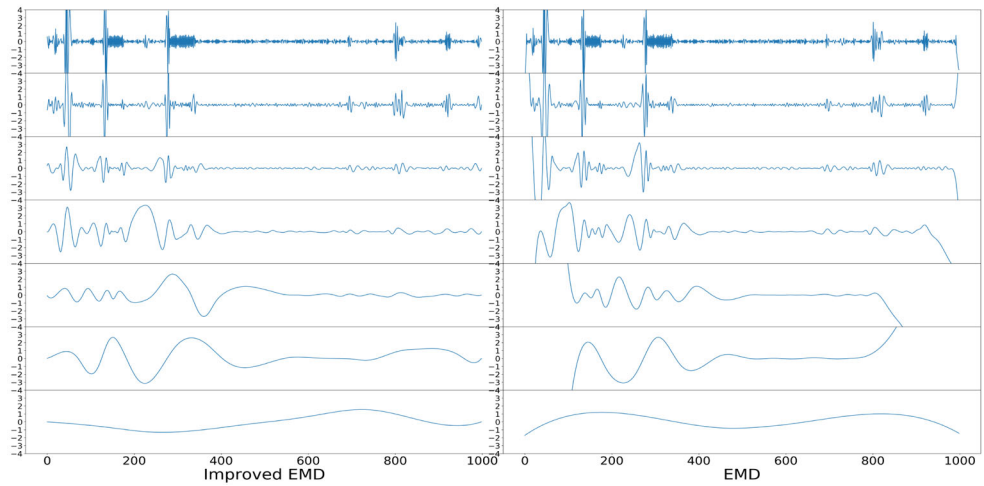
- Perform EMD decomposition on the  $H_i^+$  and  $H_i^-$  sequences proposed in the previous step, then limit the number of decompositions to  $m$  (The number of decompositions of the same type of data is the same), then the IMF of the first stage can be obtained as:

$$IMF_i^+ = \{imf_{i1}^+, imf_{i2}^+, \dots, imf_{ij}^+\}, j = 1, \dots, m;$$

$$IMF_i^- = \{imf_{i1}^-, imf_{i2}^-, \dots, imf_{ij}^-\}, j = 1, \dots, m;$$

- Cycle steps 1, 2, and 3 for  $N$  times, and add a new white noise sequence to each cycle. Finally, a series of IMF components can be obtained as follows:

**Fig. 2** Improved EMD (left) and EMD (right)



$$IMF_i^+ = \{imf_{ij}^+\}, i = 1, \dots, N; j = 1, \dots, m$$

$$IMF_i^- = \{imf_{ij}^-\}, i = 1, \dots, N; j = 1, \dots, m;$$

- After averaging the  $IMF_i^+$  and  $IMF_i^-$  series in step 4, the final IMF component of the original data  $X$  is:

$$IMF = \{imf_j\}, j = 1, \dots, m;$$

Where

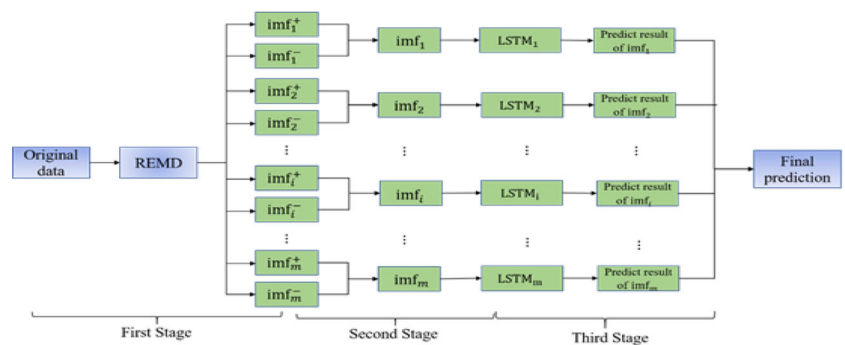
$$imf_j = \sum_{i=1}^N \left( (imf_{ij}^+ + imf_{ij}^-) / 2N \right).$$

### 3.2 REMD-LSTM algorithm

#### 3.2.1 Algorithm flow

The REMD-LSTM can be obtained by combining the REMD and LSTM model as described above. Overall, REMD-LSTM can be divided into three steps: 1) REMD data decomposition; 2) LSTM prediction; 3) Combines the prediction results of

**Fig. 3** REMD-LSTM model structure diagram



each IMF to obtain the final prediction value. Figure 3 shows the overall architecture of the REMD-LSTM model.

In Section 3.1, we can use the REMD to decompose the input data  $X = \{x_1, x_2, \dots, x_n\}$ , into  $IMF = \{imf_j\}, j = 1, \dots, m$ . Then, we use the LSTM algorithm to predict each  $imf_j, j = 1, \dots, m$  and get the predicted value of each IMF as follows.

$$L_j = LSTM(imf_j), j = 1, \dots, m \tag{3}$$

After obtaining the predicted value  $L_j$  of each  $imf_j$  component, the final predicted value of data  $X$  is expressed as:

$$Fx = \sum_{j=1}^m (L_j) \tag{4}$$

Where  $L_j$  represents the predicted value of  $imf_j$ , because  $X = \sum(imf_j)$ , the predicted value of  $X$  is shown in the above formula.

#### 3.2.2 Algorithm complexity

Our hybrid model REMD-LSTM is composed of REMD and LSTM. The complexity of the traditional EMD model is  $O(n^2)$ , where  $n$  is the amount of input data. The REMD we

proposed has carried out multiple loop operations and additions and deletions of other extreme points on the basis of EMD. So the complexity of REMD is  $O(k * n^2)$ , where  $k$  is the number of cycles. The complexity of LSTM is  $4(m * n + m^2 + m)$ , where  $n$  is input size and  $m$  is hidden size. So the complexity of the entire algorithm is  $O(k * n^2 + m * n + m^2)$ .

## 4 Experiment results

### 4.1 Basic setting

**Data set** To verify that the REMD-LSTM model has a good prediction effect on different types of time series data streams, we selected four types of data: stock price data, restaurant sales data, commodity sales data, and satisfaction rate data. Figure 4 shows the diagram of the four types of time series data stream. Table 1 gives some statistics about the four sets of data.

- Stock: The stock price data is selected from the daily closing price of a stock on the Shanghai Stock Exchange A-share market. The data length is 2000.
- Restaurant Sales: The restaurant’s sales data comes from the statistics of the Kaggle database. It records the restaurant’s sales data for the past five years. Since the data record in the initial stage is incomplete, the data will have certain errors.
- Commodity: Commodity data is downloaded from Kaggle, which records the sales volume of shampoo in a mall in California in the past five years.
- Satisfaction rete: Satisfaction reate is the customer satisfaction data surveyed by a shopping mall in Shanghai. The specific satisfaction value is obtained through the results of the statistical questionnaire.

It can be seen from Fig. 4 that the four sets of data have obvious non-linear and non-stationary characteristics, which is consistent with the types of problems this article hopes to solve.

**Evaluation index** In order to verify the accuracy of the prediction results, this article selects the MSE (Mean Square Error),

RMSE (Root Mean Square Error), MAE (Mean Absolute Error) three common indicators as the evaluation criteria.

The three indicators are calculated as follows

$$MSE = \frac{\sum_{i=1}^N (y_i - \hat{y}_1)^2}{N}$$

$$RMSE = \frac{1}{N} \sqrt{\sum_{i=1}^N (y_i - \hat{y}_1)^2}$$

$$MAE = \frac{\sum_{i=1}^N (y_i - \hat{y}_1)}{N}$$

Where  $y_i$  is the prediction result and  $\hat{y}_i$  is the real data.

**Parameter setting** REMD parameter setting: 1) Limit the number of IMF decompositions to 8(In the course of the experiment, it was found that when the number of decompositions reaches 8 times, the final subsequence already has strong linear characteristics); 2) Set the number of outer loops to 10 (during the experiment, we found that the number is too large and the accuracy improvement is not obvious).

LSTM parameter setting: batch size is set to 1, epoche is set to 100, and the number of neurons is 40.

SVR parameter setting: The kernel is *rbf*; degree is 3, gamma is 0.2 and C = 100. ARIMA parameter setting:  $p$  is 2,  $q$  is 2 and  $d$  is 0.

**Experiment environment** The experiments in this article are all done on HP computers. The specific computer configuration is: Processor Intel(R)Core(TM) i7-6700 CPU @ 3.4GHz; RAM 16GB; RAM 1 T; Windows10 64-bit operating system.

**Data set division** The first half of the data is the training set, and the second half is the test set.

### 4.2 Analysis of experiment results

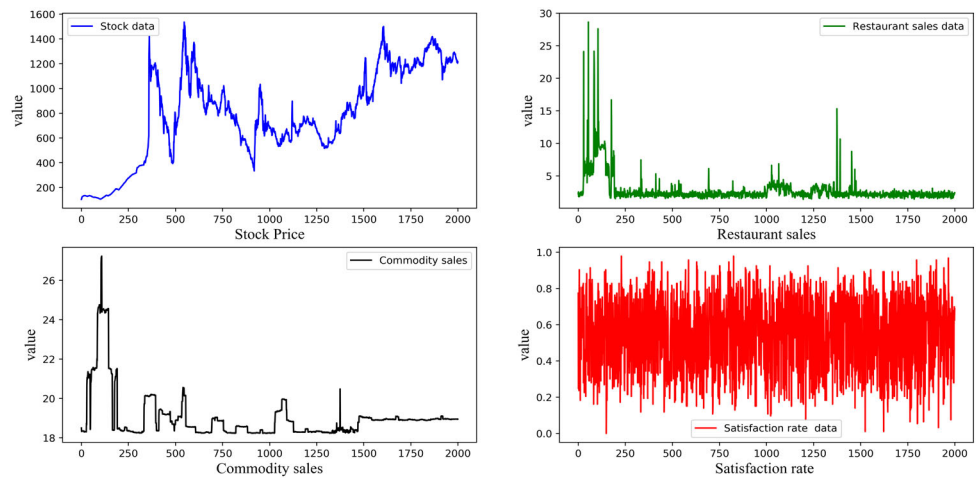
#### 4.2.1 Time comparison

Since eight independent LSTM models are used in the REMD-LSTM model, the overall running time will increase. To solve this problem, we designed a set of parallel solutions: 1) Decompose the data into eight sets of IMF data on the first

**Table 1** Statistics of four types of time series data

Data	Max	Min	Mean	Var	Std	Median
Stock	1536.8	104.39	788.3333	134,557.5	366.82	772.915
Restaurant Sales	28.641	1.3916	2.798125	4.559266	2.1352	2.241667
Commodity	27.229	18.219	19.01126	1.505774	1.2271	18.83379
Satisfaction Rate	0.9784	0	0.546747	0.040573	0.2014	0.58065

**Fig. 4** The time-series diagram of four types of data

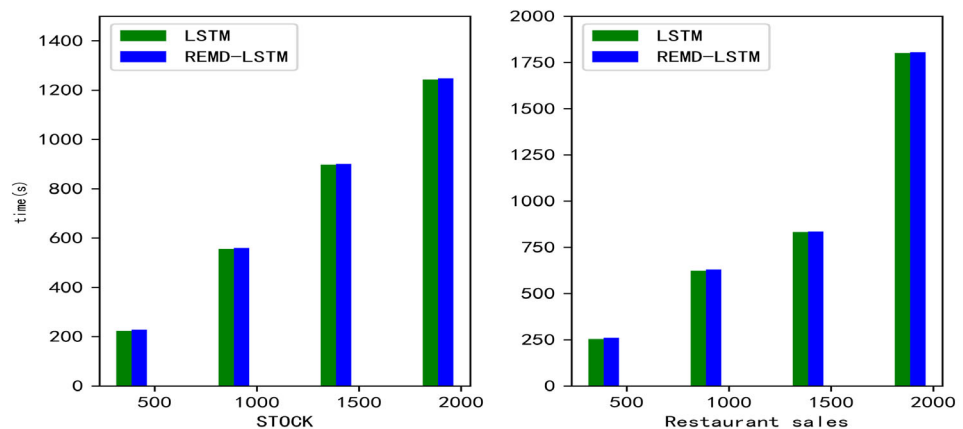


host; 2) Transfer the eight sets of IMF data to four different hosts (each host is assigned two sets of IMF data), and then on each host Run LSTM to make predictions and get the corresponding prediction results; 3) Return the prediction results of the four hosts to the first host to obtain the final prediction results.

To verify the feasibility of this program, we designed the following experimental program: 1) Select 500, 1000, 1500, and 2000 data from stock and restaurant sales respectively as Input data set; 2) Set the same size of RMSE for four data sets with different sizes, and then train the model; 3) Finally, comprehensively compare the time consumption of REMD-LSTM and LSTM when they reach the same RMSE (Note: To control the experimental time, the RMSE under each data set here is: 0.6). Figure 5 shows the specific comparison effect.

From the results in Fig. 5, it can be seen that the time consumed by REMDLSTM after parallel operations increases less than 1% of the time consumed by single LSTM. This conclusion fully validates the defects of deep learning we proposed in related work. At the same time, this conclusion fully shows that REMDLSTM can overcome the problem of high time consumption through the parallel operation.

**Fig. 5** Comparison of REMD-LSTM and LSTM time consumption



### 4.2.2 Comparison of prediction results

To verify that the REMD-LSTM model proposed in this paper has a controversial improvement in prediction accuracy compared to the LSTM model, Fig. 6 shows the comparison between the prediction results of REMD-LSTM and LSTM on the four types of time series data sets: stock, restaurant sales, commodity, and satisfaction rate.

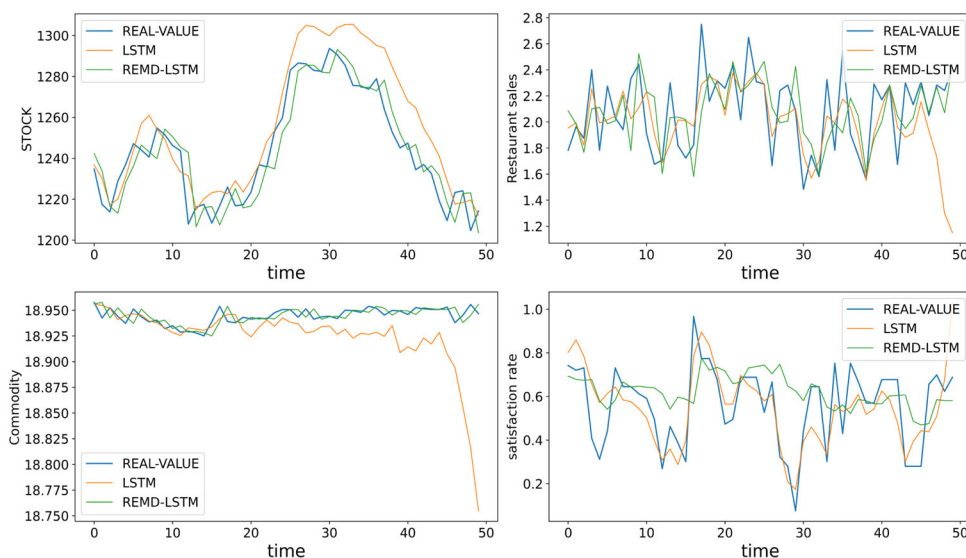
From the prediction results in Fig. 6, we can find that the overall prediction result of REMD-LSTM is closer to the real data. Therefore, it can be considered that the prediction effect of REMD-LSTM is better than LSTM.

Next, in order to verify that the REMD-LEMD model proposed in this paper has higher prediction accuracy than the commonly used prediction models, we select the ARIMA, SVM, LSTM, Prophet, and XGboost models for comparison, and predict the four data sets respectively. Figure 7 shows the prediction results of REMD-LSTM and other five models on the four data sets.

It can be seen from the time series diagram of the prediction results in Fig. 7 that compared with the other five prediction models, the trend of the REMDLEMD prediction results fits



**Fig. 6** The prediction results of REMD-LSTM and LSTM



the original data more closely, the error with the original data is smaller, and the overall prediction effect is the best. It fully shows that the REMD-LEMD model has a great advantage over the current algorithm in terms of prediction accuracy.

To further intuitively quantify the prediction effect of each model, Table 2 shows the MSE, RMSE, and MAE of the prediction results of the six algorithms on the four data sets.

From the results in Table 2, we can find that the REMD-LSTM is superior to other algorithms in the calculation results of these three errors for four different types of data. To intuitively experience the prediction effect of REMD-LSTM, Fig. 8 plots the data in Table 2 into a histogram.

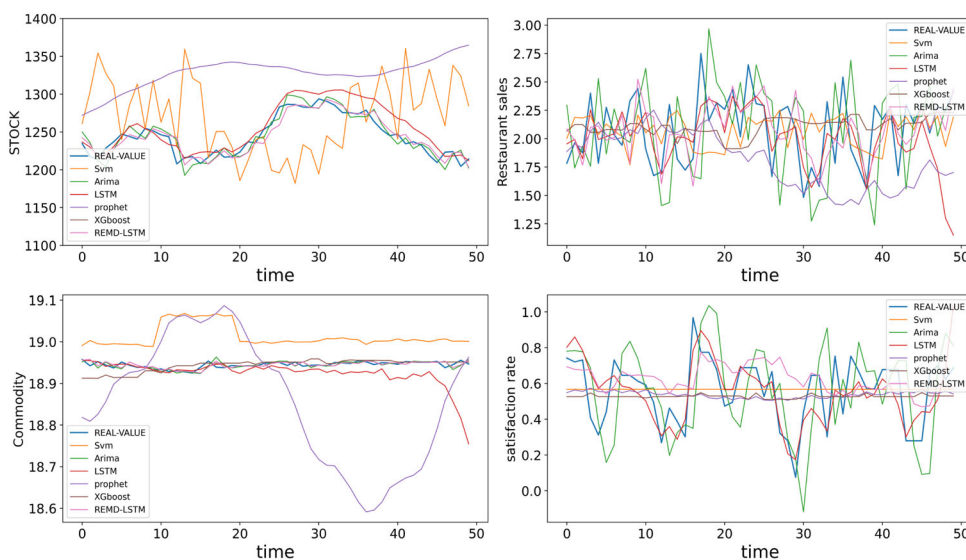
Combining Table 2 and Fig. 8, it can be seen that the REMD-LSTM model has the highest prediction accuracy

among the four groups of models. This result fully shows that the REMD-LSTM model is a high-precision prediction model and can be applied to a variety of application scenarios.

### 4.2.3 Friedman test

In the above content, we have judged the performance of the REMD-LSTM algorithm through specific experiments. In this section, we will use the Friedman test to verify our proposed algorithm and the comparison algorithm, in order to judge the pros and cons of this algorithm. We take the MSE indicator in Table 3 as an example to calculate the Friedman values of the six algorithms on the four data sets.

**Fig. 7** The prediction results of the four algorithms



**Table 2** The prediction error results of four algorithms

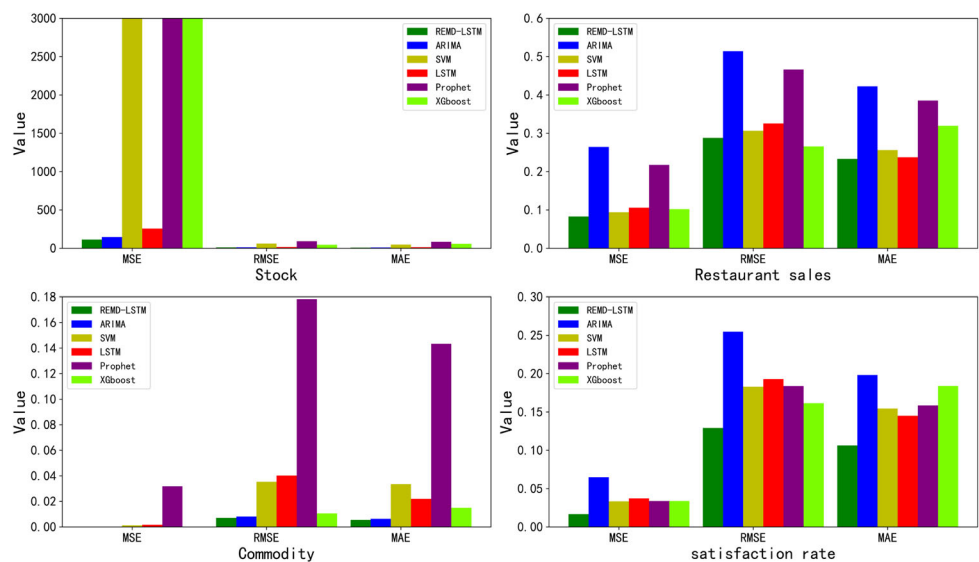
Algorithm	Index	Stock	Restaurant	Commodity	Satisfaction
MSE	REMD-LSTM	113.04	0.08	5.03553E-05	0.016
	ARIMA	146.32	0.264	6.73191E-05	0.064
	SVM	3812.9	0.094	0.00124	0.033
	LSTM	256.71	0.1059	0.00161	0.037
	Prophet	8212.28	0.21742	0.03179	0.03378
	XGboost	3280.33	0.10211	0.000222	0.03381
RMSE	REMD-LSTM	10.63	0.28	0.007096145	0.129
	ARIMA	12.096	0.514	0.008204823	0.254
	SVM	61.749	0.3066	0.03532	0.182
	LSTM	16.022	0.32543	0.04023	0.192
	Prophet	16.022	0.32543	0.04023	0.192
	XGboost	45.5547	0.26565	0.01056	0.16152
MAE	REMD-LSTM	8.164	0.23	0.005454375	0.106
	ARIMA	10.074	0.422	0.006328749	0.198
	SVM	48.457	0.256	0.03354	0.154
	LSTM	13.072	0.23741	0.02196	0.145
	Prophet	83.7201	0.3853	0.14336	0.15852
	XGboost	57.2741	0.31954	0.01492	0.18388

From the results in Table 3, the Friedman mean values of the six algorithms on the four data sets are all different. Among them, the REMD-LSTM proposed in this article is the smallest, followed by SVM, XGboost, ARIMA, LSTM, and Prophet (this article uses an ARIMA single-step prediction model, so its prediction accuracy is high). This conclusion once again clarifies from a statistical point of view that the model proposed in this paper has a greater advantage over these comparison algorithms.

#### 4.2.4 Impact of parameter

Five parameters are involved in algorithm 1, but the three parameters involved in LSTM can be generated adaptively, and the number of decompositions is also automatically generated by REMD. Therefore, only the number of cycles has an unknown effect on the results. Based on this, this section will study the influence of the number of cycles in REMD on the final prediction result.

**Fig. 8** The prediction effect of four algorithms on four sets of data



**Table 3** Friedman test result

Algorithm Data	REMD-LSTM	ARIMA	SVM	LSTM	Prophet	XGboost
Stock	1	2	5	3	6	4
Restaurant	1	6	2	4	5	3
Commodity	1	2	4	5	6	3
Satisfaction	1	6	2	5	3	4
Average value	1	4	3.25	4.25	5	3.5

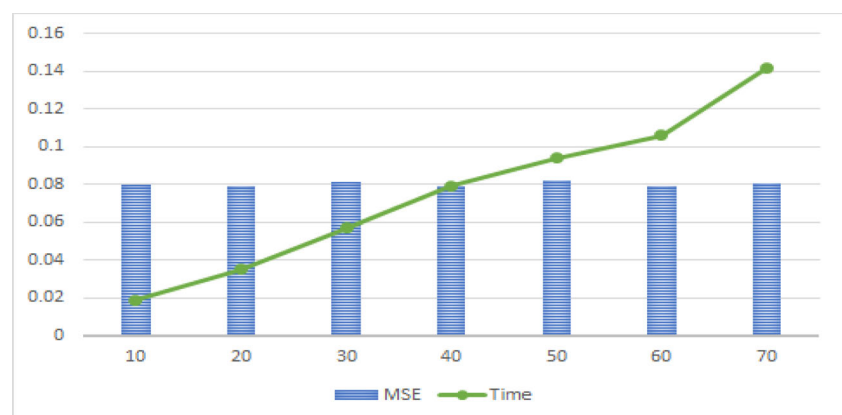
First, we use Restaurant data as the experimental data, MSE as the criterion, and set the number of cycles to [10,20,30,40,50,60,70]. Next, we calculate REMDLSTM under different cycles. Time-consuming and precision. The specific results are shown in Fig. 9.

From the results of Fig. 9, the prediction accuracy of the model will not be significantly improved with the increase of the number of cycles, but the corresponding prediction time consumption is continuously increasing. This conclusion fully shows that only a small number of cycles needs to be set, which can ensure the prediction accuracy and reduce time consumption.

Under the above experimental results show that REMD-LSTM has unique advantages in the following aspects:

- Accuracy: From the results of Table 2 and Fig. 8, we can see that the REMDLSTM model is superior to the common prediction models in terms of prediction accuracy.
- Versatility: From the results in Table 2, Table 2 and Fig. 8, it can be seen that the REMD-LSTM model proposed in this paper has the best predictive effect on the four sets of data. This result fully proves that REMD-LSTM has strong versatility and can be applied to multiple types of scenes.
- Adaptability: Although there are five parameter variables involved in REMDLSTM, four of them can be adaptive to get specific values, and the loop variable has a limited impact on the final result through experimental verification.

**Fig. 9** The influence of the number of cycles (real time = time\*100)



## 5 Conclusion

In this paper, we propose a novel hybrid time series prediction model based on REMD and LSTM. In this model, we first construct a new decomposition algorithm REMD. Next, use the algorithm to decompose the input data into multiple IMF components. Then use the LSTM model to predict each IMF component separately and get the prediction result of the corresponding component. Finally, the final prediction result of the input data is obtained by reconstructing the results of all components. In the experimental part, in order to verify that the model proposed in this paper has good prediction results, we selected 4 kinds of real-time series data and 3 commonly used prediction models. The final experimental results prove that the models proposed in this paper are superior to these models in terms of prediction accuracy, which is a better prediction model.

**Further research 1.** Using LSTM will consume more time. We consider designing a machine learning algorithm that can approach the performance of LSTM in the next research (there may be limitations in the research field) to further shorten the time sequence prediction, and then achieve stronger real-time prediction. **2.** This article is aimed at the time series prediction of a single-dimensional data stream, but there are a lot of multi-dimensional data in the real scene. Therefore, we consider to use the research conclusions of this article as the basis for the follow-up research on multi-dimensional data from the aspects of data relevance and density.

**Acknowledgments** Yun Yang is very grateful to her for the help from teachers and other students in the process of completing the thesis. Special thanks to Liang Chen for his guidance in programming and senior Hongling Xiong for his guidance on thesis topic selection.

## References

1. Esling P, Agon C (2012) Time-series data mining. *ACM Comput Surv (CSUR)* 45(1):1–34
2. Martínez-Álvarez F, Troncoso A, Asencio-Cortés G, Riquelme JC (2015) A survey on data mining techniques applied to electricity-related time series forecasting. *Energies* 8(11):13162–13193
3. Wang Y, Liu Q, Tang J, Cao W, Li X (2014) Optimization approach of background value and initial item for improving prediction precision of gm (1, 1) model. *J Syst Eng Electron* 25(1):77–82
4. Hernandez-Matamoros A, Fujita H, Hayashi T, Perez-Meana H (2020) Forecasting of covid19 per regions using Arima models and polynomial functions. *Appl Soft Comput* 96:106610
5. Amini MH, Kargarian A, Karabasoglu O (2016) ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. *Electr Power Syst Res* 140:378–390
6. Wang W-c, Chau K-w, Xu D-m, Chen X-Y (2015) Improving forecasting accuracy of annual runoff time series using Arima based on eemd decomposition. *Water Resour Manag* 29(8):2655–2675
7. Xu W, Peng H, Zeng X, Zhou F, Tian X, Peng X (2019) Deep belief network-based ar model for nonlinear time series forecasting. *Appl Soft Comput* 77:605–621
8. Sarica B, Eğrioglu E, Aşıkil B (2018) A new hybrid method for time series forecasting: Ar-anfis. *Neural Comput & Applic* 29(3):749–760
9. Kim HY, Won CH (2018) Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple garch-type models. *Expert Syst Appl* 103:25–37
10. Kristjanpoller W, Minutolo MC (2016) Forecasting volatility of oil price using an artificial neural network-garch model. *Expert Syst Appl* 65:233–241
11. Taylor SJ, Letham B (2018) Forecasting at scale. *Am Stat* 72(1):37–45
12. Lin T, Horne BG, Giles CL (1998) How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. *Neural Netw* 11(5):861–868
13. Sagheer A, Kotb M (2019) Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing* 323:203–213
14. Chang Z, Yang Z, Chen W (2019) Electricity price prediction based on hybrid model of Adam optimized lstm neural network and wavelet transform. *Energy* 187:115804
15. Wang F, Xuan Z, Zhen Z, Li K, Wang T, Shi M (2020) A day-ahead pv power forecasting method based on lstm-rnn model and time correlation modification under partial daily pattern prediction framework. *Energy Convers Manag* 212:112766
16. Luo H, Huang M, Zhou Z (2019) A dual-tree complex wavelet enhanced convolutional lstm neural network for structural health monitoring of automotive suspension. *Measurement* 137:14–27
17. Jaseena KU, Kovoov BC (2021) Decomposition-based hybrid wind speed forecasting model using deep bidirectional lstm networks. *Energy Convers Manag* 234:113944
18. Chang Y-S, Chiao H-T, Abimannan S, Huang Y-P, Tsai Y-T, Lin K-M (2020) An lstm-based aggregated model for air pollution forecasting. *Atmos Pollut Res*
19. Moreno SR, da Silva RG, Mariani VC, dos Santos Coelho L (2020) Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network. *Energy Convers Manag* 213:112869
20. Kim K-j (2003) Financial time series forecasting using support vector machines. *Neurocomputing* 55(1–2):307–319
21. Qiu X, Ren Y, Suganthan PN, Amaratunga GAJ (2017) Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. *Appl Soft Comput* 54:246–255
22. Lin W, Zeng Y, Chen T (2015) Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Syst Appl* 42(2):855–863
23. Li S, Chen T, Lin W, Ming C (2018) Effective tourist volume forecasting supported by pca and improved bpnn using baidu index. *Tour Manag* 68:116–126
24. Yang H-j, Hu X (2016) Wavelet neural network with improved genetic algorithm for traffic flow time series prediction. *Optik* 127(19):8103–8110
25. Minu KK, Lineesh MC, John CJ (2010) Wavelet neural networks for nonlinear time series analysis. *Appl Math Sci* 4(50):2485–2495
26. Zhou Y, Li T, Shi J, Qian Z (2019) A CEEMDAN and XGBOOST-based approach to forecast crude oil prices. *Complexity* 2019
27. Ribeiro MHD, Mariani VC, dos Santos Coelho L (2020) Multi-step ahead meningitis case forecasting based on decomposition and multi-objective optimization methods. *J Biomed Inform* 111:103575
28. Li H, Jin F, Sun S, Li Y (2020) A new secondary decomposition ensemble learning approach for carbon price forecasting. *Knowl.-Based Syst* 214:106686
29. da Silva RG, Ribeiro MHD, Moreno SR, Mariani VC, dos Santos Coelho L (2020) A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting. *Energy* 216:119174
30. Agana NA, Homairfar A (2018) Emd-based predictive deep belief network for time series prediction: an application to drought forecasting. *Hydrology* 5(1):18
31. Wang W-c, Xu D-m, Chau K-w, Chen S (2013) Improved annual rainfall-runoff forecasting using pso-svm model based on eemd. *J Hydroinf* 15(4):1377–1390
32. Zhang Y, Chen B, Pan G, Zhao Y (2019) A novel hybrid model based on vmd-wt and pca-bp-rbf neural network for short-term wind speed forecasting. *Energy Convers Manag* 195:180–197
33. Wang Z-X, Li Q, Pei L-L (2018) A seasonal gm (1, 1) model for forecasting the electricity consumption of the primary economic sectors. *Energy* 154:522–534
34. Liu H, Yin S, Chen C, Zhu D (2020) Data multi-scale decomposition strategies for air pollution forecasting: a comprehensive review. *J Clean Prod* 277:124023
35. Awajan AM, Ismail MT, Wadi SA (2017) Forecasting time series using EMD-HW bagging. *Int J Stat Econ* 18(3):9–21
36. Cheng J, Yu D, Yu Y (2006) Research on the intrinsic mode function (IMF) criterion in EMD method. *Mech Syst Signal Process* 20(4):817–824
37. Kim D, Hee-Seok O (2009) EMD: a package for empirical mode decomposition and hilbert spectrum. *The R Journal* 1(1):40–46
38. Xingjian S, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: *Advances in neural information processing systems*, pages 802–810

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.