香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

A Hybrid Method for Stock Index Price Prediction in the Chinese Stock Market Using CEEMDAN and LSTM Neural Network.

Lingxiao XU

119020061

119020061@link.cuhk.edu.cn

Contents

1 INTRODUCTION

This paper discusses the application of a hybrid method of CEEMDAN (Complete Ensemble Empirical Mode Decomposition with Adaptive Noise) and LSTM (Long Short-Term Memory) neural network in the stock index prediction. Stock index prediction has been a classical yet difficult task since its time series has a complex, noisy, volatile, and non-parametric nature. To build an effective model to predict the stock index, both linear and non-linear tools have been implemented for the past couple of decades. Many existing models have the problems that the prediction is not sensitive to the booms or plunges of the stock market. To solve this problem, this paper proposed a hybrid prediction method using CEEMDAN and LSTM. This paper collected 4 years of data of SSE Composite Index (000001.SS) and Shenzhen Component Index (399001.SZ) from the Chinese stock market. The proposed model considers the stock index price as a non-stationary and non-linear time series signals which accords with the intrinsic characteristics of the stock market. After pre-processing of the data, the model performs CEEMDAN and decomposite the time series signals into several Intrinsic Mode Functions(IMF). Taking the IMFs as the inputs, LSTM neural network and Fully-connected neural network are implemented and output the predicted time series for each IMF. Finally, the solution sum up the outputs and scale back the data to the original representation to generate a prediction for the stock index. This paper also compare the prediction ability of CEEMDAN-LSTM model with the classical LSTM model.

2 THE DATA SET

This section details the data that was extracted from the public data sources. This paper uses the SSE Composite Index (000001.SS) and Shenzhne Component Index (399001.SZ) from 2016 to 2021, which covers two stock market in China. The data set contains 6 categories: close price, the highest price, the lowest price, trade amount, and trade volume.

3 METHODOLOGY

3.1 THE PROCEDURE OF THE METHOD

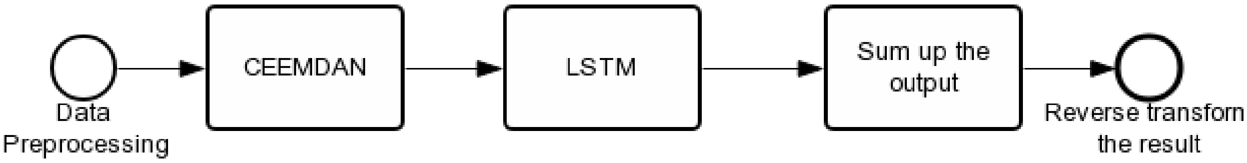

FIGURE 1: THE PROCEDURE OF THE METHOD

3.2 EMD, EEMD, AND CEEMDAN

EMD is a method of breaking down a signal into intrinsic mode functions without leaving the time domain. Formally, an IMF is a function whose upper and lower envelopes are symmetric, and the number of zero-crossings and the number of extremes are equal or differ at most by one. EMD makes no assumption about the composition of the signal. Instead, it successively deconposite the original signal into IMFs by interpolating between maxima and minima. The algorithm for the extraction of IMFs from a given signal $x(t)$ consists following steps:

1. Initialize $r_0(t) = x(t)$. Set the index of IMF k = 1.
2. Set i = 1 and $h_0(t) = r_{k-1}(t)$.
3. Find all minima and maxmima of $h_{i-1}(t)$ by peak finding algorithm.
4. create an envelope $e_{min,i(t)}$ of minima and $e_{max,i(t)}$ of maxima from the array of minima and maxima by Cubic Spline Interpolation.
5. Calculate the mean values of the upper envelope and the lower envelope: $m_i(t) = \frac{e_{min,i(t)}+e_{max,i(t)}}{2}$.
6. $h_i(t) = h_{i-1}(t) - m_{i-1}(t)$ and set $i = i + 1$.
7. Check if $h_i(t)$ is a IMF. If not, repeat step 3-5 until h_it being an IMF. Set $IMF_k(t) = d_i(t)$. Update the residual $r_k(t) = r_{k-1}(t) - IMF_k(t)$. Update $k = k + 1$.
8. Repeat step 2-6 until the final residual becomes a monotone function.

After the whole procedure of EMD, the initial signal $x(t)$ could be decomposed into several IMF as follows:

$$x(t) = \sum_{k=1}^{K} IMF_k + r \tag{1}$$

Although EMD could perform decomposition without assumption about the composition of the signal, the major shortcoming of EMD is the effect of mode mixing, which occurs when oscillation with different time scales are preserved in one IMF, or the oscillations with the same time scale are splitted into different IMFs To overcome the effect of mode mixing, Wu and Huang [4] proposed a noise-assisted EMD algorithm, named EEMD. The EEMD adds different series of white noise into the signal in several trials. Since the added noise is different in each trial, the correlation between the resulting trial with the corresponding IMFs from one trial to another could be eliminated. With the adequate number of trials, the added noise could be eliminated by averaging the obtained IMFs related to different trials. The algorithm of EEMD is performed as follows [1]:

1. In the nth trial, generating a new time series by adding a white noise time series $u_n(t)$ to a given signal $x(t)$:

$$Y_n(t) = x(t) + u_n(t) \tag{2}$$

   where $n = 1, 2, ..., N$ with N the ensemble number.

2. Perform EMD on $Y_n(t)$, the noise-contaminated signal $Y_n(t)$ is decomposed into several IMFs and a monotone residual.

$$Y_n(t) = \sum_{m=1}^{M-1} IMF_m^n(t) + r^{(}_{M}n)(t) \tag{3}$$

3. The step 1 and step 2 are reiterated for N trials and different white noise series are added to the original signals for each trial.

4. The final IMF of EEMD algorithm is obtained by averaging the total m IMF related to N trials:

$$IMF_m^{ave}(t) = \frac{1}{N} \sum_{n=1}^{N} IMF_m^{(n)}(t) \tag{4}$$

Although EEMD could solve the problem of mode mixing in EMD, it leads to a problem that the noise is not fully elimated by averaging the obtained IMFs, which means the resulting IMFs contain mixture of noise and signal. To solve this problem, Torres et al. [1] have proposed CEEMDAN algorithm that provides spectral seperation of the modes. The CEEMDAN algorithm could be performed as follows:

1. Performs EEMD algorithm and get the resulting $IMF_m^{ave}(t)$ and set the first resulting IMF as:

$$\widetilde{IMF}_1(t) = IMF_1^{ave}(t) \tag{5}$$

2. The first residue could be computed as:

$$r_1(t) = x(t) - IMF_1^{ave}(t) \tag{6}$$

3. Repeat the step 2 for N times. For each iteration, decompose the signal $r_1(t) + E_1(u_1(t))$, where $E_k(.)$ is an operator to extract k^{th} IMF from the given signal by EMD algorithm. The iteration stops when the resulting time series satisfies the requirement of an IMF.

$$\widetilde{IMF}_2(t) = \frac{1}{N}\sum_{m=1}^{N} E_1(r_1(t) + E_1(u_1(t))) \tag{7}$$

4. For the remaining IMFs, calculate the i^{th} residue and repeat the step 3 to get $\widetilde{IMF_i} + 1$:

$$r_i(t) = r_{i-1}(t) - \widetilde{IMF}_i(t) \tag{8}$$

$$\widetilde{IMF}_{i+1} = \frac{1}{N}\sum_{m=1}^{N} E_1(r_i(t) + E_i(u_i(t))) \tag{9}$$

5. Check whether the maxima and minima of the residue time series is more than 2. If so, stop the algorithm and the resulting residue could be represented as:

$$R_(t) = x(t) - \sum_{m=1}^{N} \widetilde{IMF}_m \tag{10}$$

By performing CEEMDAN algorithm, the time series $x(t)$ could be decomposed to:

$$x(t) = \sum_{m=1}^{N} \widetilde{IMF}_i + R(n) \tag{11}$$

This paper performs CEEDMAN method on the close price of the stock index. The distribution of the close price of 000001.SS and 399001.SZ are shown figure 1 and figure 2.

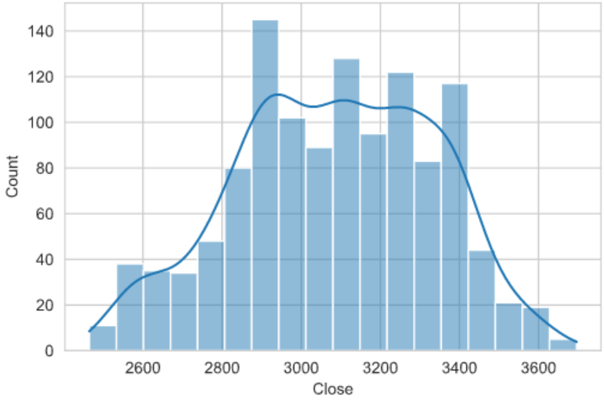

FIGURE 2: CLOSE PRICE OF 000001.SS

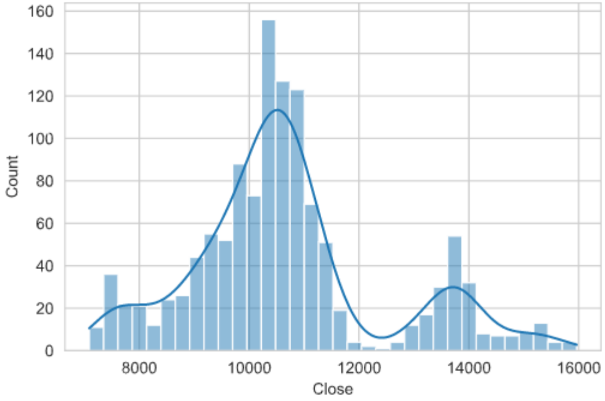

FIGURE 3: CLOSE PRICE OF 399001.SZ

To eliminate the effect of the magnitude of data, a min-max scaling method was firstly applied to the close price of two indexes. The algorithm is as follows:

$$\widetilde{x_i} = \frac{x_i - min(x)}{max(x) - min(x)} \tag{12}$$

After data scaling, CEEMDAN was implemented to decompose the close price time series signal into several IMFs.

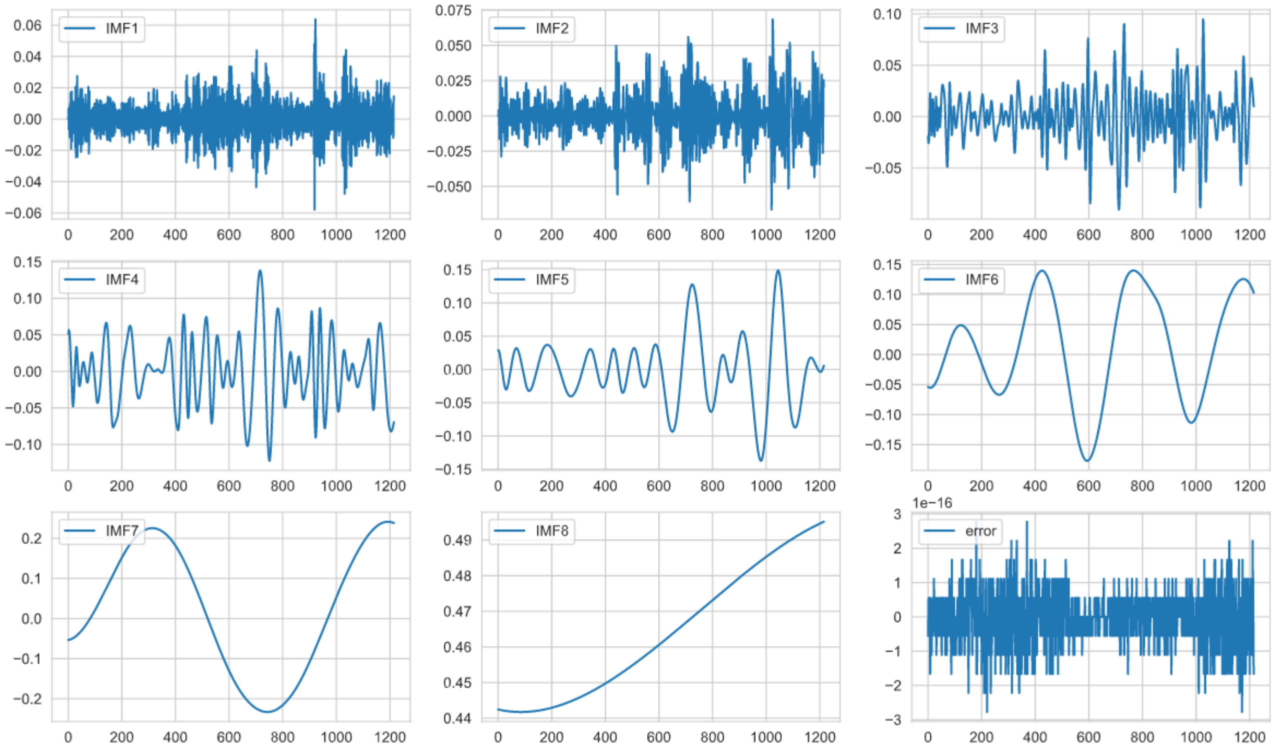

FIGURE 4: IMFS OF THE CLOSE PRICE OF 000001.SS

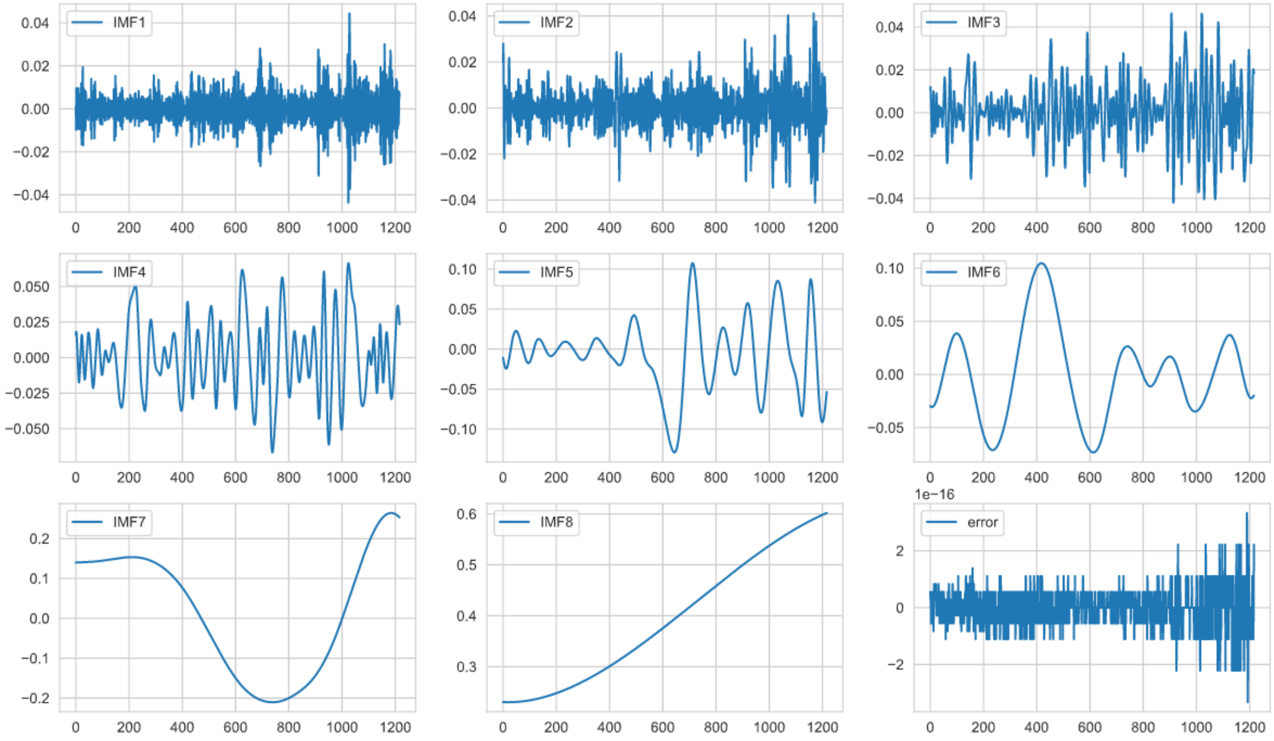

FIGURE 5: IMFS OF THE CLOSE PRICE OF 399001.SZ

3.3 LSTM NEURAL NETWORK

After obtaining the IMF of the close price, this paper uses LSTM neural network to predict each IMF in the future time series. Long short-term memory (LSTM) networks are a state-of-the-art technique for sequence learning. This paper's LSTM neural network follows Graves' description [2].

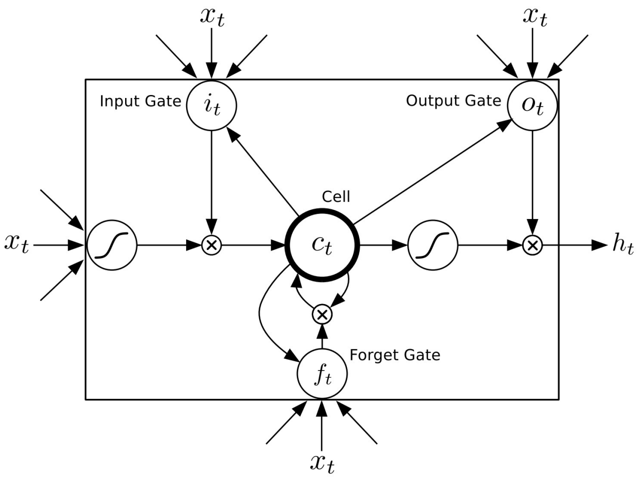

FIGURE 6: STRUCTURE OF LSTM MEMORY CELL FOLLOWING GRAVES (2013)

LSTM networks belongs to the class of recurrent neural networks(RNNs). The typical RNN suffers from the problems of vanishing gradients, which could be solved by LSTM by introducing gates to control the state c_t of the cell. Graves' LSTM memory cell has 3 gates:

1. Forget gate (f_t): The forget gate defines which information is removed from the cell state.
2. Input gate (i_t): The input gate determines which information is added to the cell state.
3. Output gate (o_t): The output gate determines which information from the cell state is used as output.

At every time t, each of the three gates is entered with the input x_t (one element of the input sequence) as well as the output h_{t-1} of the memory cells at the previous time $t-1$. The following equations are vectorized to illustrate the forward propagation of a LSTM memory cell at each time t. Hereby, the following notations will be used:

- x_t: the input data at time t.
- $W_{f,x}$, $W_{f,h}$, $W_{\tilde{c},x}$, $W_{\tilde{c},h}$, $W_{i,x}$, $W_{i,h}$, $W_{o,x}$, and $W_{o,h}$: weight matrices.
- b_f, $b_{\tilde{c}}$, b_i, and b_o: biased vectors.

In the first step, the LSTM memory cell would determine which information should be removed from the previous cell states by passing the output of the last memory cell h_{t-1} to the forget gate:

$$f_t = sigmoid(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \tag{13}$$

In the second step, LSTM memory cell defines which information should be added to the cell state C_t. This procedure contains two steps: 1. Calculate the candidates value $\widetilde{C_t}$ that would potentially be added to the cell state; 2. The input gate determines which values would be added to the cell state.

$$\widetilde{C_t} = tanh(W_{\widetilde{c},x}X_t + W_{\widetilde{c},h}h_{t-1} + b_{\widetilde{c}}) \tag{14}$$

$$i_t = sigmoid(W_{i,x}x_t + W_{i,h}h_{t-1} + b_{i-1}) \tag{15}$$

In the third step, the new cell state will be calculated by Hadamard(elementwise) product of $\widetilde{C_t}$ and i_t:

$$C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C_{t-1}} \tag{16}$$

In the last step, the output of the memory cell h_t is derived by:

$$o_t = sigmoid(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \tag{17}$$

$$h_t = o_t \circ C_t \tag{18}$$

For training LSTM neural network, this paper used Adam optimizer [3] and mini-batch stochastic gradient descent to update the parameter values. Besides, this paper implemented dropout regularized layer within the recurrent layer to avoid over-fitting. This paper used time lag of 20 and look forward 5 days, which means for each time t, this paper used $t - 20$ to $t - 1$ as the input data and predict the time series for time t to $t + 4$. The neural network of this paper composed two layers: one LSTM layer with 20 neurons and one fully-connected layer. The dropout rate for each layer is 0.9 and 0.99 respectively. The input of LSTM neural network is the resulting IMFs of the close price, and the output dimension of the fully-connected layer equals to the number of IMFs, which means that the network predict the future time series for each IMFs. The loss function of this neural network is the root mean square error(RMSE).

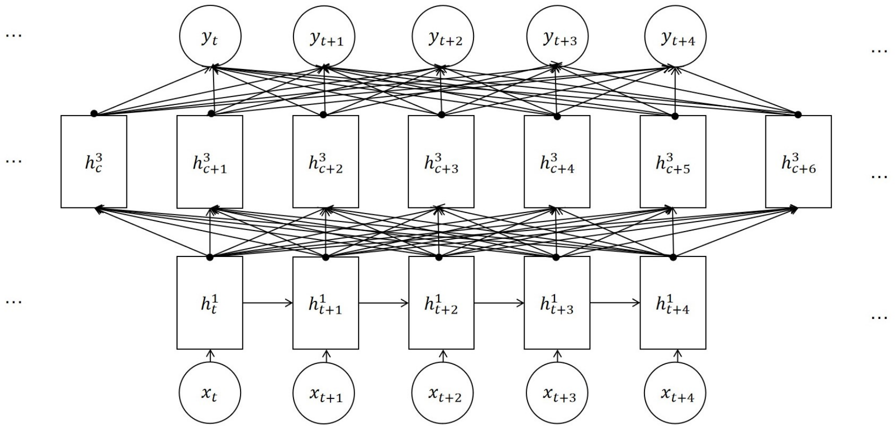

FIGURE 7: STRUCTURE OF THE NEURAL NETWORK

This paper divide data into two parts: training data (80% of the total data set) and testing data (20% of the total data set). In training data, this paper use 10% as validation set to adjust the hyper parameters. The loss of the neural network converges in 100 epochs.

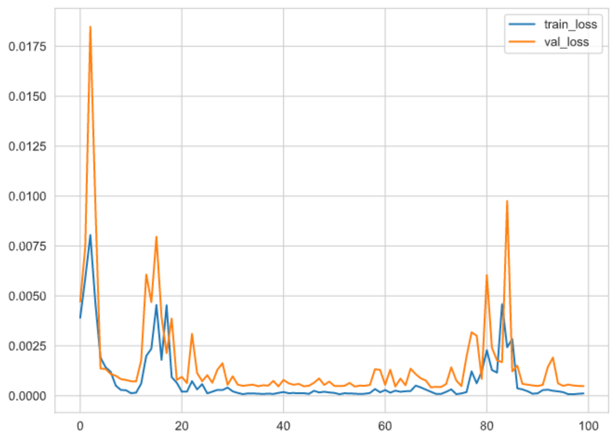

FIGURE 8: LOSS OF 000001.SS

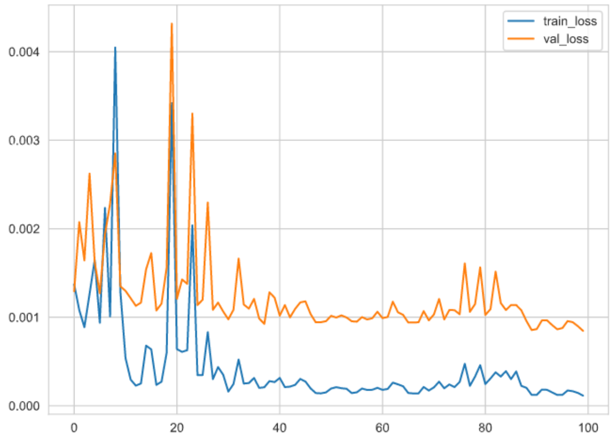

FIGURE 9: LOSS OF 399001.SZ

This paper then applied the model to the training data set to validate model's ability of fitting the data.

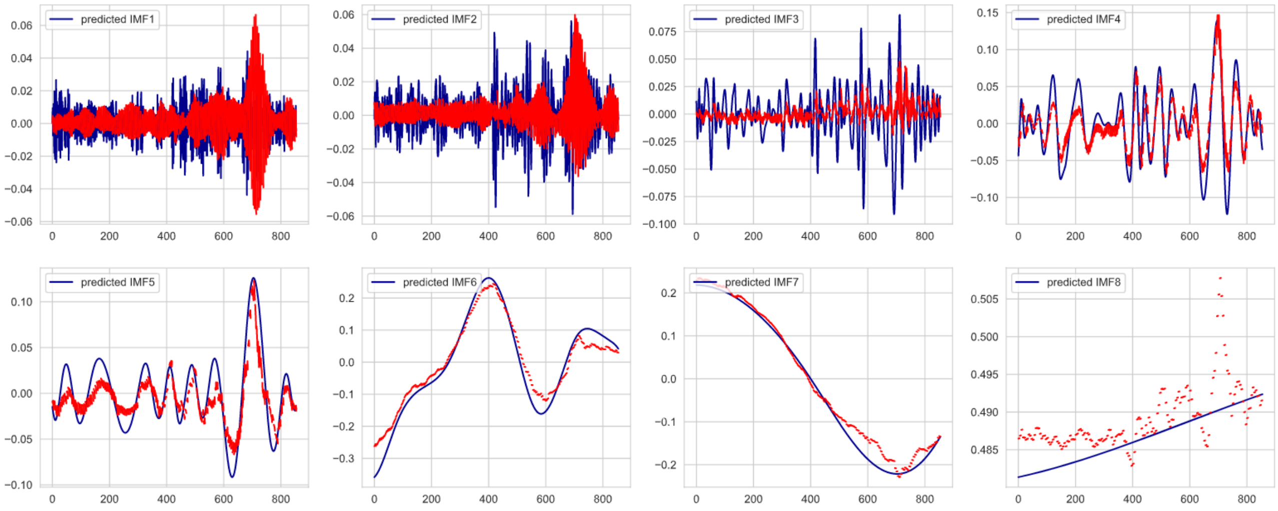

FIGURE 10: THE PREDICTED IMFS ON TRAINING DATA OF 000001.SS

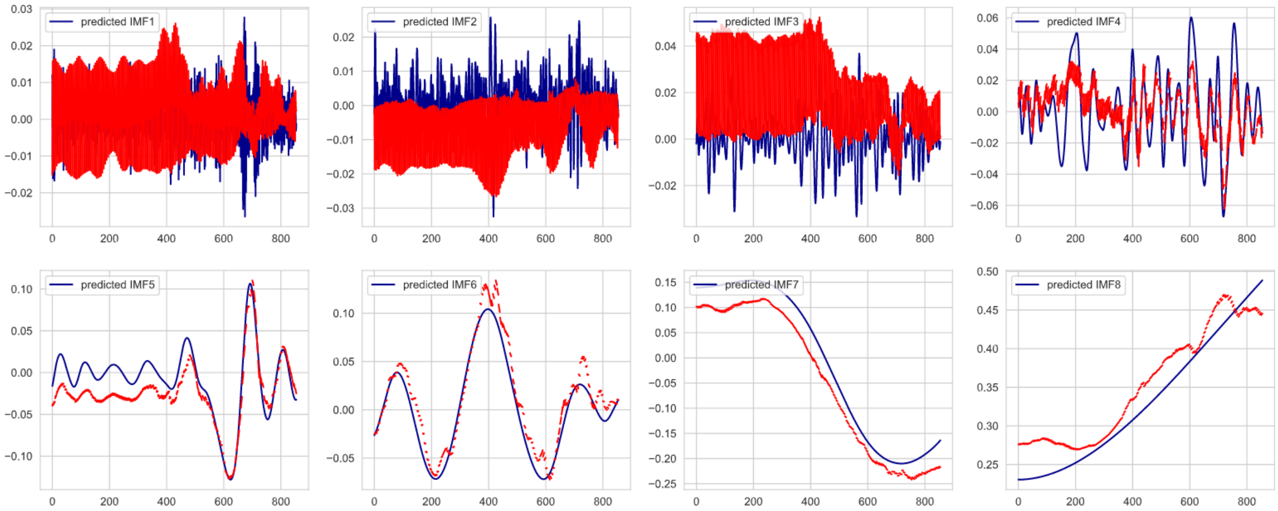

FIGURE 11: THE PREDICTED IMFS ON TRAINING DATA OF 399001.SZ

3.4 FORECASTING

After model training, the predicted IMFs $\widetilde{IMF_i}$ were predicted on the testing data set. The predicted price $\widetilde{Y_t}$ is obtained by sum the predicted IMFs and invert min-max algorithm.

$$\widetilde{y_t} = \sum_{i=1}^{N} \widetilde{IMF_i}(t) \tag{19}$$

$$\widetilde{Y_t} = \widetilde{y_t} \times (max(x) - min(x)) + min(x) \tag{20}$$

4 RESULTS

4.1 PERFORMANCE REVIEW

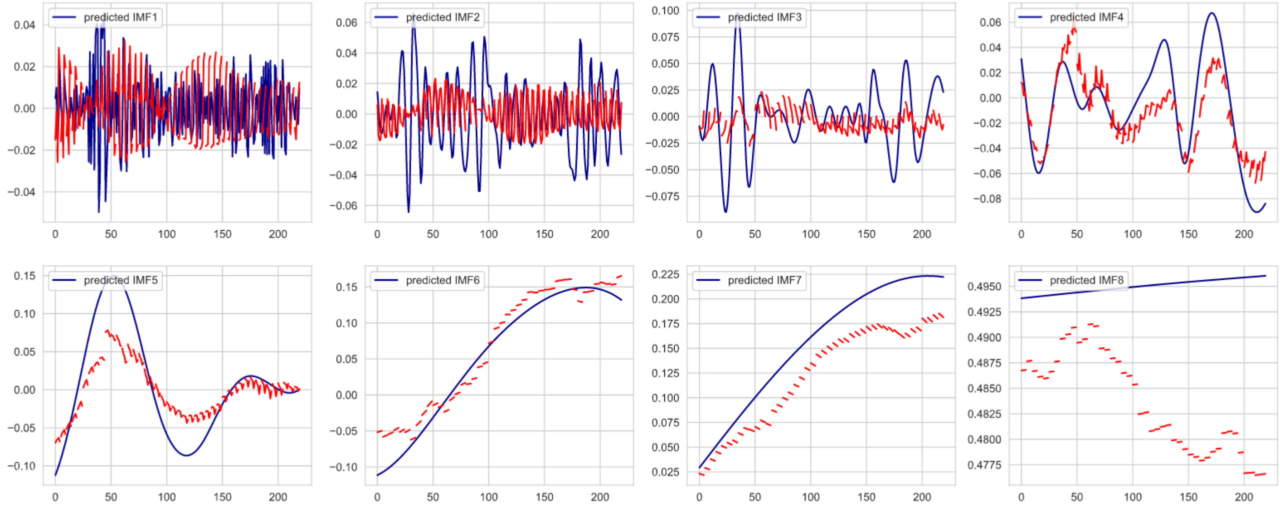

FIGURE 12: THE PREDICTED IMFS ON TESTING DATA OF 000001.SS

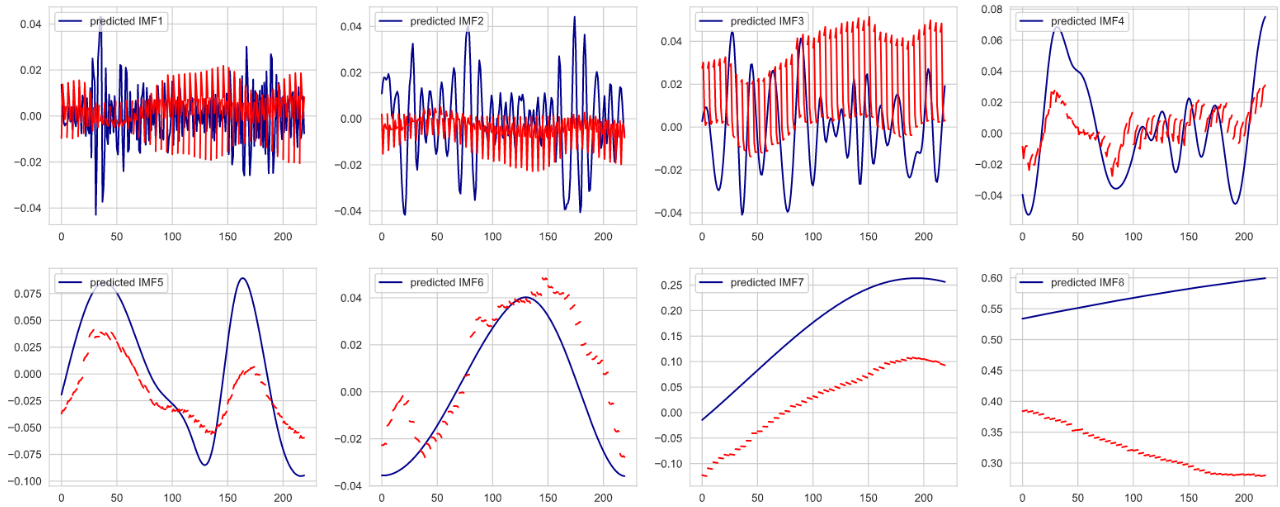

FIGURE 13: THE PREDICTED IMFS ON TESTING DATA OF 399001.SS

From figure 10 to figure 13, this paper found that the LSTM neural network is relatively incapable of fitting and predicting the time series with high frequency. The predicted price on training and testing data was then obtained by following the forecasting procedure.

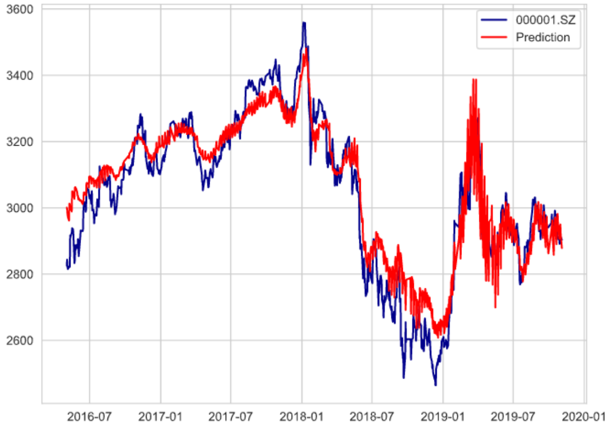

FIGURE 14: PREDICTED PRICE ON TRAINING DATA OF 000001.SS

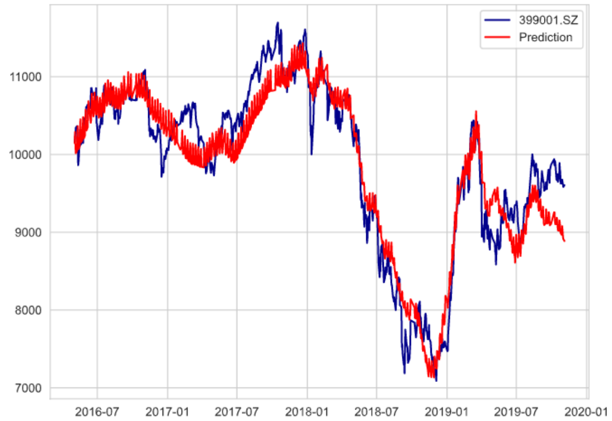

FIGURE 15: PREDICTED PRICE ON TRAINING DATA OF 399001.SZ

To evaluate the fitting ability of the proposed model,the average RMSE and R squared of the model were calculated by performing the model for 10 times.

TABLE 1: EVALUATION OF THE TRAINING DATA

	RMSE	R-squared
000001.SS	2320.42	69.96%
339001.SZ	10025.73	89.92%

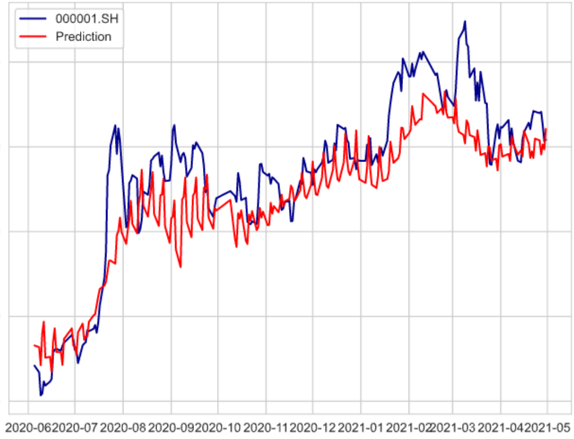

FIGURE 16: PREDICTED PRICE ON TESTING DATA OF 000001.SS

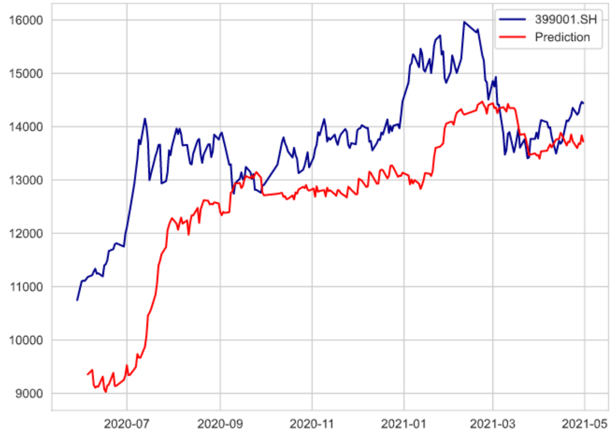

FIGURE 17: PREDICTED PRICE ON TESTING DATA OF 399001.SZ

TABLE 2: EVALUATION OF THE TESTING DATA

	RMSE	R-squared
000001.SS	1500.26	64.19%
339001.SZ	21746.53	184.44%

4.2 COMPARISON WITH TYPICAL LSTM NEURAL NETWORK MODEL

To better illustrate the prediction ability of the proposed model, this paper compared the prediction and fitting ability with the typical LSTM neural network model, which uses only pre-processed stock index as the input and predict the future stock index price. The LSTM neural network used as comparison in this paper has the same structure as the LSTM neural network in the proposed model: One LSTM layer with 20 neurons, and one fully-connected layer.

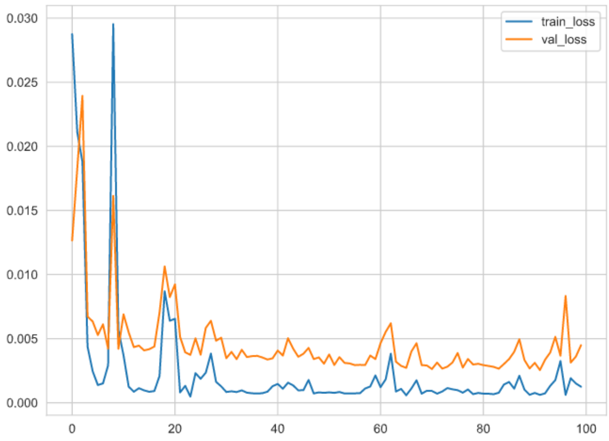

FIGURE 18: LOSS OF LSTM ON 000001.SS

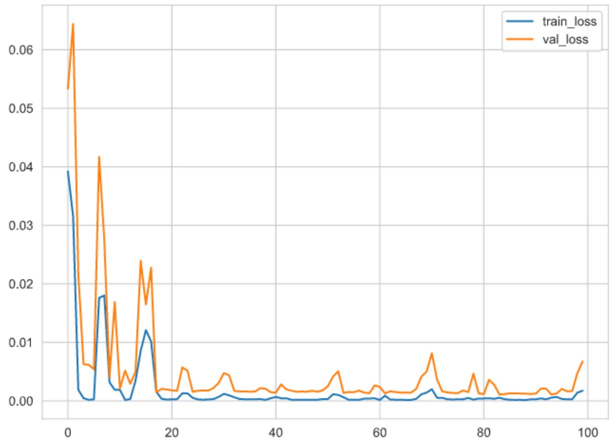

FIGURE 19: LOSS OF LSTM ON 399001.SZ

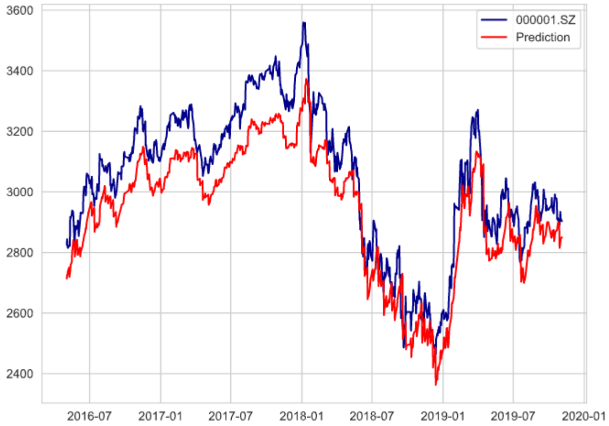

FIGURE 20: PREDICTION OF LSTM ON TRAINING DATA FROM 000001.SS

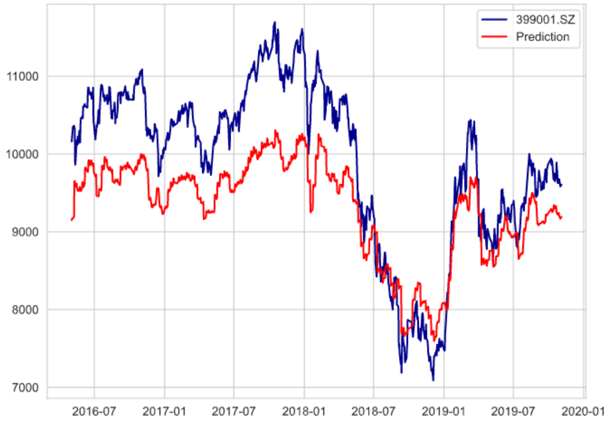

FIGURE 21: PREDICTION OF LSTM ON TRAINING DATA FROM 399001.SZ

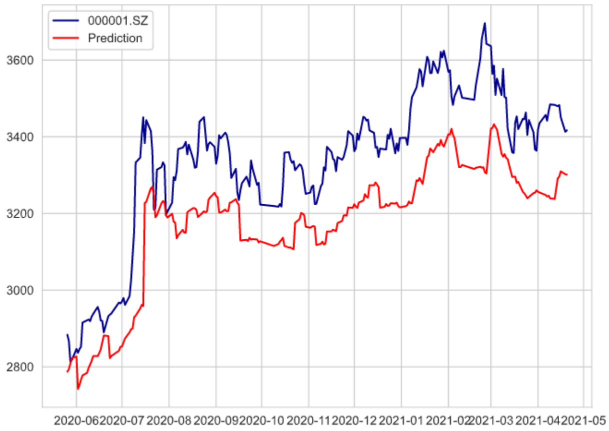

FIGURE 22: PREDICTION OF LSTM ON TESTING DATA FROM 000001.SS

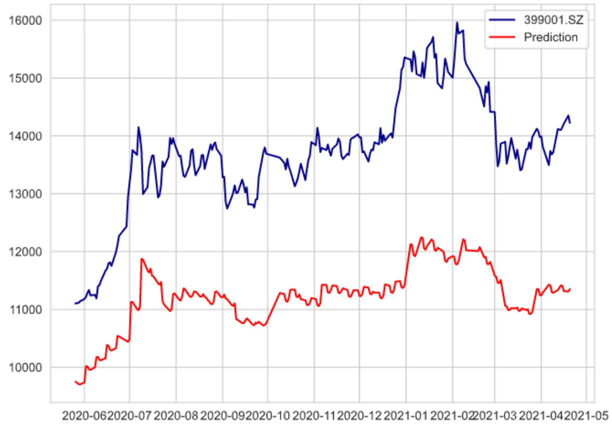

FIGURE 23: PREDICTION OF LSTM ON TESTING DATA FROM 399001.SZ

To compare the prediction ability between the model, the average RMSE were calculated by performing the models by 10 times.

TABLE 3: COMPARISON OF RMSE BETWEEN LSTM AND CEEMDAN-LSTM

	LSTM	CEEMDAN-LSTM
000001.SS	2623.31	2320.42
339001.SZ	37746.60	21746.53

4.3 THE PERFORMANCE FOR DIFFERENT LENGTH OF TIME WINDOW

To study model's prediction ability for different length of time window. This paper mainly considers three cases:

- 5-1: Look back for 5 days and look forward for 1 day.
- 20-5: Look back for 20 days and look forward for 5 days, which is the time widow used in the proposed model.
- 90-30: Look back for 90 days and look forward for 30 days.

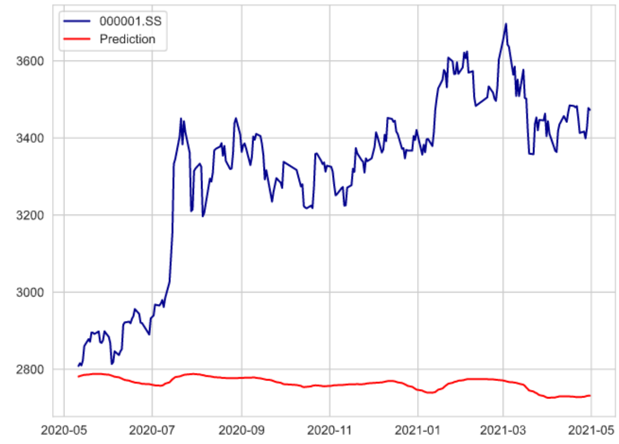

FIGURE 24: 5-1 USED ON TESTING DATA FROM 000001.SS

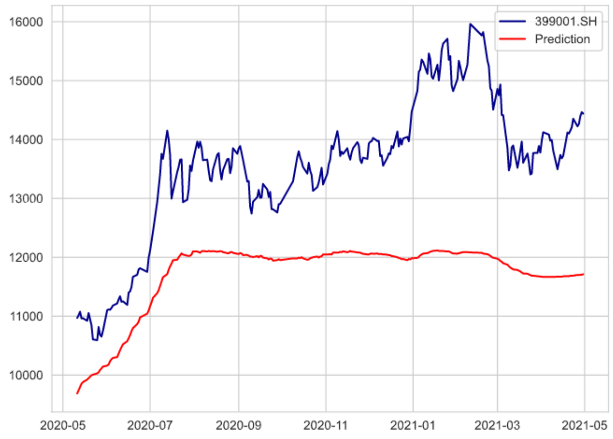

FIGURE 25: 5-1 USED ON TESTING DATA FROM 399001.SZ

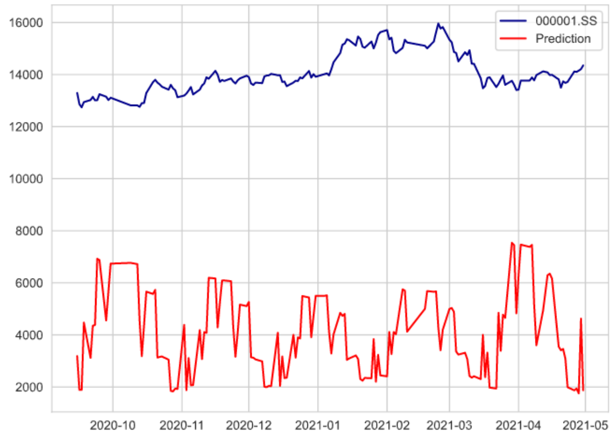

FIGURE 26: 90-30 USED ON TESTING DATA FROM 000001.SS

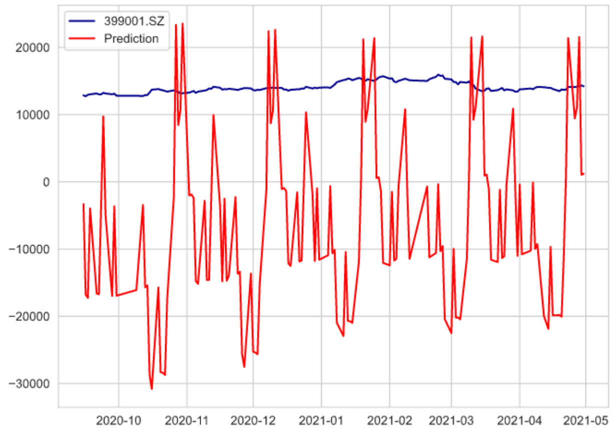

FIGURE 27: 90-30 USED ON TESTING DATA FROM 399001.SZ

TABLE 4: COMPARISON OF RMSE BETWEEN DIFFERENT TIME WINDOW

	5-1	20-5	90-30
000001.SS	9085.27	2320.42	124925.94
339001.SZ	30424.62	21746.53	295201.04

5 CONCLUSION

In this paper, we developed a hybrid method with CEEMDAN and LSTM neural network to predict the stock index price. Firstly, the proposed model enhanced the prediction ability of the classical LSTM neural network in forecasting the stock index price. The underlying logic of the proposed model is to use CEEMDAN to decompose the price signal into several IMFs that are more stationary and follow a cyclical pattern before performing LSTM neural network on the time series. LSTM neural network has a better performance when predicting the stationary time series. Besides, by decomposing the time series into the signals that are changing with high frequency and the signals that are changing with low frequency, the proposed model could better capture the long-tern trend of the stock index price. By comparing with the classical LSTM neural network, this paper found that the proposed model has a better performance in both fitting and predicting the stock index price. Secondly, by comparing the prediction ability between different time windows, this paper found that the proposed model has a better performance when predicting the stock index for a week(5 days) by using the data of nearly one month(20 days). This indicates that the proposed model is more suitable for constructing a trading strategy with weekly frequency. Overall, this paper illustrate that CEEMDAN-LSTM model is capable of fitting and predicting the stock index price in the Chinese stock market. Compared with the classical LSTM neural network, the proposed model is a method of choice to extract meaningful information from noisy financial time series data to construct a weekly-based trading strategy.

References

[1] Diez, P. F., Torres, A., Avila, E., Laciar, E. & Mut, V. (2009), Classification of mental tasks using different spectral estimation methods, INTECH Open Access Publisher.

[2] Graves, A. (2013). Generating sequences with recurrent neural networks.

[3] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[4] Wu ZH & Huang NE. ‘’Ensemble empirical mode decomposition: a noise-assisted data analysis Method’’ AADA: Advances in Adaptive Data Analysis, vol. 1, pp.1-4. 2009.