Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Foreign exchange rates forecasting with an EMD-LSTM neural networks model

To cite this article: Boning Zhang 2018 *J. Phys.: Conf. Ser.* **1053** 012005

View the article online for updates and enhancements.

Related content

- Time scale defined by the fractal structure of the price fluctuations in foreign exchange markets
Yoshiaki Kumagai

- Description of dynamics of stock prices bya Langevin approach
Huang Zi-Gang, Chen Yong, Zhang Yong et al.

- Two layers LSTM with attention for multi-choice question answering in exams
Yongbin Li

IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

This content was downloaded from IP address 118.70.67.78 on 04/12/2019 at 02:04

Foreign exchange rates forecasting with an EMD-LSTM neural networks model

Boning Zhang[1]

Department of Software Engineering, Shan Dong University, Jinan, China

[1]Email:zhang_bning@126.com

Abstract. Since financial time series are inherently noisy and unstable, it is quite challenging to enhance the forecasting accuracy. To alleviate the adverse effect, in this study, the combined model of Empirical Mode Decomposition (EMD) and Long Short-Term Memory (LSTM) is proposed. In this model with the EMD, the original time series are decomposed into IMF sequence and residual sequence. LSTM is applied to the financial forecasting model. The proposed model present superior forecasting capacity, which proves that machine learning performs better in nonlinear time series forecasting.

1. Introduction

For decades, studies in time series forecasting greatly support predictions in various real-world applications [1]. When most research emphasize the critical value of on time series forecasting, those nonlinear and non-stationary time series forecasting become interesting and draw increasing attention in academia [2]. Therefore, more scholars devote themselves to study the effectiveness of time series forecasting model. Statistical models and artificial intelligence models most popular forecasting models in research [3]. However statistical models do have drawn backs due to their innate linear form. As a result, nonlinear models are worthy to study their extraordinary capability of constructing nonlinear model. [4][5][6]. Empirical studies indicate that neural networks perform better than statistical models occasionally [7][8]. Of all learning algorithm, recursive neural network (RNN) works better with sequence dependency. Long Short-Term Memory (LSTM) is such kind of RNN algorithm, which may greatly facilitate exchange rates forecasting. Since LSTM is widely used to predict, classify and generate sequence data, it is widely used in business [9]. It combines the scoring process and threshold value selecting process instead of separate processes [10]. Therefore, all these unique features enable LSTM to perform well in time series forecasting.

However, when it comes to random walk models, where drawbacks of nonlinear model remain, neural network models cannot be used as well [11]. Therefore, it is inappropriate to apply one single forecasting model in time series forecasting in real-world situations [12]. Even though LSTM could be a better solution, it is still difficult to deal with processing noise and non-stationary. Hence, the preprocessing of time series becomes significant in time series forecasting.

Empirical mode decomposition (EMD) can present inherent characteristics of a signal, which makes it a good solution to decompose those noisy and non-stationary time series [13][14]. With EMD, a group of intrinsic mode functions and a residual are decomposed. IMF makes the forecasting easier and more accurate because of compositions and tight correlations [15]. Hence, EMD is widely used to preprocess nonlinear original time series which makes the forecasting more accurate. Moreover, EMD-LSTM model shows a good performance in the prediction of electric load [16].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

In this paper, a better forecasting model was proposed to enhance the accuracy and to alleviate the impact of noise. The model is applied in two steps. At first, time series are decomposed into some IMFs and a residual. Then, with different IMFs and residual, LSTM are used in modeling. In the following four sections of this paper, EMD and LSTM are reviewed in section 2, the proposed model is explained in section 3, the empirical time series forecasting is conducted in section 4 and in section5, conclusions and suggestions in this research are discussed.

2. Preliminaries

2.1. Empirical Mode Decomposition (EMD)

The core algorithm of HHT is EMD. The EMD is used to decompose the poor performance signal in to some intrinsic mode functions (IMFs) and a residue.

The calculation described in the following section. Firstly, we search for all spots in both maximum and minimum numbers in the original time series. In this manner, the upper and lower envelope of the initial sequence are determined. The mean is appointed m_1 and the initial sequence minus $\mathrm{m}_1$will gain a low frequency data sequence $\mathrm{h}_1(\mathrm{t})$, which is the result of $\mathrm{x(t)}$ minus $\mathrm{m}_1(\mathrm{t})$.However, $\mathrm{h}_1(\mathrm{t})$ is not always a stationary data sequence, so we replicate the process for several times. Iterate as needed so that the first extracted IMF $\mathrm{c}_1(\mathrm{t})$ from time series of the highest frequency is subtracted from the initial data. With $\mathrm{x(t)}$ minus $\mathrm{c}_1(\mathrm{t})$, a fresh data array$\mathrm{r}_1(\mathrm{t})$ that takes out the high frequency component. Consider $\mathrm{r}_1(\mathrm{t})$ as the original sequence and reapply above operation. When the last data sequence $\mathrm{r}_\mathrm{n}(\mathrm{t})$ cannot be decomposed, the operation of extracting new IMFs can stop. Thus, the original sequence is decomposing like following:

$$x(\mathrm{t}) = \sum_{\mathrm{j=1}}^{\mathrm{n}} c_j(\mathrm{t}) + r_n(\mathrm{t}) \tag{1}$$

$\mathrm{x(t)}$ is the original time series. $\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{c}_\mathrm{j}(\mathrm{t})$ are the IMFs and $\mathrm{r}_\mathrm{n}(\mathrm{t})$ is the residual.

2.2. Long Short-Term Memory (LSTM)

LSTM is a kind of time recursive neural network (RNN), which can learn long-term dependency information. The LSTM has an additional "processor" called cell in the algorithm to determine whether information is useful or not. One cell has three doors called the input gatei_t, the forgetting gatef_t and the output gateo_t, respectively, which make the weight of the self-circulation change dynamically at different moments when parameters are fixed. At time t, every cell records the cell state C_t, which will impact the rear cell block. The cell will read inputs from two external sources: the previous hidden state h_{t-1} and the input vector x_t. The whole process is showed in figure 1.

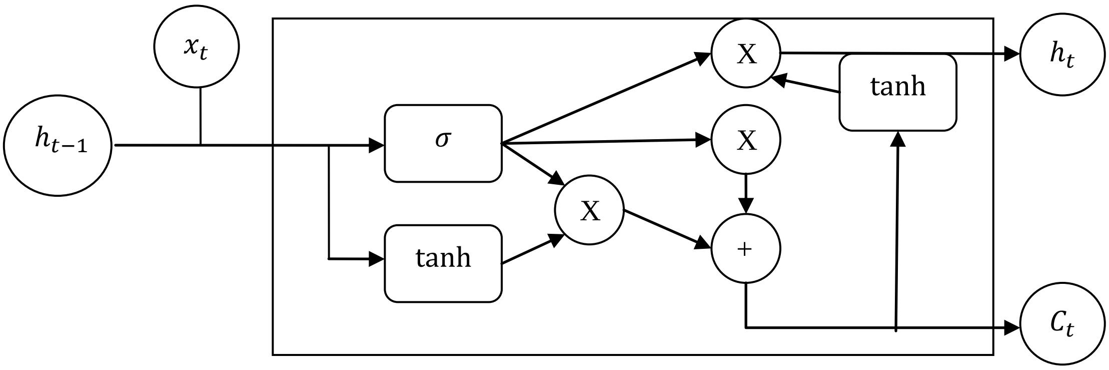

Figure 1. Overall processed for LSTM model.

Firstly, we need to determine which information should be abandon from the cell state. The forgetting gate readsh_{t-1} and x_t, and assigns a number for each value in the former cell statec_{t-1} as the output. "0" means "completely abandonment" and "1" means "complete reservation".

$$f_t = \sigma[W_f(h_{t-1}, x_t) + b_f] \quad (2)$$

Then, we need to decide the new information which will store in the cell state. Sigmoid function will decide what value we need to update i_t and tanh function will create a new candidate vector $\tilde{C}_t$.

$$i_t = \sigma[W_i(h_{t-1}, x_t) + b_i] \quad (3)$$

$$\tilde{C}_t = tanh[W_C(h_{t-1}, x_t) + b_C] \quad (4)$$

Afterwards, the old cell state should be updated. The first step has already determined the numberf_t which needs to multiply the old stateC_{t-1}, which decides the information we need to discard. Adding the result of the second step, we can gain the new cell state C_t.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

Finally, the output value needs to be determined. First, we use sigmoid function to decide which part of the state cell o_t will be output. Then, o_t need multiple with the cell state that processed by tanh function as the final output h_t.

$$o_t = \sigma[W_o(h_{t-1}, x_t) + b_o] \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

In the above equation, σ() is the sigmoid function and tanh() is the tanh function. W_f, W_iandW_c are peep hole connection. $b_f,\ b_i$and b_c are the input weight matrix.

3. Proposed model

The proposed model is a combination of EMD and LSTM. The whole operations are consists of three main procedures: decomposition of the original sequence, fitting of the model and final forecasting. Each procedure in the proposed model is illustrated in figure 2.

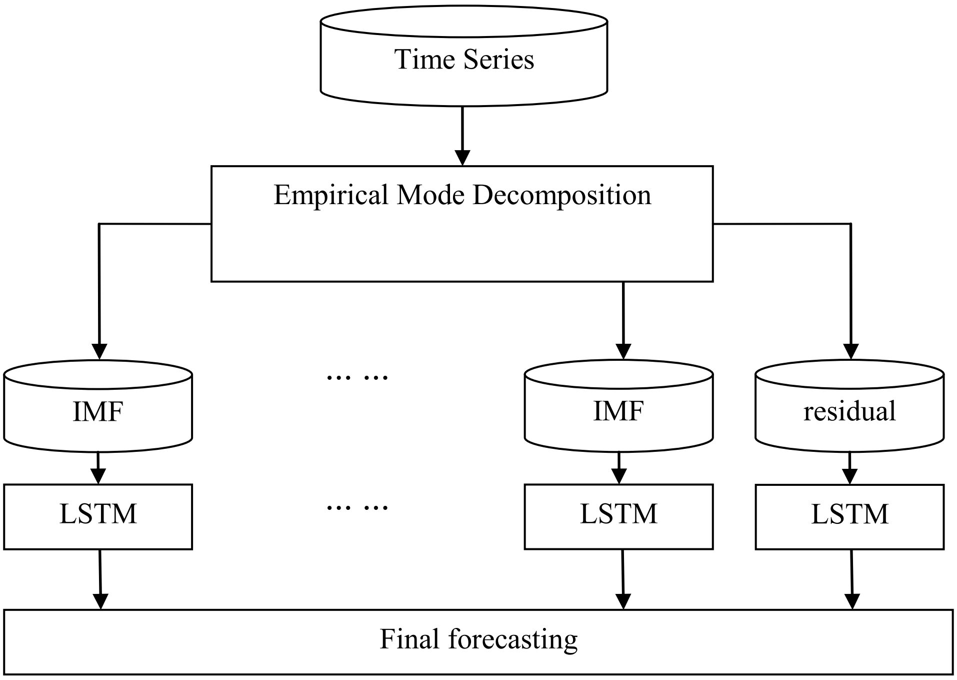

Figure 2. Overall processed for proposed model.

3.1. Decomposition of original sequence

The actual time seriesx_t, t = 1, 2,..., N is decomposed by EMD to IMFs time series$c_j(t)$, j = 1,2,...,n and one residual time series $r_n(t)$. First, upper envelopes and lower envelopes should be identified. Then we should figure out their mean and extract it from original sequence. The shifting process must be repeated to reduce the extracted signal. After finding the last mode, a final residue will exist. In this step, the input is the original time series and we don't need to set other parameters.

3.2. Fitting of the model

Each IMF and residual datasets is broken into training and testing dataset. Firstly, every training and testing dataset need to be standardized. And then, preset the needed variable and build neural networks based on the preset weight. We should try several times to ensure more appropriate preset parameters. Because better preset parameters can improve the performance of the final forecasting. In this step, we need to set the hidden layer unit as 10, the input and output size as 1, the learning rate as 0.06, the batch size as 80 and the time step as 15.By using the LSTM, the model should be trained by training dataset and repeat the process of training in order to improve the accuracy. Finally, testing dataset is used as the input to the predict model which has already constructed the parameters and obtain the forecasting data which should be compared with the original dataset to assess the capacity of prediction accuracy. Results of IMF time series $y_j(t)$ and residual time series $y_r(t)$ export as the respectively output. Equation (8) and (9) are the detailed function, which produced from the LSTM.

$$y_j(t) = f[c_j(t)] \tag{8}$$

$$y_r(t) = f[r_n(t)] \tag{9}$$

3.3. Final forecasting

The forecasting results are reconstruct the forecasting values from each LSTM model, including the LSTM models of IMFs time series $y_j(t)$ and the LSTM model of residual model.Equation (10) is the final forecasting result.

$$Y(t) = \sum_{j=1}^{n} y_j(t) + y_r(t) \tag{10}$$

After that, different models are built separately to deal with noise. The outputs of those models present better forecasting capacity. Finally, the final forecasting is implemented by combining LSTM models of IMFs and residual, which were decomposed by EMD. Aiming to assess the accuracy of the result, images of predicted values and original values are observed. Moreover, the result of them is verified by three well-known measures.

4. Experiments and results

4.1. Data sets

To evaluate the performance of the forecasting model, daily exchange rate values for USD/CNY, EUR/CNY, and USD/EUR were selected from the database provided by Investing.com and utilizedas original datasets, with 4770 data points in total. The three initial data sets divided into training data sets and testing data sets respectively, which avoid to choose the turning point. Training data set are the 4500 data (94% of the entire data set) from January 1, 2000 to April 4, 2017 and the testing data set are the 270 data (6% of the entire data set) from April 4, 2017 to April 17, 2018. The scale of the training data set is sufficient with enough data to fit a good model and to avoid excessive fitting. Since the sample training is sufficient, the number of training sets is also appropriate. Table 1 shows statistics for the original, training and test data set.

Table 1. Basic statistics of three exchange rate data set.

Model	Under USD/EUD dataset			Under USD/CNY dataset			Under EUD/CNY dataset		
	Original	Training	Testing	Original	Training	Testing	Original	Training	Testing
Numbers	4770	4500	270	4770	4500	270	4770	4500	270
Mean	0.845	0.844	0.856	7.237	7.274	6.618	8.768	8.827	7.782
Max	1.216	1.216	0.946	8.280	8.280	6.915	11.314	11.314	8.001
Min	0.630	0.630	0.803	6.044	6.044	6.270	6.601	6.601	7.328

4.2. Benchmark prediction models

ARIMA: ARIMA is composed of three parts including autoregressive, integration, and moving average. Applying ARIMA to the original time series lead to the final prediction. Being accurate and flexible, the prediction of ARIMA model ensures the fitness of different time series [17]. We assume that parameters of p, q, r and 1 are equals.

ANN: a part of the initial sequences is used to build ANN models and the other part of the initial data set is used to get the final result. ANN is widely used in data modeling and forecasting in different areas [18]. Also, ANN shows a great performance in the time series forecasting. We assume that the hidden layer size is 100, the batch size is 200 and the learning rate is 0.001.

LSTM: The same method of ANN is applied. With the growing information, LSTM can enhance the prediction accuracy by learning from those fluctuations [19]. In addition, we test the LSTM model to examine the impact of the EMD model. The parameters used in this model are the same as those in the proposed model.

Proposed model: First, we use EMD to decompose the initial time series into some IMF sequences and one residual sequence. Afterwards, we fit these time series irrelevant LSTM models independently.

4.3. Evaluation criteria

We use three well-known measures to evaluate the forecasting capability of the proposed model. The three measures are mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error(MAPE) , which are defined as the following formulas:

$$\mathrm{MSE} = \mathrm{N}^{-1}\sum_{t=1}^{N}(\mathrm{Y}_{(t)} - \widehat{\mathrm{Y}_{(t)}})^2 \tag{11}$$

$$\mathrm{MAE} = \mathrm{N}^{-1}\sum_{t=1}^{N}\left|\mathrm{Y}_{(t)} - \widehat{\mathrm{Y}_{(t)}}\right| \tag{12}$$

$$\mathrm{MAPE} = \mathrm{N}^{-1}\sum_{t=1}^{N}\left|(\mathrm{Y}_{(t)} - \widehat{\mathrm{Y}_{(t)}})/\mathrm{Y}_{(t)}\right| \tag{13}$$

Let $\mathrm{Y}_{(t)}$ denotes the actual value at period t, $\widehat{\mathrm{Y}_{(t)}}$ denotes the forecasted value at period t and N denotes the sample size. Thus, a lower value suggests better forecasting performance.

4.4. Forecasting results

According to table 2, the proposed model performs the best results in most case. More specially, when there is a relatively low exchange rate, the proposed model is superior in predication accuracy than that of ANN and ARIMA. The data proves that LSTM model is applicable in model fitting and forecasting. Meanwhile, the EMD is helpful to alleviate the noise in original time series and thus lead to better prediction outcome.

The appropriate 50 data is selected from January 23, 2018 to April 3, 2018. The selected data minus the original data and the absolute value is taken to show in the figure 3, 4 and 5. The result shows the proposed model has the closest result with the original data and the comparative model has more errors.

Table 2. Performance of different models.

Model	Under USD/EUD dataset			Under USD/CNY dataset			Under EUD/CNY dataset		
	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE
ARIMA	0.000025	0.003000	39.4%	0.000049	0.004000	5.50%	0.002401	0.035000	4.04%
ANN	0.000012	0.002536	0.300%	0.000523	0.019139	0.292%	0.000871	0.022163	0.285%
LSTM	1.04937e-05	0.002447	0.286%	0.000295	0.013064	0.197%	0.000921	0.022356	0.288%
EMD+LSTM	3.09723e-06	0.001294	0.151%	7.60310e-05	0.006456	0.098%	0.000179	0.009998	0.129%

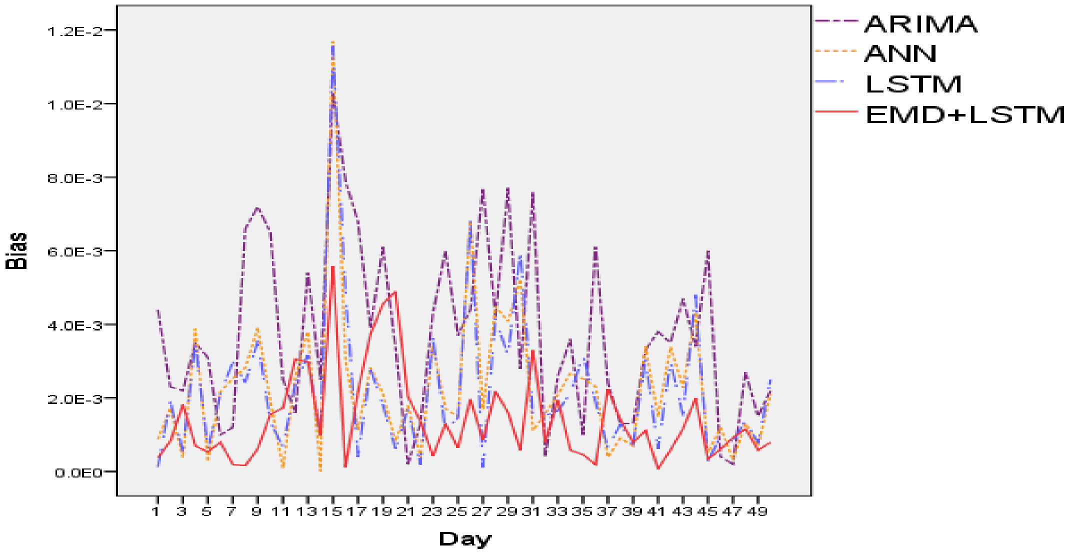

Figure 3. Under USD/EUD exchange rate forecasted values.

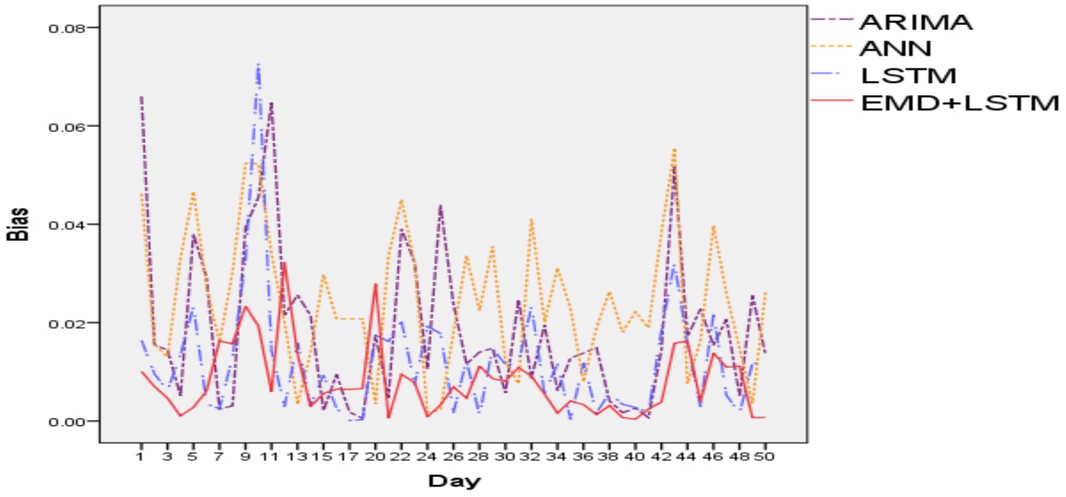

Figure 4. Under USD/CNY exchange rate forecasted values.

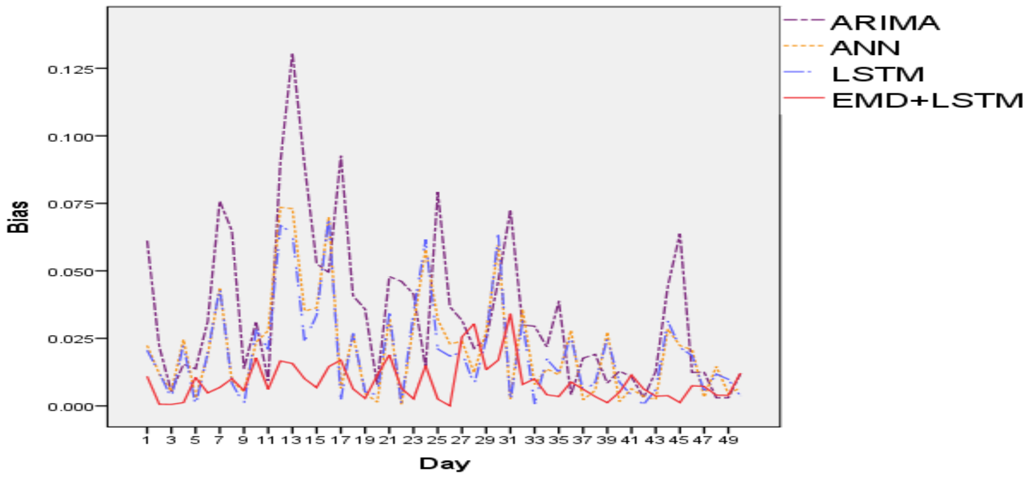

Figure 5. Under EUD/CNY exchange rate forecasted values.

5. Conclusions

By combining EMD and LSTM, the proposed model greatly enhance prediction capacity, forecasting accuracy and model robustness. It could better deal with the noisy and non-stationary original time series and could keep learning the generated fluctuation.

In most cases, the proposed model achieves the best results among all three measures. In particular, when the value of the exchange rate is low, the prediction of the proposed model performs significantly better. Moreover, thanks to in pre-training the intrinsic mode functions are extracted, thus providing an appropriate solution to fluctuations. Since the proposed model performs the best in all three measures, it is obvious that EMD can improve forecasting accuracy. Furthermore, the LSTM demonstrates a better capacity in nonlinear time series.

In future, phase space reconstruction is worthy to study. We could reconstruct a phase space from the one-dimensional chaotic time series that is identical to the original dynamical system. It is quite challenging to determine the two key parameters of phase space reconstruction: embedded dimension and delay time.

References

[1] De Gooijer J G and Hyndman R J 2006 *International Journal of Forecasting* **22** 443-473
[2] Yang H L and Lin H C 2017 *Computational Economics* **49** 99-116
[3] Zhu B Z and Wei Y M 2013 *Omega* **41** 517-524
[4] Deng S K, Yoshiyama K, Mitsubuchi T and Sakurai A 2015 *Computational Economics* **45** 49-89
[5] Zhang Y D and Wu L N 2013 *Expert Systems with Applications* **36** 8849-8854
[6] Vasilakis G A, Theofilatos K A, Georgopoulos E F, Karathanasopoulos A and Likothanassis S D 2012 *Expert Computational Economics* **42** 415-431
[7] Fletcher D and Goss F 1993 *Information and Management* **24** 159-167
[8] Leonard G, Alam P and Medy G R 1991 *Decision Science* **23** 896-916
[9] Navin K M 2018 *Deep Learning with Applications Using Python* 115-126
[10] Yan Y, Gao W C, Zhang B W, Yang C and Yin X C 2017 *Neural Processing Letters* **47**117-138
[11] Qi M and Wu Y 2003 *International Journal of Empirical Finance* **10** 623-640
[12] Zhang G P 2003 *Expert Neurocomputing* **50** 159-175
[13] Huang N E, Shen Z, Long S R, Wu M C, Shih H H, Zheng Q N,Yen N C, Tung C C and Liu H H 1998 *Proceedings of the Royal Society A* (London: Royal Society) **454**
[14] Huang N E, Wu M L C, Long S R, Shen S P S, Qu W D, Gloersen P and Fan K L 2003

Proceedings of the Royal Society A (London: Royal Society) **459**
[15] Jaeger H and Haas H 2004 *Science* **304** 78-80
[16] Zheng H L, Yuan J B and Chen L 2017 *Energies* **10**1-20
[17] Ina K, Ratnadip A and Ghanshyam V 2017 *Procedia Computer Science* **48** 173-179
[18] Ahmed T, Hesham H and Amr B 2017 *Future Computing and Informatics Journal* **2** 39-47
[19] Ha Y K and Chang H W 2018 *Expert Systems with Applications* **103** 25-37