A Novel Hybrid Short-Term Load Forecasting Method of Smart Grid Using MLR and LSTM Neural Network

Jian Li, *Senior* Member, *IEEE*, Daiyu Deng, Junbo Zhao, *Senior* Member, *IEEE*, Dongsheng Cai, Member, *IEEE*, Weihao Hu, *Senior* Member, *IEEE*, Man Zhang, Member, *IEEE*, Qi Huang, *Senior* Member, *IEEE*

***Abstract*—The short-term load forecasting is crucial in power system operation and control. However, due to its non-stationary and complicated random features, accurate forecast of load behavior is challenging. An improved short-term load forecasting method is proposed in this paper. At first, the load is decomposed into different frequency components varying from low to high levels realized by the ensemble empirical mode decomposition (EEMD) algorithm. Then, the smooth and periodic low-frequency components are predicted by the multivariable linear regression (MLR) method while maintaining efficient computation capacity, while the high-frequency components with strong randomness are forecasted by the long short-term memory (LSTM) neural network algorithms. Thus, the actual load behavior is obtained by combining these two methods. Finally, the proposed method is validated by experiments, in which the tested data from the west area of China, Uzbekistan, and PJM are used. The prediction of load behavior is accurate globally along with local details as presented in the experiments, which verify the effectiveness of the proposed method.**

***Index Terms*—Short-term load forecasting, ensemble empirical mode decomposition, long short-term memory neural networks, multivariable linear regression**

I. INTRODUCTION

Accurate load forecasting plays an important role in power system planning and operation, such as the development of unit commitment plan and making appropriate spinning reserve and maintenance plans [1][2]. With the development of the smart grid as well as the integration of distributed renewable energy resources and power electronics-based devices, the load behavior becomes more and more complicated, causing huge challenges to load forecasting approaches [3-5].

Load forecast can be divided into long-term forecasting, medium-term forecasting, and short-term forecasting. The medium and long-term load forecasting range from several weeks to several years. They are mainly used for long-term planning and seasonable demand analysis. By contrast, the short-term load forecasting (STLF), which ranges from minutes to one week ahead, is the crucial part of the daily operation of the power grid. Note that most loads have basic periodic features, while they are affected by many factors, such as temperature, the aging process of the devices, etc. Those factors bring strong uncertainties and non-periodic effects, which cause a decrease of forecast accuracy, especially in the case of an individual load. To mitigate the uncertainties in the short-term load forecasting process, many researches have been done. They can be categoried into two groups, namely the statistics analysis-based methods, such as multivariable linear regression (MLR), time series analysis method [6], exponential smoothing [7], and the machine learning algorithms-based methodologies, such as neural networks [8-11] and support vector regression [12][13].

For statistics analysis-based methods, MLR is fast to execute. In [14], the time series analysis method is proposed for short time load forecast. However, these methods do not yield good performances in the presence of large uncertainties. By contrast, machine learning algorithm, such as the support vector regression (SVR), has good generalization ability that allows obtaining the global solution in a fast manner. However, its scalability for the large dataset is limited. To solve that, the neural networks are proposed thanks to its powerful multivariable mapping capability. But the traditional back-propagation neural network easily traps into the local optimal solution and the internal influence of time series is not considered properly. To address that, recurrent neural network (RNN) [15-17] is introduced that is able to take into account the time series characteristics via the self-connection between hidden layers. But the gradient explosion problem of RNN [18] restricts its applicability. Inspired by [19], a long short-term memory (LSTM) neural network mitigating the issue is achieved by adding a special unit structure to RNN in [20].

Although machine learning algorithms have several advantages over the traditional statistics analysis-based ones and are widely used [21-23], it is challenging to establish appropriate hyper-parameters [24] and their computation speeds for large-scale datasets are slow. By contrast, for the dataset without complicated patterns, the time-series analysis is fast to

Manuscript received March 7, 2019; revised May 20, 2019, July 2, 2019, September 23, 2019, December 10, 2019, February 1, 2020, April 30, 2020; accepted June 1, 2020.This research was supported by the Natural Science Foundation of China (NFSC, Grant No. 51977025), and supported by the Science and Technology Support Program of Sichuan Province (2020YJ0251).

Jian Li, Daiyu Deng, Dongsheng Cai, Weihao Hu, Man Zhang, and Qi Huang are with the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731 China. (e-mail: leejian@uestc.edu.cn, hwong@uestc.edu.cn)

Junbo Zhao is with the Mississippi State University, Starkville, MS, USA, USA (email:zjunbo@vt.edu).

execute with good accuracy. For the practical electric load [25], its behavior is usually non-stationary and contains complicated random patterns. If only machine learning-based algorithms are used for load forecasting, the computational burden might be high. While if only statistics-analysis based methods are used, some complicated load patterns may not be considered, resulting in decreased forecasting accuracy. To deal with that, this paper proposes a decomposition-based hybrid forecasting method that integrates the advantages of both time-series and machine learning algorithm. In the literature, the most common decomposition methods for time series data are wavelet transform, seasonal-trend decomposition procedure based on Loess (STL), and empirical mode decomposition (EMD). The decomposition results of wavelet transform for non-stationary sequence depend heavily on the choice of basis function and order, which is not easy to tune for the actual prediction process. For STL, when the data itself has a good seasonal periodicity, it is an ideal method. However, that is difficult to satisfy for practical load data as it has strong randomness. Compared with wavelet transform and STL, EMD decomposition is an adaptive method that is well suited for processing nonlinear and non-stationary time series. Thus, a EMD-based method is adopted to decompose the electric load in this paper.

In summary, this paper has the following contributions:

- The ensemble empirical mode decomposition (EEMD) algorithm is adopted to decompose the electric load into different frequency components varying from low to high levels. These components have linear correlation and can be superposed linearly. Therefore, the prediction results obtained by LSTM and MLR for different components can be treated separately. This significantly improves computational efficiency for large data using distributed computing technique;
- MLR method and the long short-term memory (LSTM) neural network algorithms are used to forecast respectively the loads corresponding to low and high-frequency components. The MLR allows to predict the overall trend of change in a fast manner while the LSTM enables the prediction of complex nonlinear local behaviors. In such a way, both the advantages of MLR and LSTM can be effectively taken by the proposed approach, yielding significantly improved results.

This paper is organized as follows: Section II presents the framework of the proposed method, where the way of using EEMD, MLR, and LSTM for load forecasting in a unified manner is shown. Section III shows and analyzes the test results and finally, Section IV concludes the paper.

II. Proposed Hybrid Short-term Load Forecasting Approach

The structure of the proposed method is shown in Fig. 1. In this paper, ELM is an abbreviation for EEMD-LSTM-MLR. It has four major components, namely load decomposition via the EEMD, the load forecasting via MLR for low-frequency load part, the load forecasting via LSTM for high-frequency load part, and the load forecasting reconstruction through the superimposition rule.

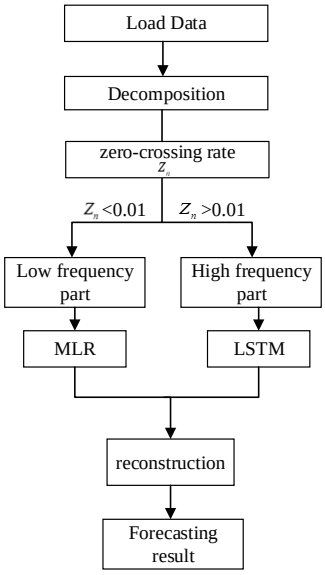


Fig. 1 The framework for the proposed ELM method.

It should be noted that the zero-crossing rate of a signal is defined as follows：

$$Z_n = \frac{n_{zero}}{N} \tag{1}$$

where Z_n represents the zero-crossing rate, n_{zero} represents the number of zero crossings, and N represents the signal length. In this paper, components with a zero-crossing rate of less than or greater than 0.01 are defined as the low-frequency and high-frequency components, respectively. The threshold selection of the zero-crossing rate needs to ensure the validity of the MLR algorithm. For some high-frequency components, the iterative prediction results of MLR may be divergent, but the LSTM neural network does not. On the other hand, the division of the components of the intermediate frequency does not have a large impact on the overall prediction result. Therefore, according to the experience of multiple groups, the zero-crossing rate is chosen to be 0.01, which can effectively divide the low-frequency components.

A. Load Decomposition

Electric load consumption is affected by many complicated factors, yielding non-stationary features. As a result, it is challenging for the traditional forecasting method. To address that, EEMD approach [26][27] is adopted to decompose the load data. EMD is one of typical adaptive data processing tools, which could decompose the signal according to its own time-scale features. It decomposes the signal into different intrinsic mode functions (IMFs) with frequency varying from high to low. The low-frequency part represents the overall trend of the load periodicity, while the higher frequency part represents the stronger randomness of the local features and the noise. Each IMF contains the local feature of the signals of different time scales.

EEMD is an improvement of EMD. It processes the signal by adding white noise. The added white noise would populate

the whole time-frequency space uniformly with the constituting components of different scales. Due to the features of zero-mean noise, after several averaging, the noise will cancel each other out and a more reasonable integrated mean value result will be obtained to achieve the intrinsic mode decomposition of the electric load. The EEMD utilizes all the statistical characteristic of the noise: it helps perturb the signal and enable the EMD algorithm to transverse all possible solutions in the finite neighborhood of the true answer; it also takes advantage of the zero mean of the noise to cancel out this noise background once it has served its function of providing the uniformly distributed frame of scales, a feat only possible in the time domain data analysis. In a way, EEMD is essentially a controlled repeated experiment to produce an ensemble mean for a non-stationary data as the final answer. Since the role of the added noise in the EEMD is to facilitate the separation of different scales of the inputted data without a real contribution to the IMFs of the data, the EEMD is a truly noise-assisted data analysis (NADA) method that is effective in extracting signals from the data.

Specifically, the use of EEMD for the time series load signal can be done via the following steps:

1) Add white noise to the original electric load data, where the standard deviation of white noise is 0.2 times [28] of the standard deviation of the signal, and the proper adjustment is made according to the high or low frequency of the principal component of the signal. The added white noise would uniformly distribute in the whole time-frequency domain with the constituting components of different scales [28]. And due to the features of zero-mean noise, after several averaging, the noise will cancel each other out and a more reasonable integrated mean value result will be obtained to achieve the intrinsic mode decomposition of the electric load.
2) Decompose the noisy data into several IMFs using EMD as shown below:

$$s(n)=\sum_{i=1}^{M}c_i(n)+r(n) \tag{2}$$

where $s(n)$ represents the electrical load, $c(n)$ represents different load components, and the decomposition residual is $r(n)$.

3) Repeat steps 1 and 2 for N times with different white noise sequence added to the signal. If N is too small, the effect of white noise may not be fully offset, while if N is too large, the computational burden will be increased. In this paper, the strategy recommended by [28] is adopted to determine N.
4) Obtain the means of corresponding IMFs as the final result once the maximum N is reached.

EMD firstly calculates the intrinsic mode by the feature time scale of the data and then decomposes it, which called “sifting” process. The specific steps are as follows:

1) Find all extreme points of electric load data $s(n)$, the upper and lower envelope curves $e_{max}(n)$ and $e_{\min}(n)$using the cubic spline interpolation function, and calculate the mean $m(n)$:

$$m(n)=\frac{e_{max}(n)+e_{min}(n)}{2} \tag{3}$$

2) Subtraction of $m(n)$ from $s(n)$:

$$d(n)=s(n)-m(n) \tag{4}$$

3) Use $d(n)$ to calculate the SD, where SD is the standard deviation:

$$SD=\sum_{n=0}^{N}\left[\frac{\left|d_{i-1}(n)-d_i(n)\right|^2}{d_{i-1}^2(n)}\right]<\sigma \tag{5}$$

when the value of *SD* is less than a predefined value σ whose empirical value is generally 0.2 to 0.3 [29], let $c_j(n)=d(n)$ and get an IMF component $c_j(n)$ after this round of "sifting", where $j = 1, 2, 3,\cdots, h$, and h represents the total number of IMF components. Otherwise, with $d(n)$ as the input signal, return to step 1 for the next round of "sifting" until *SD* is less than σ after multiple rounds.

4) Let $s(n)=s(n)-c_j(n)$. If the resulting $s(n)$ is a monotonous function, or when $s(n)$ or $c_j(n)$ is less than or equal to a given value, the residual $r(n)=s(n)$ and the iteration is stopped. Otherwise, continue to the next iteration with the resulting $s(n)$ and return step 1 to calculate the next IMF component $c_{j+1}(n)$. By analogy, the original signal $s(n)$ can be finally decomposed into the following IMFs $c_1(n), c_2(n), \cdots, c_h(n)$with the residual $r(n)$.

B. Multivariable Linear Regression

After decomposition via the EEMD method, the low-frequency components that are smooth and periodic can be obtained. For this type of signal, the MLR can be easily implemented to perform the load forecasting. MLR is a traditional method that is fast to execute and has good accuracy when the signal patterns are simple [30]. Compared with neural networks and SVR, the training process speed of MLR has obvious advantages. At the same time, for periodic strong and smooth curves, MLR is easier to obtain accurate prediction values than neural networks and SVR. It is similar to that of a neural network using a linear function as the activation function, but it does not require cumbersome iterative training processes and parameter adjustments. Therefore, for smooth low-frequency load components, MLR is a more suitable choice than other methods. Its mathematical model can be represented as follows:

$$Y=X\times\beta+\mu \tag{6}$$

$$\begin{bmatrix}y_1\\y_2\\\vdots\\y_n\end{bmatrix}=\begin{bmatrix}1&x_{11}&\cdots&x_{1n}\\1&x_{21}&\cdots&x_{2n}\\\vdots&\vdots&\ddots&\vdots\\1&x_{n1}&\cdots&x_{nn}\end{bmatrix}\times\begin{bmatrix}\beta_0\\\beta_1\\\vdots\\\beta_n\end{bmatrix}+\begin{bmatrix}\mu_1\\\mu_2\\\vdots\\\mu_n\end{bmatrix} \tag{7}$$

where y_i is the value of load data, x_{ij} represent the factors that affect the load, β_0 is a constant, β_i (i=1,2…,n) is the regression coefficient, μ_i is the random disturbance. The solution of (6) can be easily obtained by using least squares method, yielding:

$$\hat{\beta}=(X^TX)^{-1}X^TY \tag{8}$$

To illustrate that, a representative example is given below. The objective is to predict the last 96 points of a group of smooth data set. In this paper, the mean absolute percentage error (*MAPE*) and the root mean square error (*RMSE*) are used

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3000184, IEEE Transactions on Industrial Informatics

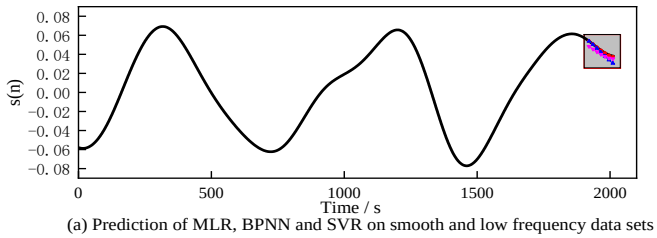
to evaluate the performance, and the forecasting accuracy is assessed by the relative error and the absolute error. The equations of *MAPE* and *RMSE* are shown as follows:

$$MAPE=100\%\times\frac{1}{N}\sum_{T=0}^{N}\frac{\left|P_{real}-P_{forecast}\right|}{P_{real}} \quad (9)$$

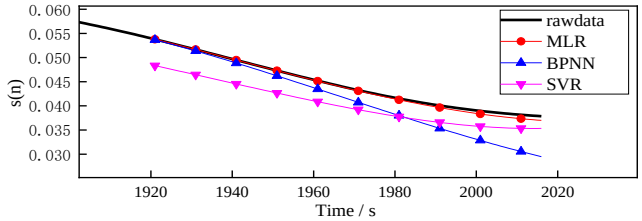
$$RMSE=\sqrt{\frac{\sum_{T=0}^{N}\left(P_{real}-P_{forecast}\right)^{2}}{N}} \quad (10)$$

where P_{real} and P_{forecast} are real and forecasted values respectively, N is the length of data set.

For the signal shown in Fig. 2(a), the performance of each method is shown in Table I. It can be found that MLR outperforms BPNN and SVR, with the MAPE of 0.638%, and the RMSE of 0.00038. Unlike the BPNN and SVR that need careful tunings, MLR is easy to be implemented and yields good results. As a result, its performance for low-frequency load part forecasting is validated.



(a) Prediction of MLR, BPNN and SVR on smooth and low frequency data sets



(b) Partial enlargement

Fig. 2 Illustration results of MLR.

TABLE I. PREDICTION ERROR OF EACH MODEL ON LOW-FREQUENCY COMPONENTS

	MLR	**BPNN**	**SVR**
MAPE	0.638%	7.400%	9.175%
RMSE	0.00038	0.00389	0.00420

C. LSTM Neural Network

For the decomposed components with high frequency and strong randomness, LSTM neural network is adopted for load forecasting. For the traditional neural network, the model does not care about what information will be used for the next moment and it only pays attention to the processing of the current moment each time. The basic steps of the back-propagation neural network include 1) output calculation from each neuron, 2) assessment of the error of each neuron output in backward and 3) reducing the error through optimization algorithms. Note that for time series load data, there are hidden features and if the time correlation of load is neglected, the accuracy of forecasting is degraded. Therefore, a LSTM taking into account the temporal correlations of the time series is developed for load forecasting.

Compared with the traditional RNN whose hidden layer has only one state (h), LSTM introduces the concept of a cell state (c) to takes into account the time correlation hidden in the long-term state. The structure of the LSTM is shown in Fig. 3. At time t, the input of LSTM includes x_{t} of the network at the current time, $h_{\mathrm{t-1}}$ of the previous time, and the cell state $c_{\mathrm{t-1}}$ of the previous time. The LSTM output includes the output value h_{t} for the current time and the cell state c_{t} for the current time. LSTM has three gates to control the cell state, namely forget gate, input gate, and the output gate. They can be described as,

$$g(x)=\sigma(Wx+b) \quad (11)$$

where W is the weight matrices of the gate, x is the input load data, b is the bias vectors, σ is the sigmoid function. The range of the latter is (0, 1), therefore, the gate's output vector is between (0, 1). When the gate's output vector is 0, the product of the control vector and the output vector is also 0, that is, the gate cannot be passed. Whereas, when the output is 1, the gate can be passed. Eq. (11) shows that the gate is always half open and half closed. This enables LSTM neurons to effectively control the amount of transmitted information, retain effective information in load time series, and remove invalid information.

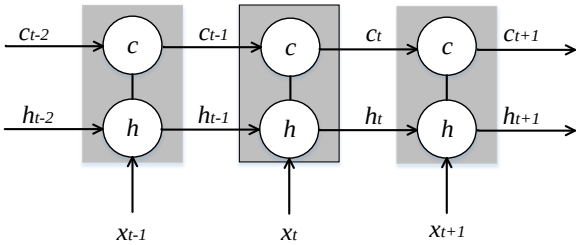


Fig. 3 The structure of LSTM in time form.

In the process of backward optimization, the weights of LSTM are updated by adaptive moment estimation (Adam) optimization algorithm [31]. An appropriate optimization algorithm is important, it can save computation resources and speed up the convergence to the optimal solution. Adam is a first-order optimization algorithm that can replace the traditional gradient descent process. It has high computational efficiency and requires less memory when solving optimization problems with large-scale data and parameters. It also has high robustness when facing high noise or sparse gradients. After many cycles, the output value of LSTM gradually approximates the true value on the training set to obtain an effective forecasting model.

LSTM neural network iteratively updates the weights to reduce errors and extracts the inherent rules of high-frequency time series to improve the forecasting accuracy of the model. It can gradually approximate high-frequency data with strong randomness. The prediction result of LSTM neural network is closely related to the setting of its hyper-parameter, including the number of layers in the network, the number of nodes in each layer, the number of cycles and the initial weights [32]. In this paper, they are tuned according to the rules introduced in [20].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3000184, IEEE Transactions on Industrial Informatics

D. Load Forecasting Reconstruction

According to eq. (2), all load components with coefficients 1 are in the same time coordinate, and EEMD is a linear decomposition, so it is only necessary to superimpose them in order to reconstruct the complete load curve. In addition, as the components obtained by EEMD have relatively stable frequencies, the prediction results obtained by LSTM and MLR can be directly superimposed via the following equation to obtain the final result:

$$P_{Load} = P_{IMF1} + P_{IMF2} + P_{IMF3} + \cdots + P_{IMFn} \quad (12)$$

where P_{Load} represents the sequence of the predicted results and P_{IMFi} represents the sequence of the predicted results of the *i*-th component, where $i = 1,2,\cdots,n$, and n is determined by the load data itself and the threshold set by the EEMD decomposition. Furthermore, since the large data set is divided into different components that can be treated separately, the computational burden can be mitigated by using distributed computing technique.

The superposition process from low frequency to high frequency is also a process of reconstruction from the overall trend to the local details. Finally, the forecasting of an electric load is achieved by combining the results of low-frequency empirical mode load forecasted based on MLR with those of the rest of frequencies based on LSTM.

III. EXPERIMENTAL RESULTS

A. Load Data Analysis

Field load data of 5000 users from west area of China are utilized for the case study varying from April 8th, 2017 to April 28th, 2017. The data is updated every 15 minutes, which means that each user has 96 data per day. Like the transfer learning, a pre-training model is first obtained from the data of 5000 users. The raw data of user's load are shown in Fig. 4, which present characteristics without obvious periodicity. In the paper, the historical data are divided into two parts: the first 90% of the historical data are used as the training set and the left 10% are used as validation set. The test set is from the real load data from users. All test errors are calculated between the prediction and reality.

When load forecasting is implemented, the historical data of previous days are batched together to train the model and forecast the load data in the following day. For example, if the historical data from April 8th to April 27th are taken as the training set in the model, the electric load for the whole day of April 28 can be predicted by iterative forecasting. Note that Python Keras is used for model building and data training in this paper.

B. EEMD-LSTM-MLR Model Prediction on Medium Data Sets

It is clear in Fig. 4 that the raw data of one user load consumptions contain a large number of high-frequency components and noise, etc. To deal with that, the load data are decomposed by EEMD first. The IMFs of the frequency from high to low are divided into 11 groups depending on the complexity of the data and the EEMD pre-threshold settings as shown in Fig. 5. According to the EEMD results, no obvious mode mixing phenomenon is found in each intrinsic mode function. The zero-crossing rate of each IMF component can be obtained as shown in Fig. 6 by eq.(1). It can be seen that the zero-crossing rate starts to be less than 0.01 from IMF7. The residual, as well as the modes from IMF7 to IMF10, are very smooth. Then, the MLR is used to forecast the loads corresponding to the low-frequency modes. By contrast, modes from IMF 1 to IMF 4 have a very high frequency. Thus, the LSTM neural network is adopted for load forecasting. The whole framework of the proposed method is shown in Fig. 7.

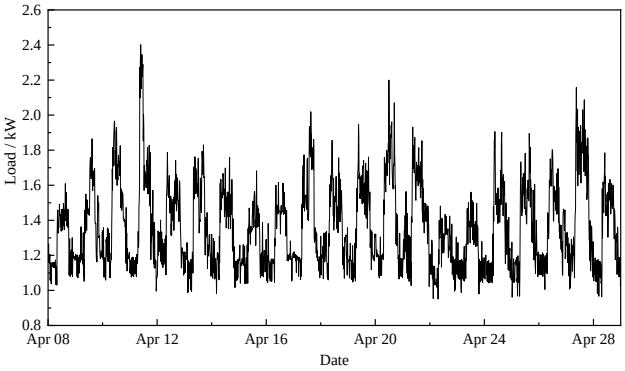


Fig. 4 Raw data of one user.

The IMFs and residual obtained after EEMD are trained into the corresponding algorithms, and the data of the next 96 time points are forecasted iteratively. The resulting forecasted series are summed to obtain the complete load forecasting. In particular, each forecasting sequence is superimposed in order varying from low frequency to high frequency, and the complete sequence can be gradually obtained.

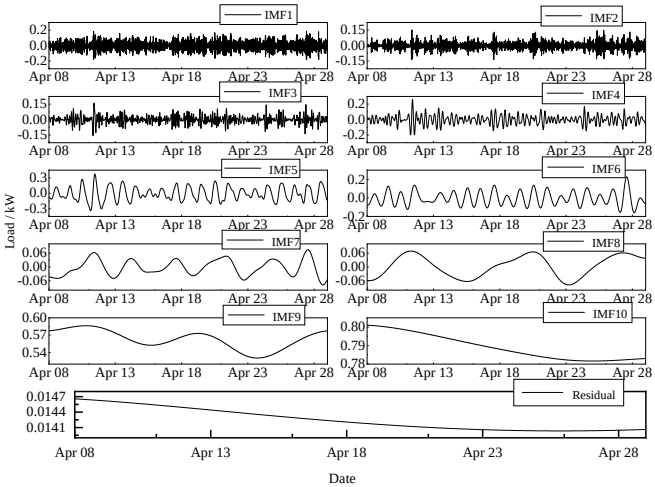


Fig. 5 The mode decomposition results of EEMD.

Table II lists the forecasting errors, where conclusions can be drawn as that the prediction accuracy is gradually improved by accounting both medium and high frequencies. According to eq. (9) and (10), the final *MAPE* is 5.45% and *RMSE* is 0.0943, respectively. It is interesting to notice that the LSTM neural network is able to capture the local details of the load time series. This allows the LSTM neural network to achieve good forecasting accuracy for complex high-frequency time series. *MAPE* of Pre5-11 is 6.34%. This demonstrates that MLR is suitable for load forecasting of low-frequency parts.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3000184, IEEE Transactions on Industrial Informatics

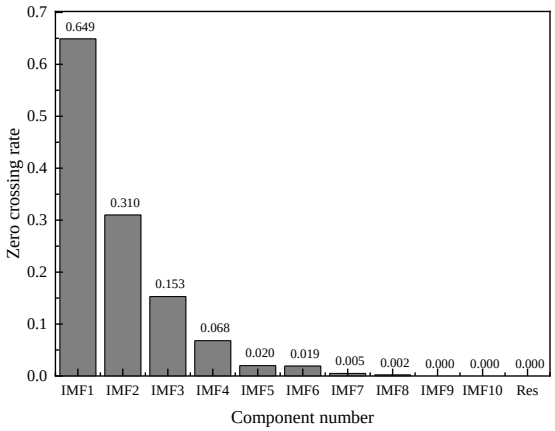


Fig. 6 Zero-crossing rate of IMFs.

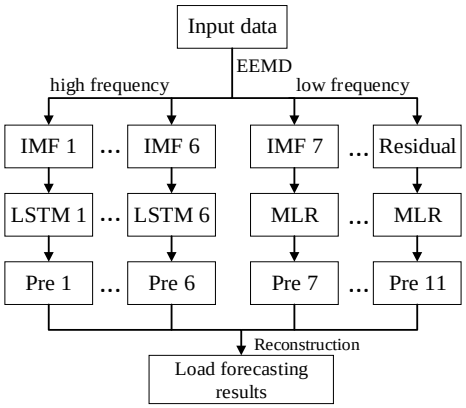


Fig. 7 The proposed model framework of ELM.

TABLE II. THE ERROR OF THE LAST FIVE SUPERPOSITION PROCESSES

	Pre5-11	**Pre4-11**	**Pre3-11**	**Pre2-11**	**Pre1-11**
MAPE	6.34%	5.69%	5.53%	5.47%	5.45%
RMSE	0.1080	0.0949	0.0946	0.0945	0.0943

To further demonstrate the benefits and performance of the proposed method, several prediction algorithms are investigated for comparisons, including the support vector regression (SVR), the back-propagation neural network (BPNN), the MLR, and LSTM neural network, and a multi-input LSTM neural network (MLSTM) that takes into account temperature, wind direction, wind speed, day of the week, and historical load. Their prediction curves and errors are shown in Fig. 8, Fig. 9 and Table III, respectively.

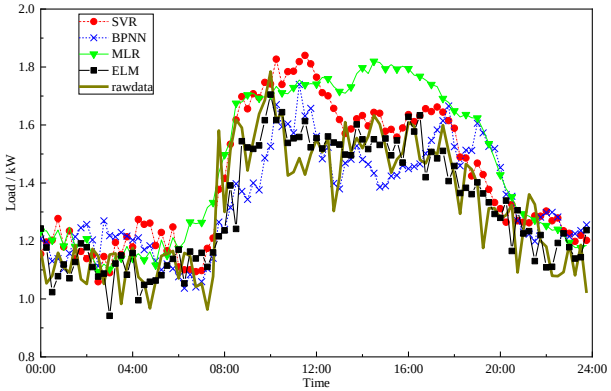


Fig. 8 Comparison of ELM model and common prediction methods.

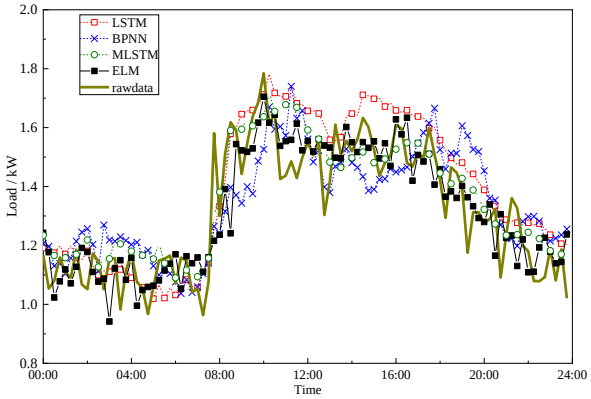


Fig. 9 Comparison of ELM model and neural network prediction methods.

TABLE III. THE ERROR OF EACH MODEL

	MLR	**SVR**	**BPNN**	**LSTM**	**MLSTM**	**ELM**
MAPE	11.96%	9.03%	8.70%	8.12%	6.57%	5.45%
RMSE	0.1866	0.1467	0.1399	0.1304	0.1061	0.0943

SVR is a common prediction algorithm for processing high-dimensional data. The radial basis function (RBF) with parameters C =50 and σ =0.01 are used. BP neural network uses the same sigmoid function as LSTM for developing the activation function. It contains two hidden layers with numbers of neurons 96 and 48 respectively. Note that more layers may lead to better prediction results but also needs more computation time. Hence, only two hidden layers are chosen to lighten the computational burden without sacrificing the accuracy of ELM. To make a fair comparison, the number of hidden layers of LSTM and MLSTM is the same as ELM. It shows that the LSTM neural network achieves better performance than the BP neural network for sequence prediction. Besides, it should be noted that in this example, the consistency of the results obtained by LSTM is poor. The best *MAPE* value is 6.57%, but the result is not reproducible. It means that the algorithm is difficult to accurately match the data with a large number of high-frequency components, yielding local optimal solutions. Through the adjustment of the hyper-parameters, the prediction error is finally stable around 8%. In summary, the proposed approach outperforms the other two alternatives as they fail to capture the high-frequency components. In particular, the ELM model not only has a significant improvement in MAPE and RMSE over the LSTM, which are 32.9% and 27.7% respectively, but also achieves much better performance in fitting the time series load curve as shown in Fig. 9. Specifically, there is no effective fitting for each method at point 10:00 to 18:00, but the ELM model achieves accurate prediction to certain degree. Moreover, except for the proposed ELM method, the other four methods have poor performance in predicting the load from 00:00 to 04:00. Finally, it is found that even with less hidden layers, the proposed method achieves high computational efficiency without sacrificing forecasting accuracy. This means that the proposed method does not need to involve too many complicated model selection process while still yielding very good performance, which is the advantage over LSTM and MLSTM.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3000184, IEEE Transactions on Industrial Informatics

C. Case Studies on Big Data Sets

In the previous case, the proposed ELM model achieves good results in the case of medium data sets. To further validate the effectiveness of the method on big data sets, load data of three users from Uzbekistan are added. The data of the first set is shown in Fig. 10. The time range is from 2017-04-16 20:00:00 to 2018-07-31 15:00:00, and the sampling interval is 30 minutes. Similar to the previous case, the 24-hour loads of three sets of loads varying from 2018-07-30 15:00:00 to 2018-07-31 15:00:00 are iteratively predicted, yielding 48 points in total. It can be seen from Fig. 10 that the data set has seasonal characteristics. The region around December is the peak period of electricity consumption in a year, and the load has large oscillation characteristics.

By using the EEMD method, the modes are extracted and shown in Fig. 11. Due to space limitation, only the load curve and EEMD decomposition diagram of the first group of data are given. The EEMD first decomposes the load data with violent oscillations into a number of relatively stable components. It can be seen that the last component (IMF13) is roughly similar to the seasonal variation of the load.

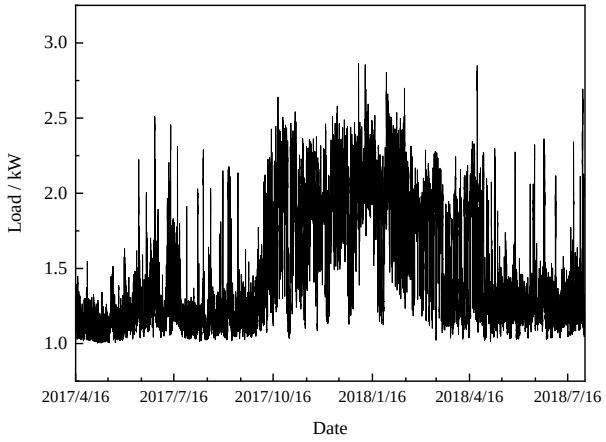


Fig. 10 The first group of load data in a certain area of Uzbekistan.

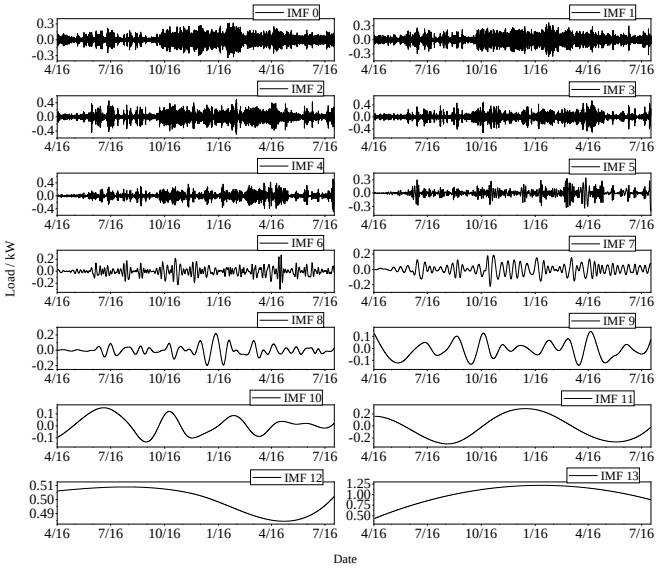


Fig. 11 EEMD decomposition results for the first group of loads.

The prediction results for the three groups of load data are shown in Fig. 12 and Table IV, respectively. LSTM neural network is used as a comparison method. From the Fig. 12, it can be seen that the three groups of loads have different characteristics, and ELM has better prediction results than the LSTM neural network.

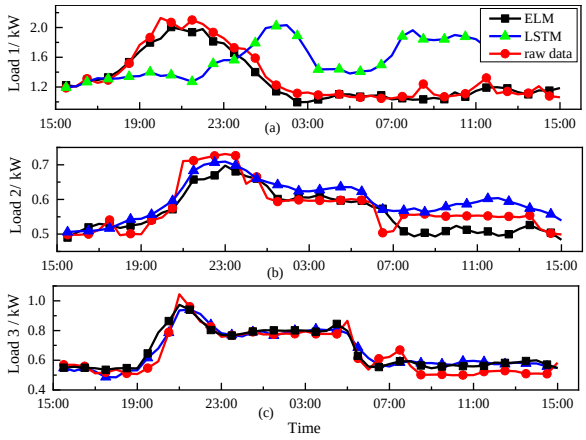


Fig 12 Three groups of load forecasting result in a certain region of Uzbekistan.

In the first group of load data, the ELM model is far better than the LSTM neural network. The latter is difficult to make effective prediction results on this load data as it can not capture well the local time series details. The MAPE error of LSTM is as high as 40.09%, while that of ELM is only 4.85%. This means the LSTM neural network cannot produce a good performance in the case of big data if not all necessary features are appropriately considered. For example, unstable data sets will have serious impacts on the LSTM neural network, making it difficult to form an accurate prediction model. However, ELM method decomposes the unstable seasonal factors and uses the advantages of LSTM and MLR to address different load characteristics.

TABLE IV. PREDICTION ERROR OF THREE GROUPS OF LOADS IN A CERTAIN AREA OF UZBEKISTAN

	Load 1		**Load 2**		**Load 3**	
	LSTM	**ELM**	**LSTM**	**ELM**	**LSTM**	**ELM**
MAPE	40.09%	4.85%	5.47%	4.86%	7.54%	7.17%
RMSE	0.5731	0.0883	0.0373	0.0362	0.0561	0.0541

In the case of the second group load and the third group load, the prediction results of ELM and LSTM are relatively close. This is because the load characteristics are not complex and LSTM is able to capture most of them. Anyway, the ELM still does a better job in capturing all necessary time series load characteristics and yields better results. This can be validated by the MAPE and RMSE errors of the two methods.

D. Case Studies on PJM power system

The data for the last two cases are from relevant institutions and companies, which are not public data sets. In order to reflect the prediction accuracy of the method from the horizontal direction, a set of hourly load data are added from PJM, which is a famous benchmark. According to the data source information, the MARKET REGION is WEST and the LOAD AREA is DAY. The time is from 2018-4-30 00:00 to 2019-4-30 00:00,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3000184, IEEE Transactions on Industrial Informatics

and the sampling interval is one hour. The specific data is shown in Fig. 13.

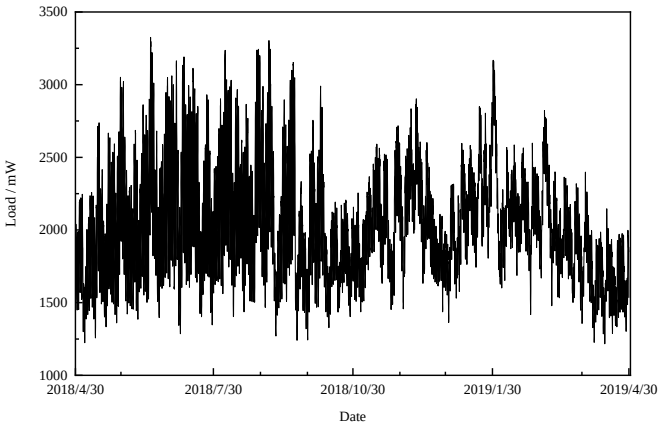


Fig 13 PJM hourly metering load data.

Similarly, the data set is disposed by a sliding window to generate a number of sub-time series, and corresponding features and labels are obtained. The length of the sub-sequence is 24, corresponding to the length of one day. The forecast is developed from two aspects, 1) the same as the previous two cases using time series-based iterative prediction and 2) the last 10% is used as a validation set for prediction while the first 90% is used as a training set. The load is decomposed by EEMD to obtain 13 components as shown in Fig. 14. To elaborate on the training and validation procedures of LSTM neural network, Fig. 15 shows the loss trend of it on IMF0 component, where the MSE criterion is used. It can be observed that the loss values of the validation set and the training set tend to be stable, and the same one after 100 epochs. As a result, no over fitting or under fitting in the training is observed.

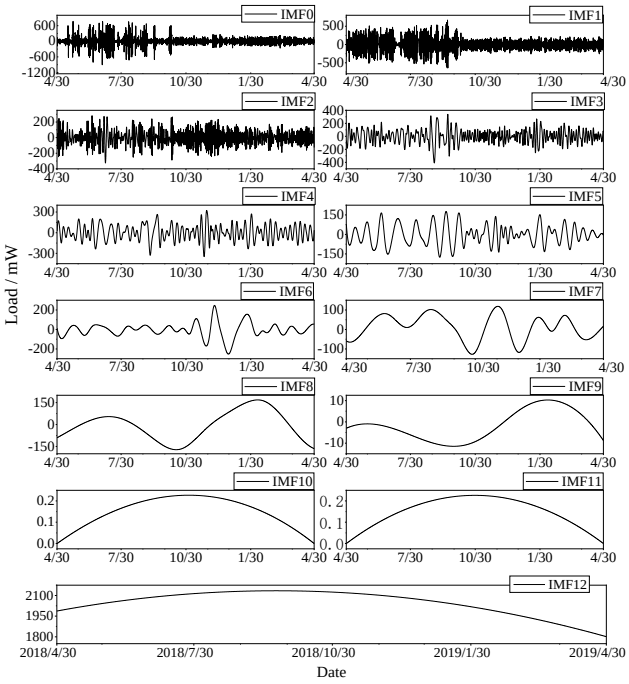


Fig 14 EEMD decomposition results of PJM data.

The prediction results based on iterative prediction are shown in Fig. 16 and the corresponding errors are shown in Table V. In the results based on iterative prediction, the MAPE of the ELM is only 2.16%, and the MAPE on the validation set is only 0.91%, indicating the accuracy and effectiveness of the ELM method.

***Remark*:** to further validate the motivation of integrating MLR with LSTM instead of using the latter only to achieve computational efficiency improvement, the PJM data set is used as an example. All testes are run on a computer with win7 64 bits, CPU i3-6320, memory 8G and the Python 3.5.4, Keras 2.1.2 is used. Other settings include two hidden layers of LSTM neural network with 48 and 24 neurons, respectively, and the iteration parameter epochs 200. The total time of LSTM training and iteration prediction is 19.78 seconds, and the error of MAPE is 0.217%, while MLR only takes 0.13 seconds and its MAPE value is only 0.016%. If the hidden layer of LSTM is reduced to one and the number of neurons is 10, the training and iterative prediction time is reduced to 7.72 seconds with MAPE of 1.190%. As a result, the computational efficiency of MLR is much higher than that of neural network algorithm. For medium and low frequency time series data, MLR achieves much better computational efficiency over LSTM in training and prediction as well. The limitation of MLR is that it is not fit for the data with high frequency randomness. This weakness has been overcome by using the proposed hybrid approach that integrates the advantages of MLR and LSTM.

Moreover, the correlations of different IMFs with their original signal have been studied for Figs. 5, 11, 14 and the results are displayed in Figs. 17-19, respectively. According to the correlation results, it can be concluded that the obtained IMFs show strong dependences on the original signal, validating the effectiveness of the EEMD method. Note that the correlation patterns shown here are similar to those shown in [28].

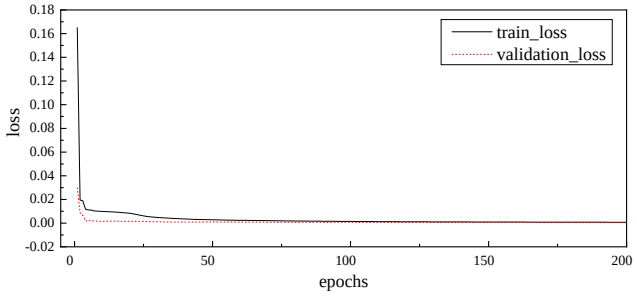


Fig 15 Loss trend of LSTM neural network on IMF0 component.

TABLE V. PREDICTION ERROR OF PJM LOAD DATA

	LSTM	ELM	VALIDATION
MAPE	6.61%	2.16%	0.91%
RMSE	136.41	50.257	20.463

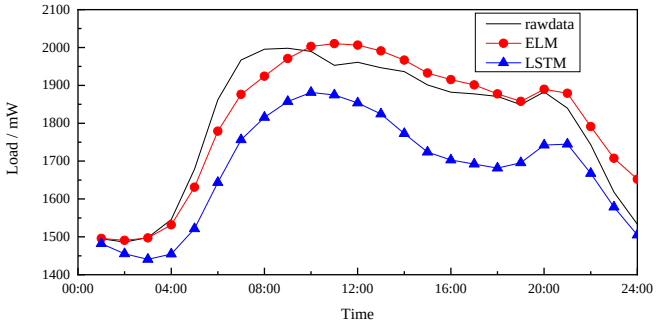


Fig 16 Iterative prediction-based results for PJM load data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2020.3000184, IEEE Transactions on Industrial Informatics

Discussions: In this paper, EEMD algorithm has been adopted to decompose the time series electric load into different components that have different frequencies varying from low to high levels. Experimental results using realistic field data have validated the its effectiveness and high accuracy in load forecasting as compared to other literatures. Since every method has both advantages and disadvantages, some possible issues of EEMD algorithm, i.e., (1) The EEMD may produce results that do not satisfy the strict definition of IMF. (2) The EEMD may have decreased efficiency in dealing with multi-mode distribution of the IMFs. (3) The end effect and the stoppage criteria of EEMD need to be carefully investigated.

For (1), it has been validated by [28] and others, although there are deviations from strict IMFs, they are small for the load forecasting applications, take the IMF residuals for example. For (2), according to the results shown in this paper, mix-mode phenomena does not happen and its impact on the load forecasting results is minor. While for (3), it is still an open problem. However, it is worth pointing out that the purpose of electric load decomposition by EEMD is to obtain different signal components with different frequencies varying from low to high levels. To this end, the MLR method and LSTM neural network algorithms can be used to forecast respectively the loads corresponding to low and high-frequency components. The MLR allows predicting the overall trend of change in a fast manner while the LSTM enables complex nonlinear local behaviors to be predicted. In such a way, both the advantages of MLR and LSTM can be effectively integrated to obtain significantly improved forecasting results. Although (2) and (3) may yield conservative decomposition results, MLR and LSTM methods can still achieve a balance in dealing with components with different frequencies. In particular, more components can be assigned to LSTM as it has powerful capabilities in dealing with complex signals. The test results have clearly demonstrated the integrated MLR and LSTM in obtaining much accurate results than other methods.

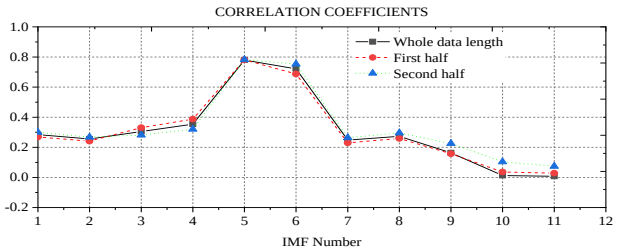


Fig. 17. The correlation coefficients of the original signal in Fig. 4 and their corresponding IMF components in Fig. 5. IMF 11 here means the Residual components.

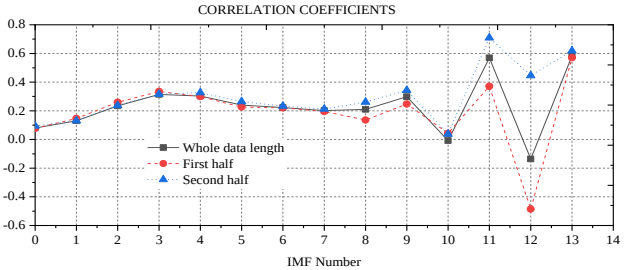


Fig. 18. The correlation coefficients of the original signal in Fig. 10 and their corresponding IMF components in Fig. 11.

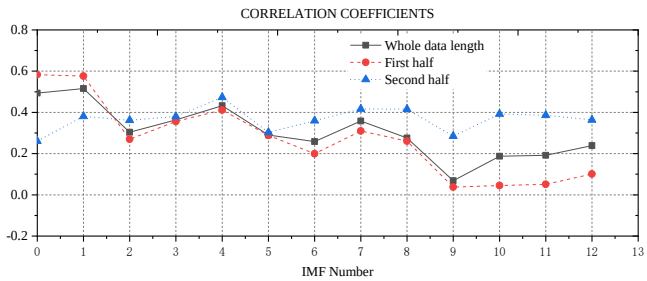


Fig. 19 The correlation coefficients of the original signal in Fig. 13 and their corresponding IMF components in Fig. 14.

IV. Conclusion

In this paper, an improved short-term load forecasting method, named ELM is proposed by taking both the advantages of MLR and LSTM algorithms. The basic principle of the load forecasting method is first introduced, where its compositions are investigated. Then, the proposed method is tested both in case of medium data sets and big data sets. ELM is compared to SVR, BPNN and other neural network-based prediction methods, such as LSTM and MLSTM. The prediction accuracy of EML is superior to all other methods. Tests on the data from PJM power system are also reported and it is found that the maximum improvement of ELM is up to 67% in MAPE compared to LSTM.

Appendix A

Pseudo-code of Decomposition Algorithm

Input: Number of ensembles *M*, the amplitude of white noise *k* and the iterations of the experiment *i*.

Output: *n* IMF components and one remaining component.

1: Initialize the population $i_m(m=1,2,\ldots n)$

2: Repeat

3: $i=0$

4: **If** $i<max$ M do

5: Add white noise to the original data $X_i(t) = x(t) + k \cdot n_i(t)$,

6: Decompose $X_i(t)$ according to the EMD method

7: **End if**

8: Calculate the means of corresponding IMFs as result

Pseudo-code of Load Reconstruction Algorithm

Input: The number of IMFs n, the predicted results P_IMFi corresponding to each IMF, Residual.

Output: Load prediction P_{Load}.

1: Initialize the load $P_{Load}=0$

2: Add predicted results of the IMF component separately

$P_{Load} = \sum_{i=0}^{n} P_{IMFi} + Residual$

References

[1]. B. A. Høverstad, A. Tidemann, H. Langseth, and P. Özturk, "Short-term load forecasting with seasonal decomposition using evolution for parameter tuning," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1904-1913, July 2015.

[2]. Y. Hsiao, "Household electricity demand forecast based on context information and user daily schedule analysis from meter data," *IEEE Trans. Ind. Informat*, vol. 11, no. 1, pp. 33-43, Feb. 2015.

[3]. X. Chen, Y. Hou, and S. Y. R. Hui, "Distributed control of multiple electric springs for voltage control in microgrid," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1350-1359, May 2017.

[4]. C. E. Borges, Y. K. Penya and I. Fernandez, "Evaluating combined load forecasting in large power systems and smart grids," *IEEE Trans. Ind. Informat*, vol. 9, no. 3, pp. 1570-1577, Aug. 2013.

[5]. X. Chen, M. Shi, H. Sun, Y. Li, and H. He, "Distributed cooperative control and stability analysis of multiple DC electric springs in a DC microgrid," *IEEE Trans. Ind. Electron*, vol. 65, no.7, pp. 5611-5622, 2018

[6]. E. Paparoditis, and T. Sapatinas, "Short-term load forecasting: the similar shape functional time-series predictor," *IEEE Trans. Power Syst*, vol. 28, no. 4, pp. 3818-3825, Nov. 2013.

[7]. Kyung-Bin Song, Seong-Kwan Ha, Jung-Wook Park, Dong-Jin Kweon, and Kyu-Ho Kim, "Hybrid load forecasting method with analysis of temperature sensitivities," *IEEE Trans. Power Syst*, vol. 21, no. 2, pp. 869-876, May 2006.

[8]. H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Trans. Power Syst*, vol. 16, no. 1, pp. 44-55, Feb 2001.

[9]. H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Trans. Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 303-315, Feb. 2014.

[10]. M. Rafiei, T. Niknam, J. Aghaei, M. Shafie-khah, and J. P. S. Catalão, "Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6961-6971, 2018.

[11]. F. Y. Xu, X. Cun, M. Yan, H. Yuan, Y. Wang, and L. L. Lai, "Power market load forecasting on neural network with beneficial correlated regularization," *IEEE Trans. Ind. Informat*, vol. 14, no. 11, pp. 5050-5059, Nov. 2018.

[12]. Bo-Juen Chen, Ming-Wei Chang, and Chih-Jen lin, "Load forecasting using support vector Machines: A study on EUNITE competition 2001," *IEEE Trans. Power Syst*, vol. 19, no. 4, pp. 1821-1830, Nov. 2004.

[13]. C.-N. Ko, and C.-M. Lee, "Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter," *Energy*, vol. 49, pp. 413-422, 2013.

[14]. R. E. De Grande, A. Boukerche, and R. Alkharboush, "Time series-oriented load prediction model and migration policies for distributed simulation systems," *IEEE Parallel and Distributed Systems*, vol. 28, no. 1, pp. 215-229, 1 Jan. 2017.

[15]. G. Capizzi, C. Napoli, and F. Bonanno, "Innovative second-generation wavelets construction with recurrent neural networks for solar radiation forecasting," *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1805-1815, Nov. 2012.

[16]. D. D. Monner, and J. A. Reggia, "Recurrent neural collective classification," *IEEE Trans. Neural Networks and Learning Systems*, vol. 24, no. 12, pp. 1932-1943, Dec. 2013.

[17]. C. Napoli, G. Pappalardo, G. M. Tina, and E. Tramontana, "Cooperative strategy for optimal management of smart grids by wavelet RNNs and cloud computing," *IEEE Trans on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1672-1685, Aug. 2016.

[18]. Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157-166, Mar 1994.

[19]. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.

[20]. T. Ergen, and S. S. Kozat, "Efficient online learning algorithms based on lstm neural networks," *IEEE Trans. Neural Networks and Learning Systems*, vo. 29, no. 8, pp. 3772-3783, 2018.

[21]. E. Arvin, and S. Salvatore, "Single-Sensor Acoustic Emission Source Localization in Plate-Like Structures Using Deep Learning," *Aerospace*, vol. 5, no. 2, pp. 50, May 2018.

[22]. A. Ebrahimkhanlou, B. Dubuc, and S. Salamone, "Damage Localization in Plate-like Structures Using Guided Ultrasonic Waves Edge Reflections," in *Proc. of 10th International Workshop on Structural Health Monitoring: System Reliability for Verification and Implementation*, vol. 2, no. 7, pp. 2521–2528, 2015.

[23]. A. Ebrahimkhanlou, and S. Salamone, "Single-sensor acoustic emission source localization in plate-like structures: a deep learning approach," in *Proc. of SPIE, Health Monitoring of Structural and Biological Systems*, 2018.

[24]. F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition," *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 939-946, 10 2018.

[25]. F. L. Quilumba, W. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 911-918, March 2015.

[26]. A. O. Boudraa, and J. C. Cexus, "EMD-based signal filtering," *IEEE Trans. Instrum. Meas*, vol. 56, no. 6, pp. 2196-2202, 2007.

[27]. Y. Ren, P. N. Suganthan, and N. Srikanth, "A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods," *IEEE Trans. Sustainable Energy*, vol. 6, no. 1, pp. 236-244, 2015.

[28]. W. U. Zhaohua, and N. E. Huang. "Ensemble empirical mode decomposition: a noise-assisted data analysis method." *Adv Adapt Data Anal*, vol. 1, no. 1, pp. 1-41, 2009.

[29]. J. Tang, T. Chai, W. Yu, Z. Liu, and X. Zhou, "A comparative study that measures ball mill load parameters through different single-scale and multiscale frequency spectra-based approaches," *IEEE Trans. Ind. Informat*, vol. 12, no. 6, pp. 2008-2019, Dec. 2016.

[30]. F. Sulla, M. Koivisto, J. Seppänen, J. Turunen, L. C. Haarla and O. Samuelsson, "Statistical analysis and forecasting of damping in the nordic power system," *IEEE Trans. Power Syst*, vol. 30, no. 1, pp. 306-315, Jan. 2015.

[31]. D. Kingma, and J. Ba, "Adam: A method for stochastic optimization," Computer Science, 2014.

[32]. H. Jahangir, H.Tayarani , S. Baghali , A. Ahmadian , A. Elkamel, M. A. Golkar and M. Castilla, "A novel electricity price forecasting approach based on dimension reduction strategy and rough artificial neural networks," *IEEE Trans. Industrial Informatics*, vol. 16, no. 14, pp. 2369-2381, 2019.