energies

MDPI

Article

Short-Term Load Forecasting Using EMD-LSTM Neural Networks with a Xgboost Algorithm for Feature Importance Evaluation

Huiting Zheng *,†, Jiabin Yuan † and Long Chen †

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; jbyuan@nuaa.edu.cn (J.Y.); chenlong0711@163.com (L.C.)

* Correspondence: zhenghuiting@nuaa.edu.cn

† These authors contributed equally to this work.

Received: 31 May 2017; Accepted: 1 August 2017; Published: 8 August 2017

Abstract: Accurate load forecasting is an important issue for the reliable and efficient operation of a power system. This study presents a hybrid algorithm that combines similar days (SD) selection, empirical mode decomposition (EMD), and long short-term memory (LSTM) neural networks to construct a prediction model (i.e., SD-EMD-LSTM) for short-term load forecasting. The extreme gradient boosting-based weighted k-means algorithm is used to evaluate the similarity between the forecasting and historical days. The EMD method is employed to decompose the SD load to several intrinsic mode functions (IMFs) and residual. Separated LSTM neural networks were also employed to forecast each IMF and residual. Lastly, the forecasting values from each LSTM model were reconstructed. Numerical testing demonstrates that the SD-EMD-LSTM method can accurately forecast the electric load.

Keywords: long short-term memory neural networks; similar day; extreme gradient boosting; k-means; empirical mode decomposition; short-term load forecasting

1. Introduction

Short-term load forecasting (STLF), which ranges from one hour to one week ahead, plays an important role in the control, power security, market operation, and scheduling of reasonable dispatching plans for smart grids. However, achieving high accuracy is difficult because of the complicated effects of a variety of attributes on the load.

Over the past few decades, scholars have developed many modus to improve the accuracy of STLF that can mainly be divided into three methods, namely, traditional, similar day (SD), and artificial intelligence (AI)-based methods. Traditional methods are based on mathematical models, including multiple linear regression [1], stochastic time series [2], exponential smoothing [3], and knowledge-based methods [4]. Traditional methods often perform poorly at nonlinear forecasting, and STLF is a nonlinear problem. Accordingly, the prediction accuracy of traditional methods is insufficient for STLF.

The SD method is based on the selection of historical days that have similar features to the forecasted days [5–9]. Mandal et al. [7] selected SDs based on the calculation of the Euclidean norm of factors between historical and forecasted days. Chen et al. [8] required SDs to have the same weekday index and similar weather to the forecasted days. Mu [9] applied a weighted average model for the historical day to determine the influence of most SDs on the forecasted day. However, using this method solely cannot sufficiently obtain high prediction accuracy. The selection of input variables plays a crucial role when modelling time series and thus should be treated as a generalization problem. Arahal [10] proposed a method consists on calculating the difference index for all variables.

The AI-based methods, such as the artificial neural networks [11–14], support vector machine [15–18], expert system models [19], fuzzy logistic methods [20], and Bayesian neural networks [21], are extensively used to handle many forecasting problems. Although wide-ranging research has been conducted, an accurate STLF remains a challenge due to its non-stationary load data and long-term dependencies forecasting horizon. Hence, we applied the long short-term memory (LSTM) [22,23], which is a special type of recurrent neural network (RNN) architecture [24], to solve the STLF problem. The vanishing gradient point is a problem for RNNs in handling time series; LSTM cells can address this issue by incorporating memory cells in the hidden layer of RNN. LSTM performs well in long time horizon forecasting than other artificial intelligence methods based on the past load data that determine the effect and relationship among time series.

However, two inherent defects exist in neural networks: slow convergence and presence of a local minimum. After extensively analyzing the structure of neural networks, scholars [25] proposed a model that combines data decomposition with neural networks to address these two defects. Empirical Mode Decomposition (EMD) [26–29] can facilitate the determination of the characteristics of the complex non-linear time series. EMD is based on local characteristics of the signal sequence to complete the signal decomposition, in other words, the method do not require any base function pre-defined. Compared with wavelet decomposition, EMD method can be applied in theory to the decomposition of any type of signal since it has the characteristics of intuitive, direct, posterior and adaptive. Because it is essentially quite different from wavelet decomposition methods based on the wavelet basis function. Briefly, The EMD takes advantage of the multi-resolution and overcomes the difficulty of selecting wavelet basis function in wavelet transformation.

Given the preceding discussion, this study presents a generic framework that combines extreme gradient boosting (Xgboost) and k-means on SD selection, empirical mode decomposition (EMD), and LSTM neural networks to forecast short-term load (i.e., SD-EMD-LSTM model). Simulation experiments that use hourly load data from New England-ISO are conducted to verify the performance of the proposed STLF framework. We compare the SD-EMD-LSTM model with several other classical STLF models, and the results demonstrate that the proposed model outperforms the others in one-day ahead and one-week ahead tests. Moreover, numerical testing confirms that the proposed SD-EMD-LSTM model is capable of forecasting accuracy, robustness, and stability.

The main contributions of this study are as follows.

1. Although the temperature, humidity, and day type have been extensively used as input features in STLF, we also recognize that STLF is sensitive to the day-ahead peak load, which has to be a supplemental input feature to the SD selection and LSTM training processes.
2. Extending from our previous work on data analysis, we independently learned the feature candidate weights for the SD selection framework based on the Xgboost algorithm to overcome the dimensionality limitation in clustering. Thus, the proposed Xgboost-based k-means framework can deal with the SD selection tasks beyond pure clustering.
3. Numerical testing demonstrates that data decomposition-based LSTM neural networks can outperform most of the well-established forecasting methods in the longer-horizon load forecasting problem.

The rest of this paper is organized as follows. Section 2 discusses the factors that affect electricity forecasting, including temperature, day-type, and day-ahead peak load factors. Section 3 presents a generic SD selection framework that combines the Xgboost and k-means algorithms. Section 4 presents the forecasting framework, which combines the EMD and LSTM neural networks. Section 5 presents the experimental design and numerical test results. Lastly, Section 6 provides the conclusions of this study.

2. Data Analysis

The analysis of the relationship between the load data and external variables that affect the electric load is necessary to achieve high forecasting accuracy. This analysis is based on the electricity load data (provided by ISO New England) measured at one-hour intervals from 2003 to 2016. This section describes the major load-affecting factors, including temperature and day-type index. We also analyze the relationship between the daily and day-ahead peak loads.

Evidently, temperature changes are the primary cause of electricity load changes. In particular, the temperature variation range often determines the variation range of the electricity load. The variation in the interval-valued load with respect to the interval-valued temperature is shown in Figure 1. In the summer season, the higher the temperature is, the larger the electricity load value becomes (see Figure 1a). That is, a positive correlation exists between the load and temperature. By contrast, this correlation becomes negative in the winter season (Figure 1b). The preceding analysis indicates the necessity of discussing the effect of temperature on electricity load from one season to another.

Different day-types have different daily load curves, and the load of different day-types, such as weekends, holidays, and working days, are also different. The load of a working day is often higher than that on the weekend due to the decrease in industrial load on weekends. Accordingly, load on Saturdays is lower compared with those on other days (see Figure 2). Mondays and Tuesdays typically have the largest energy consumption over the week. Evidently, non-working days have considerably low energy consumption. Therefore, day-types are an important feature that cannot be ignored.

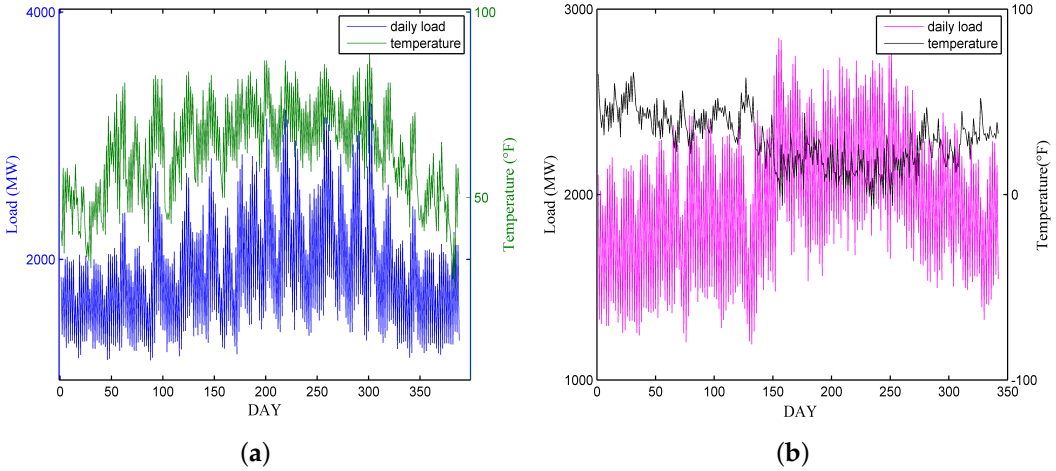


Figure 1. The daily interval-valued load and temperature: (**a**) from 11 April to 21 Octorber 2015; (**b**) from 22 Octorber 2015 to 10 April 2016.

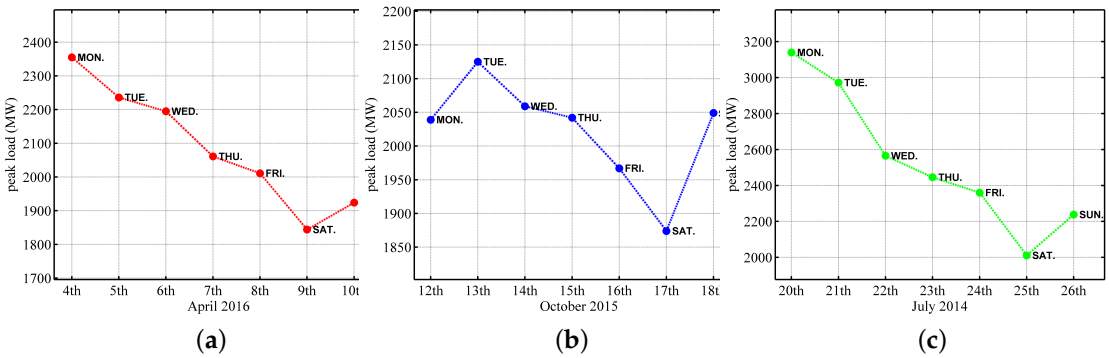


Figure 2. The daily peak load has different values depending on the day-type: (**a**) from 4 to 10 April 2016. (**b**) from 12 to 18 October 2016. (**c**) from 20 to 26 July 2014.

Although we have already identified several features that affect load forecasting, prediction errors may still be large during peak hours in the STLF process. Thus, we suppose that the day-ahead peak load is an important feature for forecasting. The Figure 3 is a scatter plot between the day-ahead peak load and daily load from 1 March 2003 to 31 October 2016. We determine that these two variables are closely related with the correlation coefficient = 0.8754. This result confirms the necessity of one day-ahead peak load to be a supplemental input feature to the SD selection and LSTM training processes.

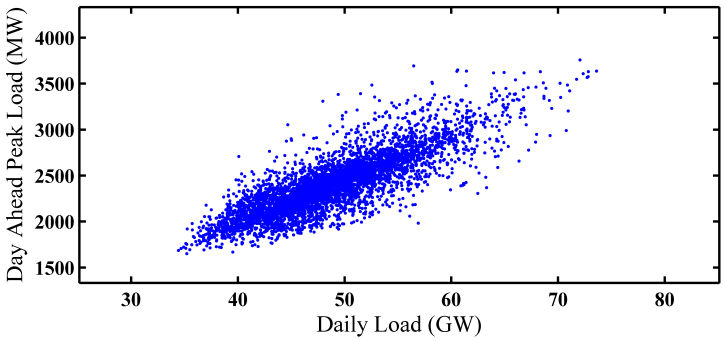


Figure 3. Correlations between daily load and one day-ahead peak load.

Precipitation and wind speed also have a bearing on the electricity load. Load on a sunny day is significantly higher than that on a rainy day. Therefore, improving the prediction accuracy is possible by selecting SD and maximizing the historical and features data.

3. Similar Day Selection: Improved K-Means with Extreme Gradient Boosting

If exogenous features, such as temperature, are included, then the traditional load forecasting model could lead to slow convergence and poor prediction accuracy. Thus, we select the SD load as the input data to improve the prediction power.

Clustering based on the feature values of the data and similar samples gathered in the same cluster can substantially improve the selection of SDs with the forecasting day. The performance of the clustering algorithm depends on the distance between records. "It is misleading to calculate the distance by measuring all attributes equally. The distance between neighbors will be dominated by the large number of irrelevant attributes, which sometimes leads to the dimensionality curse" [30]. An effective method to overcome this problem is to add a weighted parameter for each feature. Hence, the more relevant the feature is, the larger the impact of this feature becomes on the clustering results.

This section presents an alternative to SD selection that calculates the weights of the features using the Xgboost algorithm and integrates the weighted features using the k-means clustering.

3.1. Feature-Weight Learning Algorithm: Extreme Gradient Boosting

Xgboost [31] is an improved algorithm based on the gradient boosting decision tree and can construct boosted trees efficiently and operate in parallel. The boosted trees in Xgboost are divided into regression and classification trees. The core of the algorithm is to optimize the value of the objective function.

Unlike the use of feature vectors to calculate the similarity between the forecasting and history days, gradient boosting constructs the boosted trees to intelligently obtain the feature scores, thereby indicating the importance of each feature to the training model. The more a feature is used to make key decisions with boosted trees, the higher its score becomes. The algorithm counts out the importance by "gain", "frequency", and "cover" [32]. Gain is the main reference factor of the importance of a feature in the tree branches. Frequency, which is a simple version of gain, is the number of a feature in all constructed trees. Cover is the relative value of a feature observation. In this study the feature importance is set by "gain".

For a single decision tree T, Breiman et al. [33] proposed

$$w_\ell^2(T) = \sum_{t=1}^{J-1} \hat{\tau}_t^2 \tag{1}$$

as a score of importance for each predictor feature X_ℓ. The decision tree has $J-1$ internal nodes, and partitions the region into two subregions at every node t by the prediction feature X_ℓ. The selected feature is the one that provides maximal estimated improvement $\hat{\tau}_t^2$ in the squared error risk over that for a constant fit over the entire region. The squared importance of the feature X_ℓ is the sum of such squared improvement over the $J-1$ nodes, for which it was selected as the splitting feature. The following formula represents the importance calculation over the additive M trees.

$$w_\ell^2(T) = \frac{1}{M} \sum_{\mathrm{m}=1}^{M} \hat{\tau}_t^2(T_m) \tag{2}$$

The importance of a feature depends on whether the prediction performance changes considerably when such feature is replaced with random noise. Given the data analysis in the previous section, we take several features as input for the Xgboost algorithm to calculate the feature importance with the electricity load. We can obtain how each feature contributes to the prediction performance in the training course of the Xgboost algorithm. Evidently, the electricity load is sensitive to temperature variables (see Figure 4). Moreover, the supplement features (i.e., day-ahead-peak load) are an important feature for load forecasting. This conclusion is consistent with the results of the data analysis. We have now derived the important values of all features, which will be used as a priori knowledge of the subsequent clustering algorithm.

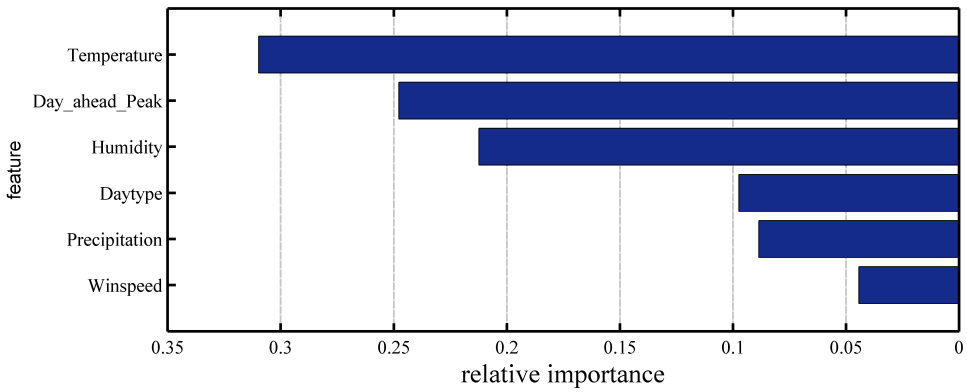


Figure 4. XGBoost feature importance.

3.2. K-Means Clustering Based on Feature-Weight

K-means, which was first proposed by MacQueen in 1967 [34], is extensively applied in many fields and sensitive to the selection of the initial cluster centroids. We selected the initial cluster centers with the maximum distance method to diminish the probability of converging to a local optimum. This section improves the k-means clustering by computing the initial cluster centers and utilizing the new distance calculation method. The steps are presented as follows.

1. Given a data set $X = \{x_1, x_2, ..., x_n\}$ and an integer value K. The data set is normalized as follows:

$$x_i = \frac{x_i - x_{i\min}}{x_{i\max} - x_{i\min}} \tag{3}$$

where $x_{i\min}$ and $x_{i\max}$ denote the minimum and maximum values, respectively, of each input factor.

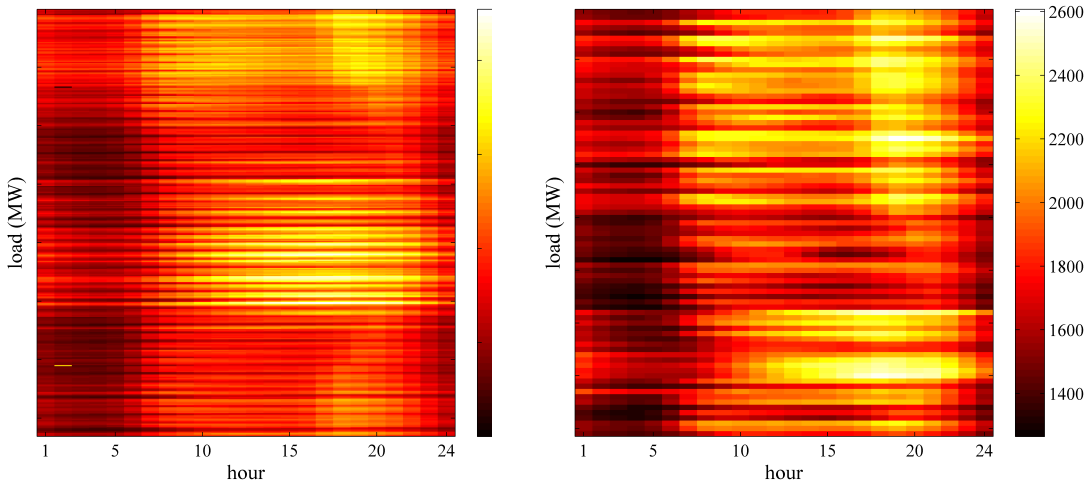
2. The forecasting day is selected as the first center u_0
3. The next center u_j is selected, where u_j is the farthest point from the previously selected cluster centers $\{u_0, u_1, ..., u_{j-1}\}$. Steps 2 and 3 are repeated until the K centers have been identified.
4. The feature weights are calculated using the Xgboost algorithm. Thereafter, the weights are attributed to each feature, thereby providing them with different levels of importance. Let w_p be the weight associated with the feature p. The norm is presented as follows.

$$d(x_i, x_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + \cdots + w_p(x_{ip} - x_{jp})^2 + \cdots + w_n(x_{in} - x_{jn})^2} \quad (4)$$

(1) Each data point is assigned to the nearest cluster.

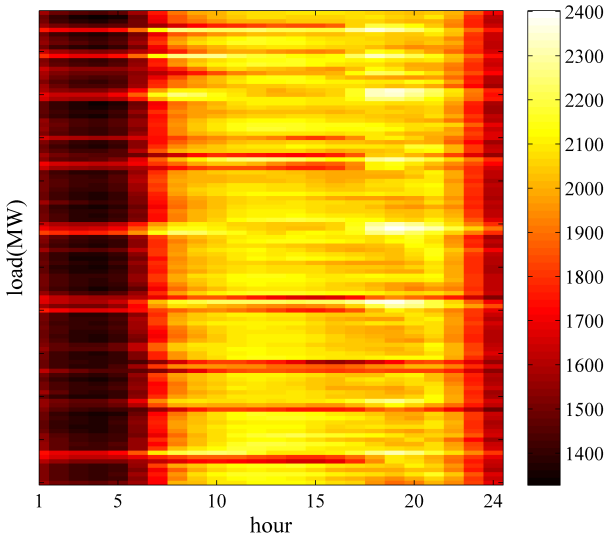
(2) The clusters are updated by recalculating the cluster centroid. The algorithm repeatedly executes (1) and (2) until convergence is reached.

The key idea in selecting SDs is to determine the attribute weights using the Xgboost algorithm and calculating the distance between the selected day and the day that relies on measuring different attributes in different weights. In Figure 5, the horizontal coordinate-axis presents the time (hour), whereas the longitudinal coordinate-axis presents the load curves. The color that changes from light to dark means that the electric load values change from large to small. Figure 5a shows the heat map for the original load data set, where every curve is evidently different in shape. Figure 5b,c are the heat maps for the original load data after the simple k-means clustering and weight k-means clustering, respectively. Our proposed Xgboost-k-means method can merge SDs into one cluster more effectively than the simple k-means algorithm does. Therefore, the SD can be the input data for subsequent load forecasting.



(**a**) (**b**)

Figure 5. *Cont.*



(**c**)

Figure 5. (**a**) Heat map for load profiles of the original data set; (**b**) Heat maps for cluster by k-means. (**c**) Heat maps for cluster by xgboost-k-means.

4. LSTM with Empirical Mode Decomposition

Neural networks are extensively employed in time series forecasting. However, determining the structure is difficult and often falls into the local minimum. The EMD method can facilitate the determination of the characteristics of the complex non-linear or non-stationary time series, i.e., it can divide the singular values into separated IMFs and determines the general trend of the real time series. This can effectively reduce the unnecessary interactions among singular values and improve the performance when a single kernel function is used in forecasting. Thus, this section proposes a model that combines the EMD and LSTM neural networks for STLF.

4.1. Empirical Mode Decomposition

EMD is a new signal processing method proposed by Huang et al. in 1998 [26]. The original signal was derived from the data's characteristics and can be decomposed into the intrinsic mode functions (IMF) by EMD. Thus, EMD can effectively decompose the singular values and avoid trapping into a local optimum, thereby improving the performance and robustness of the model.

All IMFs must meet the following conditions:

a. For a set of data sequences, the number of extremal points must be equal to the number of zero crossings or, at most, differ by one.
b. For any point, the mean value of the envelope of the local maxima and local minima must be zero.

For the original signal $x(t)$, EMD decomposes $x(t)$ through the "sifting" process, which is described as follows.

1. Identify all the maxima and minima of signal $x(t)$.
2. Through the cubic spline interpolation fitting out the upper envelope $u(t)$ and lower envelope $l(t)$ of signal $x(t)$ .The mean of the two envelopes can be the average envelope curve $m_1(t)$:

$$m_1(t) = \frac{u(t) + l(t)}{2} \tag{5}$$

3. Subtraction of m from $x(t)$ to obtain an IMF candidate :

$$h_1(t) = x(t) - m_1(t) \tag{6}$$

4. If $h_1(t)$ does not satisfy the two conditions of the IMF, then it should take $h_1(t)$ as original signal and repeat above calculate k times. At this point, $h_{1k}(t)$ could be as shown in Equation (7):

$$h_{1k}(t) = h_{1(k-1)}(t) - m_{1k}(t) \tag{7}$$

$h_{1(k-1)}(t)$ and $h_{1k}(t)$ present the signal after shifting $k-1$ times and k times,respectively. $m_{1k}(t)$ is the average envelope of $h_{1k}(t)$

5. If $h_{1(k-1)}(t)$ satisfies the conditions of the IMF, define $h_{1k}(t)$ as $c_1(t)$. Standard deviation is defined by Equation (8):

$$\mathrm{S}_d = \sum_{t=0}^{T} \frac{|h_{1(k-1)}(t) - h_{1k}(k)|^2}{{h_{1k}}^2(t)} \in (0.2, 0.3) \tag{8}$$

6. Subtraction of $c_1(t)$ from $x(t)$ to obtain new signal $r_1(t)$

$$r_1(t) = x(t) - c_1(t) \tag{9}$$

7. Repeat previous steps 1 to 6 until the $r_n(t)$ cannot be decomposed into the IMF. $r_n(t)$ is the residual of the original data $x(t)$. Finally, the original signal $x(t)$ can be presented as a collection of n components $u_i(t)$ (i = 1, 2, ..., n) and a residual $r_n(t)$:

$$\mathrm{x}(t) = \sum_{i=1}^{n} u_i(t) + r_n(t) \tag{10}$$

The preceding steps show that the EMD method is employed to decompose the SD load at low and high frequencies, respectively. Figure 6 evidently shows the decomposition of the eight IMF extractions and residuals. Furthermore, all graphs in Figure 6 are shown in the same scale, thereby enabling the assessment of the contribution of each extracted IMF.

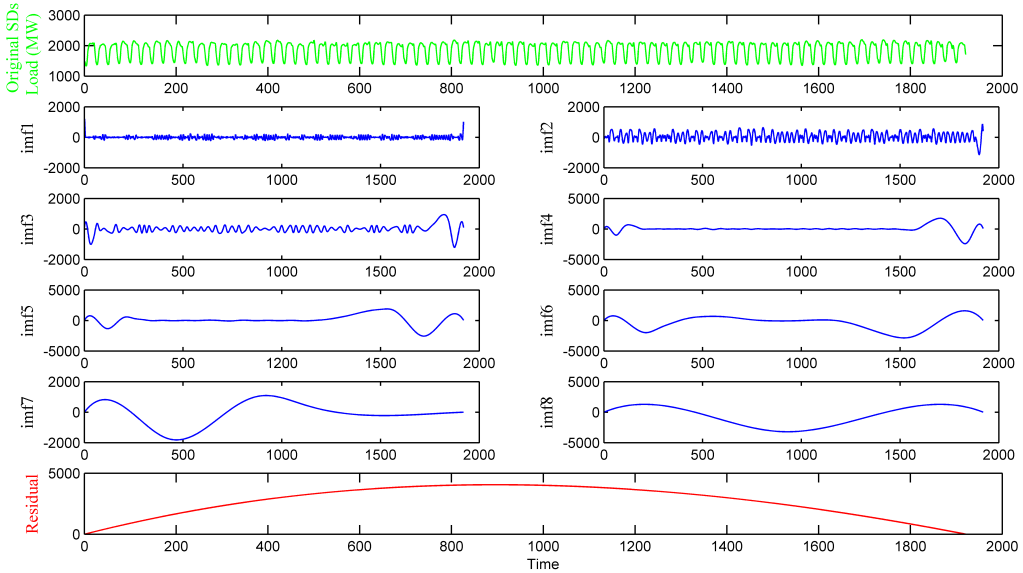


Figure 6. The original data sequence of the similar daily load and the result of empirical mode decomposition.

4.2. Lstm-Based Rnn for Electric Load Forecasting

LSTM was proposed by Hochreiter et al. in 1997 [35] as a type of efficient RNN architecture, and has been extensively applied in various fields. Moreover, LSTM is a popular time series forecasting model and can expertly deal with long-term dependencies data.

A. Recurrent Neural Networks (RNNs)

RNNs are designed to operate over the non-linear time-varying problem [24]. The RNN internal connections enable signals to travel forward and backward, thereby making RNNs substantially suitable for time series prediction.

RNNs can mine the rules from the time sequences to predict the data that have yet to occur [36,37]. The reason for this characteristic lies in the feedback connections that can facilitate updating the weights based on the residual in each forward step (Figure 7). The forecasting day load in STLF is bound up with the SD load. Therefore, if we provided the SD time sequences, then obtaining high accuracy on the forecasting day becomes possible.

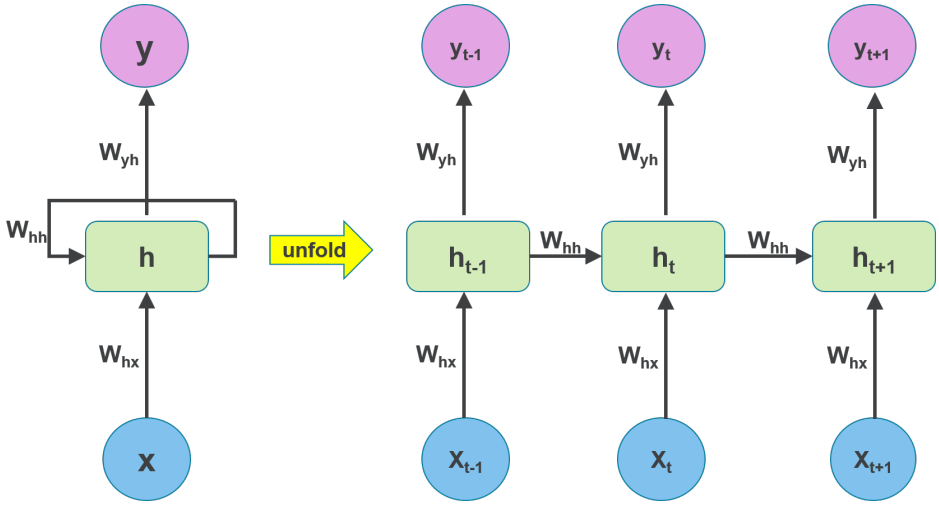


Figure 7. The architecture of RNN.

RNN proves to be suitable for this problem [38]. However, RNNs tend to suffer heavily from gradient vanishing, which may increase indefinitely and eventually cause the network to break down. Therefore, simple RNNs may not be the ideal option for forecasting problems with long-term dependencies.

B. LSTM-Based RNN Forecasting Scheme

LSTM was mainly motivated and designed to overcome the vanishing gradients problem of the standard RNN when dealing with long term dependencies.This section leads to the long short-term memory neural network. The LSTM model add the input gate,output gate and forgetting gate to the neurons in RNN. Such a structure can effectively mitigate the vanishing gradient problem [39]. This makes LSTM an architecture suitable for problems with long term dependencies.

The major innovation of LSTM is its memory cell, which essentially acts as an accumulator of the state information. First, as shown in Figure 8, the forget gate is applied to decide what information to get rid of the cell state. A sigmoid function is used to calculate the activation of the forget gate f_t.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{11}$$

The second step is to determine what new information should be stored in the cell state. To start with, a sigmoid layer named the "input gate layer" decides which information should be update. Then, a tanh layer creates a vector $\tilde{c}_t$ of new candidate values should be updated in the next state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{12}$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{13}$$

Next, we will update the old cell state c_{t-1} into the new cell state c_t. We multiply c_{t-1} by f_t for throw away the information from old cell. Then we add $i_t * \tilde{c}_t$. There are the new candidate values, scaled by how much information should be updated to each state value.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{14}$$

Lastly, we need to decide the output. This has two parts: we run a sigmoid layer as output gate to filter the cell state firstly. Then, we put the cell state through $\tanh(\cdot)$ and multiply it by the output o_t to calculate the desired information.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{15}$$

$$h_t = o_t * \tanh(c_t) \tag{16}$$

In Equations (11)–(16), W_i, W_f, W_c, W_o represents the appropriate weight matrices. The vectors b_i, b_f, b_c, b_o denote the corresponding bias vectors.

This study presents the experiments that apply the separated LSTM neural networks scheme for the SD load's IMF and residual forecasting. The training process inputs include the temperature, day-ahead-peak, humidity, day-type index, precipitation, wind speed, and IMF component of the SD load. The model framework is shown in Figure 9. In order to further improve the accuracy and practicability of the prediction model, we establish the architecture based on LSTM, named sequence to sequence (S2S). Sequence to sequence structure can adjust the length of the input and output sequences flexibly, that is appropriate to perform different time scales load forecasting.

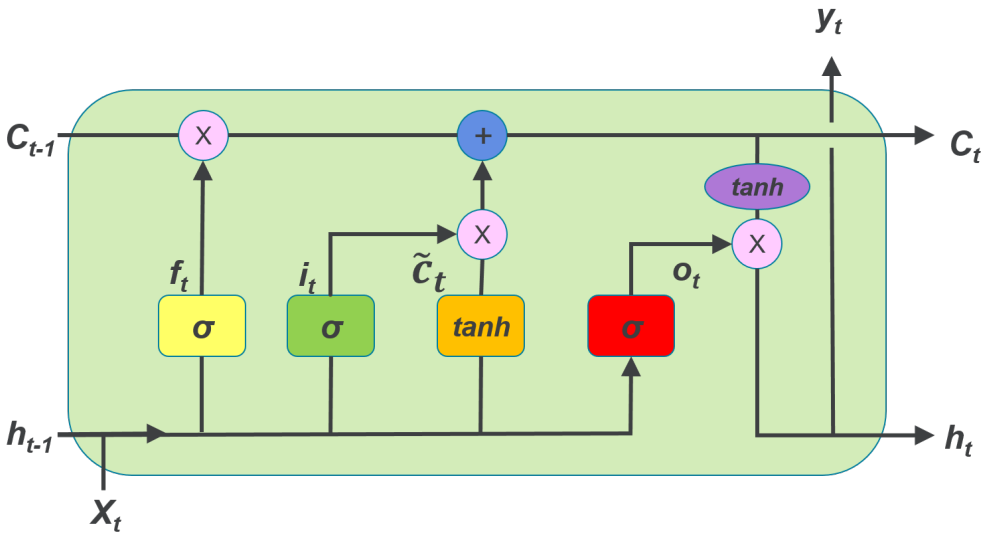


Figure 8. The architecture of LSTM memory block.

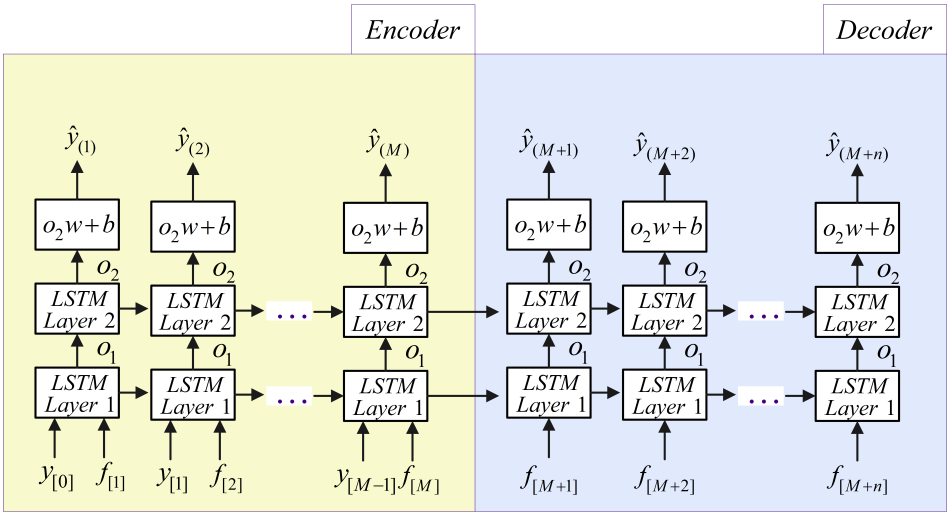


Figure 9. LSTM neural networks model for hourly load forecasting.

It is obvious that this architecture consists of two parts: encoder and decoder. The encoder aims to prediction the load we have already known, yet $y_{[M-1]}$ and $f_{[M]}$ present the load's IMF component of previous hour and the features of this hour, respectively. $f_{[M]}$ can be expressed as:

$$\begin{aligned} f_{[M]} = [&Temperature_{[M]} \quad Humidity_{[M]} \quad Day-type_{[M]} \quad Precipitation_{[M]} \\ &Windspeed_{[M]} \quad Day-ahead-peak \quad load_{[M]}] \end{aligned} \tag{17}$$

Then, the decoder generates an output sequence $\hat{y} = \left\{\hat{y}_{[M]}, \hat{y}_{[M+1]}, ..., \hat{y}_{[M+n]}\right\}$, that is the prediction of IMF component of the load for the next $n+1$ hours.

Standard backpropagation can be applied to train the network using a gradient based method called Stochastic Gradient Descent (SGD). Table 1 shows the Mean Absolute Percentage Error (MAPE) on training and testing datasets for different number of layers and units using the S2S architecture for data with one-day-ahead forecasting.

It can be seen that the proposed architecture is able to produce very low errors on training dataset. Further, it is noticed that increasing the capacity of the network by increasing the number of layers and units only improves error on training dataset. The model performs well on training dataset using a 2-layer network with 50 units in each layer. However, increasing the capacity of the network does not improve performance on testing data. In order to improve accuracy on testing data Dropout is used as regularization methodology.

Table 1. MAPE(%) FOR S2S ARCHITECTURE (one-day-ahead load forecasting).

Layers	Units	MAPE (Training)	MAPE (Testing)
1	5	1.13	1.026
1	20	1.079	1.043
1	50	1.023	1.072
1	100	0.944	1.115
2	5	1.095	1.028
2	20	1.021	1.061
2	50	0.962	1.012
2	100	1.078	1.103
3	5	1.083	1.116
3	20	1.05	1.051
3	50	0.958	1.138
3	100	0.934	1.342

4.3. The Full Procedure of SD-EMD-LSTM Model

The complete procedure of the proposed SD-EMD-LSTM model is presented as follows and illustrated in Figure 10.

1. Similar days selection. Calculate the features weight by xgboost method, then combined with K-means algorithm to determine the similar days cluster.
2. Data decomposition. Take similar days load as input data, and decompose the input data into several intrinsic mode functions with EMD algorithm.
3. Forecasting. Separated LSTM neural networks employed to forecast each IMF and residual, respectively. Reconstruct the forecasting values from each single LSTM model.

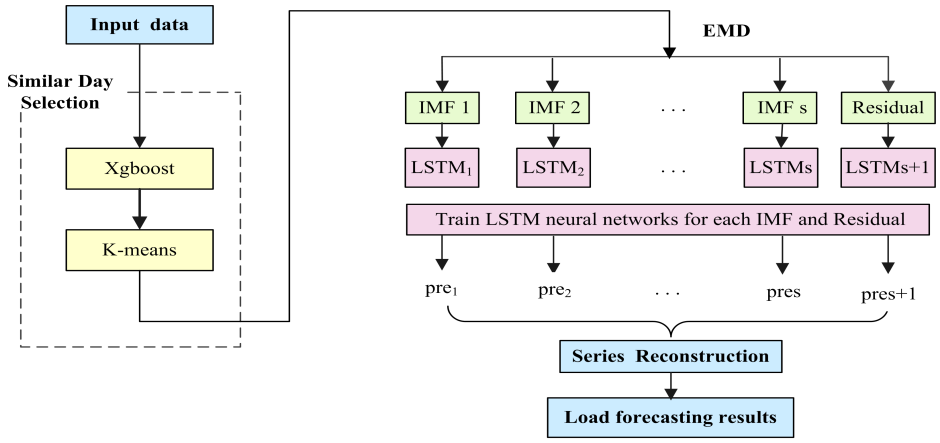


Figure 10. The full flowchart of the SD-EMD-LSTM model flowchart.

5. Numerical Experiments

This section presents the forecasting performance of the proposed SD-EMD-LSTM model. The hourly electric load data from NE-ISO 2003 to 2016 is employed for the models. The forecasting has been conducted in two time scales, namely, one day ahead (24 h) and one week ahead (168 h).

First, we present the experiments on applying the weighted k-means-based SD selection algorithm for load forecasting, as well as analyze the optimum value of the clusters k. Second, we verify the clustering effect of the proposed SD selection method and the need of using the supplemental feature. Third, experiments in two time scales are conducted to compare the proposed model with the standalone LSTM, SD-LSTM, and EMD-LSTM models to show the fitting effect of the hybrid model. Lastly, we compare the forecasting performance with three other models (i.e., ARIMA, BPNN, and SVR) to illustrate the forecasting accuracy and stability of the SD-EMD-LSTM model. The structure of BPNN model comprises of 3 layers viz. input, hidden and output layers(6-20-1), where the transfer functions of hidden layer and output layer are tansig and purelin, respectively. While the training function uses traingdm, the learning function of threshold and weights use learnged. SVR with LIBSVM package with $C = 8.4065$, $\gamma = 0.0869335$, $\varepsilon = 0.000118$.

5.1. Evaluation Indices for the Forecasting Performance

The mean absolute percentage error (MAPE) is employed as a criterion of error evaluation to analyze the forecasting performance.

$$MAPE = \frac{1}{m}\sum_{j=1}^{m}\left|\frac{X_j - \widehat{X}_j}{X_j}\right| \quad (18)$$

where $\widehat{X}_j$ is the forecasting value, X_j is the actual value, and m is the total number of forecasting points. For the two forecasting time scales, m is set at 24 h and 168 h.

5.2. Empirical Results and Analysis

We perform simulations of the four examples to verify the predictive ability of the proposed method:

Example 1: Through the enumeration method, k ranges from 5 to 12, the run is repeated several times in each k value using the Xgboost-k-means-based SD-EMD-LSTM model. Thereafter, the prediction accuracy of each k is calculated. Experiments of the 24-h-ahead forecasting in different seasons are performed to analyze the best k value with the highest prediction accuracy. Figure 11 shows that when the number of clusters equals 9, the prediction curve most closely follows the raw curve in four days, including 30 October 2016, 5 July 2015, 13 April 2014, and 22 February 2013, which represent the four seasons.

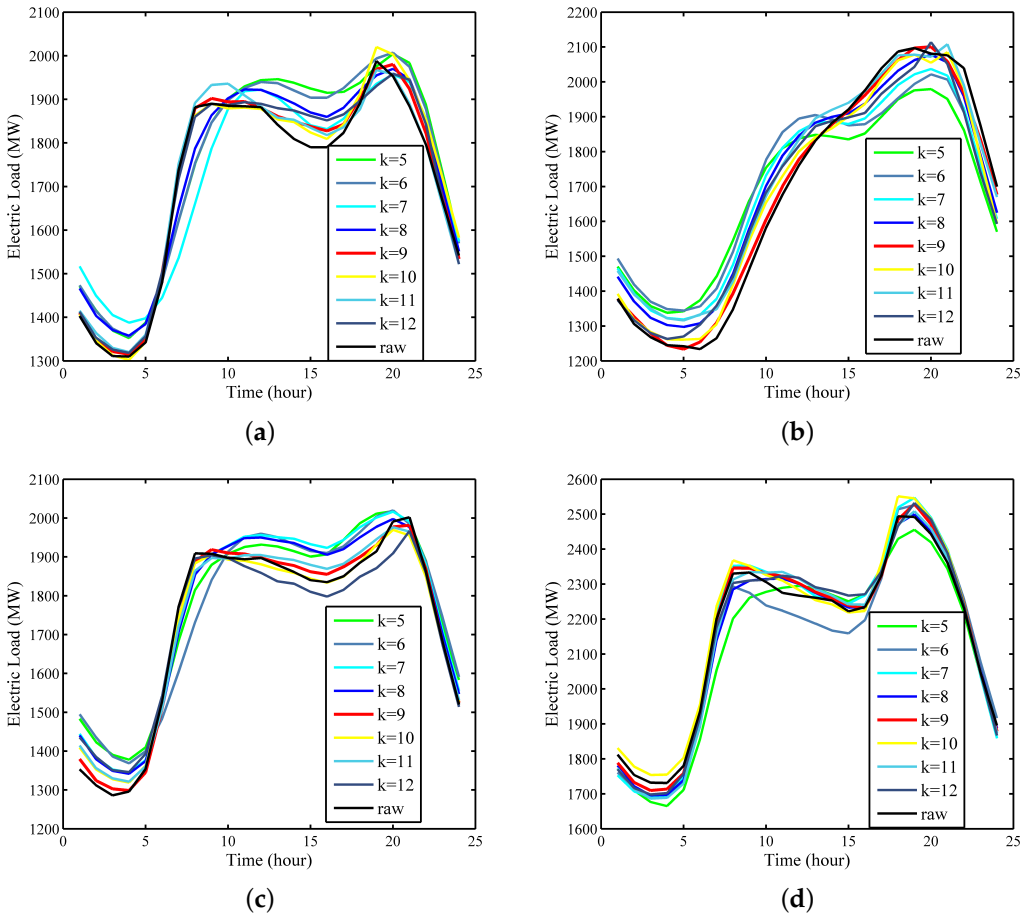


Figure 11. SD-EMD-LSTM forecast with respect to the number of clusters: (**a**) 30 October 2016; (**b**) 5 July 2015; (**c**) 13 April 2014; (**d**) 22 February 2013.

MAPE can also be used to determine the ideal number of clusters. Comparison results (see Table 2) show that the proposed model with 9 clusters outperformed all the other cluster numbers with the smallest forecasting MAPE of 0.97%. That is, the proposed Xgboost-k-means method can effectively merge SDs into one cluster. Consequently, we define k = 9 as a priori knowledge in the proposed SD-EMD-LSTM model to select SD for the subsequent load forecasting.

Table 2. MAPE(%) for the different number of clusters in one-day ahead prediction.

k	5	6	7	8	9	10	11	12
MAPE(%)	3.81	3.45	3.08	2.05	0.97	1.42	1.94	2.12

Example 2: This example includes two cases. Case 1 verifies the clustering effect of the proposed SD selection method, and the simple k-means algorithm is used to the SD selection for comparison with the proposed Xgboost-k-means model. Case 2 demonstrates the importance of using the supplemental feature, namely, day-ahead peak load. The training period in this example is from 2003 to 2015, and the prediction period is 2016.

Case 1: EMD-LSTM is combined with the proposed SD selection method and simple k-means algorithm respectively to verify its performance. In the one-day ahead load forecasting as shown in Figure 12, Xgboost-k-means hybrid with the EMD-LSTM model fits the raw data better than the simple k-means clustering algorithm. That is, the Xgboost-k-means algorithm could merge SDs into one cluster more effectively, thereby improving the prediction accuracy.

Table 3 also verifies this scenario, which shows that the SD-EMD-LSTM model achieved an improved forecasting performance with a considerably small MAPE, as well as agrees with the conclusion presented in Section 3. The reason lies in that the Xgboost algorithm has considerable ability to access each feature's weight, the limitation of the dimensionality is generally reduced, and the models are obtained with increasing the forecasting accuracy.

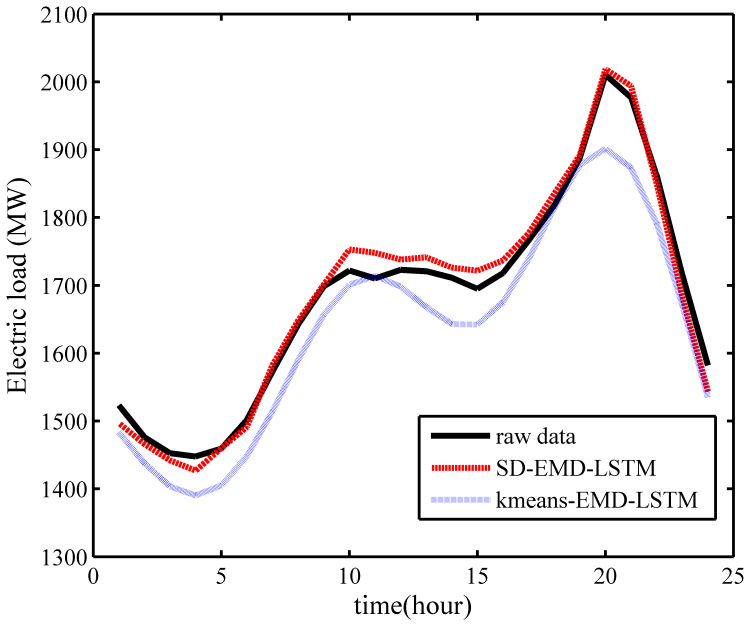


Figure 12. One-day ahead prediction of 20 March 2016.

Table 3. Monthly MAPE(%) of validation phase.

Model	January	February	March	April	May	June	July	August	September	October	November	December
SD-EMD-LSTM	1.03	0.96	0.83	1.09	0.77	0.93	0.96	0.88	1.06	1.12	0.91	1.12
kmeans-EMD-LSTM	2.11	2.17	1.98	2.56	1.83	2.16	1.96	2.03	2.08	1.79	2.04	2.18

Case 2: The SD-EMD-LSTM model is used with and without the supplemental feature (i.e., day-ahead peak load) to analyze the prediction accuracy on the one-day ahead load forecasting. Further details are shown in Figure 13.

The most significant forecasting errors often occur at the peak points of the forecast load curve. The reason is that the proposed model with the supplemental feature (i.e., day-ahead peak load) can achieve an improved forecasting performance at the peak points.

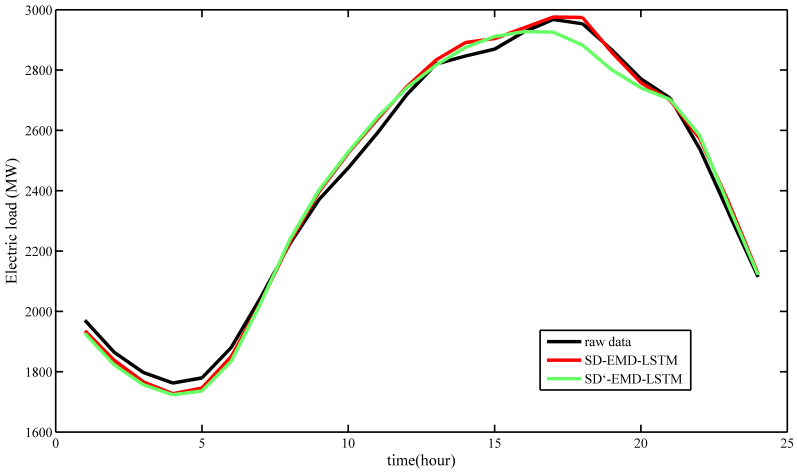


Figure 13. One-day ahead prediction of 26 August 2016.

The hourly mean absolute percentage errors listed in Table 4 indicate that the proposed model with supplemental input feature (i.e., day-ahead peak load) obtained an average MAPE of 1.10%. This value is lower than the 1.44% obtained in the model without the supplemental input feature. Furthermore, SD-EMD-LSTM with supplemental input feature has good prediction accuracy during peak hours (i.e., from 15:00 to 20:00). Therefore, the day-ahead peak load should be the supplemental input feature for load forecasting.

Table 4. 24-h forecast MAPE (%) of validation phase.

Hour	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
SD-EMD-LSTM	1.77	1.42	1.73	2.00	1.87	1.63	0.81	0.20	1.09	1.99	1.75	0.96
SD'-EMD-LSTM	2.20	2.25	2.26	2.21	2.44	2.41	0.93	0.61	1.33	2.13	1.99	0.82
13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**Average**
0.49	1.57	1.22	0.47	0.27	0.71	0.29	0.46	0.25	1.33	1.48	0.52	1.10
0.12	1.01	1.46	1.05	1.41	2.40	2.32	1.07	0.15	1.73	1.02	0.33	1.44

Example 3: This example compares the forecasting value of the proposed SD-EMD-LSTM model to the single LSTM, SD-LSTM, and EMD-LSTM models. The experiments have a forecasting horizon of h = 24 h and h = 168 h. The training period is from 2003 to 2014, and the prediction period is 2015.

Figure 14a,b show the one-day ahead and one-week ahead forecasting results of the single LSTM, SD-LSTM, EMD-LSTM, and proposed SD-EMD-LSTM models, respectively.

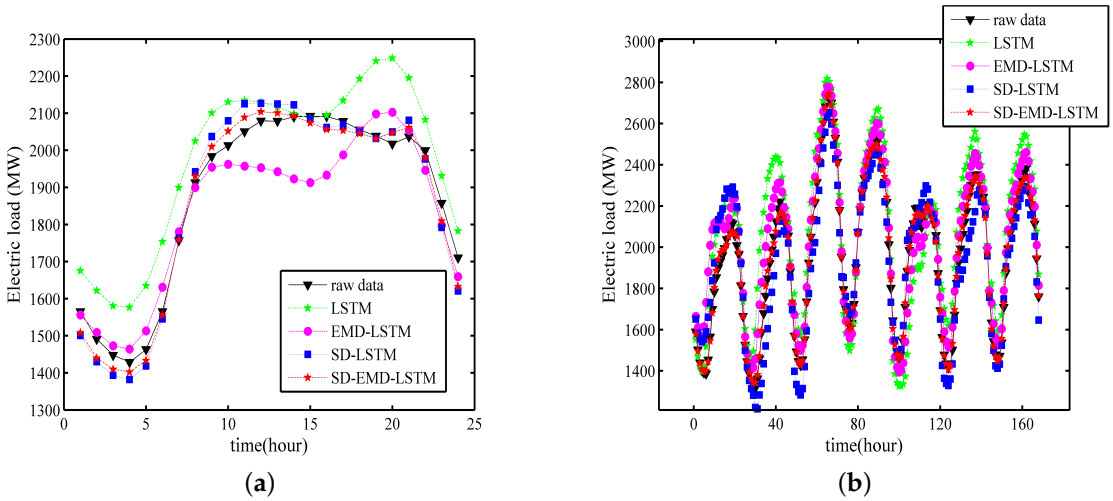


Figure 14. (**a**) One-day ahead prediction of 3 June 2015. (**b**) One-week ahead prediction from 18 June 2015 to 24 June 2015.

We can conclude from Figure 14 that the forecasting curve of the proposed SD-EMD-LSTM model follows the raw data better than the other alternative models for the two forecasting horizon in Example 3. Evidently, comparing the LSTM curve with those of SD-LSTM and EMD-LSTM shows that the SD selection can generally enhances the accuracy of the load forecasting in the one-day-ahead and one-week-ahead forecasting. EMD can also effectively determine the general trend of the real time series.

Table 5 shows the MAPE values per month of all the models in Example 3. The last row of Table 4 lists the average MAPE values for the experiment based on 12 months. The LSTM neural networks combined with the Xgboost-k-means-based SD selection method is better than the LSTM neural networks combined with the EMD model but is slightly inferior to the SD-EMD-LSTM model. The evaluation results of the MAPE indexes and prediction curves for the four models tend to be consistent.

Example 3 enables us to conclude the following points.

(1) The fitting effect of the hybrid model is evidently better than that of the single LSTM neural networks model in both time scales.
(2) The Xgboost-k-means method can effectively merge SDs into one cluster and prevent the LSTM neural networks from being trapped into a local optimum, thereby substantially improving the prediction accuracy.
(3) The data decomposition method divides the singular values into separated IMFs and determines the general trend of the real time series, thereby effectively improving the performance and robustness of the model.

In general, the SD-EMD-LSTM model significantly outperforms the three other methods and achieves a good prediction effect in STLF.

Table 5. Mape(%) for Load Forecasting in Example 3.

	24-h Ahead				168-h Ahead			
	LSTM	**SD-LSTM**	**EMD-LSTM**	**SD-EMD-LSTM**	**LSTM**	**SD-LSTM**	**EMD-LSTM**	**SD-EMD-LSTM**
January	5.97	2.7	4.39	1.19	9.86	3.37	6.46	1.57
February	4.3	2.13	3.73	0.98	7.88	2.43	5.69	1.22
March	6.89	1.65	4.58	0.66	6.26	2.62	4.77	1.25
April	5.19	2.46	3.63	0.72	7.93	4.34	4.93	2.08
May	6.85	3.97	5.17	1.26	9.65	3.21	6.65	1.79
June	5.64	1.02	3.4	0.83	7.49	3.1	5.12	0.91
July	5.9	2.72	4.09	1.24	9.19	3.13	7.11	1.04
August	4.59	2.16	2.44	1.12	8.3	4.97	6.72	1.63
September	4.31	1.88	3.11	0.85	8.69	4.08	6.07	1.59
October	6.82	2.89	5.94	1.65	8.53	5.02	6.69	1.87
November	4.2	2.84	3.15	1.35	11.81	6.16	8.03	2.31
December	4.46	2.01	3.68	1.05	9.31	3.88	7.96	1.79
Average	**5.43**	**2.37**	**3.94**	**1.08**	**8.74**	**3.86**	**6.35**	**1.59**

Example 4: This example compares the forecasting results of ARIMA, BPNN, SVR, and the proposed SD-EMD-LSTM model. For a fair comparison, we compare their performance with the same input data sets (i.e., SDs). Figure 15a,b show the one-day ahead and one-week ahead forecasting results, respectively. The training period is from 2003 to 2013, and the prediction period is 2014.

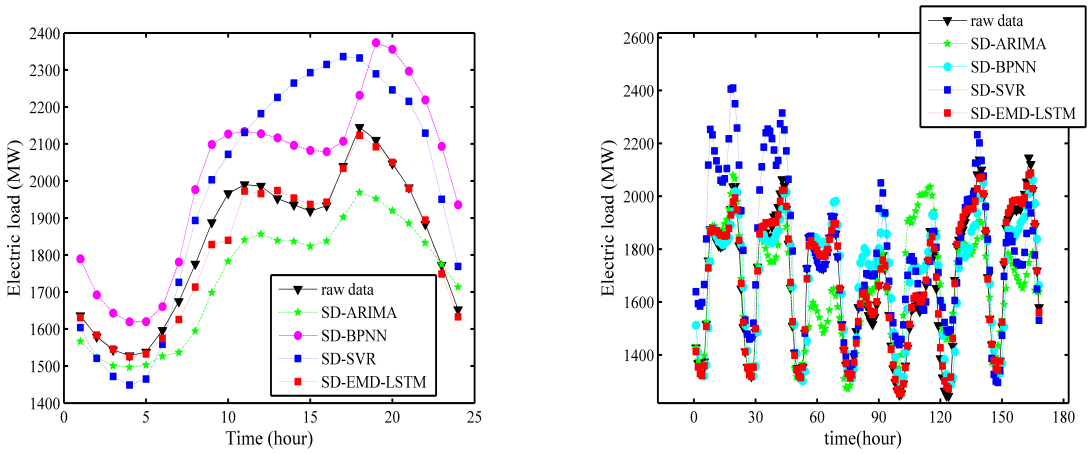


Figure 15. (**a**) One-day ahead prediction of 9 January 2014 are performed by example 4; (**b**) One-week ahead prediction from 12 October 2014 to 18 October 2014 are performed by example 4.

Figure 15 shows that the forecasting curve of the proposed SD-EMD-LSTM model is closer to the raw load curve than the other alternative models in Example 4. The performance results of the three other methods are insufficient for STLF.

From the MAPE values in Table 6, the experiment results indicate that the proposed model is significantly superior to the SVR, ARIMA, and BPNN models. MAPE of the SD-EMD-LSTM model is the lowest among all the models. Its prediction accuracy also reaches 98.96% and 98.44% in the 24-h-ahead and 168-h-ahead forecasting, respectively. ARIMA has the maximum MAPE value. Although the three other models determined the general trend of the raw data, their forecasting errors were extremely high.

The comparison between the two different forecasting time scales demonstrate that the accuracy of the proposed hybrid model exhibit minimal changes because the LSTM neural networks can maximize the long-term dependencies in the electric load time series for substantially accurate forecasting. That is, the SD-EMD-LSTM model can perform longer-horizon load forecasting. Overall, the proposed hybrid model provides a powerful method that can outperform many other forecasting methods in the challenging STLF problem.

Table 6. MAPE(%) For Load Forecasting in Example 4.

	24-h Ahead				168-h Ahead			
	SD-ARIMA	**SD-BPNN**	**SD-SVR**	**SD-EMD-LSTM**	**SD-ARIMA**	**SD-BPNN**	**SD-SVR**	**SD-EMD-LSTM**
January	5.47	4.27	4.29	1.23	11.55	7.52	8.3	2.36
February	4 .43	1.79	1.96	0.75	6.97	9.03	2.29	1.05
March	3.34	2.84	3.5	1.12	5.04	3.42	5.25	1.29
April	4.05	3.74	6.08	1.04	7.94	4.56	7.52	1.09
May	7.88	4.43	2.43	0.95	9.36	5.26	2.41	1.83
June	5.49	5.68	2.93	0.82	6.44	9.26	6.96	1.92
July	3.61	1.92	2.07	1.12	8.5	3.81	3.18	1.69
August	5.86	2.68	3.58	1.87	10.65	9.12	6.29	2.03
September	4.51	1.86	3.68	0.98	9.72	2.19	4.33	1.26
October	6.07	4.28	6.51	0.76	7.98	5.77	8.85	1.28
November	8.51	3.85	2.13	0.99	6.11	4.52	3.72	1.75
December	10.05	5.71	3.47	0.87	11.73	5.44	9.5	1.22
Average	**5.48**	**3.39**	**3.55**	**1.04**	**8.50**	**5.83**	**5.72**	**1.56**

6. Conclusions

This study presents an LSTM neural network model hybridized with the SD selection and EMD methods for STLF. The key idea in selecting SDs is to determine the attribute weights using the Xgboost algorithm and calculate the distance between the selected day and the day that relies on the different measured attributes in different weights. Thereafter, the k-means algorithm merges SDs into one cluster as input data for the subsequent forecasting based on the Xgboost distance. EMD eventually determines the key features of the SD load at low and high frequencies. Lastly, the separated LSTM neural networks are used to forecast the future values in low-frequency and high-frequency time series. The proposed method has been compared with the LSTM, SD-LSTM, EMD-LSTM, ARIMA, BPNN, and SVR models in real-load data obtained from the NE-ISO for the one-day ahead and one-week ahead load forecasting. Comparison results demonstrate that the proposed Xgboost-k-means method can effectively merge SDs into one cluster. Moreover, the EMD-LSTM model has a good ability to accurately forecast the complex non-linear electric load time series over a long horizon. The aforementioned analysis implies that the proposed SD-EMD-LSTM framework can be a promising alternative approach to STLF.

Acknowledgments: This work is supported by the National Key Research and Development Program of China (No. 2017YFB0802303), Application platform and Industrialization for efficient cloud computing for Big data of the Science and Technology Supported Program of Jiangsu Province (BA2015052), Funding of National Natural Science Foundation of China (Grant No.61571226), Jiangsu Program for the transformation of scientific and technological achievements (BA2015051).

Author Contributions: In this research activity, all the authors were involved in the data collection and preprocessing phase, model constructing, empirical research, results analysis and discussion, and manuscript preparation. All authors have approved the submitted manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dudek, G. Pattern-based local linear regression models for short-term load forecasting. *Electr. Power Syst. Res.* **2016**, *130*, 139–147.
2. Shenoy, S.; Gorinevsky, D.; Boyd, S. Non-parametric regression modeling for stochastic optimization of power grid load forecast. In Proceedings of the American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 1010–1015.
3. Christiaanse, W. Short-term load forecasting using general exponential smoothing. *IEEE Trans. Power Appar. Syst.* **1971**, *PAS-90*, 900–911.
4. Kandil, M.; El-Debeiky, S.M.; Hasanien, N. Long-term load forecasting for fast developing utility using a knowledge-based expert system. *IEEE Trans. Power Syst.* **2002**, *17*, 491–496.
5. Sun, X.; Luh, P.B.; Cheung, K.W.; Guan, W.; Michel, L.D.; Venkata, S.; Miller, M.T. An efficient approach to short-term load forecasting at the distribution level. *IEEE Trans. Power Syst.* **2016**, *31*, 2526–2537.

6. Ghofrani, M.; Ghayekhloo, M.; Arabali, A.; Ghayekhloo, A. A hybrid short-term load forecasting with a new input selection framework. *Energy* **2015**, *81*, 777–786.
7. Mandal, P.; Senjyu, T.; Funabashi, T. Neural networks approach to forecast several hour ahead electricity prices and loads in deregulated market. *Energy Convers. Manag.* **2006**, *47*, 2128–2142.
8. Chen, Y.; Luh, P.B.; Guan, C.; Zhao, Y.; Michel, L.D.; Coolbeth, M.A.; Friedland, P.B.; Rourke, S.J. Short-term load forecasting: Similar day-based wavelet neural networks. *IEEE Trans. Power Syst.* **2010**, *25*, 322–330.
9. Mu, Q.; Wu, Y.; Pan, X.; Huang, L.; Li, X. Short-term load forecasting using improved similar days method. In Proceedings of the 2010 Asia-Pacific Power and Energy Engineering Conference (APPEEC), Chengdu, China, 28–31 March 2010; pp. 1–4.
10. Arahal, M.R.; Cepeda, A.; Camacho, E.F. Input variable selection for forecasting models. *IFAC Proc. Vol.* **2002**, *35*, 463–468.
11. Raza, M.Q.; Khosravi, A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renew. Sustain. Energy Rev.* **2015**, *50*, 1352–1372.
12. Velasco, L.C.P.; Villezas, C.R.; Palahang, P.N.C.; Dagaang, J.A.A. Next day electric load forecasting using artificial neural networks. In Proceedings of the 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu City, Philippines, 9–12 December 2015; pp. 1–6.
13. Hernández, L.; Baladrón, C.; Aguiar, J.M.; Calavia, L.; Carro, B.; Sánchez-Esguevillas, A.; Pérez, F.; Fernández, Á.; Lloret, J. Artificial neural network for short-term load forecasting in distribution systems. *Energies* **2014**, *7*, 1576–1598.
14. Buitrago, J.; Asfour, S. Short-term forecasting of electric loads using nonlinear autoregressive artificial neural networks with exogenous vector inputs. *Energies* **2017**, *10*, 40.
15. Kaytez, F.; Taplamacioglu, M.C.; Cam, E.; Hardalac, F. Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *Int. J. Electr. Power Energy Syst.* **2015**, *67*, 431–438.
16. Selakov, A.; Cvijetinović, D.; Milović, L.; Mellon, S.; Bekut, D. Hybrid pso-svm method for short-term load forecasting during periods with significant temperature variations in city of burbank. *Appl. Soft Comput.* **2014**, *16*, 80–88.
17. Niu, D.; Dai, S. A short-term load forecasting model with a modified particle swarm optimization algorithm and least squares support vector machine based on the denoising method of empirical mode decomposition and grey relational analysis. *Energies* **2017**, *10*, 408.
18. Liang, Y.; Niu, D.; Ye, M.; Hong, W.-C. Short-term load forecasting based on wavelet transform and least squares support vector machine optimized by improved cuckoo search. *Energies* **2016**, *9*, 827.
19. Kim, K.-H.; Park, J.-K.; Hwang, K.-J.; Kim, S.-H. Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems. *IEEE Trans. Power Syst.* **1995**, *10*, 1534–1539.
20. Suganthi, L.; Iniyan, S.; Samuel, A.A. Applications of fuzzy logic in renewable energy systems—A review. *Renew. Sustain. Energy Rev.* **2015**, *48*, 585–607.
21. Niu, D.-X.; Shi, H.-F.; Wu, D.D. Short-term load forecasting using bayesian neural networks learned by hybrid monte carlo algorithm. *Appl. Soft Comput.* **2012**, *12*, 1822–1827.
22. Sak, H.; Senior, A.; Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.
23. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with lstm. *Neural Comput.* **2000**, *12*, 2451–2471.
24. Williams, R.J.; Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.* **1989**, *1*, 270–280.
25. Yang, Y.; Yu, D.J.; Cheng, J.S.; Shi, M.L.; Yu, Y. Roller bearing fault diagnosis method based on emd and neural network. *J. Vib. Shock* **2005**, *1*.
26. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.-C. Tung, C.C.; Liu, H.H. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *R. Soc. Lond. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995.
27. Huang, B.; Kunoth, A. An optimization based empirical mode decomposition scheme. *J. Comput. Appl. Math* **2013**, *240*, 174–183.

28. An, X.; Jiang, D.; Zhao, M.; Liu, C. Short-term prediction of wind power using emd and chaotic theory. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 1036–1042.
29. Dong, Y.; Ma, X.; Ma, C.; Wang, J. Research and application of a hybrid forecasting model based on data decomposition for electrical load forecasting. *Energies* **2016**, *9*, 1050.
30. Mitchell, T.M. *Machine Learning*; McGraw Hill: Burr Ridge, IL, USA, 1997; Volume 45, pp. 230–247.
31. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378.
32. Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*; Springer Series in Statistics; Springer: Berlin, Germany, 2001; Volume 1.
33. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Wadsworth International Group, Chapman and Hall/CRC: Belmont, CA, USA, 1984.
34. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 21 June 1967; Volume 1, pp. 281–297.
35. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
36. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann:Burlington, MA, USA, 2016.
37. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 28, pp. 1310–1318.
38. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166.
39. Graves, A. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin, Germany, 2012; pp. 5–13.