Electrical Load Forecasting Model Based on Long-Short Term Memory in Day-Ahead Electricity Markets

Duc Nguyen Huu*, Dinh Van Thin
Electric Power University, Hanoi, Vietnam
*Corresponding Author Email: ducnh@epu.edu.vn

ABSTRACT

Electrical load forecasting plays an important role for participants in electricity markets. In electricity markets such as the day-ahead market and the intraday market, it is required to forecast load demand 24 hours in advance for participating in the day-ahead electricity market. Thus, the problem of predicting the electricity load 24 hours in advance plays a vital role in participating into the day-ahead market. Due to the non-linearity and instability under natural conditions of electrical loads in small-scale power systems, accurate forecasting is still a challenge. This paper introduces an estimation model of the electrical load using Long-Short-Term-Memory (LSTM) short-term memory structure based on feedback neural network structure to predict electrical loads 24 hours in advance.

KEYWORDS

Time series, day-ahead electricity market, recurrent neural network (RNN), long-short-term-memory (LSTM).

INTRODUCTION

Electrical load forecasting plays an important role in efficiently operating the electrical power system including generation, transmission and distribution systems. Accurate electrical load forecasting improves the safety of the electrical power system, optimizes costs of the generation, and reduces breakdowns of power transmission and distribution systems. In the electricity markets, load forecasting helps sellers and buyers determine the optimal electrical price for transactions. Electricity markets can be divided into day-ahead markets and intraday markets [1]. In order to help participants in the day-ahead market, it is necessary to forecast load demand 24 hours in advance. The electricity load forecasting problem is divided into 3 types depending on the forecast time: Long-term forecasting; Medium-term load forecast; and Short-term load forecasting. Short-term electricity load forecasting is the problem of predicting the electricity load from 15 minutes to the previous hour and day [2]. There have been many studies on electrical load forecasting models in recent years and some techniques are used in developing load forecasting models. A widely used technique for short-term load forecasting is the Support Vector Regression (SVR) method [3, 4].

This is a method developed from the Support Vector Machine (SVM) algorithm. The researchers create an SVR model for each hour of the day, thus 24 SVR models for a day. The input variables include: (1) historical electrical load data; (2) temperature; (3) humidity; (4) day of the week; (5) months. The advantage of this model is that the error is quite small. Due to the nature of the electrical loads being cyclical over time, many forecasting methods use historical data to forecast electrical loads such as AR (Autoregressive), MA (Moving Average), ARIMA (Autoregressive Integrated Moving Average) methods [5-7]. Among them, the ARIMA method is the most common and popular method. However, these methods all assume that the historical and future load data have a linear relationship with each other. This can lead to large errors in cases where the relationship is nonlinear. The authors in [8] improved the ARIMA method to predict nonlinear electrical loads according to the GARCH (Autoregressive Conditional Heteroskedastic) model. However, this method gives good results only for static data files [9, 10], while load data can be dynamic.

The above mentioned problems are solved with the application of artificial neural networks in electrical load forecasting. With the development of computer science, another technique that is widely used is artificial neural
network [11-13]. Artificial neural network is built with many layers including an input layer, an output layer, and hidden layer (in between an input layer and an output layer). In each hidden layer the user decides the number of neurons (number of nodes). Each neuron is set up an activation function to match the nonlinearity of the data, here is the electrical load. The weights of the neurons in the hidden layers are updated by using the Back Propagation algorithm. Papers [11, 14, 15, 16] present an improved application of artificial neural networks and Wavelet Neural Networks (WNN) in the problem of electrical load forecasting. The researchers found out historical data of a date that most closely resembles the forecast date, and then applied a filter to that day's electrical load (wavelet decomposition).

In these documents, the author decomposes the electrical load of a day into low- and high-order wave components, each of which is combined with weather data such as temperature, wind speed, humidity to form the input vector for two neural networks for each low- and high-order wave components, respectively. Then take the total output of these two neural networks to forecast the electrical load of the day. The above mentioned methods have some existence as several input data such as temperature, humidity are not available, and electrical load data are linear and fixed. Disadvantage of these models is complex because it is required to build the model for each step individually. Some models are only applied to a specific data set, not likely to apply to other datasets. Evaluation of results is sometimes based on only a small data set, which does not reflect the overall accuracy. From the above analysis, this paper proposes a short-term power load forecasting model applied to the day-ahead electricity market using a neural network architecture to generate feedback based on short- long term memory.

This proposed model can help overcome some of the above problems:

- The model does not use temperature data.
- The model applies to all days of the week, month of the year with a forecast step of an hour.
- The model can be used for different electrical load data sets (linear and non-linear, also dynamic loads).
- The error of the model is evaluated with the amount of data for all days of the week.

The structure of the paper is presented as follows: Section 2 presents an overview of some short-term load forecasting methods. Section 2 presents the electrical load forecasting method based on the short-term memory feedback neural network (LSTM). This method is used in this paper to build a predictive model of electricity load in 24 hours advance before participating in the electricity market during the day. The test simulation results are presented in Section 4. Some conclusions are made in Section 5.

RESEARCH METHOD AND MODEL

Recurrent Neural Networks

Many feed-forward neural networks, such as MLP (Multilayer Perceptron), DNN (Deep Neural Network), CNN (Convolutional Neural Network), etc., have achieved high performance in machine learning applications. The success of the models depends heavily on the assumption of independence between the training and test data sets [17]. When the data in a time series depend on each other or the assumption of independence is incorrect, the learning performance of the models is reduced due to the inability to model long-term dependencies. Time series forecasting is a typical scenario in which current data points are related to previous data points. Long time dependence is the basis for time series forecasting. In addition, feedforward neural networks restrict inputs and outputs to fixed-length vectors [18], which also makes them unsuitable for time series learning. In contrast, RNNs are specifically designed to operate on sequential or time series data [19]. Compared to a feedforward neural network that only allows signals to travel from input to output, RNN allows signals to go both forward and backward. RNNs introduce loops in the network and allow internal connections between hidden nodes. With the help of such internal connections, RNNs are more suitable for learning historical information of the electrical load to forecast electrical load data. In particular, RNNs make it possible to discover temporal relationships between distant data [20]. Figure 1 shows the unfolded architecture of the RNN.
Given the input time series $x = \{x_1, x_2, \ldots, x_T\}$, the RNN computes the hidden state sequence $h = \{h_1, h_2, \ldots, h_T\}$, as well as the output sequence $y = \{y_1, y_2, \ldots, y_T\}$ iterate using the following set of equations:

$$h_t = f(W_{hx} x_t + W_{hh} h_{t-1} + b_h)$$  
$$y_t = g(W_{yh} h_t + b_y)$$  

In equations (1) - (2), $W_{hx}$, $W_{hh}$ and $W_{yh}$ represent the input-hidden weight matrix, the hidden-hidden weight matrix and the output-hidden weight matrix, respectively. The vectors $b_h$ and $b_y$ represent the offset of the hidden layer and the output layer, respectively. In addition, $f(\cdot)$ and $g(\cdot)$ are activation functions for the hidden layer and the output layer, respectively. The RNN uses the hidden state $h_t$ at step $t$ to remember the network. The hidden state captures all the information contained in the previous time steps. The multi-step pre-time series prediction shows multi-step dependence because the prediction of the out-of-sample data $x_t + h$ depends on the input data observed at previous time points $t_e$, where $t_e < t + h$. However, as the data-dependent time period increases, simple RNNs tend to be increasingly affected by the gradient vanishing problem [21]. In other words, the influence of the input data at $t_e$ on the forecast data $x_{t + h}$ decreases rapidly with time $t + h - t_e$. Therefore, simple RNNs may not be the most suitable method in forecasting problems with long-term dependencies.

**Long-Short-Term Memory Method**

LSTM is an efficient RNN architecture introduced by Hochreiter and Schmidhuber in 1997 [22] and improved by several other researchers since then [23]. LSTM is primarily promoted and designed to overcome the vanishing gradient problem of the standard RNNs when dealing with long-term dependencies. In the standard RNN, the overall neural network is a sequence of repeating modules formed as a sequence of simple hidden networks, such as a single sigmoid layer. In contrast to the standard RNN which has a series of repeating modules with a relatively simple structure, the hidden layers of the LSTM have a more complex structure. In particular, LSTM introduces the concepts of gates and cells in each hidden layer. A memory block mainly consists of four parts: input port $i$, forget port $f$, output port $O$, and self-connecting memory cells $C$. The input gate controls the input of triggers into the memory cell. The output port learns which cell to activate for filtering and outputs to the next network. The forget gate helps the network to forget historical input data and reset the memory cells. Additionally, kernel gates are carefully applied so that memory cells can access and store information for a long period of time. Such a structure can effectively reduce the vanishing derivative problem [24].

**Recurrent Neural Network based on Long-Short-Term Memory**

Considering the advantages of the LSTM method in time series forecasting, this paper uses the RNN scheme based on LSTM to predict electrical load in day-ahead.
Given an input time series \( x = \{x_1, x_2, \ldots, x_T\} \), the LSTM maps the input time series to two output time series \( h = \{h_1, h_2, \ldots, h_T\} \) and \( y = \{y_1, y_2, \ldots, y_T\} \) iterates by updating the state of the memory cell with the following procedure. First, as shown in Figure 2, a forget gate is applied to help the LSTM decide what information to remove from the cell state. A sigmoid function \( \sigma (\cdot) \) is used to calculate the activation of the forget gate as:

\[
 f_t = \sigma(W_c x_t + W_h h_{t-1} + W_C C_{t-1} + b_f)
\]  

The \( f_t \) output of Equation (3) is a value between 0 and 1 that corresponds to the final cell state \( C_{t-1} \). A value of 0 means a completely final state, while a value of 1 stands for the final state.

Next, it is required to inform the LSTM what new information should be stored in the new cell state. To begin with, the LSTM uses a sigmoid layer, named the input gate layer, where:

\[
 i_t = \sigma(W_i x_t + W_h h_{t-1} + W_i C_{t-1} + b_i)
\]  

Equation (4) decides what information to update. The sigmoid class \( g(\cdot) \) constructs a vector \( U_t \) to store the new candidate values to be added to the new cell state as the following equation:

\[
 U_t = g(W_c x_t + W_h h_{t-1} + W_C C_{t-1} + b_f)
\]  

Then the old cell state \( C_{t-1} \) is updated to the new cell state \( C_t \) with estimated \( f_t \) and \( U_t \). In particular, the old cell state is multiplied by \( f_t \) to forget the information from the last state. The candidate values are multiplied by the input gate layer to decide how much new information should be updated to the new cell state, which leads to the following equation:

\[
 C_t = U_t i_t + C_{t-1} - f_t
\]  

Another sigmoid layer \( \sigma(\cdot) \) is then used as the output port to filter and output the cell state as \( o_t \), where:

\[
 o_t = \sigma(W_o x_t + W_h h_t + W_o C_{t-1} + b_o)
\]  

Furthermore, a cell output sigmoid activation function \( \ell(\cdot) \) is applied on the cell state, which is then multiplied by the output \( o_t \) to give the desired information shown by the following equation:

\[
 h_t = \sigma_o \ell(C_t)
\]
For the output of the memory block, an output activation function $k(\cdot)$ is used as the following equation:

$$y_t = k(W_{yh}h_t + b_y)$$  \((9)\)

In equations (3) - (9), the matrices $W_{ix}$, $W_{fx}$, $W_{ox}$, $W_{cx}$ are the appropriate input weight matrices, $W_{ih}$, $W_{fh}$, $W_{oh}$, $W_{ch}$ are the repeated weight matrices; $W_{yh}$ represents the hidden output weight matrix; the vectors $b_i$, $b_f$, $b_o$, $b_c$ are the bias vectors, respectively.

RESULTS AND DISCUSSION

Test Data

In order to verify the proposed model, the data used to run the model is the electrical load data of Italy from 2020 to 2021 [25]. Data is sampled every hour. The dataset is composed of two parts: set of training data (Italian electrical load in 2020), test data (Italian electrical load for 1 week from 25/01/2021 to 01/02/2021). The LSTM algorithm is applied to predict the power load for a next time step $X_{t+1}$, using the previous time steps $\{X_{t-d}, X_{t-d+1},..., X_t\}$. The model then updates the actual value at time step $X_{t+1}$ to predict the next step $X_{t+2}$. And so on until the last time step in the test dataset.

The two evaluation criteria used are RMSE (root mean square error) and MAPE (mean absolute percentage error) between the actual value and the predicted value.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} [L_{\text{pred}}(i) - L_{\text{true}}(i)]^2}$$ \((10)\)

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^{N} \frac{|L_{\text{pred}}(i) - L_{\text{true}}(i)|}{L_{\text{true}}(i)}$$ \((11)\)

Where: $N$ is number of steps in the Test dataset; $L_{pred(i)}$: predictive load at step i; $L_{true(i)}$: actual load at step i.

Case Study

The simulation results to verify the proposed model and compare with the actual power load data of the days in a week are shown from Figure 4 to Figure 12. Figure 4 gives the forecast results of the model in a week from January 25, 2021 to February 01, 2021. The blue line is the actual load data and the red line is the model's forecast results in 15 minute increments. Figure 5 shows the forecast results of the model and actual data on Monday (January 25, 2021). We can see that the largest error is concentrated on the time of 12:30 to 13:30 and the time frame from 17:00 to 18:00 of the day. Figure 6 shows the forecast results for Tuesday (January 26, 2021). Similarly, the forecast results for Wednesdays, Thursdays, and Fridays are shown in Figures 7, 8, and 9 respectively. Forecast results for weekends are shown in Figures 10 and 11.
Figure 4. Comparison chart between the predicted power load value from the model and the actual load in 1 week from January 25, 2021 to February 01, 2021

Figure 5. Comparison chart between the predicted power load value from the model and the actual load on Monday, January 25, 2021

Figure 6. Comparison chart between the predicted electric load value from the model and the actual load on Tuesday, January 26, 2021
Figure 7. Comparison chart between the predicted power load value from the model and the actual load on Wednesday January 27, 2021

Figure 8. Comparison chart between the predicted power load value from the model and the actual load on Thursday, January 28, 2021

Figure 9. Comparison chart between the predicted electric load value from the model and the actual load on Friday, January 29, 2021
Electrical Load Forecasting Model Based on Long-Short Term Memory in Day-Ahead Electricity Markets

**Figure 10.** Comparison chart between the predicted power load value from the model and the actual load on Saturday, January 30, 2021

**Figure 11.** Comparison chart between the predicted power load value from the model and the actual load on Sunday, February 01, 2021

**Table 1.** Errors of the model's forecasted load with 1-week data set

<table>
<thead>
<tr>
<th>Types</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>1.463</td>
</tr>
<tr>
<td>MAPE</td>
<td>3.03%</td>
</tr>
</tbody>
</table>

The simulation results of the model are summarized in Table 1. Table 1 shows the error value of the model's forecast results on the 1-week data of the verification. The RMSE error value is 1.463% and the MAPE error value is 3.03%.

**CONCLUSIONS**

This article proposes and builds a model to forecast the electricity load 24 hours in advance to participate in the day-ahead electricity market. The model uses a neural network based on long-short-term memory (LSTM) feedback. The advantage of the model is that it simply does not require data such as temperature, humidity, etc. The model can be applied with only historical data on the power consumption of the load. Thus, the model can be applied in the case of electrical load forecasting where data on temperature and humidity are not available. Power load characteristics of a historical day are collected and inputted into the model so that the electricity load can be forecast 24 hours in advance. The simulation results of the model is verified with Italy's 1-week power load data set with predicted errors of about 3%. The model is simple to apply because it does not need data on temperature and humidity, thus it is easy to apply to practical conditions. The model can also be applied to medium and long-term load forecasting (in weeks) by using forecast data of the $N$th step, to generate input data for the $(N+1)$th step, and so on until the $(N+n)$th step. The model is especially necessary when power buyers participate in the day-ahead electricity market. In addition, this forecasting model can be integrated into the
power management systems of factories and buildings (PMS) to predict electricity consumption for the coming hours.

REFERENCES


