Electric Load Forecasting in Smart Grid Using Long-Short-Term-Memory based Recurrent Neural Network

Jian Zheng, Cencen Xu, Ziang Zhang and Xiaohua Li
Department of Electrical and Computer Engineering
State University of New York at Binghamton
Binghamton, NY 13902
Email: {jzheng65,cxu32,zhangzia,xli}@binghamton.edu

***Abstract*—Electric load forecasting plays a vital role in smart grid. Short term electric load forecasting predicts the load that is several hours to several weeks ahead. Due to the nonlinear, non-stationary and nonseasonal nature of the electric load time series, accurate forecasting is challenging. This paper explores Long-Short-Term-Memory (LSTM) based Recurrent Neural Network (RNN) to deal with this challenge. LSTM-based RNN is able to exploit the long term dependencies in the electric load time series for more accurate forecasting. Experiments are conducted to demonstrate that LSTM-based RNN is capable of forecasting accurately the complex electric load time series with a long forecasting horizon. Its performance compares favorably to many other forecasting methods.**

***Index Terms*—Electric load forecasting, univariate time series, smart grid, recurrent neural network (RNN), long-short-term-memory (LSTM)**

I. Introduction

Compared with the traditional power grid, smart grid is capable of providing more intelligent, efficient, sustainable and reliable power service by making use of the advanced infrastructure and information technology. Electric load forecasting plays an increasingly indispensable role in smart grid. It is of fundamental importance for utility providers to model and forecast power loads in advance, to strike a balance between production and demand, to decrease the production cost, and to implement various pricing schemes for demand response.

Based on the duration of the forecasting horizon, electric load forecasting is classified into four categories, i.e., long term load forecasting, medium term load forecasting, short term load forecasting, and very short term load forecasting [1]. Short term load forecasting predicts load that is several hours to several weeks ahead based on the observed load time series data [2].

Since electric load is primarily an univariate time series [3], many general time series forecasting methods can be applied for electric load forecasting. A general class of methods is the statistical forecasting models that include Autoregressive (AR), Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA) models [4] [5] and a number of their variants [6]. In particular, the ARIMA is one of the most popular and commonly used methods for time series forecasting. However, these methods work under the assumption that the observed time series and the future time series are linearly related, which makes them less effective for time series with significant nonlinear characteristics. There have been plenty of work to extend their applications into nonlinear forecasting, such as the Autoregressive Conditional Heteroskedastic (GARCH) model [7]. Nevertheless, these statistical models only have good prediction performance over stationary data [8] [9], while the electric load time series may be nonstationary.

Another class of forecasting methods are based on artificial Neural Networks (ANNs). ANNs have become immensely popular in electric load forecasting in the past decade. Basically, ANNs mimic the human brain to learn regularities and patterns automatically from the past experience and produce generalized results. In contrast to the ARIMA-based linear forecasting methods, ANNs are a set of nonlinear self-adaptive methods that are driven by data, which means there is no need of any prior knowledge of the relationship between the models and the data variables. It is well known that ANNs are capable of approximating any nonlinear function. ANNs can usually achieve reasonable results, especially for complicated models and time series [10]. There is an extensive literature on using ANNs for electric load forecasting, such as feed-forward Multilayer Perceptron (MLP) [11], nonlinear autoregressive models with exogenous input (NARX) Neural Network [12], Generalized Regression Neural Network (GRNN) [13], Support Vector Regression (SVR) [14], etc.

Although extensive research have been done, accurate electric load forecasting remains a challenge in smart grid. Electric load forecasting is usually an univariate time series forecasting problem that is more challenging than the corresponding multivariate time series forecasting problem. Because there is no additional information from other data sources that can be utilized for learning [15]. In addition, compared with linear, stationary and seasonal time series, electric load time series are nonlinear, nonstationary and nonseasonal, where nonseasonal means without apparent periodicity in time. It is difficult to forecast accurately such time series in a long time horizon. Therefore, more efforts are needed to develop more effective forecasting methods.

This work is supported by the National Science Foundation under Grant CNS-1443885.

In this paper, we tackle this challenge using the Long-Short-Term-Memory (LSTM) [16], which is a special recurrent neural network (RNN) architecture that can be utilized to learn temporal sequences and long term dependencies more accurately than Deep Neural Networks (DNNs) and conventional RNNs [17]. We will develop the novel electric load forecasting scheme based on LSTM. This scheme is capable of forecasting accurately the complex nonlinear, non-stationary and nonseasonal univariate electric load time series over a long forecasting horizon.

The remainder of this paper is organized as follows. The univariate time series forecasting problem is introduced in Section II. In Section III, the electric load forecasting scheme with LSTM-based RNN is presented. Experiments and conclusions are given in Sections IV and V, respectively.

II. Multi-step Ahead Time Series Forecasting

Considering an univariate electric load time series with N observations $\{X_{t_1}, X_{t_2}, \cdots, X_{t_N}\}$, the task of multi-step ahead forecasting is to utilize these N recorded data points to predict the next H data points $\{X_{t_{N+1}}, X_{t_{N+2}}, \cdots, X_{t_{N+H}}\}$ in future of the existing time series. The parameter $H > 1$ is the forecasting horizon. Even in short term electric load forecasting, the forecasting horizon can be very large, i.e., $H \gg 1$. Typically, the smart grid electric load data are obtained via smart meters or PMU (power measurement unit). If the smart meters have a sampling interval of 15 minutes, the forecasting horizon is $H = 96$ for 24-hour (one-day) ahead load forecasting. PMU has even higher sampling frequency with sampling interval in sub-second, and the forecasting horizon can be extremely long.

A. Recursive Strategy

There are three strategies that are commonly used to conduct multi-step ahead time series forecasting, i.e., recursive strategy, direct strategy, and multiple-input and multiple-output (MIMO) strategy [18]. The most intuitive and traditional forecasting strategy is the recursive strategy [19], where a one-step ahead time series forecasting method is applied with a single forecasting model $f(\cdot)$. Specifically,

$$X_{t+1} = f(X_t, X_{t-1}, \cdots, X_{t-d+1}) + \epsilon, \tag{1}$$

where $t \in \{d, d+1, \cdots, N\}$, d is the dimension of the estimator, ϵ is the additive noise, the estimator $f : \mathbf{R}^d \to \mathbf{R}$ and $\mathbf{R}$ denotes the real field. In order to forecast H steps ahead, we first forecast the one-step ahead estimation $X_{t_{N+1}}$ using (1). Then, with the forecasted $X_{t_{N+1}}$ as part of the input time series, the next step is to estimate $X_{t_{N+2}}$ using the same one-step ahead forecasting model (1). This procedure runs recursively until we have estimated $X_{t_{N+H}}$.

Although the recursive strategy is intuitive and is easy to apply, it is sensitive to the accumulation of forecasting errors, especially when the forecasting horizon is large. In the recursive strategy, the forecasting errors in previous steps are propagated and accumulated to deteriorate the subsequent forecasting accuracy [18].

B. Direct Strategy

Another strategy for multi-step ahead forecasting is the direct strategy [20]. Unlike the recursive strategy, the direct strategy constructs H different forecasting models for each forecasting horizon based on the observed time series data. Specifically,

$$X_{t+h} = f_h(X_t, X_{t-1}, \cdots, X_{t-d+1}) + \epsilon_h, \tag{2}$$

where $h \in \{1, 2, \cdots, H\}$, f_h is the hth forecasting model, and ϵ_h is the additive noise associated with the hth model. Since the direct strategy does not use any forecasted value as input for forecasting, it is not prone to accumulated errors. However, the H forecasting models are trained separately and independently from each other, which may result in conditional independence among the H forecasted values [21]. Such an independence effect will prevent the forecasting methods from reflecting the statistical dependency among the forecasted data, which will degrade the forecasting performance.

C. Multiple-input and Multiple-output (MIMO) Strategy

Both the recursive strategy and the direct strategy are considered as single output strategy because they map multiple inputs (a vector) to a single output (a scalar) [18]. In contrast, the MIMO strategy is a forecasting strategy that uses multiple inputs to create multiple outputs [22]. With the MIMO strategy, the forecasting result is a time series (a vector) instead of a scalar. All the data in this output vector are generated by the same model trained using the same observed time series data. Specifically,

$$\{X_{t+1}, X_{t+2}, \cdots, X_{t+H}\} = F(X_t, X_{t-1}, \cdots, X_{t-d+1}) + \epsilon, \tag{3}$$

where ϵ is the noise vector and $F : \mathbf{R}^d \to \mathbf{R}^H$. Compared with the single output strategies, the MIMO strategy is capable of mitigating the conditional independence problem. It has the advantage of preserving the temporal statistical dependency in the forecasted time series. However, since the MIMO strategy forecasts all the data with the same forecasting model, its flexibility and variability may not be as strong as the other forecasting strategies [23].

III. LSTM-based RNN for Electric Load Forecasting

A. Recurrent Neural Networks (RNNs)

Many feedforward neural networks, such as MLP (Multi-layer Perceptron), DNN (Deep Neural Network), CNN (Convolutional Neural Network), etc., have achieved state-of-the-art performance in various supervised or unsupervised machine learning applications. Their success highly depends on the independence assumption among the training and test data [24]. When the data in a time series depend on each other or the independence assumption fails, their learning performance will degrade due to their insufficient capability of modeling long term dependencies. Time series forecasting is such a typical scenario where the current data points are related with the previous data points. Long time dependence is in fact the basis for time series forecasting. In addition, the feedforward neural networks constrain the inputs and targets to be vectors of fixed length [25], which also makes them inconvenient for sequence (such as time series) learning.

In contrast, RNNs are designed specifically to operate over sequential data or time series [26]. Compared with the feedforward neural networks that only allow signals to travel forward from the input to the output, RNNs allow signals to travel both forward and backward. They introduce loops in the network and allow internal connections among hidden units. With the help of such internal connections, RNNs are more suitable for exploiting the information in the past data to forecast the future data. In particular, RNNs make it possible to explore temporal relationships among the data that are far away from each other [27].

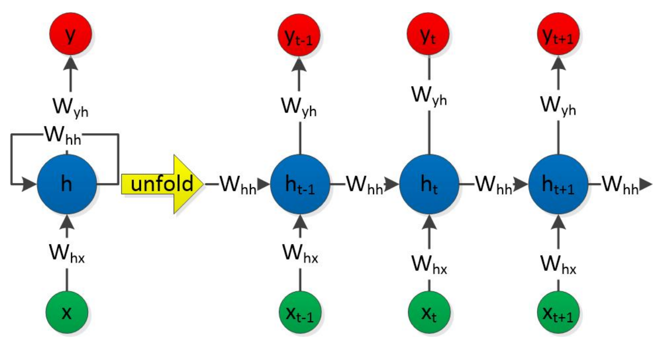

Fig. 1: The architecture of RNN.

Fig. 1 shows the architecture of RNN. Given an input time series $\mathbf{x} = \{x_1, x_2, \cdots, x_T\}$, the RNN computes the hidden state sequence $\mathbf{h} = \{h_1, h_2, \cdots, h_T\}$ as well as the output sequence $\mathbf{y} = \{y_1, y_2, \cdots, y_T\}$ iteratively using the following set of equations

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{4}$$
$$y_t = g(W_{yh}h_t + b_y). \tag{5}$$

In (4)-(5), W_{hx}, W_{hh} and W_{yh} denote the input-hidden weight matrix, the hidden-hidden weight matrix, and the hidden-output weight matrix, respectively. The vectors b_h and b_y represent the bias of the hidden layer and the output layer, respectively. In addition, $f(\cdot)$ and $g(\cdot)$ are the activation functions for the hidden layer and the output layer, respectively. The RNN uses the hidden state h_t at time step t to memorize the network. The hidden state captures all the information included in the previous time steps.

Multi-step-ahead time series forecasting shows the multi-step dependencies because the forecast of the out-of-sample data x_{t+h} depends on the input data observed at much earlier time t_e, where $t_e \ll t+h$. Nevertheless, when the interval of data dependencies increases, the simple RNNs tend to suffer increasingly heavily from the gradient vanishing problem [28]. In other words, the influence of the input data at t_e to the forecasted data x_{t+h} decays quickly over time $t+h-t_e$. Therefore, the simple RNNs may not be the best choice in forecasting problems with long term dependencies.

B. *LSTM (Long-Short-Term-Memory) Architecture*

LSTM is an efficient RNN architecture introduced by Hochreiter and Schmidhuber in 1997 [16] and refined by many people since then [29]. LSTM was mainly motivated and designed to overcome the vanishing gradients problem of the standard RNN when dealing with long term dependencies.

In the standard RNN, the overall neural network is a chain of repeating modules formed as a series of simple hidden networks, such as a single sigmoid layer. In contrast to the standard RNN which has a series of repeating modules with relatively simple structure, the hidden layers of LSTM have a more complicated structure. Specifically, LSTM introduces the concepts of gate and memory cells in each hidden layer. A memory block mainly consists of four parts: an input gate i, a forget gate f, an output gate o, and self-connected memory cells C. The input gate controls the entry of the activations to the memory cell. The output gate learns when to output the activations to the successive network. The forget gate helps the network to forget the past input data and reset the memory cells. In addition, multiplicative gates are applied carefully to make it possible for the memory cells to access and store the information over a long time interval. Such a structure can effectively mitigate the vanishing gradient problem [30]. This makes LSTM an architecture suitable for problems with long term dependencies.

Since the gates can not get any information from the memory cell output when the output gate is closed, the LSTM does not know how long the memory should be for the model. To resolve this problem, peephole connections can be added to the LSTM memory cells. Working as an immediate supervisor, peephole connections make it possible for all the gates to inspect the cell states [31]. Fig. 2 shows the architecture of a general LSTM memory block with peephole connections added.

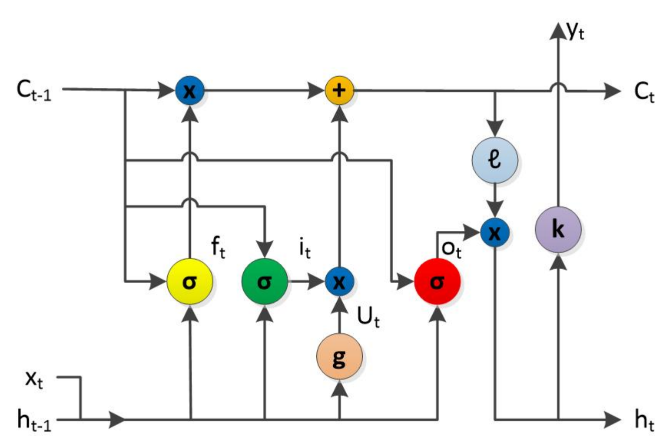

Fig. 2: The architecture of LSTM memory block with peephole connections.

C. *LSTM-based RNN Forecasting Scheme*

Considering the advantages of LSTM in time series forecasting, we use the LSTM-based RNN scheme in this paper for electric load time series forecasting. The scheme applies the LSTM with peephole connections.

Given an input time series $\mathbf{x} = \{x_1, x_2, \cdots, x_T\}$, the LSTM maps input time series to two output time sequences $\mathbf{h} = \{h_1, h_2, \cdots, h_T\}$ and $\mathbf{y} = \{y_1, y_2, \cdots, y_T\}$ iteratively by updating the states of memory cells with the following procedure.

First, as per Fig. 2, the forget gate is applied to help the LSTM to decide what information to throw away from the cell state. A sigmoid function $\sigma(\cdot)$ is used to calculate the activation of the forget gate as

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}C_{t-1} + b_f). \tag{6}$$

The output f_t of (6) is a value between 0 and 1 corresponding to the last cell state C_{t-1}. The value 0 means forgetting the last state completely, while the value 1 stands for keeping the last state completely.

Next, we need to let the LSTM know what new information is going to be stored in the new cell state. To begin with, the LSTM uses a sigmoid layer, which is named as the input gate layer i_t, where

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}C_{t-1} + b_i), \tag{7}$$

to decide what information to update. The sigmoid layer $g(\cdot)$ constructs a vector U_t to store the new candidate values to be added to the new cell state as

$$U_t = g(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{8}$$

Then, the old cell state C_{t-1} is updated to a new cell state C_t with the estimated f_t and U_t. Specifically, the old cell state is multiplied with f_t in order to forget information from the last state. The candidate values is multiplied with the input gate layer to decide how much new information to be updated to the new cell state, which gives

$$C_t = U_t i_t + C_{t-1} f_t. \tag{9}$$

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

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}C_{t-1} + b_o). \tag{10}$$

Furthermore, a cell output sigmoid activation function $\ell(\cdot)$ is applied over the cell state, which is then multiplied by the output o_t to give the desired information

$$h_t = o_t \ell(C_t). \tag{11}$$

As for the output of the memory block, an output activation function $k(\cdot)$ is used, i.e.,

$$y_t = k(W_{yh}h_t + b_y). \tag{12}$$

In (6)-(12), 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 recurrent weight matrices, W_{yh} represents the hidden output weight matrix, W_{ic}, W_{fc}, W_{oc} denote the weight matrices of peephole connections. The vectors b_i, b_f, b_o, b_c, b_y are the corresponding bias vectors.

IV. Experiment Evaluations

A. Experiment Setup

In this section, we present our experiments on applying the LSTM-based RNN scheme for electric load forecasting. We compared the proposed **LSTM**-based RNN scheme with the following methods: **SARIMA** which is the Seasonal Autoregressive Integrated Moving Average model [32]; **NARX** which is a nonlinear autoregressive neural network model with exogenous inputs [12]; **SVR** (Support Vector Regression) which is a very popular model in financial time series forecasting [14]; and **NNETAR** which is a feed-forward neural network model for univariate time series forecasting with a single hidden layer and lagged inputs. Two evaluation criteria were used as performance metric: root mean square error (RMSE) and mean absolute percentage error (MAPE) between real values and forecasting results.

Since most of the methods in comparison are developed for general time series forecasting, for fair comparison we compared their performance with two data sets: an electric load data set that we collected, and an airline passengers data set that is used widely as benchmark for algorithm evaluation. The airline passengers data set is a time series describing the monthly totals of the international airline passengers [33]. This data set includes 144 observations in total for 12 years. As shown in Fig. 3, there is an apparent upward trend and strong seasonal variations. As a result, this data set is helpful for us to examine the performance of our scheme in forecasting short time series with multiplicative seasonal patterns.

The electric load data set is an univariate time series describing the electricity consumption in our school's engineering building. This data set contains power consumption samples of the building recorded every 15 minutes. As shown in Fig. 5, this data set is a strong non-stationary and non-seasonal time series, which poses a great challenge for long-horizon time series forecasting.

B. Experiment Results with the Airline Data Set

The experiments on the international airline passengers data set had a forecasting horizon $H = 12$. It can be seen from Fig. 3 that, compared with the original time series, all the five methods followed the upper trend and the seasonal pattern to variant extents. Fig. 4 shows more details of the forecasting results. We can clearly see that **LSTM** and **SARIMA** both had superior performance over **SVR** and **NNETAR**. This is also verified from Table I which shows that **LSTM** and **SARIMA** achieved better forecasting performance with smaller RMSE and MAPE. In particular, although **NARX** had a similar RMSE score as **LSTM**, **LSTM** obtained better MAPE score and thus had better forecasting accuracy.

The **SARIMA** had the best performance in this case. The reason lies in that the airline passenger data set has a strong multiplicative seasonal pattern and a clear upper trend. The **SARIMA** exploited these regular seasonal patterns by using logarithmic transformation as well as seasonal differencing [34]. These special and data-dependent pre-processing techniques made it easier for SARIMA to forecast. Obviously, such advantage is highly data dependent, not general for

TABLE I: Forecasting RMSE and MAPE for the Airline Passenger Data Set

FORECASTING METHODS	RMSE	MAPE
SVR	0.0717	0.0556
NNETAR	0.0799	0.0595
LSTM	**0.0435**	**0.0345**
NARX	0.0452	0.0403
SARIMA	**0.0359**	**0.0256**

TABLE II: RMSE and MAPE results for the Electricity Consumption Data Set

FORECASTING METHODS	RMSE	MAPE
SVR	0.2044	0.1775
NNETAR	0.1952	0.1689
LSTM	**0.0702**	**0.0535**
NARX	0.1446	0.1192
SARIMA	**0.2537**	**0.2001**

other data sets. In terms of generality, **LSTM** should still be better because it achieved the performance closest to SARIMA without applying any special data-dependent pre-processing.

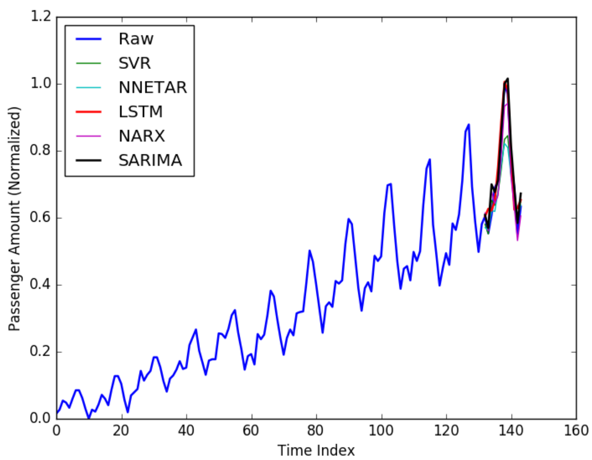

Fig. 3: In-sample and out-of-sample comparison for the international airline passengers data set.

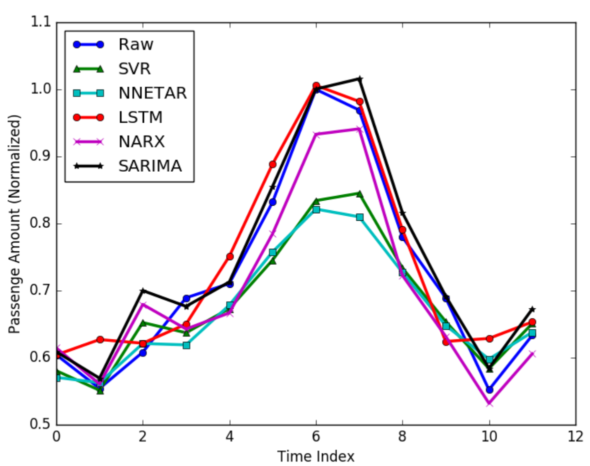

Fig. 4: Forecasting comparison for the international airline passengers data set.

C. Experiment Results with the Electricity Consumption Data Set

In the experiments, we used 904 data samples of the electricity consumption data set to forecast the electrical consumption $H = 96$ steps ahead. In other words, we used the electricity consumption of the past 10 days to forecast the electricity consumption of the next day. It can be seen from Fig. 5 that, compared with the airline passenger time series, this electric load time series is more complex, without any obvious seasonal pattern or trend. The non-stationarity and non-seasonality pose severe challenge for conventional methods to forecast. The relatively longer forecasting horizon, i.e., $H = 96$, makes accurate forecasting even more challenging.

Experiment results in Fig. 5 show that **LSTM** forecasted quite well compared with the original time series. Fig. 6 shows more clearly that **LSTM** outperformed all the other methods with the best forecasted time series. Table II shows that **LSTM** outperformed all the other methods with the smallest forecasting errors. In this complex electric load forecasting scenario, the performance of the other four methods was considered in general unsatisfactory. Although **NARX** captured the general trend of the real time series, the forecasting result was quite spurious, resulting in inaccurate forecasting with large RMSE and MAPE. **SARIMA** did not work well either in this longer-horizon forecasting due to non-stationarity and non-seasonality.

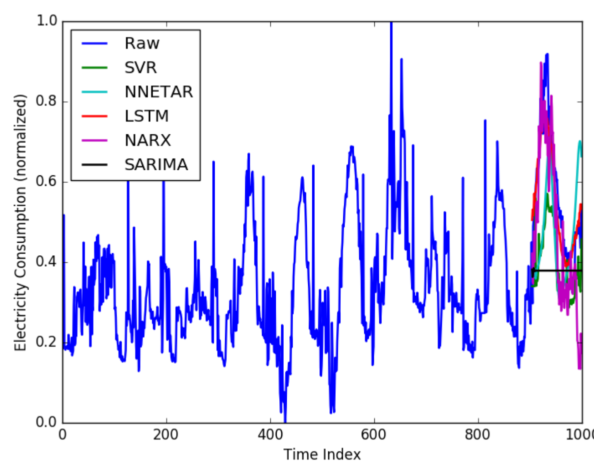

Fig. 5: In-sample and out-of-sample comparison for the electricity consumption data set.

V. Conclusions

In this paper, we propose to use the Long-Short-Term-Memory (LSTM) based Recurrent Neural Network (RNN) to address the challenging short term electric load forecasting problem. By exploiting the long term dependencies in the

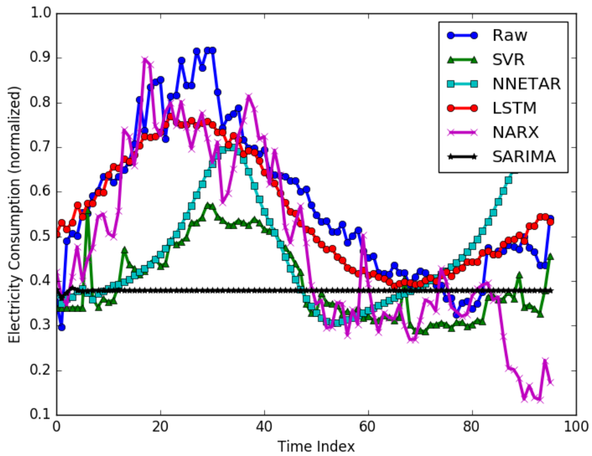

Fig. 6: Forecasting comparison for the electricity consumption data set.

time series, LSTM is capable of forecasting complex univariate electric load time series with strong non-stationarity and non-seasonality. Experiments are conducted with a short benchmark international airline passenger data set and a long electricity consumption data set. Experiment results show that the LSTM-based forecasting method can outperform most traditional forecasting methods in the challenging short term electric load forecasting problem.

References

[1] C. Nataraja, M. Gorawar, G. Shilpa, and J. S. Harsha, "Short term load forecasting using time series analysis: A case study for karnataka, india," *International Journal of Engineering Science and Innovative Technology*, vol. 1, pp. 45–53, 2012.

[2] L. Hernández, C. Baladrón, J. M. Aguiar, L. Calavia, B. Carro, A. Sánchez-Esguevillas, F. Pérez, Á. Fernández, and J. Lloret, "Artificial neural network for short-term load forecasting in distribution systems," *Energies*, vol. 7, no. 3, pp. 1576–1598, 2014.

[3] E. Almeshaiei and H. Soltan, "A methodology for electric power load forecasting," *Alexandria Engineering Journal*, vol. 50, no. 2, pp. 137–144, 2011.

[4] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[5] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and arima models for time series forecasting," *Applied Soft Computing*, vol. 11, no. 2, pp. 2664–2675, 2011.

[6] K.-Y. Chen and C.-H. Wang, "A hybrid sarima and support vector machines in forecasting the production values of the machinery industry in taiwan," *Expert Systems with Applications*, vol. 32, no. 1, pp. 254–264, 2007.

[7] C.-L. Hor, S. J. Watson, and S. Majithia, "Daily load forecasting and maximum demand estimation using arima and garch," in *Probabilistic Methods Applied to Power Systems, 2006. PMAPS 2006. International Conference on*. IEEE, 2006, pp. 1–6.

[8] K. M. Kam, *Stationary and non-stationary time series prediction using state space model and pattern-based approach*. The University of Texas at Arlington, 2014.

[9] M. Lineesh, K. Minu, and C. J. John, "Analysis of nonstationary nonlinear economic time series of gold price: A comparative study," in *International Mathematical Forum*, vol. 5, no. 34. Citeseer, 2010, pp. 1673–1683.

[10] S. Panigrahi, Y. Karali, and H. Behera, "Time series forecasting using evolutionary neural network," *International Journal of Computer Applications*, vol. 75, no. 10, 2013.

[11] C. García-Ascanio and C. Maté, "Electric power demand forecasting using interval time series: A comparison between var and imlp," *Energy Policy*, vol. 38, no. 2, pp. 715–725, 2010.

[12] M. Espinoza, J. A. Suykens, R. Belmans, and B. De Moor, "Electric load forecasting," *IEEE Control Systems*, vol. 27, no. 5, pp. 43–57, 2007.

[13] Y. Li and J. Wang, "The load forecasting model based on bayes-grnn," *Journal of Software*, vol. 7, no. 6, pp. 1273–1280, 2012.

[14] 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.

[15] J. Du Preez and S. F. Witt, "Univariate versus multivariate time series forecasting: an application to international tourism demand," *International Journal of Forecasting*, vol. 19, no. 3, pp. 435–451, 2003.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *INTERSPEECH*, 2014, pp. 338–342.

[18] S. B. Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert systems with applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[19] C. Hamzaçebi, D. Akay, and F. Kutay, "Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3839–3844, 2009.

[20] D. R. Cox, "Prediction by exponentially weighted moving averages and related methods," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 414–422, 1961.

[21] N. H. An and D. T. Anh, "Comparison of strategies for multi-step-ahead prediction of time series using neural network," in *2015 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE, 2015, pp. 142–149.

[22] G. Bontempi, "Long term time series prediction with multi-input multi-output local learning," *Proc. 2nd ESTSP*, pp. 145–154, 2008.

[23] S. B. Taieb, A. Sorjamaa, and G. Bontempi, "Multiple-output modeling for multi-step-ahead time series forecasting," *Neurocomputing*, vol. 73, no. 10, pp. 1950–1957, 2010.

[24] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[26] L. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, 2001.

[27] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks." *ICML (3)*, vol. 28, pp. 1310–1318, 2013.

[28] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[29] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[30] A. Graves, "Neural networks," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 15–35.

[31] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[32] Y. Chakhchoukh, P. Panciatici, and L. Mili, "Electric load forecasting based on statistical robust methods," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 982–991, 2011.

[33] J. Faraway and C. Chatfield, "Time series forecasting with neural networks: a comparative study using the airline data," *Applied statistics*, pp. 231–250, 1998.

[34] R. Adhikari and R. Agrawal, "An introductory study on time series modeling and forecasting," *arXiv preprint arXiv:1302.6613*, 2013.