2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Banff Center, Banff, Canada, October 5-8, 2017

Long Short Term Memory Networks for Short-Term Electric Load Forecasting

Apurva Narayan* and Keith W. Hipel†
* Department of Electrical and Computer Engineering,
† Department of Systems Design Engineering,
University of Waterloo, Waterloo, ON Canada N2L 3G1
Email: *apurva.narayan@uwaterloo.ca, †kwhipel@uwaterloo.ca

***Abstract*—Short-term electricity demand forecasting is critical to utility companies. It plays a key role in the operation of power industry. It becomes all the more important and critical with increasing penetration of renewable energy sources. Short-term load forecasting enables power companies to make informed business decisions in real-time. Demand patterns are extremely complex due to market deregulation and other environmental factors. Although there has been extensive research in the area of short-term electrical load forecasting, difficulties in implementation and lack of transparency in results has been cited as a main challenge.**

Deep neural architectures have recently shown their ability to mine complex underlying patterns in various domains. In our work, we present a deep recurrent neural architecture to unearth the complex patterns underlying the regional demand profiles without specific insights from the utilities. The model learns from historical data patterns. We show that deep recurrent neural network with long-short term memory architecture presents a robust methodology for accurate short term load forecasting with the ability to adapt and learn the underlying complex features over time. In most cases it matches the performance of the latest state-of-the-art techniques and even supercedes it in a few cases.

I. Introduction

Short-term load forecasting is a critical component for the reliable and secure operation of large power systems. High accuracy in load forecasting for power systems improves efficiency of operation. The power systems are required to operate not only efficiently by utilizing the available resources but also by maintaining system security and reliability. Load forecasting is crucial for power companies in meeting their objectives.

The problem of short-term load forecasting has traditionally been viewed as a modeling problem, where load has to be modeled as a function of time of day, day of the week, weather, and other social factors. The majority of operational and control decisions such as load dispatch, reliability analysis, and maintenance planning are all based on the load forecasts.

Electrical load can be represented as a time-series. Time-series analysis is a vast field for sequential data analysis. Every data set has unique properties associated with their source that make them challenging to analyze and model. Pre-existing assumptions about data, such as noise levels, redundancies, and temporal dependencies are often used in the chosen model or feature representation [1]. The alternate approach to using hand-crafted features is to automatically learn them by interacting with the data.

Modern machine learning techniques, such as expert systems [2], Artificial Neural Network (ANN) [3], [4], wavelets [5] have been deployed and shown encouraging results. Among these, ANNs have performed well given their ability to handle the non-linear relationships between the load and various underlying factors. The demand profiles are generally taken as sequences over time. Hence, it is a sequence-prediction task given only the historical and current information about the sequence. Recently, another class of ANN models have emerged, often referred to as *deep learning*. These models are similar to ANN with increased depth and specialized approaches for training these networks are currently being developed. Most of these models fall under the category of unsupervised learning. Unsupervised feature learning [6] performs this function by learning feature representations from unlabeled data. Usually, these layers of representations are stacked to create deep networks that are capable of modeling complex structures in the data. Unsupervised feature learning and deep learning have presented a success story for feature representation for static data [1].

In the context of short-term electrical load forecasting (hourly time scale or even shorter) the problem can be considered as a sequence prediction problem. A specialized ANN architecture referred to as recurrent neural network (RNN) has been successful in predicting sequences accurately. Moreover, RNNs have advanced to an extent where they have memory which has the ability to learn as the data arrives and have shown considerable success in the domain of learning relationships and text prediction based on context and are called the Long-Short-Term Memories (LSTM). Electrical load can be considered as a sequence (time series) which can be modeled using LSTM to accurately predict the future demand.

In this paper we develop a LSTM-Recurrent Neural Network for short term electrical load forecasting. The model is general enough and can be adapted for other time series where sufficient historical data is available. The results obtained are promising and present the generalized nature of LSTMs for time series forecasting problems.

The remainder of the paper is organized as follows, Section II presents a literature review of the recent work in electrical load forecasting. In Section III, we present a deep learning neural network architecture for short-term electrical load forecasting. In Section IV, we explain the various experiments. Lastly, in Section V we present our conclusions and directions for future research.

II. Background

Short-term load forecasting is a well studied problem. Accurate forecasts for loads are useful in the planning and operation of large [7] and micro-power systems [8]. There have been

978-1-5386-1645-1/17/$31.00 ©2017 IEEE

numerous models proposed for accurate modeling and prediction of demand. Electricity demand forecasting is considered a time-series modeling problem [9]. Various time series models such as auto regressive moving average (ARMA), generalized auto regressive conditional heteroskedasticity (GARCH), and intervention time series models have been used to model electrical load forecasting [10], [11]. Artificial neural networks (ANN) and other machine learning algorithms have proven successful in various tasks such as classification, regression and time series modeling [12]. Besides ANN, support vector regression is known as a strong predictor for achieving global optimum solutions.

In [13], the authors use support vector regression (SVR) for short-term load forecasting with two additional improvements in procedure for generation of model inputs and subsequent model input selection using feature selection algorithms. Another recent work [13] combines price and load forecasting using a hybrid time-series and adaptive wavelet neural network. In [14], they evaluate the effectiveness of some of the newest designed algorithms in machine learning to train typical radial basis function (RBF) networks for 24-h electric load forecasting: support vector regression (SVR), extreme learning machines (ELMs), and decay RBF neural networks (DRNNs). A comprehensive review of various tools for short term load forecasting has been done in [11].

ANNs were extensively used for regression, classification, and time series modeling until the mid 1990, but then got left behind with the advent of other novel regression methods. In 2006, the interest in neural network research was rekindled by Hinton et. al [15]. They showed that much better performance could be achieved using neural networks with multiple hidden layers or deep networks. Numerous efforts have been made to use the power of deep neural networks for time series modeling and forecasting.

In [16], a deep belief network with multiple restricted Boltzmann machines is proposed for time series forecasting. The researchers optimized the model's performance using particle swarm optimization (PSO) and present superiority of their approach over standard feed forward neural networks and other statistical models such as ARIMA models. In [17], the authors conducted simulations to compare deep learning architectures with standard neural networks for time series forecasting. In [18], several machine learning algorithms are presented to address the time series forecasting problem, such as multi-layer perceptron, Bayesian networks, K-nearest neighbor regression, support vector regression, and Gaussian processes. Whereas, in [19] the researchers presented the impact and usefulness of local learning techniques in dealing with temporal data.

Superficial methods, such as ensemble methods, where the goal is to improve the performance of the "unstable" predictors (decision trees and neural networks) have been used in [20]. Research in forecasting the reliability of a mining machine using ensemble methods is presented in [21]. The authors used least-square support vector machine with parameter estimation using genetic algorithm and they used standard benchmark data sets for reliability forecasting and fault prediction.

Overall, it is found that numerous variants of the above mentioned machine learning algorithms have been experimented for short term electrical load forecasting, which is a challenging task given the complex and large number of underlying features affecting the process. Recent advances in deep learning have proven to be useful in pattern identification for such complex cases. Till now no literature can be found regarding deep learning algorithm being used for modeling and forecasting short term electrical load at the regional level for varying time scales. Our work uses the fundamentals of deep learning to automatically identify the complex features in electrical load and predict accurately and robustly at varying time scales using LSTM-recurrent neural networks.

It is quite evident that no research has been done in short term load forecasting using deep learning techniques. The power and expressibility of these techniques still remains untouched. Our work here tries to explore the power of deep networks in context of short term electrical load forecasting.

III. Recurrent Neural Network

Recurrent neural networks with Long Short-Term Memory have emerged as a reliable tool for sequential data series modeling, analysis, and forecasting [22]. Usually, techniques solving problems associated with sequential data such as language, and audio etc. used hand-crafted features. LSTMs are found to be effective at capturing long-term temporal dependencies without suffering from the optimization hurdles that plague simple recurrent networks (SRNs) [23], and they have been used to advance the state of the art for many difficult problems.

The key component of a LSTM architecture is a memory cell which retains its state over time, and non-linear gating units which regulate the information flow in the cell.

LSTMs enable backpropagation of error across the network and in time. A controlled error propagation allows for networks to learn over large time steps thereby enabling relational learning across vast time differences.

The basic unit in the hidden layer of a LSTM network is the memory block. A memory block contains one or more memory cells and a pair of adaptive, multiplicative gating units which gate input and output to all cells in the block. Memory blocks allow cells to share the same gates, thus reducing the number of adaptive parameters.

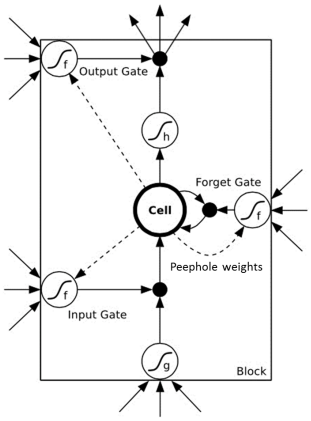

Fig. 1: LSTM Block with SRN as a hidden layer

The equations for LSTM under consideration are split into two parts: forward pass and the back-propagation through time.

Forward Pass

Let *N* be the number of LSTM blocks and *M* be the number of inputs. We have the following weights: The alphabets W, R, p, and b denote the weight associated with input, recurrent, peephole, or bias connections. The subscripts z, s, f, and o refer to the weights connecting the input gate, blocking gate, forget gate, and the output gate, respectively.

- Input Weights: $W_z, W_s, W_f, W_o \in \mathbb{R}^{N \times M}$
- Recurrent Weights: $R_z, R_s, R_f, R_o \in \mathbb{R}^{N \times M}$
- Peephole Weights: $p_s, p_f, p_o \in \mathbb{R}^{N}$
- Bias Weights: $b_z, b_s, b_f, b_o \in \mathbb{R}^{N}$

All of the arrows entering the memory block are comprised of the recurrent connection weights and the input weights.

A recurrent neural network with a memory block *M*, can be represented as shown in Figure 2. Here, x_t represents the input, h_t represents the output, *M* denotes the memory block. The curved arrow connecting the memory block to itself is the recurrent connection.

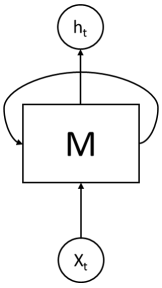

Fig. 2: RNN with a memory block

To visualize the RNN for sequence learning at time instance t, it can be unfolded into a network with horizontal connections as shown in Figure 3. The subscripts 0, 1, ..., $t-1$, t represent the time lag and are a result of unfolding the recurrent connection to the memory cell.

The output at any time instant from the LSTM-RNN is dependent on previous inputs as the horizontal connection or recurrent connection weights govern the strength of dependence. The colored blocks in Figure 3 show one possible case where the output h_t is dependent on x_{t-1}, ..., x_1, and x_0.

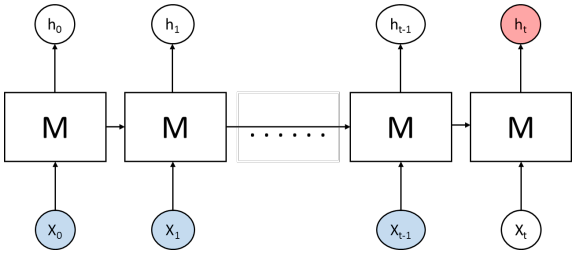

Fig. 3: Unfolded RNN with LSTM memory block where the output at time t (red) is dependent on previous inputs (blue)

The input vector to the network is given as x^t at time t, σ, g, h are point-wise non-linear functions with $\sigma(x) = \frac{1}{1+e^{-x}}$ being the logistic sigmoid, tangent hyperbolic is used for block input and output activation functions respectively. We denote the point-wise multiplication of two vectors by $\odot$. The first phase of LSTM training is the forward pass which is presented in Algorithm 1 as adapted from [24].

Algorithm 1 Forward Pass

procedure FORWARD-PASS
  $\overline{z}^t = W_z x^t + R_z y^{t-1} + b_z$
  $z^t = g(\overline{z}^t)$ block input
  $\overline{i}^t = W_i x^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i$
  $i^t = \sigma(\overline{i}^t)$ input gate
  $\overline{f}^t = W_f x^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f$
  $f^t = \sigma(\overline{f}^t)$ forget gate
  $c^t = z^t \odot i^t + c^{t-1} \odot f^t$ cell
  $\overline{o}^t = W_o x^t + R_o y^{t-1} + p_o \odot c^t + b_o$
  $o^t = \sigma(\overline{o}^t)$ output gate
  $y^t = h(c^t) \odot o^t$ block output
end procedure

The second part of the algorithm is the backpropagation through time, we consider here Δ^t is a vector of deltas (or gradients) passed down from the previous layer. Let us consider E to be the loss function, and is generally referred to as $\frac{\partial E}{\partial y^t}$, which does not include the recurrent dependencies. Therefore, there is a need for a special procedure to evaluate the deltas inside the LSTM block presented in Algorithm 2 as adapted from [24].

Algorithm 2 Backpropagation Through Time

procedure BACKPROP
  $\delta y^t = \Delta^t + R_z^T \delta z^{t+1} + R_i^T \delta i^{t+1} + R_f^T \delta f^{t+1} + R_o^T \delta o^{t+1}$
  $\delta o^t = \delta y^t \odot h(c^t) \odot \sigma'(\overline{o}^t)$
  $\delta c^t = \delta y^t \odot o^t \odot h'(c^t) + p_o \odot \delta o^t + p_i \odot \delta i^{t+1} + p_f \odot \delta f^{t+1} + \delta c^{t+1} + \delta c^{t+1} \odot f^{t+1}$
  $\delta f^t = \delta c^t \odot c^{t-1} \odot \sigma'(\overline{f}^t)$
  $\delta i^t = \delta c^t \odot z^t \odot \sigma'(\overline{i}^t)$
  $\delta z^t = \delta c^t \odot i^t \odot g'(\overline{z}^t)$
end procedure

Finally, the deltas for the inputs are only required if there is a training layer below them and is given by the Equation 1.

$$\delta x^t = W_z^T \delta z^t + W_i^T \delta i^t + W_f^T \delta f^t + W_o^T \delta o^t \tag{1}$$

The gradients of the weights are calculated based on the Equation 2 - 7.

$$\delta W_{k \in z,i,f,o} = \sum_{t=0}^{T} \langle \delta^t_{k \in z,i,f,o}, x^t \rangle \tag{2}$$

$$\delta R_{k \in z,i,f,o} = \sum_{t=0}^{T-1} \langle \delta^{t+1}_{k \in z,i,f,o}, y^t \rangle \tag{3}$$

$$\delta b_{k \in z,i,f,o} = \sum_{t=0}^{T} \delta^t_{k \in z,i,f,o} \tag{4}$$

$$\delta p_i = \sum_{t=0}^{T-1} c^t \odot \delta i^{t+1} \tag{5}$$

$$\delta p_f = \sum_{t=0}^{T-1} c^t \odot \delta f^{t+1} \tag{6}$$

$$\delta p_o = \sum_{t=0}^{T} c^t \odot \delta o^t \tag{7}$$

There are numerous variants of LSTM which have been studied in the literature such as: no input gate, no forget gate, no output gate, no input activation function, no output activation function, coupled input and forget gate, no peepholes, and full gate recurrence [24]. It is observed through experiments that for our purpose of electrical load forecasting, LSTM in its vanilla form performs well.

In the next section, we present the application of LSTM networks to the task of sequential data modeling and time series forecasting in our case study for short-term electrical load for the Province of Ontario, Canada.

IV. Experiments

LSTM-RNN for short term electric load forecasting is used as an autoregressive model where it can only access input from the current time step. Various competitors to LSTM such as multi-layer perceptron see several consecutive inputs in a given time window when trained by back-propagation.

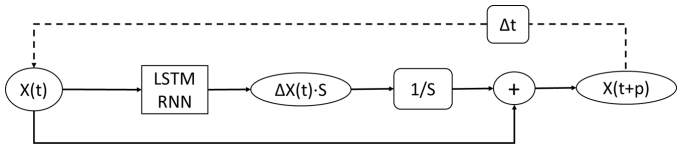

Fig. 4: LSTM as time series predictor

In Figure 4, $X(t)$ denotes the input, S is a scaling factor, and $X(t+p)$ is the prediction p time steps ahead.

Data

The data for electrical load are usually maintained on an hourly basis. Electrical load has complex and non-linear relationships. The electrical load data for Ontario is obtained from Independent Electricity System Operator (IESO) [25]. It is observed in Figure 5 that maximum demand occurs between 4:00 pm and 8:00pm whereas the minimum demand happens between 3:00 am and 5:00 am. We used 10 years of (from 2006 to 2016) historical data for training and validation of our LSTM network in the ratio of 70% for training and 30% for testing and validation.

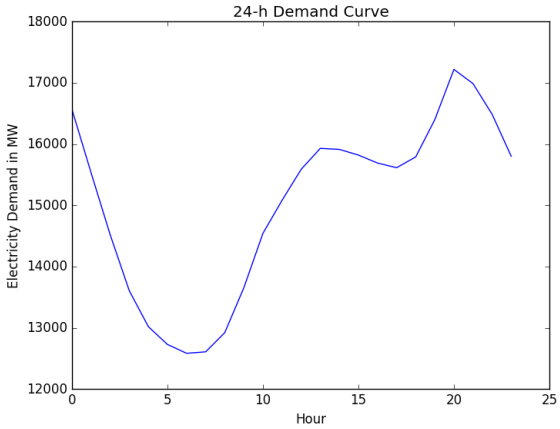

Fig. 5: Hourly power demand for 1-day in Ontario, Canada

The data can be analyzed on an hourly time scale but large scale (annual) visualization of historical data reveals other hidden characteristics which remain unknown at small scale (daily). As in Figure 6 we see a pattern in the energy demand during varying seasons on visualizing annual hourly data.

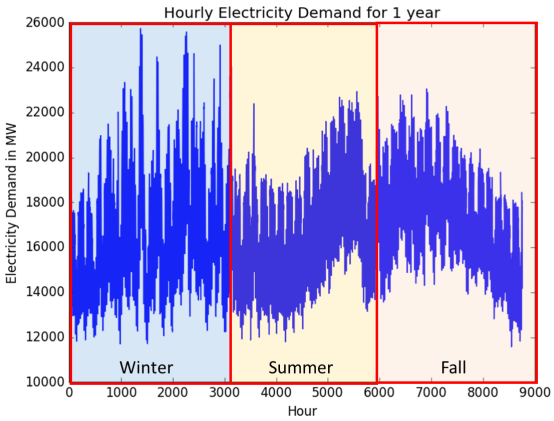

Fig. 6: Hourly power demand for 1-year in Ontario, Canada

It is important to capture both the seasonal, daily, and hourly variations in the power demand. Most of these features need to be modeled separately when using statistical time series modeling approaches or even standard feed forward neural networks. In LSTM, these features are learned by the network automatically as the data is presented to the network.

Network Topology

The input units are fully connected to a hidden layer consisting of memory block with 1 cell each. The cell outputs are fully connected to the cell inputs, to all gates, and to the output units. All the gates, the cell, and the output units are biased. Bias weights are initialized in steps of -0.5 starting from -0.5 for each block with forget gates having symmetric biases on the positive side. All other weights are initialized in the range [-0.1,+0.1]. The cell input function g is sigmoid in the range [-1,+1] and the output function h is tangent hyperbolic. We used a constant learning rate of $\alpha = 10^{-3}$. We used mean squared error as our loss function for optimizing the parameters of our

Month	ANN	ARIMA	**LSTM-RNN**
January	4.6	5.7	4.4
May	6.3	8.2	5.9
September	3.8	3.9	3.8

TABLE I: NRMSE in % for ANN, ARIMA and LSTM-RNN for different months in a year

LSTM-RNN. We used *Adam* as the optimization algorithm for our loss function. It is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [26].

Results

The overall goal of the network N is to predict the future load at time instance $t+T$, where t is the current time. The target for the network N is the difference between the values $x(t+p)$ of the electrical load time series p time steps ahead and the current value, with a scaling factor S.

Therefore $N(t) = S.(x(t+p) - x(t)) = S.\Delta x(t)$. The scaling factor is used to bring all the values in the range [-1,+1]. THe same scaling factors are used for training and testing purposes. We do the reverse when evaluating the predicted values.

The square root of the mean/average of the square of all of the error is a good measure of performance of the algorithm. The use of root mean square error (RMSE) is very common and it makes an excellent general purpose error metric for numerical predictions.

Compared to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors. If the RMSE is normalized then it is referred to as Normalized RMSE or NMRSE and is given by Equation 8.

$$NMRSE = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_{max} - y_{min}}\right)^2\right)} \qquad (8)$$

In Figure 7 we show the original values with the forecast values obtained using the training data for the past 10 years.

We present our results for three different months in different seasons (i.e. Fall, Winter and Spring) and for a year.

In Figure 7, the annual hourly power demands and forecasts are presented.

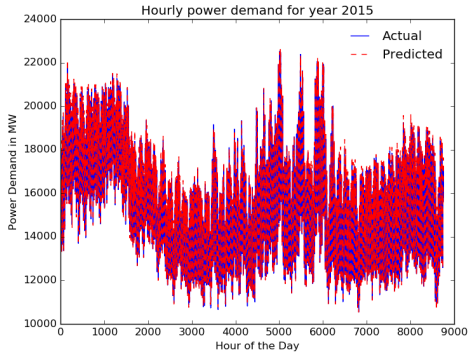

Fig. 7: Hourly power demand and forecast using LSTM for the year 2015

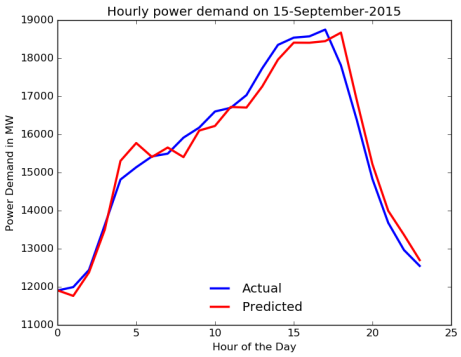

Fig. 8: Hourly power demand and forecast using LSTM for a day in the Fall of 2015

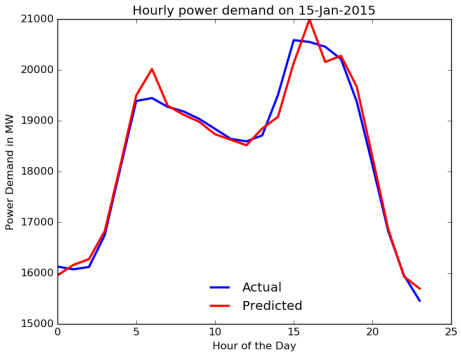

Fig. 9: Hourly power demand and forecast using LSTM for day in the Winter of 2015

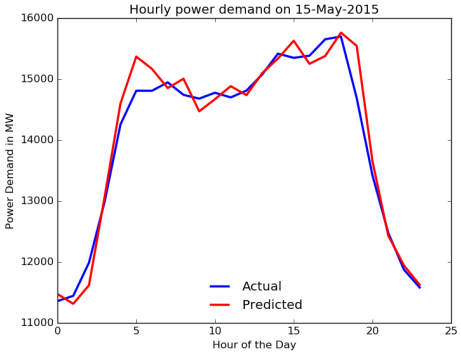

Fig. 10: Hourly power demand and forecast using LSTM for day in the Spring of 2015

In Figures 8, 9, and 10 the actual and predicted power demand for a day in three different seasons is presented. It is clear from the predictions and NRMSE values as presented in Table I, that LSTM-RNN peforms better or equivalent to existing techniques. It is also important to note that we do not require to segregate the data based on seasonality, time of day, and other factors for training the network. The network trains and learns for itself the underlying features from the data.

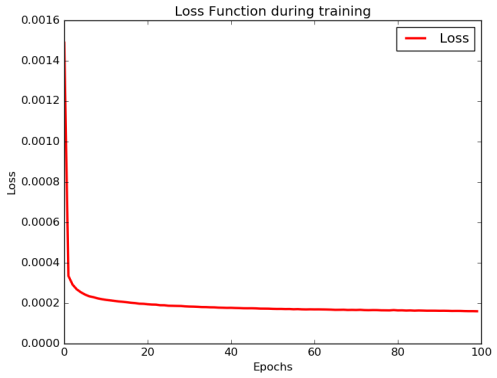

Fig. 11: Decay of loss function value with epochs

In Figure 11, the decay of the loss function with epochs is shown. The loss function used is the mean squared error which decays quite rapidly to a low value given the large training set.

V. Conclusions

In our work we implemented LSTM deep recurrent neural network for modeling short-term electrical load time series for the Province of Ontario, Canada. Short-term electrical load modeling has been a challenging task and is important for reliable and profitable operation of electricity markets and utilities . It is well established that electrical load is dependent on a large number of underlying factors which are hard to identify and model using currently established techniques. Deep learning has become a very promising approach for pattern recognition for complex and high dimensional data sets. We used the deep learning approach to model sequential short-term electrical demand time series. The results from LSTM-RNN are reliable, robust, and comparable with other state of the art techniques.

The power of LSTM can be utilized for even more complex tasks given the ability of these networks to mine complex hidden patterns in unlabeled data which is present in large quantities. We wish to extend the LSTM model for forecasting not only short term loads to even finer levels of granularity such as 5 min and 10 min. This would enable ease in integration of the renewable source of energy with the existing power systems.

References

[1] P. Geurts, "Pattern extraction for time series classification," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2001, pp. 115–127.

[2] A. Mardani, A. Jusoh, and E. K. Zavadskas, "Fuzzy multiple criteria decision-making techniques and applications–two decades review from 1994 to 2014," *Expert Systems with Applications*, vol. 42, no. 8, pp. 4126–4148, 2015.

[3] S. I. Vagropoulos, E. G. Kardakos, C. K. Simoglou, A. G. Bakirtzis, and J. P. Catalão, "Artificial neural network-based methodology for short-term electric load scenario generation," in *Intelligent System Application to Power Systems (ISAP), 2015 18th International Conference on*. IEEE, 2015, pp. 1–6.

[4] A. Narayan, K. W. Hipel, K. Ponnambalam, and S. Paul, "Neuro-fuzzy inference system (asupfunis) model for intervention time series prediction of electricity prices," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2121–2126.

[5] S. Li, P. Wang, and L. Goel, "Short-term load forecasting by wavelet transform and evolutionary extreme learning machine," *Electric Power Systems Research*, vol. 122, pp. 96–103, 2015.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] U. Mukherjee, A. Maroufmashat, A. Narayan, A. Elkamel, and M. Fowler, "A stochastic programming approach for the planning and operation of a power to gas energy hub with multiple energy recovery pathways," *Energies*, vol. 10, no. 7, p. 868, 2017.

[8] A. Narayan and K. Ponnambalam, "Risk-averse stochastic programming approach for microgrid planning under uncertainty," *Renewable Energy*, vol. 101, pp. 399–408, 2017.

[9] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[10] S.-J. Huang and K.-R. Shih, "Short-term load forecasting via arma model identification including non-gaussian process considerations," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 673–679, 2003.

[11] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making," *European Journal of Operational Research*, vol. 199, no. 3, pp. 902–907, 2009.

[12] Z. Aung, M. Toukhy, J. Williams, A. Sanchez, and S. Herrero, "Towards accurate electricity load forecasting in smart grids," in *The Fourth International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA*, 2012.

[13] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4356–4364, 2013.

[14] C. Cecati, J. Kolbusz, P. Różycki, P. Siano, and B. M. Wilamowski, "A novel rbf training algorithm for short-term electric load forecasting and comparative studies," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6519–6529, 2015.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[16] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using restricted boltzmann machine," in *International Conference on Intelligent Computing*. Springer, 2012, pp. 17–22.

[17] E. Busseti, I. Osband, and S. Wong, "Deep learning for time series modeling," Technical report, Stanford University, Tech. Rep., 2012.

[18] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.

[19] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *Business Intelligence*. Springer, 2013, pp. 62–77.

[20] L. Beriman, "Bias, variance, and arching classifiers," Technical Report, Tech. Rep., 1996.

[21] S. Chatterjee, A. Dash, and S. Bandopadhyay, "Ensemble support vector machine algorithm for reliability estimation of a mining machine," *Quality and Reliability Engineering International*, vol. 31, no. 8, pp. 1503–1516, 2015.

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

[23] 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.

[24] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *arXiv preprint arXiv:1503.04069*, 2015.

[25] IESO, "Independent electricity system operator," http://www.ieso.ca/Pages/Power-Data/Demand.aspx.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980