Electricity Demand Forecast with LSTMs

Ossman, M., & Bi, Y. (2022). Electricity Demand Forecast with LSTMs. In *The 21st UK Workshop on Computational Intelligence, 2022*

Published in:
The 21st UK Workshop on Computational Intelligence, 2022

Publication Status:
Published (in print/issue): 01/09/2022

Document Version
Peer reviewed version

Electricity Demand Forecast with LSTMs

Mazen Ossman and Yaxin Bi

School of Computing, Ulster University, Belfast, UK
{ossman_m,y.bi}@ulster.ac.uk

Abstract. Long-Short Term Memory (LSTM) networks are able to learn the complicated relationships between variables from previous and current timesteps over time series data and use them to do specific forecast tasks. LSTMs are basically stacks of perceptron algorithms, the more stacks a neural network has, the deeper the neural network. There are two types of gradient propagations over LSTM networks – forward and backward. However there is a common vanishing issue when developing LSTM networks. This paper proposes two LSTM models developed with sequence-to-sequence and sequence-to-vector frames and investigates possible empirical solutions to the vanishing issue, particularly, in the context of predicting multivariate and univariate power demands through comparative evaluation on electricity demand data.

Keywords: Long-Short Term Memory networks, Power demand forecast, Sequence-to-sequence, Sequence-to-vector.

1 Introduction

Electricity is widely used energy. The main resources of electricity generation are mainly from fossil fuels, nuclear energy, etc. non-renewable sources. Due to the demanding for zero carbon, the world started to turn to renewable sources for generating electricity, including solar, hydro, and wind [1]. Since renewable sources introduced, the electricity prices have decreased considerably. For example, in Germany the electricity price has been widely decreased due to usage of wind energy [2] [3]. In most of the situations, the electricity generated from renewable energy sources imported to national grids are intermittent which need to be managed with the distribution of renewable energy for the local energy demanding [1] [4]. Smart grid systems can collect data for the management of national grids and for the improvement of the services [4] [5]. The data collected along with the renewable energy supplied can be grouped into long, middle and short-terms based on time intervals, which can be used for developing forecast models for responding the demand side management. Specifically, the short-term is referred from one hour to a week, mid-term is from one week to one month, a long-term is referred to more than a year [6]. However the definition of these terms is not unique in the literature, for instance, one research work suggested that short term forecasting should be between an hour to three months [7].

There recently have been a body of studies on forecasting power consumption, including the use of LSTM to forecast energy consumption for short term and its comparison with support vector machine and rainforest algorithm [6]. Another research created a model via combining GUI (Graphic User Interface) and LSTM, and compared the LSTM performance to the real readings [4]. Many other papers have used statistical methods to estimate the coefficient of variables, such as regression, multiple linear regressions (MLR), and autoregressive integrated moving average (ARIMA) [7]. The method of backward gradient propagation faces a very common problem in LSTM networks called the vanishing and exploding problems, which make the network not learning fast and affect the parameters of the layers of the network [8]. The exploding problem can be solved with different techniques such as, HE initialization and using normalization and regularization, while the vanishing problem could be solved by improved LSTM or GRU [9]. LSTM could be efficient at learning long sequence problems, so as the short sequence problems.

2 Methods

In a LSTM, perceptrons are stacked together, in which layers of a perceptron share with the same weights, bias and structure. They are rolled in together to a single recurrent layer. A LSTM has the capability of learning from data for a specific task, particularly learn the relationship among variables within data for a forecast task, and then make predictions for any new data [8]. There are two types of weight gradient propagation in neural networks in terms of forward and backward propagation. The forward propagation networks are most likely used in classification tasks that might not be directly associated with timestamps like image classification or regression problem. The backward gradient propagation could be more related to time series problems [9].

There are common issues of vanishing and exploding during weight propagation in LSTM neural networks, which are mainly related to timestamps of sequential data, and variance difference between the input data before the layers and the output after layer [9]. These issues prevent weights to be updated effectively in the process of gradient propagation and optimization step. Specifically when the gap between two timesteps is too large, the previous information cannot be carried to the next step. In meanwhile, if weights are either too big(exploding) or too small(vanishing), that will result in the weights not undated. In [13], the authors presented the propagation scenarios occurred in four hidden layers with sigmoid activation functions as illustrated in Fig.1, the means and standard deviations of these activations show that the weights over the last hidden layer could be updated until 100 episodes and become saturation after 140 episodes. As the last hidden layer is immediately linked the output layer, the saturation would significantly affect the performance of neural networks.

Different methods have been proposed to solve vanishing and exploding problems. For instance, to avoid variance changes the gradient weights could be initialized using the input and the output variances [9], applying non-linear activation functions like Relu,

leakyRelu reduces saturation as occurred when using a sigmoid function [13]. This study aims to investigate the impact of different lengths of sequence to sequence and sequence to vector to resolve vanishing and exploding issues in LSTM neural networks, particularly in the context of electricity consumption forecast.

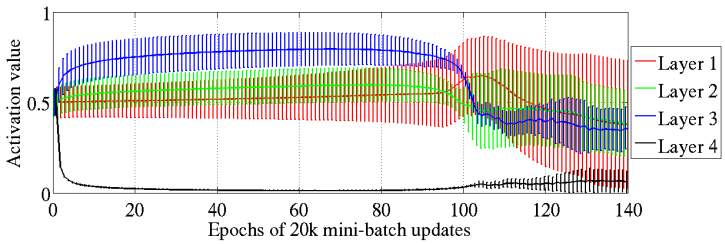


Fig.1. Mean and standard deviation (vertical bars) of the activation values (output of the sigmoid) during supervised learning.

2.1 LSTM Model Design

The process of a LSTM in dealing with time series sequences follows a loop from input to output over timesteps. The process goes through the gates of forget, input and output to train a memory state that remembers the important information and filtering out unnecessary data, preserving discriminative features in the final decision stage. In each timestep, the LSTM produces an output and accepting it as input for next timestep [9], for the first timestep, a previous timestep is considered zero.

In this study time series sequences are modelled as sequence-to-sequence and sequence-to-vector for forecasting tasks.

- sequence-to-sequence: dividing time series data into a collection of sub-sequences corresponding timesteps, a LSTM accepts a sequence as an input and produces an output sequence. For instance, treating the price of power demand over 10 days as an input sequence, and then outputting a forecasted price for next 10 days.
- sequence to vector: a LSTM accepts a sequence as an input, then produces a vector of elements at the last time step. For example, a sentiment analysis accepts a vector of words and outputs a probability distribution over the sentiment categories of positive, neutral and negative [9].

To study impact of different parameters, six LSTM models have been proosed as shown in Table 1, three models are sequence-to-sequence based, another three are built on sequence to vector based. These models have different numbers of neurons and layers in addition to different activation functions. The models are trained with Mean Absolute Error (MAE) and Mean Squared Error (MSE) cost functions, their performance is measured by Root Mean Squared Error (RMSE), MAE and MSE metrics.

In particular the number of output layers within these models depend on the tasks in case of multivariate (multiple inputs and multiple outputs) forecasting, where the output layer is defined with dense10 neurons for 10 days forecasting and a single neuron for univariate forecasting.

Table 1. Details of designed models

Sequence to vector based model				Sequence-to-sequence based model			
Layers	Model A	Model B	Model C	Layers	Model D	Model E	Model F
Layer 1	32	64	64	Layer 1	30	32	32
Dropout 1	None	30%	30%	Dropout 1	None	30%	30%
Layer 2	None	32	32	Layer 2	None	64	64
Dropout 2	None	None	30%	Dropout 2	None	None	30%
Layer 3	None	None	16	Layer 3	None	None	64

3 Experimental Results

3.1 Electricity Demand Data, Pre-processing and Experimental Setting

The dataset used for the evaluation is downloaded from the website [15], which records electricity demands since January 2019 to May 2022 across the United Kingdom, it consists of 3 columns:

- National Demand (ND): the sum of metered generation, but excluding generation required to meet station load, pump storage pumping and interconnector exports.
- Transmission system demand (TSD): being equal to the ND plus the additional generation required to meet station load, pump storage pumping and interconnector exports.
- England and Wales Demand: same as ND above but only for England and Wales.

The data has been preprocessed as follows:

- converting every 30 minutes level of granularity data into the format of YYYY-MM-DD-HH-mm, and creating additional column named by *Date*. Fig. 2 presents electricity demands in every 30 minutes before the conversion and Fig.3 presents the sum of all columns of the dataset in the new format.
- The dataset has been normalized as the LSTM network requires decimal input in the range of 0 and 1.
- The input data has been formatted in the form of (batch, timestep, features). The batch is a number of total days; the timestep is a number of days that the models take for prediction, and the features are a number of outputs. In case of multiple inputs prediction, the input features are three, while for a single input prediction, the input features is one.

The data labels have been encoded for predicting a next day or next 10 days of electricity demand. For the sequence to vector based, the models receive the input of 30

timesteps or 50 timesteps, and then outputs a vector. The models, for instance, receive 50 days and then predict day 51. Hence the labels were encoded in the form of (batch, features). On the other hand, for the sequence-to-sequence based model, the labels have been encoded in the form of (batch, timesteps, features). Every input timestep will output the next timestep. If the models, for example, receive an input of day one data, the model should predict day two, or next 10 days.

After preprocessing, (934, 50, 3) is for multiple inputs forecasting by the LSTM, (934, 50, 1) for a single input forecasting, and for the 30 timesteps forecasting it is in the form of (944, 30, 1). Meanwhile the output label (934, 3) is for multiple inputs sequence to vector forecasting and (934, 49, 3) for sequence-to-sequence forecasting. The output labels for a single input sequence to vector is (934,1) and (934,49,1) for sequence-to-sequence.

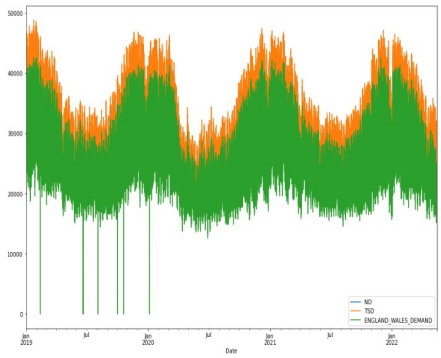


Fig.2 Electricity demands every 30 minutes

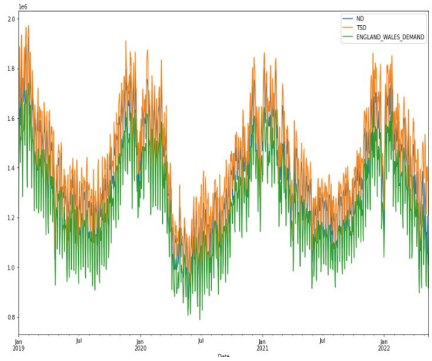


Fig.3 Electricity demands every day.

3.2 Experimental Results

To evaluate six LSTM models described in Table 1, a range of experiments have been conducted over the processed dataset, the trained models have been applied to forecast a single day ahead and 10 days ahead with different input timesteps. The results of MSE, MAE and RMSE have been compared in the following tables.

Table 2. Performance of multiple inputs models with MSE loss function.

Sequence to vector				Sequence-to-sequence			
Models	MSE	MAE	RMSE	Models	MSE	MAE	RMSE
Model A	0.0083	0.0746	0.0911	Model D	0.0066	0.0657	0.813
Model B	0.0078	0.0741	0.0884	Model E	0.0054	0.0580	0.0737
Model C	0.0161	0.1067	0.1269	Model F	**0.0051**	0.0555	0.0716

Table 3. Performance of multiple inputs models trained with MAE loss function

Sequence to vector	Sequence-to-sequence

Models	MAE	MSE	RMSE	Models	MAE	MSE	RMSE
Model A	0.0759	0.0085	0.0923	Model D	0.0601	0.0056	0.0753
Model B	0.0656	0.0062	0.0786	Model E	0.0608	0.0059	0.0767
Model C	0.1296	0.0240	0.1549	Model F	0.0579	0.0055	0.0740

For single input models and for prediction 10 days ahead with different timesteps, the transmission system demand has been used, since it includes the national grid demand plus pumps and loads demands. The performance of the models is evaluated over the dataset as mentioned same way as the multiple inputs and with the same models.

Table 4. Performance of single inputs models with MSE loss function.

Sequence to vector				Sequence-to-sequence			
Models	MSE	MAE	RMSE	Models	MSE	MAE	RMSE
Model A	0.0087	0.0744	0.0934	Model D	0.0064	0.0638	0.0802
Model B	0.0064	0.0670	0.0802	Model E	0.0060	0.0593	0.0776
Model C	0.0165	0.1012	0.1284	Model F	0.0058	0.0614	0.0765

Table 5. Performance of univariate models with MAE loss function.

Sequence to vector				Sequence-to-sequence			
Models	MAE	MSE	RMSE	Models	MAE	MSE	RMSE
Model A	0.0797	0.0095	0.0975	Model D	0.0665	0.0071	0.0845
Model B	0.0656	0.0062	0.0786	Model E	0.0589	0.0057	0.0754
Model C	0.0934	0.0139	0.1180	Model F	0.0569	0.0054	0.0733

Table 6. Performance of 10 days ahead 50 timesteps with MSE loss function.

Sequence to vector				sequence-to-sequence			
Models	MSE	MAE	RMSE	Models	MSE	MAE	RMSE
Model A	0.0164	0.1017	0.1281	Model D	0.0106	0.0106	0.1029
Model B	0.0116	0.0894	0.1077	Model E	0.0113	0.0886	0.1061
Model C	0.0170	0.1013	0.1303	Model F	0.0120	0.0922	0.1097

Table 7. Performance of 10-days ahead with 50 timesteps with MAE loss function.

Sequence to vector				sequence-to-sequence			
Models	MAE	MSE	RMSE	Models	MAE	MSE	RMSE
Model A	0.1036	0.0173	0.1315	Model D	0.0866	0.0109	0.1045
Model B	0.0857	0.0108	0.1039	Model E	0.0805	0.0095	0.0975
Model C	0.1016	0.0173	0.1314	Model F	0.0778	0.0090	0.0951

Table 8. Performance of 10 days ahead 30 timesteps with loss function

Sequence to vector				sequence-to-sequence			
Models	MSE	MAE	RMSE	Models	MSE	MAE	RMSE
Model A	0.0152	0.0985	0.1233	Model D	0.0117	0.0117	0.1081
Model B	0.0110	0.0891	0.1048	Model E	0.0096	0.0805	0.0982
Model C	0.0169	0.1017	0.1301	Model F	0.0142	0.0991	0.1190

Table 9. Performance of 10-days ahead 30 timesteps with MAE loss function

Sequence to vector				sequence-to-sequence			
Models	MAE	MSE	RMSE	Models	MAE	MSE	RMSE
Model A	0.0992	0.0157	0.1254	Model D	0.0874	0.0115	0.1072
Model B	0.0823	0.0100	0.0999	Model E	0.0795	0.0094	0.0969
Model C	0.1022	0.0168	0.1298	Model F	0.0855	0.0108	0.1040

4 Discussions

From the results illustrated in Tables 2-9 above, it can be observed that the sequence-to-sequence-based models show better performance than the sequence-to-vector-based models, indicating that the sequence-to-sequence-based models could be more effective in predicting time series problems.

The loss function used has different effects on the models. MSE is sensitive to outliers, but MAE is not instead it takes more time during the model training. In general, these models have better performance when the timesteps is less, and the less forecasting outputs are, which is due to the vanishing problem of LSTM. The sequence-to-sequence-based models have drastically achieved less error in each of the experiments, where Model B has better forecasting performance in most of experiments, which comprises two hidden layers. Investigating further, Model B has scored better than Model A and Model C in all cases, in terms of overfitting and underfitting parameters. Model A has underfitting parameters which causes poor accuracy in training set and test set, while Model C has overfitting which has a good accuracy in training set and bad accuracy in test set.

Model E consists of 2 hidden layers as well, which is used to as a benchmark model to compare the performance of sequence-to-sequence with sequence-to-vector. The following figures show the performance of Models B and E trained on MAE only in each of the experiments. For the sequence-to-sequence based models, Model E and Model F has scored better than model D. Model F has fit the data better than Model E. But for the sake of argument, we focus on comparing two hidden layers models of sequence-to-sequence and two hidden layers of the sequence-to-vector model.

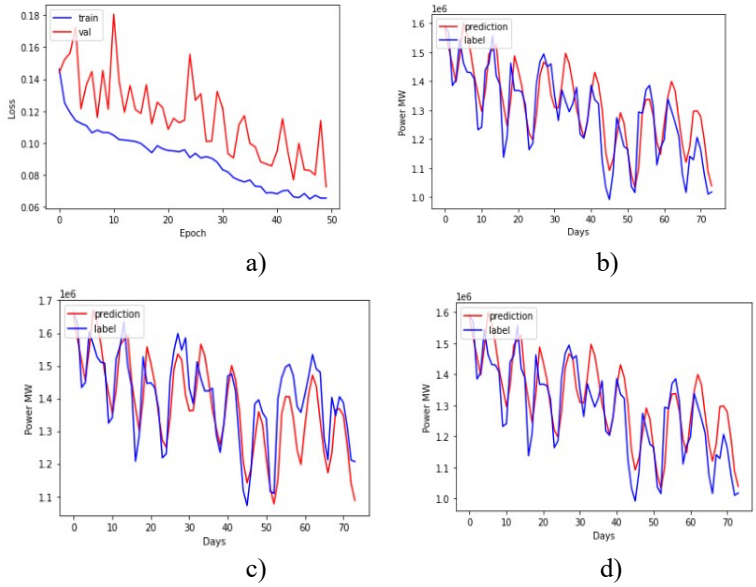


Fig. 4. Model B performance for multiple inputs: a) train and validation over each epoch; b) predictions and labels for the test set for ND; c) predictions and labels for the test set for TSD, d) predictions and labels for the test set for England and Wales

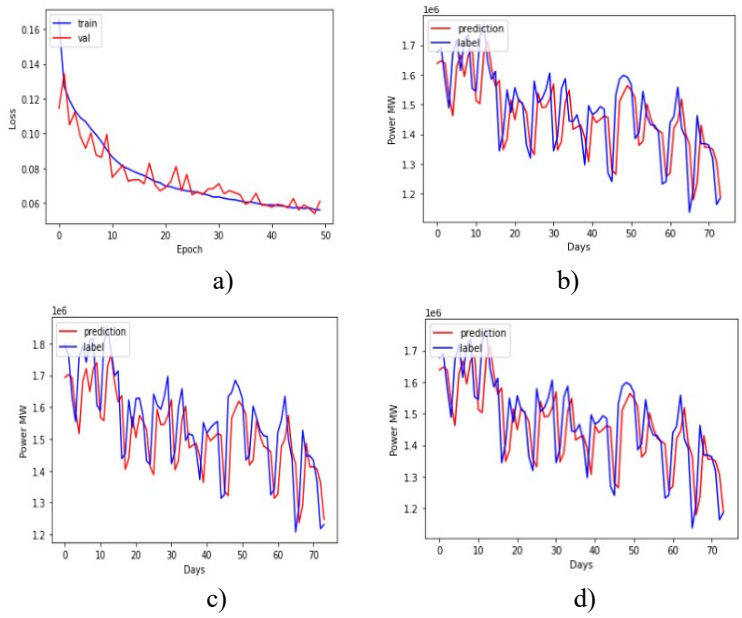
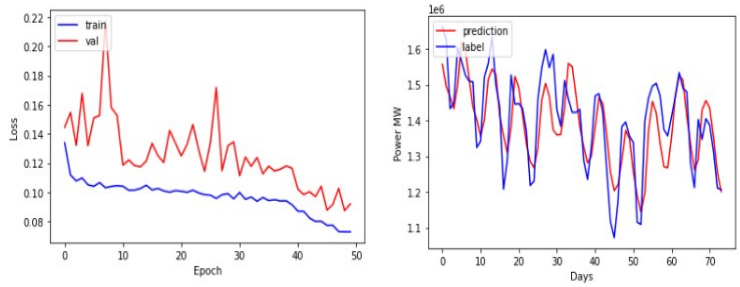
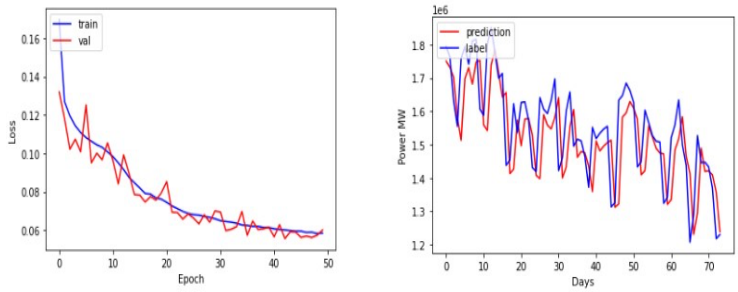


Fig 5. Model E performance for multiple inputs: a) train and validation over epoch; b) predictions and labels for the test set for ND; c) predictions and labels for the test set for TSD; d) prediction and labels for the test set for England and Wales



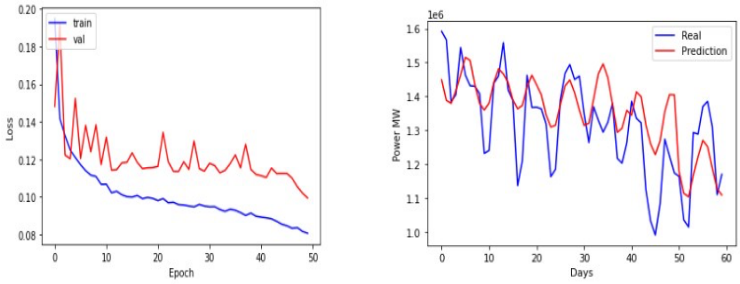
a) b)

Fig 6. Model B performance for single input with 50 timesteps: a) train and validation over epoch; b) predictions and labels for the test set for TSD



a) b)

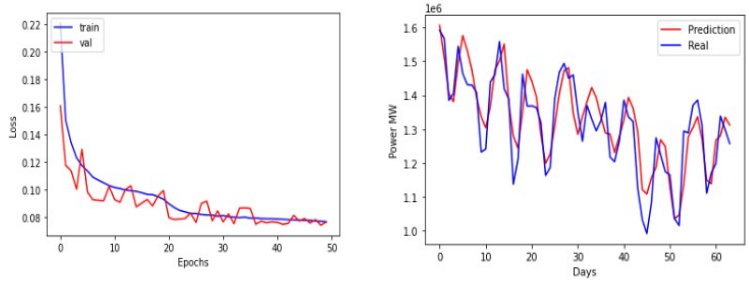
Fig 7. Model E performance for single input problem with 50 timesteps: a) train and validation epoch model b) predictions and labels for the test set for TSD



a) b)

Fig 8. Model B performance for predicting 10 days ahead with 50 timesteps: a) train and validation on epoch b) predictions and labels for the test set for TSD

Figs. 4-11present comparative analysis, from these figures it can be seen that Model E has higher accuracy and less loss on the validation set.



a) b)

Fig 9. Model E performance for predicting 10 days ahead with 50 timesteps: a) train and validation on epoch b) predictions and labels for the test set for TSD

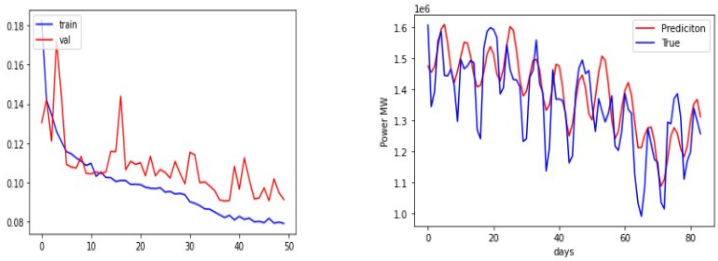
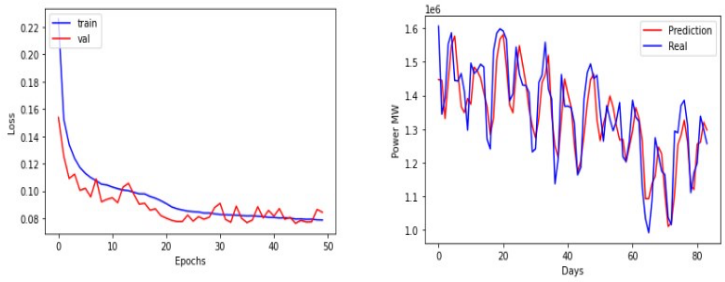


Fig 10. Model B performance for predicting 10 days ahead with 30 timesteps: a) train and validation on epoch b) predictions and labels for the test set for TSD



a) b)

Fig 11. Model E performance for predicting 10 days ahead with 30 timesteps problem: a) train and validation on epoch b) predictions and labels for the test set for TSD

By contrast Model B has less accuracy on the validation set. As shown in Tables 4, 5, 6 and 7, the LSTM performance is drastically reduced, due to the output length, which were single day output in Tables 4, 5, and 10 days output in Tables 6 and 7. Model B has scored 0.0802 and 0.0786 RMSE when trained using MSE and MAE, respectively, in a single day input. As opposed to the single input, in a single input 10 days ahead, Model B has scored 0.1077, and 01039 RMSE. For

Model E which it scored 0.0776 and 0.0754 when trained using MSE and MAE. However, in 10 days ahead prediction the error has increased from 0.0776 to 0.1061 and from 0.0754 to 0.0975 on average as shown in Table 10.

Table10. Performance comparison between Model E and Model B for 10 days ahead, and single day ahead.

Models trained on MSE			Models trained on MAE		
Models	1 day	10 days	Models	1 day	10 days
Model B	0.0802	0.1077	Model B	0.0786	0.1039
Model E	0.0776	0.1061	Model E	0.0754	0.0975

For predicting 10 days ahead with 30 timesteps and 50 timesteps, Model B has scored better in case of 30 timesteps, because the more timesteps is involved, the more the vanishing problem the model will suffer. As opposed to this the sequence-to-sequence models didn't suffer much from the timesteps, since the sequence-to-sequence models predict outputs at each timestep, which is not affected much by timesteps shown in Table 11.

Table 11. Performance comparison between Model E and Model B for 10 days ahead with 30 time-steps, and 60 timesteps.

Models trained on MSE			Models trained on MAE		
Models	10 day-30	10 days-60	Models	10 days-30	10 days-60
Model B	0.1048	0.1077	Model B	0.0999	0.1039
Model E	0.0982	0.1061	Model E	0.0969	0.0975

Generally, sequence-to-sequence models perform better than sequence-to-vector in timeseries analysis, but vice versa in some cases. This can be also observed from our experiments. In Experiment 2 of predicting 1 day ahead for TSD, Model B has scored pretty much close to Model D. For Experiment 3 of predicting 10 days ahead with 50 timesteps, Model B has better performance than Model F with the MSE metric, than Model D with the MAE. For Experiment 4 of predicting 10 days ahead with 30 timesteps, Model B performs better than Model D and Model F with the MSE. The possible reason for this could be that if the loss function and model complexity do not fit the data well, a sequence-to-vector model could perform better.

5 Conclusion

The paper investigates the six LSTM models developed on the basis of sequence-to-sequence and sequence-to-vector models for dealing with vanishing problems in forecasting power demands. The loss functions of MAE and MSE have been incorporated into the development of LSTMs. The LSTM models were evaluated on predicting multiple outputs using multiple inputs with 50 timesteps.

Multiple days ahead with 30 timesteps and 50 timesteps, single day ahead with 50 timesteps. MAE had better performance for most of the models compared with than MSE. The study also reveals that sequence-to-sequence based models has scored better than the sequence-to-vector, providing an empirical solution to cope with the vanishing problem in LSTM family neural networks.

References

1. M.S. Hossain, N. A. Madlool, N. A. Rahim, J. Selvaraj, A. K. Pandey, and A. F. Khan, "Role of smart grid in renewable energy: An overview," Renew. Sustain. Energy Rev., vol. 60, pp. 1168-1184, 2016.
2. U. Ugurlu, I. Oksuz, and O. Tas, "Electricity price forecasting using recurrent neural networks," Energies, vol. 11, no. 5, pp. 1–23, 2018.
3. Díaz, G.; Planas, E. A note on the normalization of Spanish electricity spot prices. IEEE Trans. Power Syst. 2016, 31, 2499–2500
4. B. Rohith, T. Santhosh, R. B. Alfred and R. R. Singh, "GUI Energy Demand Forecast using LSTM Deep Learning Model in Python Platform," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), 2021, pp. 1-6, doi: 10.1109/iPACT52855.2021.9696760.
5. L. Li, H. Xiaoguang, C. Ke and H. Ketai, "The applications of WiFi-based Wireless Sensor Network in Internet of Things and Smart Grid," 2011 6th IEEE Conference on Industrial Electronics and Applications, 2011, pp. 789-793, doi: 10.1109/ICIEA.2011.5975693.
6. E. Yuniarti, N. Nurmaini, B. Y. Suprapto and M. Naufal Rachmatullah, "Short Term Electrical Energy Consumption Forecasting using RNN-LSTM," 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), 2019, pp. 287-292.
7. P. Bunnoon, K. Chalermyanont, and C. Limsakul, "Mid-term load forecasting: Level suitably of wavelet and neural network based on factor selection," Energy Procedia, vol. 14, pp. 438–444, 2012
8. F. Chollet, Deep learning with Python, 1st ed. Shelter Island, NY 11964: Manning Publications Co., 2018, pp. 31-37,46-52, 72, 97, 98, 101-110.
9. A. Géron and R. Demarest, Hands-on machine learning with Scikit-Learn and TensorFlow, 2nd ed. Sebastopol (Clif.) [etc.]: O'Reilly, 2019, pp. 135-138, 293, 333-338, 365-367, 466470, 497-518.
10. D. Foster, in Generative deep learning, Sebastopol, CA: O'Reilly Media, Inc., 2019, pp. 37–44.
11. C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.
12. Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258)
13. Glorot, X. and Bengio, Y., 2010, March. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256). JMLR Workshop and Conference Proceedings.
14. Lane, H., Howard, C. and Hapke, H., 2019. Natural Language Processing in Action Video Edition. Shelter Island, NY 11964: Manning Publications Co., pp.274-285.
15. https://data.nationalgrideso.com/demand/ historic-demand-data> [Accessed 13 June 2022].
16. Gavin, C., 2014. Seasonal variations in electricity demand. [online] Assets.publishing.service.gov.uk.https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/295225/Seasonal_variations_in_electricity_demand.pdf.