EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 6 No. 1 (2025) 58-66
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Electrical Load Forecasting using Long Short-Term Memory (LSTM)

Tee Xian Qing[1], Danial Md Nor[1]*

[1] *Faculty of Electrical and Electronics Engineering*
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, MALAYSIA

*Corresponding Author: danial@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2025.06.01.008

Article Info

Received: 9 January 2025
Accepted: 4 February 2025
Available online: 9 May 2025

Keywords

Forecasting, Electrical Load, Long Short-Term Memory, Machine Learning, Artificial Intelligence

Abstract

Accurate electric load forecasting is essential for effective energy management and optimization. This project uses Long Short-Term Memory networks, a type of Recurrent Neural Network, to predict electric load due to their ability to capture long-term dependencies in time-series data. Features such as weather and holidays affect load consumption. This project aims to determine whether the number of features and the number of Long Short-Term Memory layers impact prediction accuracy. The project develops and evaluates 1-layer, 2-layer, and 3-layer models for forecasting. The two years of data used for this study is sourced from Kaggle, provided by a power supply company in Johor, Malaysia. Additional weather data is collected from the Weather Underground website, and holiday information includes public holidays and weekends. The Long Short-Term Memory models are trained using electric load data combined with features such as time, holiday, temperature, and humidity. Model performance is evaluated using metrics like Mean Absolute Error, Root Mean Squared Error, Mean Absolute Percentage Error, and Mean Squared Error. This project compares results across the three LSTM architectures and discusses the effects of data modifications, model architecture, and the training/validation process on forecasting outcomes. The findings show that the 1-layer Long Short-Term Memory model achieves the best accuracy, with a Mean Absolute Percentage Error of 15%, outperforming the 2-layer and 3-layer models, which achieve 16% and 18% Mean Absolute Percentage Error, respectively. These results can help utility companies optimize electricity generation, plan equipment maintenance, and improve resource procurement strategies.

1. Introduction

The world's energy systems have grown rapidly, leading to higher electricity usage and more complex system designs. In Malaysia, companies like TNB and consumers need to predict daily electricity use to plan better and reduce costs. With rising electricity bills, managing energy efficiently is important to save money and ensure stable power supply [1]. Electrical load forecasting is a key part of this, helping power companies make decisions about producing and distributing electricity more effectively [2]. Accurate load forecasting is essential for electric grid operations, such as generating unit start-up and shutdown schedules, and overhaul planning [3]. It also helps electric utilities in making important decisions, such as planning energy purchases and production, load switching, and infrastructure development [4][5].

The first one who introduced the concept of electrical load forecasting is Samuel Insull, a Chicago based investor and innovator who contributed to the development of integration of the United States' electrical infrastructure. He analysed various load use trends, including residential and commercial end users. With this, he concluded that commercial consumption peaks at night and residential consumption peaks during the day [6]. Traditional forecasting methods, such as regression, multiple regression, exponential smoothing, and iterative reweighted least-squares, have been enhanced over time with advanced tools and research, leading to more accurate predictions. These methods evolved into automated models that adjust to changing environmental factors, with approaches like stochastic time series, support vector machines, and adaptive load forecasting emerging as modified versions [7]. Today, AI techniques, including LSTM models, are widely used for load forecasting, offering advantages like overcoming issues of vanishing and exploding gradients, and excelling in long-term load forecasting (LTLF).

This paper uses deep learning, a type of artificial intelligence, to forecast electricity demand. Specifically, it uses Long Short-Term Memory (LSTM), a special type of neural network that can learn patterns over time. LSTM is ideal for predicting time-based data like electricity usage. Hochreiter et al. [8] and Bengio et al. [9] addressed the issues of learning long-term dependencies (due to vanishing and exploding gradients) in their papers, soon after backpropagation training of the first Elman-style RNNs [10]. To create this model, data such as temperature, humidity, and electricity usage is needed. The goal is to train and test the model to ensure it can accurately predict future electricity needs.

This paper collects 2 years of data from Kaggle and weather underground. The data will include datetime, humidity, temperature, holiday and load. Electricity consumption in Malaysia follows distinct patterns based on seasonal changes, public holidays, and consumer behaviour. For instance, electricity demand tends to be higher during festive seasons such as Chinese New Year, Deepavali, and Hari Raya Aidilfitri due to increased household and commercial activity. Conversely, certain public holidays may see a temporary reduction in industrial energy consumption. Weather conditions also play a significant role in electricity usage, with higher demand typically observed during hotter months when air conditioning usage increases. This study considers these seasonal variations to improve forecasting accuracy.

This paper aims to compare the result for different number of LSTM layer of models to know whether the number of features and number of LSTM layers will affect the result of the accuracy when testing the model. Understanding the trends is crucial for improving forecasting accuracy and ensuring efficient energy management. This paper will design and develop 1-Layer, 2-Layer and 3-Layer LSTM models, validate these models and test the model by comparing their prediction results with the actual value. The results of different models will be compared by using error metrics which are RMSE, MSE, MAPE and MAE.

2. Methodology

Fig. 1 is the block diagram of the methodology for this paper. We expect the MAPE for the models will lower than 20% so the predictions can be more accurate.

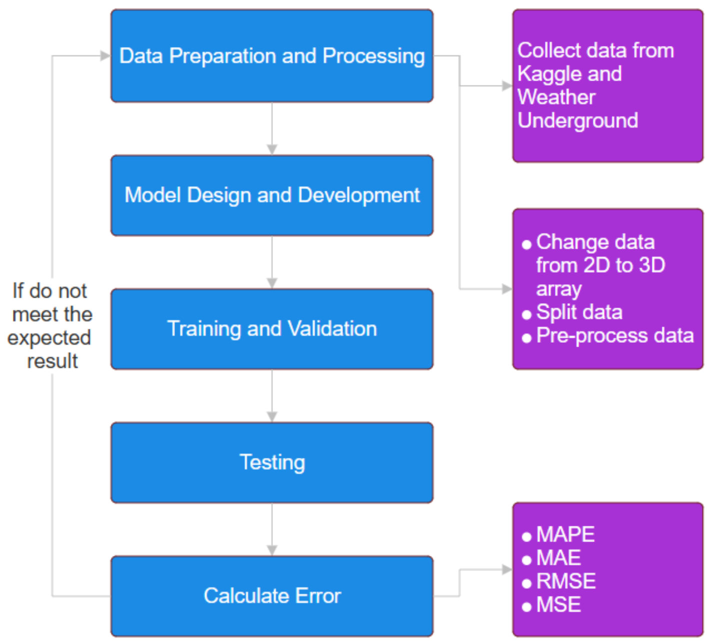


Fig. 1 *Block diagram of electrical load forecasting using LSTM*

2.1 Data preparation and preprocessing

To start this, the data need to be collected to train, validate and test the models. We collected the load data from Kaggle and the weather data from Weather Underground. The holiday includes the public holiday and the weekend. These data will be imported when running the simulation. Fig. 2 shows the data collected from 1 January 2009 1 A.M. to 1 January 2011 12 A.M.

	date	time	temperature ©	humidity(%)	holiday	load
0	2009-01-01	01:00:00	24	100	0	30360
1	2009-01-01	02:00:00	24	100	0	29155
2	2009-01-01	03:00:00	24	94	0	28086
3	2009-01-01	04:00:00	24	94	0	28031
4	2009-01-01	05:00:00	24	94	0	27730
...	...	...	...	...	...	...
17515	2010-12-31	20:00:00	26	89	0	53819
17516	2010-12-31	21:00:00	26	83	0	51543
17517	2010-12-31	22:00:00	25	89	0	44961
17518	2010-12-31	23:00:00	24	94	0	38484
17519	2011-01-01	00:00:00	24	94	1	36729

Fig. 2 *Dataset collected from different sources.*

Jupyter and python were used to run the simulation. The time data will be normalised by Sine and Cosine. Sine and cosine normalization capture the periodic nature of features like time, ensuring that the model understands relationships. The index for the data was changed to datetime. The equations used to normalize are shown in eq. (1) and (2). Fig. 3 is the final dataset used in this paper.

$$time\_sin = \sin\left(\frac{2 * \pi * time}{24}\right) \tag{1}$$

$$time\_cos = \cos\left(\frac{2 * \pi * time}{24}\right) \tag{2}$$

date	time_sin	time_cos	humidity	temperature	holiday	load
2009-01-01 01:00:00	0.258819	0.965926	100	24	0	30360
2009-01-01 02:00:00	0.500000	0.866025	100	24	0	29155
2009-01-01 03:00:00	0.707107	0.707107	94	24	0	28086
2009-01-01 04:00:00	0.866025	0.500000	94	24	0	28031
2009-01-01 05:00:00	0.965926	0.258819	94	24	0	27730
...	...	...	...	...	...	...
2010-12-31 20:00:00	-0.866025	0.500000	89	26	0	53819
2010-12-31 21:00:00	-0.707107	0.707107	83	26	0	51543
2010-12-31 22:00:00	-0.500000	0.866025	89	25	0	44961
2010-12-31 23:00:00	-0.258819	0.965926	94	24	0	38484
2011-01-01 00:00:00	0.000000	1.000000	94	24	1	36729

Fig. 3 *The change of dataset*

Evolution in Electrical and Electronic Engineering Vol. 6 No. 1 (2025) p. 58-66

Due to the LSTM only accept 3D data array as its input, we converted the 2D data array to 3D which 100 timestep, 6 features. Then, we had been split to train, validate, and test data. Fig. 4 is the process of splitting the data; 12542 of the data is the training data, 3136 of the data is the validation data and 1742 of the data is the testing data. The split data will be pre-processed using the equation of standardization. This is used to increase the accuracy of predictions. The equation of standardization is as in eq. (3).

$$x = \frac{(x - mean\ of\ x)}{standard\ derivation\ of\ x} \tag{3}$$

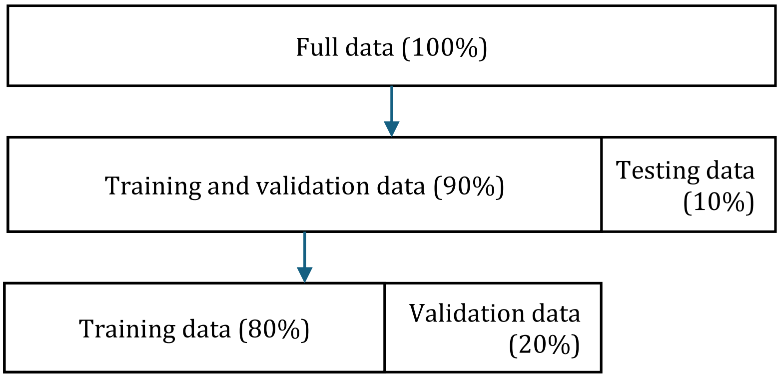


Fig. 4 *Splitting process*

2.2 Model design and development

In this phase, the model is designed and developed using Keras. Fig. 5 is the flow chart for how the models work. Input layer is used to received and reshape the data. Since our x data include 100 timesteps and 6 features, the input layer needs to be set to receive these data. Then, LSTM layer and dropout layer need to be added into the models as 1 layer. While developing 2-Layer model, it will be doubled (LSTM → Dropout →LSTM → Dropout). LSTM is the main layer in the model. It is used to learn long-term dependencies. Drop out layer can be used to avoid overfitting. Dense layer is used to capture complex in the data. In our model, a dense layer with linear activation is added to make the output of the layer to be a linear combination of its inputs, without any non-linearity applied.

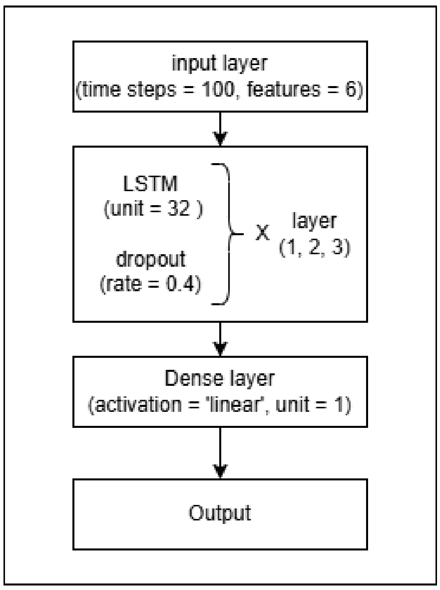


Fig. 5 *Flow chart of the models*

2.3 Training and validation

The Adam optimizer is added to use to train neural networks. It adjusts the learning rate for each parameter based on past gradients and their size. This makes training faster and more stable. MAPE is set as the metrics to determine the best result of the models. The training and validation are set to 200 epochs so it can train the models more to get the best result. Batch size is set to 40 to increase the training speed and accuracy which larger batches provide smoother updates but may miss certain patterns. The best MAPE result will be saved and used while testing the models.

2.4 Testing

After the models are trained and validated, the models will be tested and compared its predicted load value with the actual values. The table and graph are created to show the comparison between the predicted load value and the actual.

2.5 Calculate Error

In this paper, the MSE, RMSE, MAE and MAPE were used as the metrics to calculate the error between the prediction value and the actual value to determine accuracy of the prediction for the models. The n is the number of data, y is the predicted value and $\hat{y}$ is the predicted value. The MSE in eq. (4) calculates a statistical model's error amount. The mean squared difference between the actual and expected values is evaluated. The MSE is equal to 0 in a model that has no errors. The value of the model increases as the error increases.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

RMSE in eq. (5) measures the average difference between a model's predicted values and the actual values. It provides an estimation of the model's accuracy, or how well it can predict the desired value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

MAE in eq. (6) represents the average of the absolute differences between the dataset's actual and predicted values. It computes the dataset's average residual.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

MAPE in eq. (7) also referred to as mean absolute percentage deviation (MAPD), represents the percentage equivalent of MAE. It quantifies the average error magnitude of a model, indicating how far predictions deviate on average. Unlike MAE or MSE/RMSE, which are scale-dependent and influenced by data magnitude, MAPE provides a scale-independent measure of error. This makes it particularly useful for comparing model performance across datasets with varying scales or units.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{7}$$

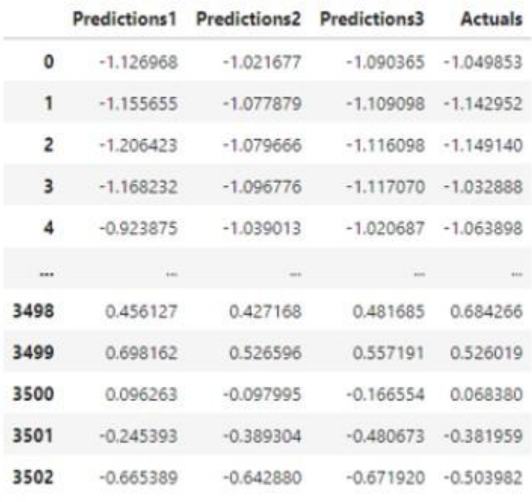
If the accuracy does not meet the expected threshold (MAPE more than 20%), the process will return to the planning phase and iterate until the expected result is achieved.

3. Result and Discussion

To do the comparison between three different models, the table and graph is generated by Python. This section shows the latest best predicted result. Fig. 6 is the comparison between the predicted value of three different models and the actual value. The metrics for three different models are listed in Table 1.

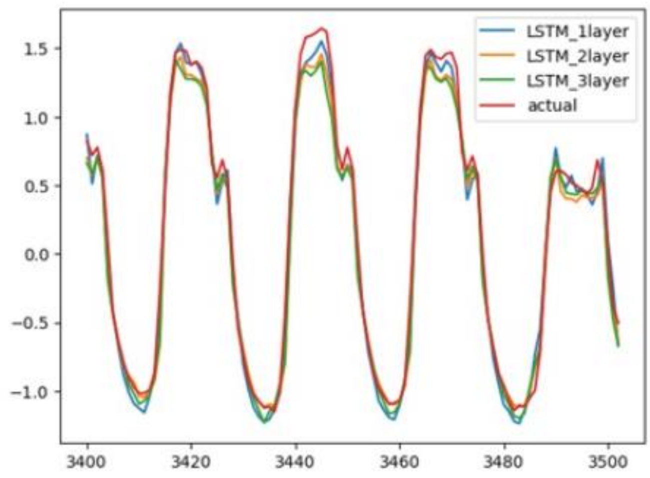
Table 1 *Metrics of three different models.*

Model	MAPE (%)	MSE	RMSE	MAE
1-Layer LSTM	15.062045	0.004255	0.065232	0.047118
2-Layer LSTM	16.017292	0.004227	0.065017	0.046469
3-Layer LSTM	17.934950	0.004452	0.066722	0.048752



	Predictions1	Predictions2	Predictions3	Actuals
0	-1.126968	-1.021677	-1.090365	-1.049853
1	-1.155655	-1.077879	-1.109098	-1.142952
2	-1.206423	-1.079666	-1.116098	-1.149140
3	-1.168232	-1.096776	-1.117070	-1.032888
4	-0.923875	-1.039013	-1.020687	-1.063898
...	...	...	...	...
3498	0.456127	0.427168	0.481685	0.684266
3499	0.698162	0.526596	0.557191	0.526019
3500	0.096263	-0.097995	-0.166554	0.068380
3501	-0.245393	-0.389304	-0.480673	-0.381959
3502	-0.665389	-0.642880	-0.671920	-0.503982

(a)



(b)

Fig. 6 *The predicted values and the actual values (Prediction 1 = 1-layer LSTM, Prediction 2 = 2-layer LSTM and Prediction 3 = 3-layer LSTM).*

The graph in Fig. 7 shows comparison of three different LSTM models' predicted load and actual data from 20th October 2010 11 A.M. to 1st January 2011 12 A.M. The 1-layer model has the best accuracy with a MAPE of 15.06%. The 2-layer model has 16.01%, and the 3-layer model has 17.93%, as seen in Table 1. The 1-layer model is the most accurate, showing that a simple model works better for this data. The 2-layer and 3-layer models are less accurate, possibly because they are too complex. All models follow the actual data patterns but miss some sharp rises and drops. This means they cannot fully capture sudden changes. The load data shows regular cycles, which all models predicted well. Adding more layers didn't improve results, so simpler models are better for this type of data. Simpler models are more efficient for periodic data. Future work could focus on adding features, improving model tuning, and testing other metrics for better evaluation.

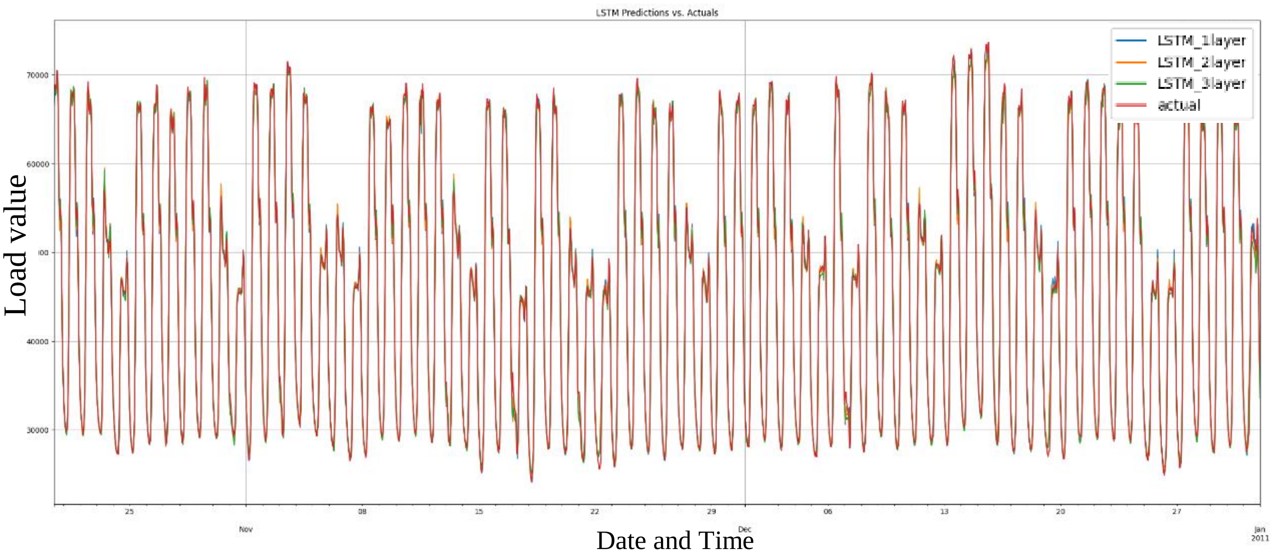


Fig. 7 *Full result for prediction of three different models*

3.1 Relationship between holiday and load

To study the relationship between various features and electricity load, a graph was created. The relationship between the electrical load and the holiday feature stands out the most in Fig. 8, compared to other features. The holiday feature is clearly linked to the load. During weekends and holidays, such as Chinese New Year, Deepavali, Eid al-Fitr (Hari Raya Aidilfitri), and other public holidays, electricity consumption drops. This is because many businesses, factories, and government offices close, leading to a significant reduction in industrial and commercial electricity demand. Additionally, during these holidays, people often travel or spend time at home, further reducing residential electricity consumption. The prediction models (1-layer, 2-layer, and 3-layer LSTMs) follow the actual load trends well, including the dips during holidays. This indicates that the holiday feature plays a crucial role in shaping load patterns and is a key factor in making accurate predictions.

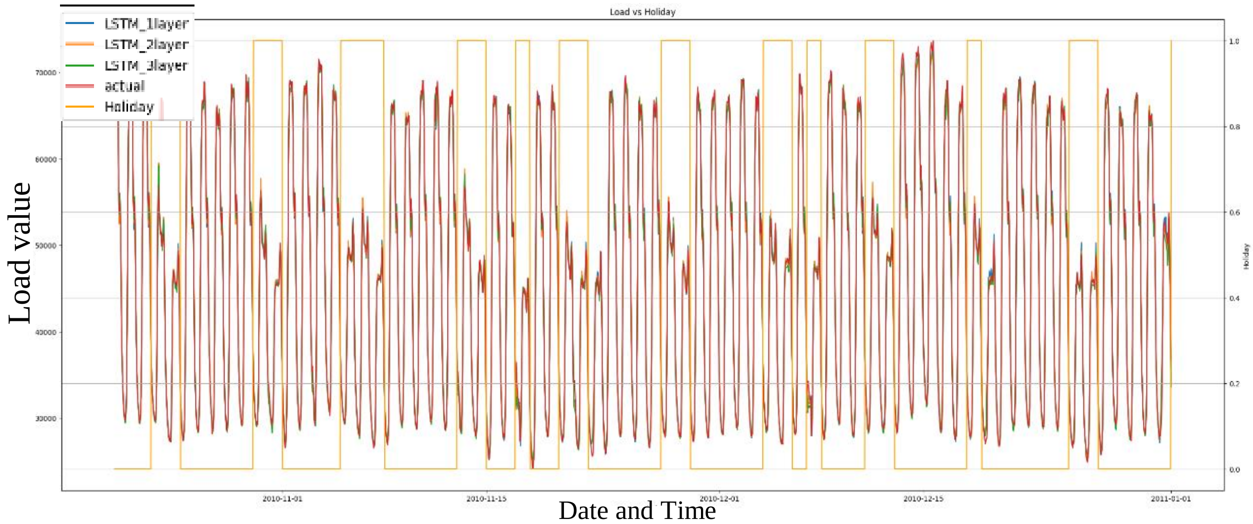


Fig. 8 *Graph for load vs holiday*

3.2 Relationship between humidity, temperature, and load

Fig. 9 shows the relationship between load, humidity, and temperature from 30th December 2010, 7 A.M., to 1st January 2011, 12 A.M. The graph reveals an inverse relationship between load and humidity or temperature. When humidity is higher and temperature is lower, the load consumed tends to be lower. This pattern may be due to end users relying more on cooling systems during hot weather, which increases electricity consumption for air conditioning and refrigeration. Additionally, industrial processes requiring cooling systems for their products might also contribute to higher energy usage during periods of high temperature and humidity. These factors highlight the strong influence of environmental conditions on electricity demand.

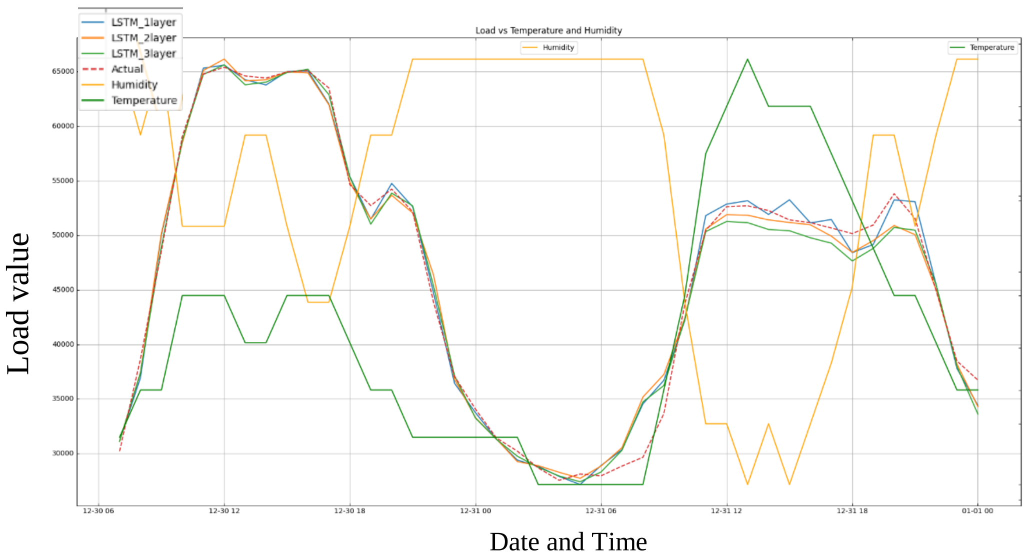


Fig. 9 *Graph for Load vs humidity and temperature from 30th December 2010 7 A.M. to 1st January 2011 12 A.M.*

3.3 Relationship between time and load

The graph in Fig. 10 shows the relationship between load and time features, time_sin and time_cos, which represent the 24-hour cycle of the day. The load follows a clear hourly pattern, with higher load during peak times like mornings and evenings and lower load during late-night hours. The sinusoidal features effectively capture this cyclic behaviour, ensuring smooth transitions between hours, such as from 11 PM to midnight. The LSTM models closely match actual load values, demonstrating that these time features are crucial for learning and predicting the periodic nature of electricity demand.

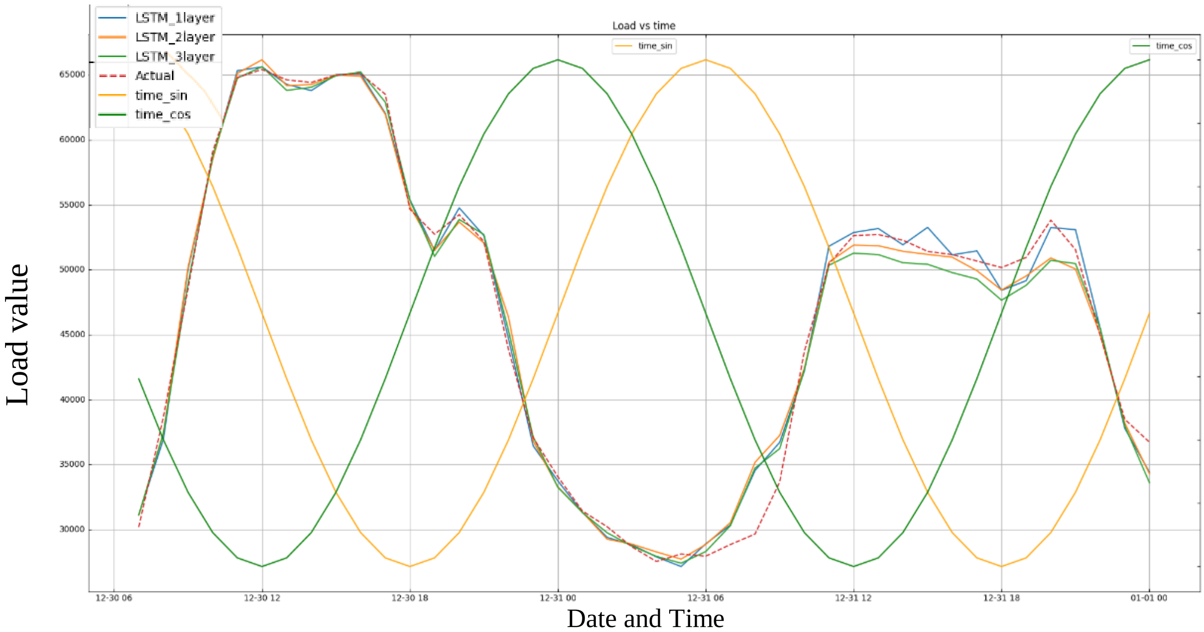


Fig. 10 *Graph for Load vs time and temperature from 30th December 2010 7 A.M. to 1st January 2011 12 A.M.*

4. Conclusion

Electricity demand is increasing annually, making accurate load forecasting essential for utility companies to optimize resource planning and for end users to manage energy costs effectively. This project developed an LSTM model in Python, achieving a MAPE of 15% using data from Kaggle and Weather Underground, validated with metrics like MAPE and RMSE. Factors like peak hours, public holidays, weekends, and weather conditions significantly influenced load consumption, while hyperparameter adjustments affected training and validation outcomes. Future improvements should focus on achieving a MAPE below 5%, accessing proprietary data through collaboration with utility companies, expanding datasets with additional features, utilizing advanced hardware like GPUs for efficient training, and implementing hybrid approaches such as combining CNN and LSTM models to enhance accuracy and robustness.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: study conception and design: Tee Xian Qing; data collection: Tee Xian Qing; analysis and interpretation of results: Tee Xian Qing, Danial Md Nor; draft manuscript preparation: Tee Xian Qing, Danial Md Nor. All authors reviewed the results and approved the final version of the manuscript.

References

[1] W. Sarapan, N. Khotsriwong, N. Boonrakchat, P. Boonraksa, T. Boonraksa, and B. Marungsri, "Weekly Load Demand Forecasting using Supervised Deep Learning Techniques: A Case Study of Suranaree University of Technology," 2023 International Electrical Engineering Congress (iEECON), Krabi, Thailand, 2023, pp. 421-424, doi: 10.1109/iEECON56657.2023.10126612.

[2] M. Abumohsen, A. Y. Owda, and M. Owda, "Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms," Energies, vol. 16, no. 5, p. 2283, 2023.
[3] G. Dudek, P. Piotrowski, and D. Baczyński, "Intelligent Forecasting and Optimization in Electrical Power Systems: Advances in Models and Applications," Energies, vol. 16, no. 7, p. 3024, 2023.
[4] Z. Liu, X. Wang, J. Xing, M. Ren, and X. Xu, "Short-Term Power Load Forecasting Based on IWOA-Attention-BiLSTM," 2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS), Chengdu, China, 2022, pp. 444-450, doi: 10.1109/CCIS57298.2022.10016322.
[5] S. Desai, T. Dalal, S. Kadam, and S. Mishra, "Electrical Load Forecasting using Machine Learning," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 2021, pp. 1-6, doi: 10.1109/ICSCAN53069.2021.9526444.
[6] M. Madhukumar, A. Sebastian, X. Liang, M. Jamil, and M. N. S. K. Shabbir, "Regression Model-Based Short-Term Load Forecasting for University Campus Load," IEEE Access, vol. 10, pp. 8891-8905, 2022, doi: 10.1109/ACCESS.2022.3144206.
[7] A. K. Singh, Ibraheem, S. Khatoon, M. Muazzam, and D. K. Chaturvedi, "Load forecasting techniques and methodologies: A review," in 2012 2nd International Conference on Power, Control and Embedded Systems, Allahabad, India, 2012, pp. 1-10, doi: 10.1109/ICPCES.2012.6508132.
[8] S. Hochreiter, A. S. Younger, and P. R. Conwell, "Learning to Learn Using Gradient Descent," in G. Dorffner, H. Bischof, and K. Hornik, eds., Artificial Neural Networks — ICANN 2001, Lecture Notes in Computer Science, vol. 2130, Berlin, Heidelberg: Springer, 2001, doi: 10.1007/3-540-44668-0_13.
[9] 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, doi: 10.1109/72.279181.
[10] J. Elman, "Finding Structure in Time," Cognitive Science, vol. 14, no. 2, pp. 179-211, 1990, doi: 10.1016/0364-0213(90)90002-E.