Proceedings of CONF-CDS 2026 Symposium: Machine Learning and Neural Network Applications in Engineering
DOI: 10.54254/2755-2721/2026.AST36187

A PyTorch-Based Long Short-Term Memory (LSTM) Model for Short-Term Electricity Load Forecasting in South Australia

Wenjun Guo

College of Engineering and Information Technology, Adelaide University, Adelaide, Australia
a3166945@adelaide.edu.au

Abstract. Accurate electricity load forecasting is important for power system operation, energy planning and demand management. With the increasing use of data-driven methods, deep learning models have become useful tools for analysing complex electricity demand patterns. This study develops a PyTorch-based LSTM model for short-term electricity load forecasting, using South Australian electricity demand data from 2024. The original dataset consists of chronological five-minute demand records, and total electricity demand is selected as the target variable. Missing values were checked, the demand series was normalised using training-set statistics, and supervised learning samples were generated by a sliding window method. In the proposed model, the previous 24 five-minute demand observations are used to predict the next demand value. The dataset is divided into training, validation and testing subsets in chronological order. The model is trained using mean squared error loss and the Adam optimiser with a step learning-rate schedule. The results show that the LSTM model can follow the main movement of electricity demand, but the prediction curve is smoother than the actual curve and some short-term fluctuations remain difficult to capture. This limitation is mainly caused by the use of a simple univariate input, because weather, calendar, electricity price and renewable generation variables are not included. Future work should introduce multivariate features, compare LSTM against benchmark models such as GRU and ARIMA, and test longer input windows to further improve forecasting accuracy.

Keywords: Electricity load forecasting, Long Short-Term Memory (LSTM), PyTorch, Time series prediction, South Australia

1. Introduction

Electricity load forecasting is an important task in modern power systems. The balance between electricity supply and demand must be maintained continuously, so system operators need reliable demand prediction to support generation scheduling, market operation and grid planning [1, 2]. Short-term load forecasting (STLF) is especially useful because electricity demand changes from hour to hour due to human activity, weather conditions, day type and operating conditions [2, 3]. Traditional STLF methods include linear regression, autoregressive integrated moving average

DOI: 10.54254/2755-2721/2026.AST36187

(ARIMA) models and other statistical time series models. These methods can be effective when the underlying patterns in the data are relatively stable, but they may be less suitable when demand contains nonlinear and time-dependent behaviour [3].

Neural network models have been widely used for STLF because they can learn nonlinear relationships from historical data [4, 5]. Among these models, Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) structure that is designed to learn sequential information and reduce the long-term dependency problem found in basic RNNs [6]. Recent studies have used LSTM, gated recurrent unit (GRU), convolutional neural network (CNN)-LSTM and other hybrid models for load forecasting, and these studies show that deep learning can be useful when temporal patterns are strong [4, 5, 7].

South Australia serves as a useful case for this study because its demand profile is affected by strong daily variation and high renewable energy penetration. These features make the demand curve less smooth than a simple daily pattern and create a need to test whether a basic LSTM model can still capture short-term movement. The difference in this study is not to build a highly complex forecasting system, but to provide a clear PyTorch-based baseline for South Australian demand data. The objective is to test whether a simple univariate LSTM model can predict the next demand value using only recent historical demand values. This can provide a practical starting point for later work that adds weather, calendar, electricity price and renewable generation information.

2. Data source and preprocessing

2.1. South Australian electricity demand data

The dataset used in this study is South Australian electricity demand data for 2024. The raw export contains chronological records for the SA1 region, including variables such as SETTLEMENTDATE, TOTALDEMAND, regional reference price and period type. Only SETTLEMENTDATE and TOTALDEMAND are used in the current forecasting model. SETTLEMENTDATE represents the timestamp of each record, while TOTALDEMAND represents the electricity demand value in megawatts (MW). The data used in this experiment has a five-minute time resolution. Since 2024 is a leap year, the full-year dataset contains 105,408 expected five-minute records. The TOTALDEMAND column was checked before modelling, and the version used for this experiment contained no missing demand values, giving a missing-value ratio of 0.00%. The records are kept in chronological order because the task is a time series forecasting problem. Therefore, training, validation , and testing subsets are split chronologically rather than by random shuffling.

2.2. Data cleaning and normalisation

Before model training, the timestamp column is converted into a datetime format and the demand column is converted into a numerical series. Although no missing TOTALDEMAND values were found in the current file, the code still includes a missing-value handling step to make the procedure more robust. If missing demand values appear in a later export, they should be filled with a value calculated from the training set only, such as the training-set mean, to avoid information leakage from the validation or testing sets.

Neural network models are sensitive to the scale of input values. If the original electricity demand values are directly used, the optimisation process may become less stable. Therefore, Min-Max normalisation is applied before training. The minimum and maximum values are calculated

DOI: 10.54254/2755-2721/2026.AST36187

from the training set, and the same values are used to normalise the validation and testing sets. This prevents the model from using information from the testing period during training. The normalisation formula is shown in Equation (1).

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x is the original demand value, x_{min} is the minimum demand value in the training set, and x_{max} is the maximum demand value in the training set. After prediction, the normalised values are converted back to the original MW units for result analysis and plotting.

2.3. Sequence construction

The cleaned and normalised demand data are converted into supervised learning samples by a sliding window method. In this study, the sequence length is set to 24. Since the original data interval is five minutes, 24 observations represent two hours of historical demand rather than one full day. The input sequence and prediction target are defined in Equation (2).

$$X_t = [x_t^{-24}, x_t^{-23}, \ldots, x_t^{-1}], y_t = x_t \quad (2)$$

This means that the model uses the previous 24 demand observations to predict the next five-minute demand value. This design is suitable for a basic one-step-ahead forecasting experiment. However, it may not fully capture longer daily or weekly patterns. For this reason, future work should test longer windows, such as 288 five-minute steps for one full day.

3. Methodology

3.1. LSTM neural network

LSTM is a special type of RNN developed to learn long-term dependencies in sequential data [6]. In contrast to vanilla RNNs, the LSTM unit employs dedicated gates to regulate the volume of information to be retained, discarded and propagated to the subsequent time step. The main components of an LSTM cell include the input gate, forget gate, cell state and output gate. The forget gate controls which information from the previous cell state should be removed. The input gate controls which new information should be added. The output gate controls the hidden state passed to the next time step.

Electricity demand has a clear time-dependent structure. For example, demand during one period is usually related to demand in earlier periods. This makes LSTM suitable for a short-term load forecasting task. In this project, the LSTM model learns the relationship between the previous 24 demand values and the next demand value.

3.2. PyTorch model structure

The forecasting model is implemented in PyTorch, an open-source deep learning framework enabling flexible neural network design and automatic differentiation [8]. The model contains one LSTM module and one fully connected linear layer. The LSTM layer extracts temporal features from the input sequence, and the linear layer maps the final hidden output to one predicted demand value. The main model parameters are shown in Table 1.

DOI: 10.54254/2755-2721/2026.AST36187

Table 1. Hyperparameters of the LSTM forecasting model

Parameter	Value
Input size	1
Hidden size	64
Number of LSTM layers	2
Output size	1
Batch size	32
Sequence length	24 five-minute observations
Optimiser	Adam
Initial learning rate	0.001
Weight decay	0.0001
Epochs	100
Loss function	Mean squared error (MSE)

The LSTM model receives input data with the shape (batch size, sequence length, and input size). Since only the demand value is used as the input feature, the input size is 1. The hidden size is set to 64, and two stacked LSTM layers are used. The output size is 1 because the model predicts one future demand value at each prediction step.

3.3. Training process

The full dataset is divided into three parts: the first 60% serves as the training set, the next 20% as the validation set, and the final 20% as the testing set. This split is made in chronological order. Random shuffling is not used because it would break the temporal structure of the electricity demand series. The model is trained for 100 epochs. MSE is used as the loss function. The Adam optimiser is selected because it is computationally efficient and commonly used for neural network training [9].

A step learning-rate scheduler is applied to improve training stability. The initial learning rate is 0.001 during epochs 1-50. After epoch 50, the learning rate is multiplied by 0.1 and reduced to 0.0001 for epochs 51-100. This allows the model to learn with a larger update step in the early stage and then fine-tune the parameters with a smaller update step later. No active early stopping is used in this experiment. Instead, the validation loss is calculated after each epoch, and the model checkpoint with the lowest validation loss is saved as the final model. This makes the experiment reproducible because the number of epochs is fixed while the selected model is still based on validation performance.

3.4. Evaluation metrics

The forecasting performance is evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). MAE measures the average absolute error in the same unit as the original demand data. RMSE penalizes large errors more heavily. MAPE gives the average percentage error, but it may become unstable when the actual value is close

DOI: 10.54254/2755-2721/2026.AST36187

to zero. For this reason, MAE and RMSE are treated as the main numerical evidence, while MAPE is reported as an additional indicator. The formulas are shown in Equation (3)- Equation (5).

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i | \tag{3}$$

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

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{5}$$

where y_i is the actual demand value, $\hat{y}_i$ is the predicted demand value, and n is the number of testing samples.

4. Results and discussion

4.1. Forecasting result

The prediction result is shown in Figure 1. The blue curve represents the actual electricity demand, and the orange curve represents the demand predicted by the LSTM model. The figure displays a selected interval from the testing set to make the comparison clear. The predicted curve follows the main trend of the actual demand curve, which shows that the model has learned useful short-term temporal information from historical demand values.

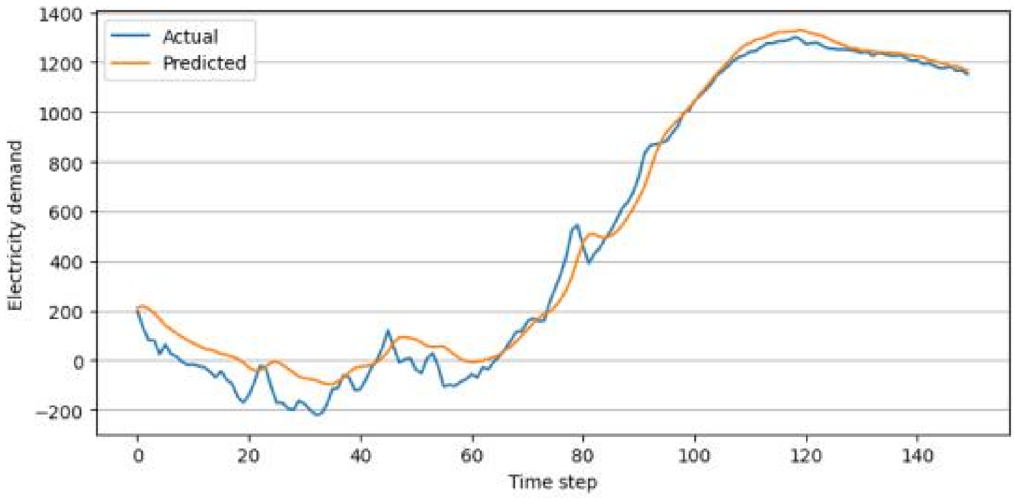

Figure 1. Actual and predicted electricity demand in a selected testing interval

The model performs better during periods when the demand changes smoothly. In the high-demand region, the predicted curve is close to the actual curve and follows the main upward and downward trend. However, the predicted curve is smoother than the actual curve. Some short-term fluctuations and local extreme values are not fully captured. This is a common limitation of a simple univariate neural network model because it mainly learns the average temporal pattern from recent historical demand values.

DOI: 10.54254/2755-2721/2026.AST36187

4.2. Error analysis

The quantitative forecasting performance is shown in Table 2. The MAE value is 30.47 MW and the RMSE value is 44.67 MW. The RMSE is larger than the MAE, which indicates that larger errors occur during rapid demand fluctuations. The MAPE value is 28.42%. This percentage value should be interpreted carefully because MAPE can be sensitive when actual demand values are close to zero or negative. Therefore, MAE and RMSE provide a more reliable interpretation for this dataset.

Table 2. Forecasting performance of the LSTM model

Metric	Value	Unit
MAE	30.47	MW
RMSE	44.67	MW
MAPE	28.42	%

The prediction error is not only caused by the model structure. It is also related to the input variables. In this experiment, the model only uses historical electricity demand. Electricity demand is affected by many external factors, such as temperature, humidity, day type, public holidays, electricity price and solar generation [2, 10]. These factors are not included in the current model. Therefore, the model can mainly capture regular short-term temporal patterns, but it may not fully account for sudden changes in demand.

4.3. Discussion of model performance

The results suggest that the LSTM model is useful as a basic baseline for short-term electricity load forecasting. It can follow the main direction of the demand curve and provide a reasonable one-step-ahead prediction based on past values. This finding agrees with previous studies showing that recurrent neural network models can be effective for electricity load forecasting [4, 5, 7]. At the same time, the model still has clear limitations. First, the model is univariate. It only uses historical demand as the input. This makes the model simple and easy to implement, but it also limits forecasting accuracy. In real power systems, demand is strongly affected by weather, calendar variables and renewable generation output. Without these variables, the model may not predict abnormal or sudden demand changes accurately.

Second, the current experiment does not compare LSTM with other models. A more comprehensive study should compare the proposed model with ARIMA, support vector regression, GRU, CNN-LSTM , or attention-based models [3, 5, 11, 12]. This would make it clearer whether LSTM offers an advantage for the South Australian dataset. Third, the sequence length is only 24 five-minute steps, which represent two hours of historical data. This is suitable for a simple short-horizon experiment, but it may be too short to capture the full daily cycle. A future model should test longer windows, including 288 observations for one full day and multi-day input windows.

Finally, Figure 1 only presents a selected interval from the testing set. This is useful for visual comparison, but it should not be treated as the full performance evidence. The full testing-set metrics in Table 2 are more important for judging model accuracy. Overall, the model captures the main load trend, but the prediction is smoother than the actual demand. This shows that the model has learned general temporal behaviour but has a limited ability to predict short-term fluctuations without additional explanatory variables.

DOI: 10.54254/2755-2721/2026.AST36187

5. Conclusion

This study developed a PyTorch-based LSTM model for short-term electricity load forecasting using South Australian electricity demand data. The original five-minute dataset was cleaned and converted into supervised learning samples through a sliding window method. The previous 24 demand values were used as the input sequence, and the following demand value was used as the prediction target. The data were divided into training, validation and testing sets in chronological order to preserve the time series structure. The model was trained using MSE loss, the Adam optimiser and a step learning-rate schedule.

The forecasting result shows that the LSTM model can capture the main movement of South Australian electricity demand. The predicted curve follows the actual demand curve in most parts of the selected testing interval. The final testing metrics are 30.47 MW for MAE, 44.67 MW for RMSE and 28.42% for MAPE. However, the model does not fully capture sudden fluctuations and local extreme values. The predicted curve is smoother than the actual curve, especially during rapid load changes.

The main limitation is that the current model uses only one input variable. Since it does not include weather, calendar, electricity price , or renewable generation data, it cannot fully account for all factors affecting electricity demand. The study should therefore be treated as a baseline experiment rather than a complete operational forecasting system. Future work should include more input features, compare LSTM with benchmark models such as ARIMA, GRU and CNN-LSTM, and test longer input windows. These improvements may reduce the prediction error and enhance the utility of the forecasting model for power system operation and planning.

References

[1] Australian Energy Market Operator. (2024). Aggregated price and demand data. *National Electricity Market Data Portal*.

[2] Akhtar, S., Ahmad, H., Khan, A., & Hassan, M. (2023). Short-term load forecasting models: A review of challenges, progress, and the road ahead. *Energies, 16*(10), 4060.

[3] Ullah, K., Qaisar, S. B., & Ali, M. (2024). Short-term load forecasting: A comprehensive review and simulation study with CNN-LSTM hybrids approach. *IEEE Access*.

[4] Abumohsen, M., Owda, A., & Owda, M. (2023). Electrical load forecasting using LSTM, GRU, and RNN algorithms. *Energies, 16*(5), 2283.

[5] Dong, Q., Wu, Y., & Pinson, P. (2024). Short-term electricity-load forecasting by deep learning. *arXiv preprint arXiv:2408.16202*.

[6] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

[7] Zhou, Y., Zhang, J., & Wang, L. (2024). A short-term electricity load forecasting method integrating empirical mode decomposition and long short-term memory neural network. *Frontiers in Energy Research, 12*.

[8] Paszke, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (Volume 32)*.

[9] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[10] Sen, D., Chakraborty, I., Kundu, S., Reiman, A. P., Beil, I., & Eiden, A. (2022). KPF-AE-LSTM: A deep probabilistic model for net-load forecasting in high solar scenarios. *arXiv preprint arXiv:2203.04401*.

[11] Yang, W., Sparrow, S. N., & Wallom, D. C. H. (2023). A generalised multi-factor deep learning electricity load forecasting model for wildfire-prone areas. *arXiv preprint arXiv:2304.10686*.

[12] [1] Zhao, X., Li, Y., & Chen, H. (2025). Research on a short-term power load forecasting method based on a three-channel LSTM-CNN hybrid model. *Electronics, 14*(11), 2262.