New Explorations in Electrical Engineering (NECE)
Department of Electrical Engineering,
Faculty of Engineering
Universiti Malaya, MALAYSIA

Vol. 2, No. 1 (2026)
pp. 41–51

Comparative Analysis of Sequential and Non-Sequential Machine Learning Models for Electrical Load Forecasting in Medical Facilities

Muhammad Alif Hidayat Zulkarnain[a], Muhammad Daniel Fahmi Mohd Razi[a], Muhammad Mokhzaini Azizan[a,*] and Azharudin Mukhtaruddin[b]

[a] *Department of Electrical and Electronic Engineering, Faculty of Engineering and Built Environment, Universiti Sains Islam Malaysia, Nilai, Negeri Sembilan, Malaysia*
**E-mail: mokhzainiazizan@usim.edu.my*

[b] *Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia*

Abstract. Medical buildings require a reliable power supply, yet their electrical demand is difficult to predict because equipment use, occupancy and air-conditioning loads change rapidly, and many accurate forecasting models are too demanding for low-cost edge devices. This study compares a sequential model, long short-term memory (LSTM), with a non-sequential model, extreme gradient boosting (XGBoost), for hour-ahead load forecasting in a medical faculty building. Public circuit-level measurements were averaged to hourly values and split chronologically into training, validation and test sets. Both models used the same 24-hour look-back window, were tuned by grid search on the validation set and were run on the same computing platform. Accuracy was assessed with the root mean square error (RMSE), the mean absolute error and the coefficient of determination, and cost with training and inference time. The tuned LSTM gave the lowest test error (RMSE 6.12 kW), ahead of XGBoost (7.51 kW), and both outperformed a persistence baseline (8.30 kW). However, XGBoost trained about 220 times faster and produced forecasts about 180 times faster. The results reveal a clear accuracy-versus-cost trade-off that can guide model selection for edge-based energy management in medical buildings.

Keywords. Electrical load forecasting, Long short-term memory, Extreme gradient boosting, Medical facilities, Edge computing

1. Introduction

1.1. Background

Electrical energy management has become a key part of operating modern smart buildings (El Husseini et al., 2025; Shah et al., 2022). Medical facilities place especially high demands on their power systems, because life-support equipment, laboratory devices, lighting and heating, ventilation and air-conditioning (HVAC) systems must run without interruption (Cao et al., 2020; Koç & Seçkiner, 2024). Artificial intelligence is now widely used to improve grid reliability (Aslam et al., 2020) and to support clinical prediction and healthcare planning (Chandragandhi et al., 2025; Nathan et al., 2024). For time-series forecasting, sequential models such as long short-term memory (LSTM) networks keep an internal

NECE. All rights reserved.

memory while reading a sequence of past values (Katsarou, 2025; Wu et al., 2021), whereas non-sequential models such as extreme gradient boosting (XGBoost) treat each time step as an independent row of input features (Mazibuko & Akindeji, 2025; Zinovyeva, 2025).

1.2. Problem statement

Few studies have examined load forecasting inside medical buildings, even though their demand differs from that of ordinary commercial buildings (Cao et al., 2020; Panagiotou & Dounis, 2022; Taha et al., 2023). Three features make this task difficult: occupancy and equipment schedules are less regular than a fixed daily profile; large items such as air-conditioning compressors start and stop suddenly, creating sharp peaks; and forecasts are increasingly produced on small edge devices close to the switchboard, where memory, processing power and response time are limited (Antaki et al., 2025; Forough et al., 2021).

Sequential models have been applied successfully at the network edge in other fields, such as security monitoring (Forough et al., 2021), and sequential and non-sequential models have been combined or compared in general energy forecasting (Dakheel & Çevik, 2025; Mazibuko & Akindeji, 2025). However, their relative accuracy and cost in medical buildings, where load records are often short and irregular, remain unclear, and a fair comparison requires both models to be tuned with equal care (Grinsztajn et al., 2022; Hewamalage et al., 2023).

1.3. Related work and research gap

Data-driven machine learning has largely replaced traditional statistical methods for building energy forecasting (El Husseini et al., 2025; Khan et al., 2021; Shah et al., 2022). For hospitals, ensemble tree models such as XGBoost and random forest have given better daily load predictions than single models (Cao et al., 2020), LSTM networks have been tested for hospital energy use (Panagiotou & Dounis, 2022), and hybrid LSTM–XGBoost models can capture both time trends and feature interactions (Ajder et al., 2025; Dakheel & Çevik, 2025; Mazibuko & Akindeji, 2025). Careful tuning of LSTM settings also has a strong effect on accuracy (Karima et al., 2026). Table 1 summarises these studies and the remaining gap.

Table 1. Summary of related predictive modelling approaches.

Research focus	Main methods	Strengths	Limitations
Smart-building profiling (El Husseini et al., 2025; Khan et al., 2021; Shah et al., 2022)	Ensemble and hybrid statistical models	Good baselines for commercial buildings	Do not reflect 24-hour medical loads
Hospital forecasting (Cao et al., 2020; Koç & Seçkiner, 2024; Panagiotou & Dounis, 2022; Taha et al., 2023)	XGBoost, random forest, ANN, LSTM	Real hospital data; value of ensembles shown	Mostly daily or longer horizons; cost ignored
Sequential and hybrid forecasting (Ajder et al., 2025; Dakheel & Çevik, 2025; Karima et al., 2026; Wu et al., 2021)	LSTM, CNN-LSTM, LSTM–XGBoost	Learn long time dependencies	Computationally heavy; sensitive to tuning
Tree ensembles versus neural networks (Grinsztajn et al., 2022)	Tree ensembles, deep networks	Show strength of trees on tabular data	Not tested on short, irregular building loads

This paper addresses that gap with a fair, tuned and repeatable comparison of both model families that considers accuracy and edge-device cost together.

1.4. Objectives and contributions

The aim of this study is to compare sequential and nonsequential machine learning models for electrical load forecasting in a medical facility. The contributions are: (i) an equitable comparison of a tuned LSTM and a tuned XGBoost model on public circuit data, using an identical observation window and test period; (ii) a fully reported protocol detailing the hyperparameter search, selected settings and computing platform so the study can be repeated (Pineau et al., 2021); and (iii) practical guidance for facility managers choosing a forecasting model for localised energy management.

The remainder of this paper is organised as follows. Section 2 describes the dataset, pre-processing, feature engineering, models and experimental protocol. Section 3 presents and discusses the results, and Section 4 concludes the paper.

2. Methodology

The framework used in this study is shown in Fig. 1.

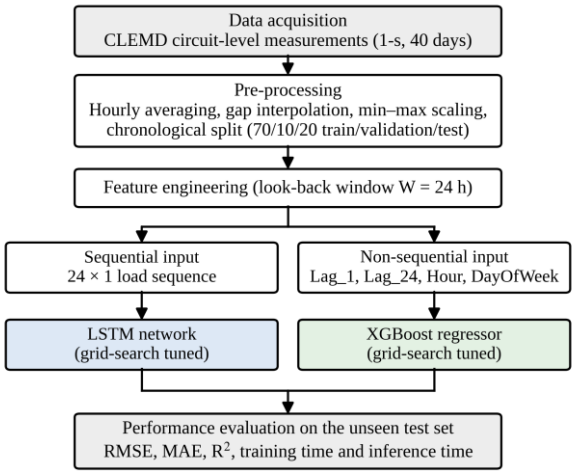


Fig. 1. Methodology framework.

2.1. Data acquisition

The data were taken from the public Circuit-Level Electrical Measurements Dataset (CLEMD) (Al-Khadher et al., 2024), recorded in the low-voltage switch room of the Faculty of Medicine and Defence Health, Universiti Pertahanan Nasional Malaysia. The building contains offices, teaching rooms, laboratories, lifts and a variable refrigerant flow (VRF) air-conditioning system, with part of its load on a generator-backed emergency switchboard (Al-Khadher et al., 2024). Power quality analysers logged voltage, current, and real, reactive and apparent power at 1-s intervals over a 40-day period, from 16 September to 25 October 2023 (Al-Khadher et al., 2024). The aggregate three-phase real power (kW) at the main switchboard was used as the forecasting target.

2.2. Data pre-processing

The 1-s readings were averaged into hourly values across the collection period. Analyser faults (Al-Khadher et al., 2024) left 284 of 914 possible hours missing, concentrated in nine gaps of 22 to 54 hours rather than scattered single hours. Gaps this long cannot be filled by interpolation without fabricating the load trend, so only gaps of three hours or less were interpolated, and any 24-hour window touching a longer gap was excluded from the sliding-window construction described in Section 2.3 (Hewamalage et al., 2023). This left 497 usable windows, split in time order into training (347, 70%), validation (49, 10%) and test (101, 20%) sets so that no future values were used during training; the test set covers 16–25 October 2023. All inputs were scaled to the range [0, 1] with the min–max normalisation in Eq. (1):

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

where x is the original value, and $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the training set. Using only the training-set limits prevents information from the test period from leaking into the model (Hewamalage et al., 2023).

2.3. Feature engineering and look-back window

A sliding window turns the load series into a supervised learning problem (Karima et al., 2026; Zinovyeva, 2025). With a look-back window of size W, the input vector at time t is given by Eq. (2):

$$X_t = \{x_{t-W}, x_{t-W+1}, \dots, x_{t-1}\} \tag{2}$$

In this study, the look-back window was set to $W = 24$, that is, 24 hourly steps covering one complete daily cycle, which is the strongest repeating pattern in building loads; a 24-hour window has also been shown to capture short-term load variation well in hourly forecasting (Karima et al., 2026). The LSTM received the full 24-step load sequence, whereas XGBoost received four features from the same window: the load one hour earlier (Lag_1), the load 24 hours earlier (Lag_24), the hour of day (Hour) and the day of week (DayOfWeek). Weather data were not included because CLEMD does not provide them, and the tropical climate at the site is fairly stable across the year (Al-Khadher et al., 2024).

2.4. Model architectures and hyperparameter optimisation

2.4.1. Sequential model: LSTM

The LSTM network represents the sequential approach because its gates allow it to retain useful information over long sequences without the vanishing-gradient problem (Katsarou, 2025). The forget gate f_t, which decides what information is removed from the cell state, is computed with Eq. (3):

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

where σ is the sigmoid function, W_f is the weight matrix of the forget gate (not to be confused with the window size W), h_{t-1} is the previous hidden state and b_f is the bias vector (Katsarou, 2025). As shown in Fig. 2, the tuned network has one LSTM layer with 128 hidden units (tanh state, sigmoid gates), a dropout layer (rate 0.2) and a linear dense output, giving 66,689 trainable parameters. Overfitting was controlled by dropout and early stopping (patience of 10 epochs), and training stopped after 85 of the 100 allowed

epochs (Karima et al., 2026). The network was trained with the Adam optimiser (learning rate 0.01) and a mean squared error (MSE) loss.

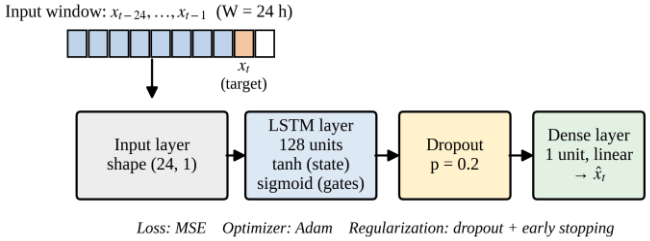


Fig. 2. Sliding-window input and architecture of the tuned LSTM network.

2.4.2. Non-sequential model: XGBoost

XGBoost builds an ensemble of decision trees one after another, and each new tree is fitted to reduce the errors left by the earlier trees (Cao et al., 2020; Mazibuko & Akindeji, 2025). Its objective penalises tree leaf weights and it can sample rows and columns at random, both of which limit overfitting to sudden load peaks. It also reports feature importance scores, which make it easy to check how the engineered lag features are used (Cao et al., 2020).

2.4.3. Hyperparameter optimisation

To give both models an equal chance, the same protocol was applied to each: a grid search over the values listed in Table 2, with each combination trained on the training set and scored on the validation set, keeping the combination with the lowest validation RMSE (Hewamalage et al., 2023; Karima et al., 2026). The selected model was then evaluated once on the unseen test set, with random seeds fixed so that the results can be repeated (Pineau et al., 2021).

Table 2. Hyperparameter search space and selected values (grid search).

LSTM setting	Search space	Selected	XGBoost setting	Search space	Selected
LSTM layers	1, 2	1	Number of trees	100, 300, 500	300
Hidden units	32, 64, 128	128	Maximum depth	3, 5, 7	5
Dropout rate	0.1, 0.2, 0.3	0.2	Learning rate	0.01, 0.05, 0.1	0.01
Learning rate (Adam)	10^{-2}, 10^{-3}, 10^{-4}	10^{-2}	Row sampling	0.8, 1.0	1.0
Batch size	16, 32, 64	16	Column sampling	0.8, 1.0	0.8
Maximum epochs (patience)	100 (10)	100 (10); 85 run	L2 penalty (λ)	1, 5	5
Activation (state/gates/output)	Fixed	tanh/sigmoid/ linear	Objective	Fixed	Squared error

2.5. Experimental environment

Because training and inference times depend strongly on the hardware (Pineau et al., 2021), all experiments were run on the single platform summarised in Table 3. The LSTM used the graphics processor available in this environment, since TensorFlow allocates one automatically when present,

whereas XGBoost trained on the central processor only, which is its default behaviour. This makes the cost comparison conservative, because any bias favours the speed of the LSTM and a real edge device is unlikely to carry a discrete graphics processor. Training time was measured as the wall-clock time needed to fit the final model, and inference time as the time needed to predict the whole test set, each averaged over five runs.

Table 3. Hardware and software environment.

Item	Specification	Item	Specification
Platform	Google Colaboratory (GPU runtime)	Processor	Intel Xeon at 2.00 GHz, 2 virtual CPUs
Memory (RAM)	12.7 GB	GPU (LSTM only)	NVIDIA Tesla T4, 15 GB
Operating system	Linux (Google Colaboratory image)	Software	Python 3.13, TensorFlow 2.20, XGBoost 3.4, scikit-learn 1.6

2.6. Evaluation metrics

Three standard metrics were used (Hewamalage et al., 2023): the root mean square error (RMSE), the mean absolute error (MAE) and the coefficient of determination (R^2), defined in Eqs (4) to (6):

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

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

where y_i is the measured load, $\hat{y}_i$ is the forecast, $\bar{y}$ is the mean measured load and n is the number of test samples. RMSE weights large errors more heavily, which matters most when a missed peak could overload a circuit.

3. Results and discussion

3.1. Feature importance of the non-sequential model

The feature importance of the tuned XGBoost model is shown in Fig. 3. The previous-hour load (Lag_1) is the most important input (49%), followed by DayOfWeek (23%), Hour (14%) and Lag_24 (13%).

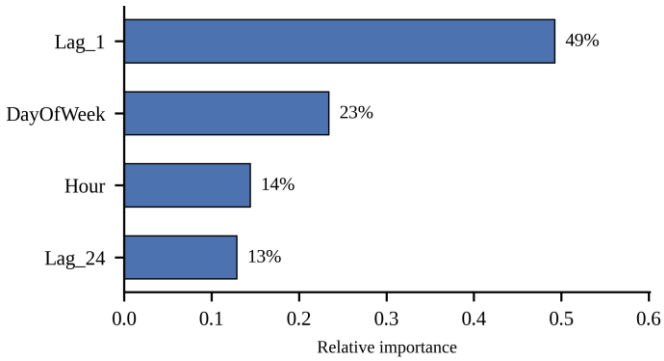


Fig. 3. Feature importance of the XGBoost model.

Feature importance has also been used in earlier hospital studies to identify the main load drivers (Cao et al., 2020), and engineered lag and calendar features give tree models the daily context they otherwise lack (Mazibuko & Akindeji, 2025). The reliance on Lag_1 reflects load persistence from one hour to the next (Hewamalage et al., 2023), while the weight on DayOfWeek reflects the fixed weekday teaching schedule and lower weekend occupancy of the building (Al-Khadher et al., 2024).

3.2. Forecasting accuracy

The test results are summarised in Table 4. The tuned LSTM gave the lowest error (RMSE 6.12 kW, MAE 3.27 kW, $R^2 = 0.905$), moderately ahead of XGBoost (RMSE 7.51 kW, MAE 3.86 kW, $R^2 = 0.857$). Both models clearly outperformed a naive persistence baseline that simply repeats the load of the previous hour (RMSE 8.30 kW, $R^2 = 0.825$), confirming that each model learns more than the trivial one-step pattern that Lag_1 alone would give.

Table 4. Performance of the two models and a persistence baseline on the test set.

Model	RMSE (kW)	MAE (kW)	R^2	Training time (s)	Inference time (s)
LSTM (sequential)	6.122	3.268	0.905	22.418	0.3269
XGBoost (non-sequential)	7.508	3.856	0.857	0.102	0.0018
Persistence (naive baseline)	8.302	4.313	0.825	–	–

This result runs against the common finding that tree ensembles match or beat neural networks on small, tabular-style problems (Grinsztajn et al., 2022). A likely reason is that the load of this building follows a regular institutional schedule: the 24-hour input sequence lets the LSTM track the daily and weekly cycle directly, whereas XGBoost must reconstruct that context from only four hand-engineered features. The gap is moderate rather than large, and the ranking was reversed on the validation set (LSTM 3.85 kW versus XGBoost 2.63 kW), which is a caution against over-trusting model selection on a short record with only 49 validation samples (Hewamalage et al., 2023).

3.3. Forecast trajectories during peak demand

Both forecasts are plotted against the measured load over a continuous 72-hour test window (22–25 October 2023) in Fig. 4. The window contains two working days, each showing a sharp rise from an

overnight baseline of about 8–10 kW to a daytime peak above 60 kW (23 October) and 50 kW (24 October), with a brief mid-peak dip consistent with a lunchtime lull (Al-Khadher et al., 2024).

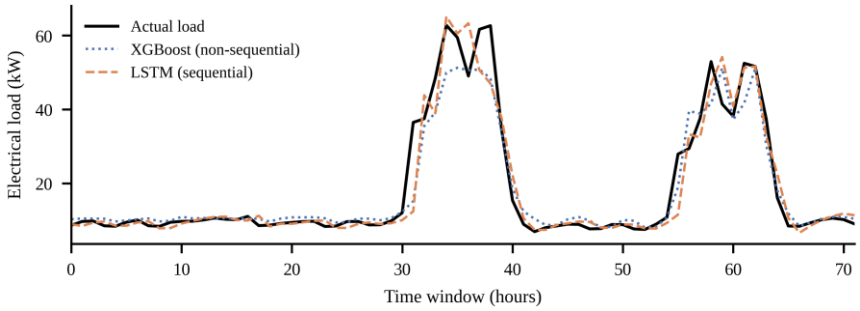


Fig. 4. Measured and forecast load over a continuous 72-hour test window.

Both models track the timing of the peaks closely. XGBoost consistently underestimates peak magnitude, reaching only about 51 kW against actual peaks of 63 kW and 53 kW, which is consistent with lag features that cannot extrapolate beyond recently seen values. The LSTM tracks the first peak more closely but overshoots on the way down, a smoothing lag typical of recurrent models (Ajder et al., 2025; Grinsztajn et al., 2022).

3.4. Distribution of forecast errors

The distribution of absolute errors on the full test set is shown in Fig. 5. The two models have almost identical median errors (XGBoost 1.14 kW, LSTM 1.18 kW), and most hours, namely the long overnight and low-demand periods, are predicted about equally well by both. The models mainly diverge in their tails: XGBoost has a wider interquartile range (0.50 to 3.41 kW versus 0.49 to 2.91 kW) and a higher maximum error (29.9 kW versus 26.8 kW), both occurring during the two demand peaks in Fig. 4. This tail behaviour, rather than typical-hour accuracy, drives the lower overall RMSE of the LSTM.

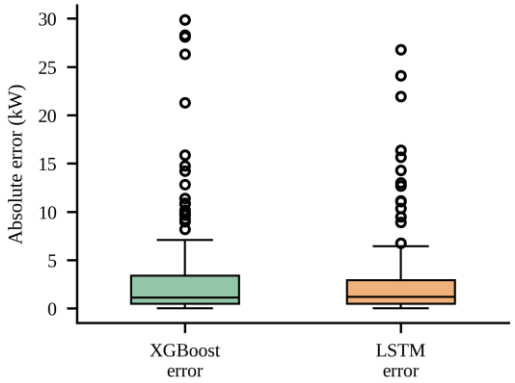


Fig. 5. Distribution of absolute forecast errors on the test set.

Because large under-predictions during peaks are the errors most likely to matter operationally, the narrower error tail of the LSTM is a practically relevant advantage, even though the two models perform almost identically most of the time.

3.5. Trade-off between accuracy and computational cost

Accuracy and computing cost are compared in Fig. 6. XGBoost trained in 0.10 s and predicted the whole test set in 0.0018 s, whereas the LSTM needed 22.42 s and 0.327 s, even with graphics processor acceleration that XGBoost did not use. XGBoost was therefore about 220 times faster to train and about 180 times faster at inference, at the cost of a moderate increase in error (RMSE 7.51 kW versus 6.12 kW).

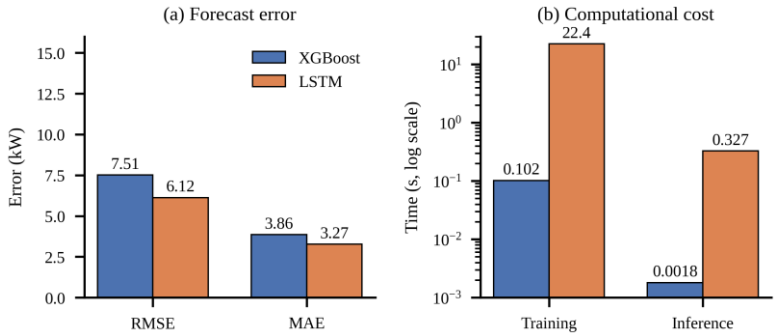


Fig. 6. Trade-off between (a) forecast error and (b) computational cost.

Edge devices have limited processing power (Antaki et al., 2025), so the choice depends on priorities: the LSTM suits accuracy-critical deployments with computing capacity to spare, while XGBoost suits constrained edge hardware or frequent retraining (Grinsztajn et al., 2022; Mazibuko & Akindeji, 2025).

4. Conclusion

This study compared a sequential LSTM network with a non-sequential XGBoost model for hourly electrical load forecasting in a medical faculty building, using public circuit-level data with 31% of hours missing in multi-hour blocks. Both models used the same 24-hour look-back window, grid-search tuning protocol and computing platform, and both clearly outperformed a naive persistence baseline. Contrary to the common assumption that tree ensembles dominate small, tabular-style problems, the tuned LSTM achieved better accuracy (RMSE 6.12 kW, R^2 = 0.905) than XGBoost (RMSE 7.51 kW, R^2 = 0.857), although XGBoost trained about 220 times faster and produced forecasts about 180 times faster, even with the LSTM benefiting from graphics processor acceleration. For facility managers, the choice therefore depends on priorities: the LSTM suits deployments where accuracy is paramount, while XGBoost suits constrained edge hardware or frequent retraining, where its much lower computing cost outweighs a moderate accuracy penalty.

The study has some limitations. The data cover only 40 days from one building with substantial gaps, leaving 497 usable windows and only 49 for validation, and the ranking of the models reversed between validation and test, which is a caution against over-trusting model selection on such a short record (Hewamalage et al., 2023). Future work should (i) use longer, multi-season records from several hospitals to check whether the advantage of the LSTM holds, (ii) add statistical significance tests (Hewamalage et al., 2023), (iii) include weather and occupancy data, and (iv) test federated learning, which allows several hospitals to train a shared model without sharing sensitive data (Briggs et al., 2022).

Acknowledgements

This study was funded by the Research and Innovation Management Centre, Universiti Sains Islam Malaysia (PPPI/FKAB/0122/USIM/12322). The authors also acknowledge the support of the International Transdisciplinary Research Conference (INTRAC) 2026, organised by Universiti Malaya. The authors declare no competing interests. Ethical approval was not required, because the study used a publicly available dataset of building electrical measurements and involved no human participants, animals or personal data.

Statement on the use of generative AI tools: during the preparation of this manuscript, the authors used Google Gemini and NotebookLM to help organise the structure of the manuscript, and Claude Opus 5 (Anthropic) to support language editing and the formatting of the manuscript to the journal template. These tools were used to improve clarity and presentation, and were not used to generate data, results or conclusions. The authors reviewed and edited all tool-assisted content and take full responsibility for the accuracy and originality of the manuscript.

References

Ajder, A., Hamza, H. A. A., & Ayaz, R. (2025). Wavelet-enhanced hybrid LSTM-XGBoost model for predicting time series containing unpredictable events. *IEEE Access, 13*. https://doi.org/10.1109/ACCESS.2025.3556540

Al-Khadher, O., Mukhtaruddin, A., Hashim, F. R., Azizan, M. M., Mamat, H., & Aqlan, A. (2024). CLEMD, a circuit-level electrical measurements dataset for electrical energy management. *Scientific Data, 11*, 594. https://doi.org/10.1038/s41597-024-03433-7

Antaki, B., Dalloul, A. H., & Miramirkhani, F. (2025). Intelligent health monitoring in 6G networks: Machine learning-enhanced VLC-based medical body sensor networks. *Sensors, 25*(11), 3280. https://doi.org/10.3390/s25113280

Aslam, Z., Javaid, N., Ahmad, A., Ahmed, A., & Gulfam, S. M. (2020). A combined deep learning and ensemble learning methodology to avoid electricity theft in smart grids. *Energies, 13*(21), 5599. https://doi.org/10.3390/en13215599

Briggs, C., Fan, Z., & Andras, P. (2022). Federated learning for short-term residential load forecasting. *IEEE Open Access Journal of Power and Energy, 9*, 573–583. https://doaj.org/article/e2e3feb418f7434e9abf62130645f261

Cao, L., Li, Y., Zhang, J., Jiang, Y., Han, Y., & Wei, J. (2020). Electrical load prediction of healthcare buildings through single and ensemble learning. *Energy Reports, 6*, 2751–2767. https://doi.org/10.1016/j.egyr.2020.10.005

Chandragandhi, S., Arvind, C., & Srihari, K. (2025). Advanced predictive disease modeling in biomedical IoT using the temporal adaptive neural evolutionary algorithm. *Scientific Reports, 15*, 20378. https://doi.org/10.1038/s41598-025-08426-z

Dakheel, F., & Çevik, M. (2025). Optimizing smart grid load forecasting via a hybrid long short-term memory-XGBoost framework: Enhancing accuracy, robustness, and energy management. *Energies, 18*(11), 2842. https://doi.org/10.3390/en18112842

El Husseini, F., Noura, H. N., Salman, O., & Chahine, K. (2025). Machine learning in smart buildings: A review of methods, challenges, and future trends. *Applied Sciences, 15*(14), 7682. https://doi.org/10.3390/app15147682

Forough, J., Bhuyan, M. H., & Elmroth, E. (2021). Detection of VSI-DDoS attacks on the edge: A sequential modeling approach. In *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES 2021)* (Article 20, pp. 1–10). Association for Computing Machinery.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022), Datasets and Benchmarks Track*. https://arxiv.org/abs/2207.08815

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: Common pitfalls and best practices. *Data Mining and Knowledge Discovery, 37*(2), 788–832. https://doi.org/10.1007/s10618-022-00894-5

Karima, N. N., Ahmad, S., Islam, A., Huda, A. S. N., Awalin, L. J., Abd Halim, S., Mokhlis, H., Ali, M. S., & Mubarok, S. (2026). Enhancing short-term load forecasting using hyperparameter-optimized deep learning approaches. *Energies, 19*(3), 705. https://doi.org/10.3390/en19030705

Katsarou, A. (2025). *Sequential machine learning for textual and time-series data* [Doctoral dissertation, Technische Universität Berlin].

Khan, A. N., Iqbal, N., Ahmad, R., & Kim, D.-H. (2021). Ensemble prediction approach based on learning to statistical model for efficient building energy consumption management. *Symmetry, 13*(3), 405. https://doi.org/10.3390/sym13030405

Koç, A., & Seçkiner, S. U. (2024). Analysing and forecasting the energy consumption of healthcare facilities in the short and medium term: A case study. *Operations Research and Decisions, 34*(3), 165–192. https://doi.org/10.37190/ord240309

Mazibuko, T., & Akindeji, K. (2025). Hybrid forecasting for energy consumption in South Africa: LSTM and XGBoost approach. *Energies, 18*(16), 4285. https://doi.org/10.3390/en18164285

Nathan, N., Saring, M., Savion-Gaiger, N., Radinsky, K., & Peri, A. (2024). Artificial intelligence in healthcare: A new frontier in predicting and managing chronic diseases. *Israel Medical Association Journal, 26*(2), 108–113. https://www.ima.org.il/MedicineIMAJ/Article.aspx?aId=6428

Panagiotou, D. K., & Dounis, A. I. (2022). Comparison of hospital building's energy consumption prediction using artificial neural networks, ANFIS, and LSTM network. *Energies, 15*(17), 6453. https://doi.org/10.3390/en15176453

Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F., Fox, E., & Larochelle, H. (2021). Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning Research, 22*(164), 1–20. https://jmlr.org/papers/v22/20-303.html

Shah, S. F. A., Iqbal, M., Aziz, Z., Rana, T. A., Khalid, A., Cheah, Y.-N., & Arif, M. (2022). The role of machine learning and the Internet of Things in smart buildings for energy efficiency. *Applied Sciences, 12*(15), 7882. https://doi.org/10.3390/app12157882

Taha, A., Barakat, B., Taha, M. M. A., Shawky, M. A., Lai, C. S., Hussain, S., Abideen, M. Z., & Abbasi, Q. H. (2023). A comparative study of single and multi-stage forecasting algorithms for the prediction of electricity consumption using a UK-National Health Service (NHS) hospital dataset. *Future Internet, 15*(4), 134. https://doi.org/10.3390/fi15040134

Wu, K., Wu, J., Feng, L., Yang, B., Liang, R., Yang, S., & Zhao, R. (2021). An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system. *International Transactions on Electrical Energy Systems, 31*(1), e12637. https://doi.org/10.1002/2050-7038.12637

Zinovyeva, E. (2025). *Deep learning for sequential data* [Doctoral dissertation, Humboldt-Universität zu Berlin].