Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Forecasting Electricity Demand Under Severe Load Shedding: An Explainable and Statistically Validated Deep Learning Benchmark

**Md Ibrahim Shekh**

shekh.ibrahim.433@gmail.com

Bangladesh University of Engineering and Technology https://orcid.org/0009-0007-5196-6453

**Rakibul Hasan Rafi**

Bangladesh University of Engineering and Technology

**Research Article**

**Keywords:** short-term load forecasting, load shedding, energy-constrained power systems, PatchTST, explainable artificial intelligence, Diebold-Mariano test

**Posted Date:** November 17th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11744091/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** The authors declare no competing interests.

# Forecasting Electricity Demand Under Severe Load Shedding: An Explainable and Statistically Validated Deep Learning Benchmark

Md Ibrahim Shekh[a*], Rakibul Hasan Rafi[a]

[a.] Department of Chemical Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh.

***Corresponding author:** Md Ibrahim Shekh (shekh.ibrahim.433@gmail.com).

**Abstract**

Electricity demand forecasting is harder on grids where supply often falls short, because load shedding changes how demand behaves. This study benchmarks one-hour-ahead forecasting on the national grid of Bangladesh from April 2015 to March 2026, a period in which load shedding rose from less than 1% of hours to 44% of hours in 2022 to 2025. Eleven models were compared under a single leak-free protocol: three naive baselines, two linear models, Random Forest, XGBoost and four deep learning architectures (PatchTST, iTransformer, CNN-BiLSTM and BiGRU). All were scored on the same 14,142 test hours from July 2024 to March 2026, 41% of which had load shedding. PatchTST was the most accurate model, with an RMSE of 203.8 MW and a MAPE of 1.26%. Its RMSE was 34% lower than that of a tuned XGBoost and 58% lower than persistence. Diebold-Mariano tests with autocorrelation-robust variance and paired block-bootstrap intervals confirmed that it was significantly better than every other model. The advantage came mainly from load-shedding hours, where the error of XGBoost rose by 36% while that of every deep learning model changed by about 5% or less. Integrated Gradients and TreeSHAP showed that the transformers used the full weekly demand cycle and the recent shedding history, whereas XGBoost could not predict demand above its training range. BatchNorm improved the transformers but not the recurrent models. Forecasting models for energy-constrained grids should therefore be evaluated separately for hours with and without load shedding.

**Keywords:** short-term load forecasting; load shedding; energy-constrained power systems; PatchTST; explainable artificial intelligence; Diebold-Mariano test

## 1. Introduction

Short-term load forecasting (STLF) underpins most grid dispatch decisions. Accurate hourly and day-ahead (1-to-24-hour) demand predictions determine unit commitment, spinning reserve allocation, and generation or maintenance scheduling. Thus, forecasting errors translate directly into unnecessary generating units, higher fuel consumption, increased operational costs, and supply instability [1-4]. On stable and well-metered grids this is a tractable problem as decades of clean training data and predictable seasonal patterns make it accessible to even simple recurrent architectures. On grids where supply does not consistently meet demand, the problem changes in a fundamental way. Utilities record the load they actually serve and estimate the shortfall, and planned outages distort the shape of the daily load curve itself [5]. Usman et al. [6] note that in many developing-country power systems, electricity shortages are routinely managed through load shedding rather than demand-side management. This makes load shedding the de facto mechanism for balancing supply and demand.

Bangladesh between 2022 and 2025 provides a particularly stark case. The country's installed generation capacity grew 285% between 2008–09 and 2020–21, driven largely by natural gas-based generation alongside oil-fired plants and power imports reliant on imported fuels [7]. That infrastructure relied heavily on liquefied natural gas, leaving the grid exposed when global fuel markets tightened after 2021 [8]. By 2023–2024, acute fossil fuel supply shortages and declining foreign exchange reserves forced power plants to operate at just over 50% of their total installed capacity. This forced the government to institute scheduled load-shedding to manage peak summer demand [9]. The demand pattern that was stable through 2021 shifted abruptly. Load shedding, which affected fewer than 1% of hours before 2022, occurred in 44% of hours between 2022 and 2025, and the shortfall was largest in the hottest hours, when demand peaks. Similar

supply-constrained operation, in which recorded load understates the electricity consumers would have used, has been reported for Iraq, Pakistan and Nigeria, where it complicates both short-term forecasting and capacity planning [10-13].

Deep learning (DL) methods have improved STLF accuracy considerably over the years. Huy et al. [14] showed that temporal fusion transformers outperform LSTM and ARIMA on Hanoi's city grid. Wen et al. [15] reduced MAPE by 39% on the GEFCom2014 international benchmark with a GRU-TCN-Attention hybrid. On the other hand, Özüpak and Mansurov [16] found hybrid CNN-GRU architectures more robust than standalone LSTM under volatile demand on the UK National Grid. Among transformer variants, feature-space attention has drawn particular interest. The iTransformer architecture operates across features rather than timesteps, avoiding the temporal overfitting that standard attention exhibits on long sequences [17]. Fang et al. [18] augmented this with an MLP feature-capture module and report improvement over the vanilla iTransformer on actual power datasets, while Huang and Yi [19] hybridized it with temporal convolutional networks on Belgium's Elia grid with similar results. PatchTST takes a different route: it splits each input series into short patches and applies attention across them, which preserves local temporal structure while keeping long look-back windows tractable [20]. At the same time, gradient-boosted trees remain difficult to beat on tabular inputs [21, 22], and a recent study on Iraq's supply-constrained grid found linear regression and XGBoost competitive with deep networks [11]. Claims of deep learning superiority therefore need tuned tree baselines and formal significance tests. The deep learning gains described above are reported exclusively on stable grids with no supply-side disruptions in scope. Most STLF benchmarks come from established regional and national grid datasets such as ISO-NE, PJM, CER (Ireland), and European grid data [2, 23]. This reflects generalized global grid operations rather than the severe supply

deficits and operational volatility characteristic of developing power systems [1]. Prior Bangladesh PGCB forecasting work sits in the same stable-data regime. Rafi et al. [24] applied CNN-LSTM on 2014-2019 data with rolling weekly evaluation. Baul et al. [25] extended this to nine PGCB zones using CNN-GRU. Islam et al. [26], the most thorough prior study, reach MAPE=2.3% with FB-Prophet across 78,906 training records but test on 45 days in January-February 2024, before load-shedding peaked. Shahiduzzaman et al. [27] tested on July 2025 data and report RMSE=709 MW, which is one of the few prior results on PGCB demand during the crisis period. No prior study compares deep learning models with tuned tree ensembles separately for hours with and without load shedding, tests whether the differences are statistically significant, or explains which inputs drive each model in each regime.

This study addresses that gap on the PGCB national grid. Using hourly records from 2015 to 2026, published as an open dataset [28], eleven forecasters are compared under a single leak-free protocol: three naive baselines, two linear models, two tuned tree ensembles and four deep learning architectures (PatchTST, iTransformer, CNN-BiLSTM and BiGRU), all scored on the same 14,142 test hours from July 2024 to March 2026. Three questions are investigated: whether deep learning models keep their accuracy in load-shedding hours, where tree ensembles degrade; whether the accuracy gaps remain statistically significant once the autocorrelation of hourly errors is taken into account; and which inputs and look-back hours each model relies on, as revealed by Integrated Gradients and TreeSHAP [29-31]. The contributions include a causal, leak-free benchmark on crisis-period data with evaluation stratified by load-shedding regime. A comparison of deep learning models and tree ensembles in hours with and without load shedding, supported by Diebold-Mariano tests and block-bootstrap confidence intervals is also included along with a

BatchNorm ablation across the four deep learning architectures and an explainability analysis of the inputs and look-back hours that each model uses in each regime.

## 2. Data and the load-shedding regime

### 2.1 Data source and target definition

This dataset was taken from Bangladesh's Power Grid Company (PGCB) public portal which contains hourly electricity demand from 2015-2026. The records were collected with a custom scraper and are published as an open dataset, where the collection procedure and every raw variable are documented in full [28]. The raw file contains 94,934 records from 17 April 2015 to 8 March 2026. Each record reports national demand, total generation, reported load shedding, generation by fuel (gas, liquid fuel, coal, hydro, solar and wind) and cross-border imports (Bheramara HVDC, Tripura, Adani and Nepal).

The forecast target is the hourly national demand reported by PGCB. In hours without load shedding, demand equals total generation within 10 MW in 99.9% of observed hours. In shedding hours, PGCB adds an estimate of the unserved load, and demand minus generation is about 1.05 times the reported load shedding (median ratio). Demand at hour t can therefore be closely reconstructed from generation and load shedding at the same hour.

Hourly temperature and relative humidity were sourced from the Open-Meteo API [32] and spatially averaged across Bangladesh’s three primary load centers (Dhaka, Chittagong, and Sylhet). From the temperature and relative humidity values, heat index was calculated using Celsius-adapted National Weather Service polynomial regression equation to capture the cooling demand. The complex equation was intentionally set to only trigger when temperatures exceeded

27°C [33, 34]. Below this temperature, heat index was assumed to be equal to the ambient temperature.

### 2.2 Cleaning and quality control

Prior to feature addition, the raw dataset was analyzed to identify physically impossible values and null values. For instance, some demand values exceeded 150,000 MW and some generation values exceeded 64 million MW. The raw file also contains 4,439 half-hourly peak readings (mostly at 19:30). These were removed so that every record refers to a full clock hour. The remaining 90,495 hourly records were then placed on a regular hourly grid of 95,477 hours. On this grid, 4,982 hours are absent from the source, 3,976 of them at midnight, because the portal rarely reports 00:00.

Firstly, a valid operational range was defined for each of the numeric values based on historical grid capacities: 500–20,000 MW for total demand; 1,000–20,000 MW for generation; 0–7,000 MW for load shedding; 0–9,000 MW for gas; and 0–7,500 MW for coal (**Table S1**). The values that were outside this range were set to missing. Isolated spikes in demand were then removed with a Hampel filter [35]. A value was flagged when it differed from the centred 7-hour median by more than six scaled median absolute deviations and by more than 15%; 94 values were removed this way (**Table S1**).

Two reporting changes required special treatment. First, solar, wind, Adani and Nepal values appear only from their first report (20 September 2017, 1 June 2023, 28 August 2024 and 14 November 2024). Before these dates the sources either did not exist or were not reported separately, so the values were set to zero instead of being filled backwards. Second, Adani imports were reported inside the coal figure until 28 August 2024. In the three weeks around the change, reported coal fell from 4,028 to 3,459 MW while a separate Adani entry of 1,114 MW appeared

(**Fig. S1b**). Coal and Adani were therefore combined into a single coal series for the whole period. Bheramara, Tripura and Nepal were summed into total imports, and solar and wind into one renewable series.

Gaps of six hours or less were filled by time-linear interpolation, and longer gaps were left missing; no value was carried forward or backward across gaps. After cleaning, 666 demand hours remained missing. Every hour whose target was missing or interpolated was excluded from model training, tuning and evaluation, so all reported errors are measured against original PGCB readings. **Fig. S1a** shows, for each year, how many hours were missing in the source and how many were removed during cleaning.

### 2.3 The load-shedding regime

Demand grew steadily over the study period, from a mean of 5.9 GW in 2015 to 11.9 GW in 2025, and the daily profile kept its evening peak at 20:00 (**Figure 1 a, b**). Supply changed more abruptly. Gas fell from 71% of supply in 2015 to 43% in 2025, while coal rose from 2% to 38% after new coal plants and Adani imports came online (**Figure 1 d**). The demand response to temperature also became steeper in recent years (**Figure 1 c**).

Load shedding was rare before 2022, affecting 0.0 to 0.4% of hours each year, but it became routine from 2022 onwards (**Table 1**). It occurred in 26.0% of hours in 2022, 57.0% in 2023, 57.2% in 2024 and 37.3% in 2025. The energy shed peaked at 2,580 GWh in 2023 (**Figure 2 a, b**). A chi-square test confirms that the shift is not due to chance: the share of hours with shedding rose from 0.16% in 2015 to 2021 to 44.4% in 2022 to 2025 ($\chi^2 = 29{,}137$, $p < 0.001$), and Cramér's V of 0.57 indicates a strong association. Shedding concentrated in the evening hours and the hot

months (**Figure 2 e**). During the crisis years, monthly peak demand exceeded peak served load by up to 723 MW (2024; **Table 1; Figure 2 c**), whereas in 2019 the two curves coincide throughout the day (**Figure 2 d**).

To test whether supply limits bound hardest in hot hours, demand, served load and shortfall were each regressed on temperature, a shedding indicator and their interaction. The regressions used hourly data for 2022 to 2025 (33,513 observed hours), with hour, month and year fixed effects and Newey-West standard errors with 24 lags [36]. The results are reported in **Table S2**. In shedding hours, demand rose by 270.9 MW per °C, but served load rose by only 224.0 MW per °C, statistically no different from its slope in normal hours (change −3.8 MW per °C, $p = 0.54$). The difference appears as shortfall, which grew by 44.2 MW for each additional °C in shedding hours ($p < 0.001$; **Figure 2 f**). In other words, once shedding begins, extra heat-driven demand is largely left unmet. This regime of rationed, temperature-sensitive supply is the setting in which the forecasting models of **Section 3** are evaluated.

Table 1 Load shedding on the national grid by year (observed hours only). 2015 starts on 17 April and 2026 ends on 8 March. Peak served is the highest hourly generation. One reading at 03:00 on 17 October 2017, 1,423 MW above demand, was excluded as a reporting error. Before 2022, shedding hours are few and some entries do not match demand minus generation, so the mean and maximum shed for those years should be read with caution.

| Year | Hours | Hours with shedding | Share (%) | Mean shed (MW) | Max shed (MW) | Energy shed (GWh) | Peak demand (MW) | Peak served (MW) |
|---|---|---|---|---|---|---|---|---|
| 2015 | 5,809 | 20 | 0.3 | 282.9 | 767 | 5.7 | 8,177 | 8,177 |
| 2016 | 8,381 | 35 | 0.4 | 208.8 | 387 | 7.3 | 9,036 | 9,036 |
| 2017 | 7,742 | 4 | 0.1 | 50.5 | 90 | 0.2 | 9,495 | 9,495 |
| 2018 | 8,314 | 10 | 0.1 | 150.7 | 540 | 1.5 | 11,571 | 11,571 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2019 | 8,308 | 12 | 0.1 | 212.3 | 900 | 2.5 | 12,893 | 12,893 |
| 2020 | 8,397 | 4 | 0.0 | 1,629.0 | 4,626 | 6.5 | 12,892 | 12,892 |
| 2021 | 8,385 | 3 | 0.0 | 201.0 | 270 | 0.6 | 13,792 | 13,792 |
| 2022 | 8,375 | 2,180 | 26.0 | 809.1 | 6,905 | 1,763.9 | 14,782 | 14,782 |
| 2023 | 8,354 | 4,759 | 57.0 | 542.1 | 3,419 | 2,579.7 | 16,221 | 15,648 |
| 2024 | 8,394 | 4,799 | 57.2 | 432.1 | 3,196 | 2,073.7 | 17,200 | 16,477 |
| 2025 | 8,390 | 3,132 | 37.3 | 115.2 | 2,223 | 360.8 | 17,100 | 16,794 |
| 2026 | 1,540 | 341 | 22.1 | 200.3 | 1,108 | 68.3 | 13,910 | 13,910 |

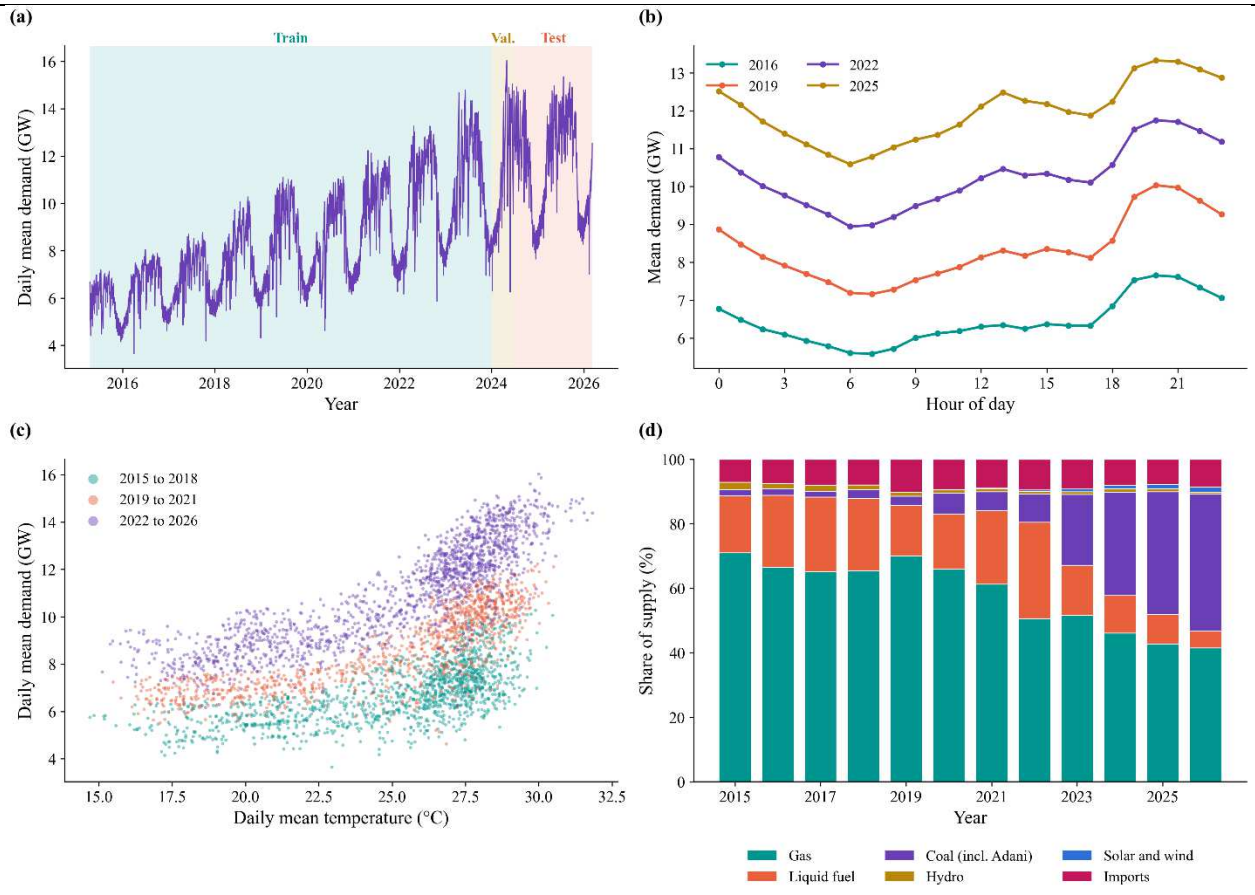



Figure 1 Overview of the PGCB demand series, April 2015 to March 2026. (a) Daily mean demand, with the training, validation and test periods shaded. (b) Average daily demand profile in 2016, 2019, 2022 and 2025. (c) Daily mean demand against daily mean temperature for 2015 to 2018, 2019 to 2021 and 2022 to 2026. (d) Annual share of supply by source.

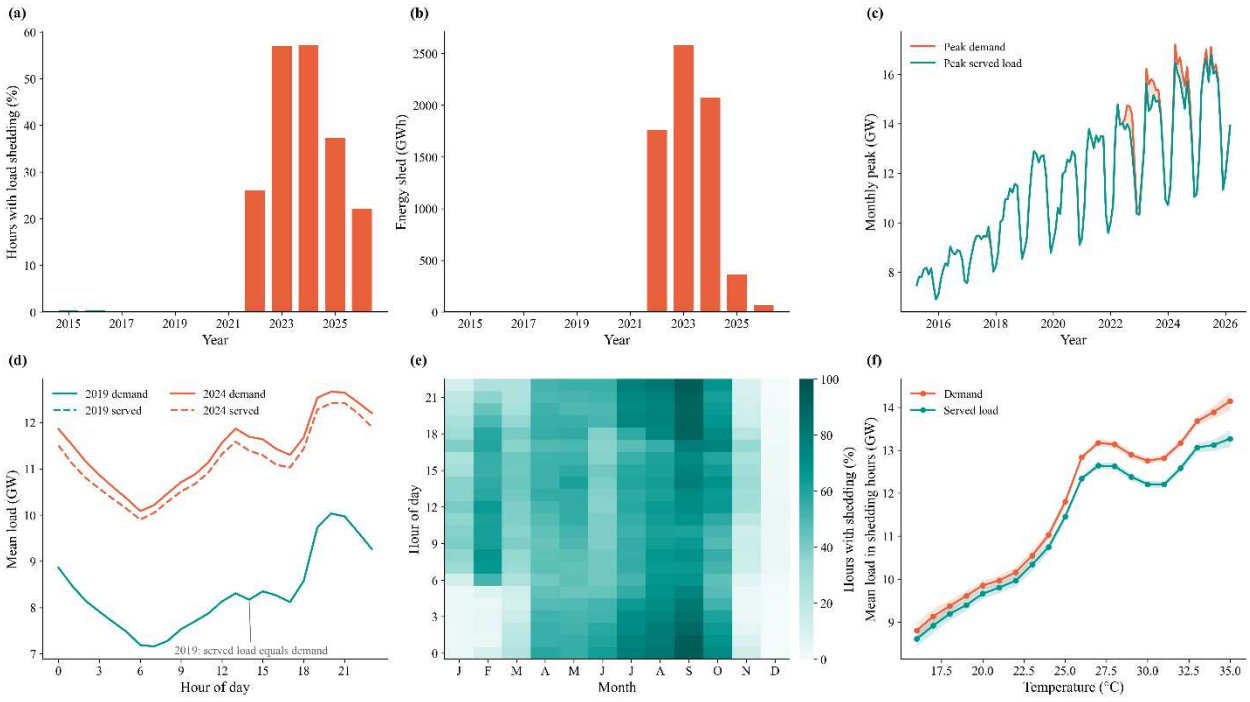



Figure 2 The load-shedding regime of 2022 to 2025. (a) Share of hours with load shedding by year. (b) Energy shed by year. (c) Monthly peak demand and peak served load; the shaded gap is unmet peak demand. (d) Average daily profiles of demand and served load in 2019 and 2024. (e) Probability of load shedding by hour of day and month, 2022 to 2025. (f) Mean demand and served load against temperature in shedding hours, 2022 to 2025.

# 3. Methodology

## 3.1 Forecasting setup and features

Each model forecasts PGCB demand one hour ahead. The forecast for hour t uses only information available at the end of hour t−1, together with calendar variables and observed weather at hour t. Using observed weather treats the weather forecast as perfect, a standard assumption in short-term load forecasting that is noted as a limitation in **Section 5.3**. The data were split chronologically. The training set runs from April 2015 to December 2023, the validation set from January to June 2024 (used only for tuning and early stopping), and the test set from 1 July 2024 to 8 March 2026. After the exclusions in Section 2.2, the three sets contain 71,311, 4,171 and 14,142 hours. Every

model is scored on the same 14,142 test hours. Thirty features in six groups were built (**Table 2**), each known at forecast time.

To capture the previous hour, the same period the previous day, and the same hour one week prior to the forecast point, six lag features were generated at 1, 2, 3, 24, 48, and 168 hours respectively. Standard temporal variables (hour, day of week, month) were transformed into continuous cyclical variables using sine and cosine encodings to prevent boundary discontinuities in deep learning encoders. Three rolling statistics (24-hour mean, 168-hour mean, and 24-hour standard deviation) were added to capture short-term load trajectory and volatility. All three windows end at hour t−1, so the current value never enters its own predictor. Weekend (Friday and Saturday), public holiday and Ramadan flags were added from the official Bangladesh calendar. Supply features describe the generation mix one hour earlier: gas, liquid fuel, coal (including Adani), hydro, solar and wind, and total imports. Load shedding enters only through past values: the shortfall at t−1, a flag for shedding at t−1, and the share of shedding hours over the previous 168 hours. Generation and load shedding at hour t were never used as inputs, because together they reproduce the target (**Section 2.1**). Features were standardized with means and standard deviations computed on usable training hours only.

Table 2 The 30 input features. All are known at forecast time; "t−1" denotes the hour before the forecast hour.

| Group | n | Features |
|---|---|---|
| Demand lags | 6 | Demand at t−1, t−2, t−3, t−24, t−48 and t−168 |
| Rolling demand | 3 | Mean and standard deviation over the previous 24 h; mean over the previous 168 h |

| | | |
|---|---|---|
| Calendar | 9 | Hour, day of week and month (sine and cosine); weekend, holiday and Ramadan flags |
| Weather | 3 | Temperature, relative humidity and heat index at t (three-city average) |
| Supply mix | 6 | Gas, liquid fuel, coal (incl. Adani), hydro, solar + wind and total imports at t−1 |
| Load shedding | 3 | Shortfall at t−1; shedding flag at t−1; share of shedding hours over the previous 168 h |

### 3.2 Baseline models

Three naive forecasts serve as reference points, which are the demand one hour earlier (persistence), 24 hours earlier and 168 hours earlier. Two regularized linear models were used: ElasticNet [37], and Bayesian Ridge [38]. ElasticNet combines both L1 and L2 regularization and Optuna was used to tune the penalty strength and the mixing parameter. Bayesian Ridge estimates the regularization strength from the data rather than searching for it. Both models were trained on the 30 features without further feature selection. Each row was treated as an independent observation, with the sequence structure embedded through lag features.

Two ensemble tree models were used as the second tier. XGBoost [39] is a gradient-boosted decision tree model that builds trees sequentially, with each tree correcting the residuals of the previous one. Random Forest [40] builds independent trees on random feature subsets and averages their outputs. Both tree models received the same 30-feature tabular input as the linear models.

The hyperparameters of ElasticNet, Random Forest and XGBoost were tuned with Optuna [41] using the tree-structured Parzen estimator [42], minimising validation RMSE under a fixed time

budget (5 min for ElasticNet, 40 min each for the tree models). XGBoost used early stopping on the validation set (100 rounds, up to 3,000 trees) and was refitted on the training set with the selected number of trees. The search spaces are listed in **Table S3**.

### 3.3 Deep learning models

Four DL architectures were trained. These contain gated RNNs, convolutional-recurrent hybrids, and transformer-based models. Each model was trained on input sequences of shape (168,30) using a 168-hour lookback window covering feature rows t−167 to t. Every row's features are known at forecast time (**Section 3.1**), so the window adds history but no information about demand at hour t.

A bidirectional Gated Recurrent Unit (BiGRU) [43] and a hybrid CNN-BiLSTM [44, 45] were used as sequential baselines. BiGRU processes the input forward and backward through two GRU layers (hidden size 64). This allows each timestep representation to draw on both earlier and later hours within the look-back window, all of which are known at forecast time. CNN-BiLSTM instead applies two 1D convolutional layers (64 channels, kernel 3) first to extract local short-range temporal patterns before passing the resulting feature sequence to a bidirectional LSTM (hidden size 64) for longer-range dependencies.

The inverted Transformer (iTransformer) [17] is different from conventional time-series transformers because it computes self-attention [46] across the feature dimension rather than the time dimension. Each of the 30 features is treated as an independent token, projected from the 168-timestep window into a d_model = 32 embedding via a linear layer, then passed through two Transformer encoder layers (4 heads, feedforward dimension 64). The flattened representation (30 × 32 = 960) is mapped to the forecast by a single linear layer. PatchTST [20] segments each univariate series into non-overlapping temporal patches, embeds each patch as a token, and applies

channel-independent self-attention across patches. Attention here operates over time segments rather than features. Patches of 24 hours were used, so each 168-hour window becomes seven daily tokens per feature. The encoder matches the iTransformer's (d_model = 32, two layers, four heads, feedforward dimension 64), followed by a flatten head.

All four models were trained with one fixed protocol and no architecture-specific tuning: the AdamW optimizer [47] (learning rate $10^{-3}$, weight decay $10^{-4}$), HuberLoss [48] on the standardised target, batch size 256, dropout of 0.1 [49], gradient clipping at norm 1.0, a learning rate halved after three epochs without validation improvement, and early stopping after eight such epochs (at most 50 epochs, 20 min per run). Each configuration was trained with several random seeds (five for iTransformer, three for the others). Results are reported as mean ± standard deviation across seeds. For the explainability and statistical analyses, each model is represented by the seed with the median validation loss. Training used an NVIDIA T4 GPU. Each architecture was trained in two variants. The default uses Layer Normalization inside the attention blocks. The BatchNorm variant replaces it with Batch Normalization [50] and adds Batch Normalization before the output layer (and after each convolution in CNN-BiLSTM). The variant with the lower mean validation loss across seeds is reported as each architecture's main result. The same variant also had the lower mean test RMSE for all four architectures. The two variants are compared statistically in **Section 4.4**.

### 3.4 Evaluation framework

Five evaluating metrics were used consistently across all the models. Standard metrics such as Root mean square error (RMSE) and mean absolute error (MAE) were used to quantify errors in megawatt unit, making them physically interpretable [51]. To express the errors in percentage, Mean absolute percentage error (MAPE) was used, which also allows better comparison with the

previous grid studies [52]. Alongside these metrics, R-square was used as it can have several advantages over traditional metrics and can provide more information [53]. To quantify the generalizability of the models, the overfitting ratio was used which is defined in **(1 (1).**

$$Overfitting\ ratio = \frac{RMSE_{test}}{RMSE_{train}} \quad (1)$$

A ratio above 1 indicates that a model performs worse on unseen data than on its training data. A ratio close to 1 indicates that the model generalizes well. This metric is particularly useful because the test period (July 2024 to March 2026) differs strongly from the training period (2015 to 2023). Errors were also computed separately for test hours with and without load shedding at the forecast hour (5,851 and 8,291 hours) and for each test year.

### 3.5 Explainability

For XGBoost, exact SHAP [30] values were computed with TreeSHAP [31] through the booster's native pred_contribs, for all 14,142 test hours. For the deep learning models, Integrated Gradients attributed each prediction to the 168 × 30 input window. A baseline equal to the training mean and 32 integration steps were used, with the models in evaluation mode. The completeness property of Integrated Gradients (attributions sum to the difference between the prediction and the baseline prediction) was checked for every explained hour. To verify that the attributions are faithful, they were compared with feature occlusion [54]. Each feature's 168-hour history was replaced by its training mean, and the resulting rise in RMSE was recorded. The deep learning analysis used a stratified sample of 2,000 test hours, 1,000 with and 1,000 without load shedding. Attributions were summed over the window to give feature and feature-group importance, and over features to

show how far back in the window each model looks. Shares of total attribution were compared across models and between the two regimes.

### 3.6 Statistical Validation

Three tests assessed whether the performance differences between models are statistically significant. All models were aligned on identical timestamps before testing. The Diebold-Mariano test [55] compared squared-error loss (h = 1) for every pair of models. Because hourly errors are autocorrelated, its variance was estimated with the Newey-West estimator [36] using 24 lags, with 168 lags as a robustness check. The Wilcoxon signed-rank test [56] compared paired daily mean absolute errors (616 test days), which reduces the dependence between consecutive hours. A moving block bootstrap [57] (168-hour blocks, 2,000 resamples) produced 95% confidence intervals for the RMSE difference between each model and the best model, preserving within-week autocorrelation structure. The same resampled hours were used for every model, so the intervals are paired. The Diebold-Mariano comparison was repeated within shedding hours, normal hours and each test year, and between the two normalization variants of each architecture. All p-values were adjusted for multiple comparisons with the Holm procedure [58]. The summary of the whole methodology is illustrated in **Figure 3**.

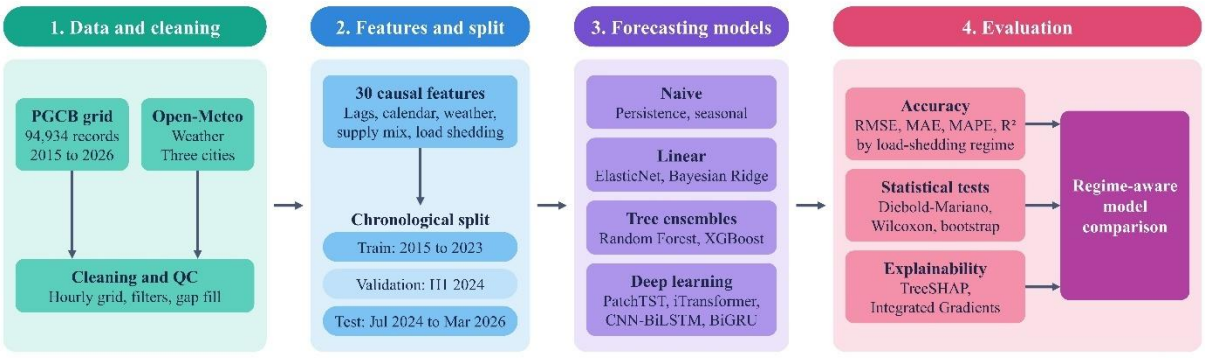



Figure 3 Summary of the methodological pipeline.

## 4. Results and Discussion

### 4.1 Overall accuracy

The performance of all models is summarized in **Table 3** and **Figure 4 a**. All four deep learning models outperformed every baseline. PatchTST achieved the lowest error, with an RMSE of 203.8 ± 3.5 MW, a MAPE of 1.26% and an $R^2$ of 0.991. It was followed by iTransformer (217.1 MW), CNN-BiLSTM (223.6 MW) and BiGRU (231.5 MW). The best tree model, XGBoost, reached 308.1 MW. The deep learning models therefore reduced RMSE by 25 to 34% compared with a tuned gradient-boosted ensemble. Compared with persistence (483.9 MW), the reduction was 52 to 58%. The ranking was stable across random seeds. The standard deviation of RMSE was at most 5.6 MW for the reported variants. Every individual seed of every deep learning model also outperformed XGBoost. The advantage of deep learning is therefore not the result of a lucky initialization.

Among the baselines, XGBoost and Random Forest performed similarly (308.1 and 318.6 MW). Both were clearly better than the linear models (368.3 MW). ElasticNet and Bayesian Ridge produced almost identical forecasts. The tuned ElasticNet penalty was very weak, so both models effectively reduced to ordinary least squares on the same features. The Overfitting Ratio separated the model families more sharply than accuracy alone. The deep learning models kept the ratio between 1.17 and 1.35. Their test error was therefore close to their training error. XGBoost (1.82) and Random Forest (2.68) fitted the training years much more closely than they generalised to the test period. The linear models had the lowest ratio (1.11), but their error was much higher. They generalized well because they are simple, not because they are accurate.

Errors grew over the test period for every non-linear model (**Section 4.3**). Between 2024 and 2026, the RMSE of PatchTST rose from 189.7 to 235.5 MW, an increase of 24%. XGBoost rose from

267.7 to 339.3 MW (27%), and Random Forest from 273.7 to 399.7 MW (46%). The linear models stayed flat at about 364 MW. The 2026 values cover only January to early March and should be read with caution. Even so, the tree models lost accuracy fastest as demand moved further from the training range. **Section 4.5** shows that this pattern is consistent with their inability to predict beyond the demand levels seen during training.

Table 3 Test-period accuracy of all models (1 July 2024 to 8 March 2026, 14,142 hours). Deep learning models show the mean ± standard deviation over seeds for the better normalization variant (BN = BatchNorm, LN = LayerNorm). The shedding and normal columns give RMSE in hours with and without load shedding.

| Model | Type | RMSE (MW) | MAE (MW) | MAPE (%) | $R^2$ | RMSE shedding (MW) | RMSE normal (MW) | Overfitting Ratio |
|---|---|---|---|---|---|---|---|---|
| **PatchTST (BN)** | Deep learning | **203.8 ± 3.5** | **142.1 ± 3.0** | **1.26 ± 0.03** | **0.991** | **196.9** | **208.6** | **1.17** |
| iTransformer (BN) | Deep learning | 217.1 ± 5.6 | 154.9 ± 5.0 | 1.36 ± 0.04 | 0.990 | 215.7 | 217.9 | 1.19 |
| CNN-BiLSTM (LN) | Deep learning | 223.6 ± 2.2 | 162.1 ± 3.6 | 1.44 ± 0.02 | 0.990 | 216.9 | 228.2 | 1.35 |
| BiGRU (LN) | Deep learning | 231.5 ± 4.3 | 167.3 ± 4.9 | 1.46 ± 0.05 | 0.989 | 238.3 | 226.5 | 1.27 |
| XGBoost | Tree | 308.1 | 222.9 | 1.89 | 0.981 | 360.9 | 264.7 | 1.82 |
| Random Forest | Tree | 318.6 | 227.2 | 2.00 | 0.979 | 326.5 | 312.9 | 2.68 |
| ElasticNet | Linear | 368.3 | 267.8 | 2.36 | 0.972 | 371.1 | 366.4 | 1.11 |
| Bayesian Ridge | Linear | 368.3 | 267.8 | 2.36 | 0.972 | 371.1 | 366.4 | 1.11 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Persistence (t−1) | Naive | 483.9 | 365.4 | 3.28 | 0.952 | 489.4 | 480.1 | 1.09 |
| Seasonal naive (t−24) | Naive | 903.6 | 629.1 | 5.56 | 0.833 | 846.4 | 941.9 | 1.16 |
| Seasonal naive (t−168) | Naive | 1342.5 | 950.6 | 8.07 | 0.631 | 1428.7 | 1278.2 | 1.19 |

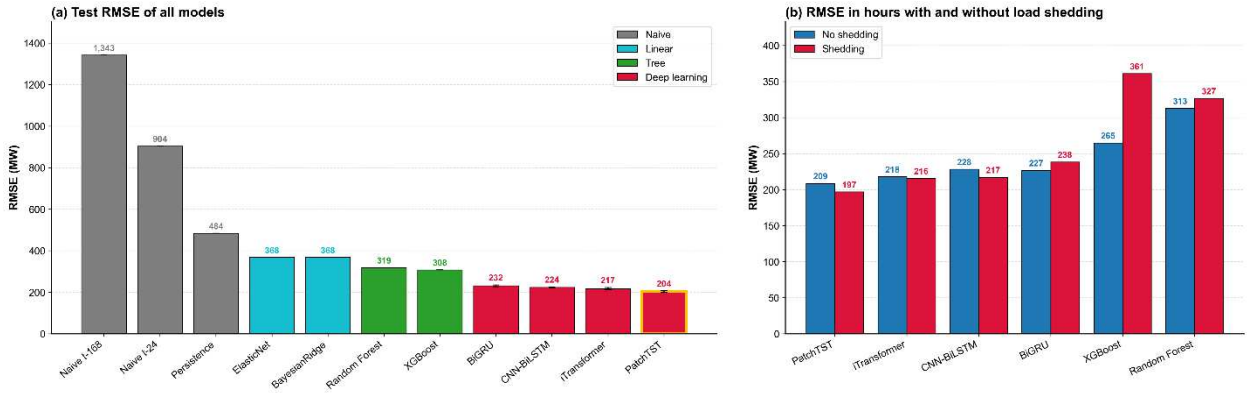



Figure 4 Test accuracy of all models. (a) RMSE over the full test period (1 July 2024 to 8 March 2026). Bars are coloured by model type. Error bars show the standard deviation over seeds for the deep learning models. The best model is outlined in gold. (b) RMSE in hours with and without load shedding at the forecast hour for the four deep learning models and the two tree ensembles.

### 4.2 Performance during load shedding

Load shedding occurred in 5,851 of the 14,142 test hours. **Figure 4 b** and **Table 3** compare the error of each model in hours with and without shedding. The deep learning models kept the same accuracy in both regimes. PatchTST was even slightly more accurate in shedding hours (196.9 MW) than in normal hours (208.6 MW). iTransformer and CNN-BiLSTM showed the same pattern. BiGRU was the only deep learning model with a higher error in shedding hours, and the increase was small (238.3 against 226.5 MW, or 5%).

XGBoost behaved very differently. Its RMSE rose from 264.7 MW in normal hours to 360.9 MW in shedding hours, an increase of 36%. In normal hours, XGBoost was within 60 MW of the deep

learning models. In shedding hours, the gap grew to between 120 and 165 MW. Most of the advantage of deep learning over XGBoost therefore comes from the load-shedding regime. Random Forest and the linear models changed little between regimes (4% and 1%). However, their errors were already high in both regimes.

The two example weeks in **Figure 5** show this behavior in the forecasts themselves. In the last week of August 2024, load was shed in every hour. PatchTST followed the evening peaks closely, while XGBoost often stopped short of them. The weekly RMSE was 178 MW for PatchTST and 255 MW for XGBoost. In a winter week without shedding in January 2026, both models followed the daily cycle, but XGBoost again undershot several peaks (166 against 264 MW).

The same pattern appears over the whole test period (**Figure S2**). The 7-day rolling RMSE of XGBoost rose sharply in the summer and autumn months, when shedding was most frequent, and reached about 550 MW. The rolling RMSE of PatchTST stayed mostly between 150 and 300 MW. The cumulative difference in squared error between XGBoost and PatchTST grew in every month of the test period. It grew fastest during the shedding seasons of July to September 2024 and April to September 2025. The advantage of PatchTST is therefore consistent over time and largest when the grid is most constrained. Section 4.5 examines why the tree models struggle in these hours.

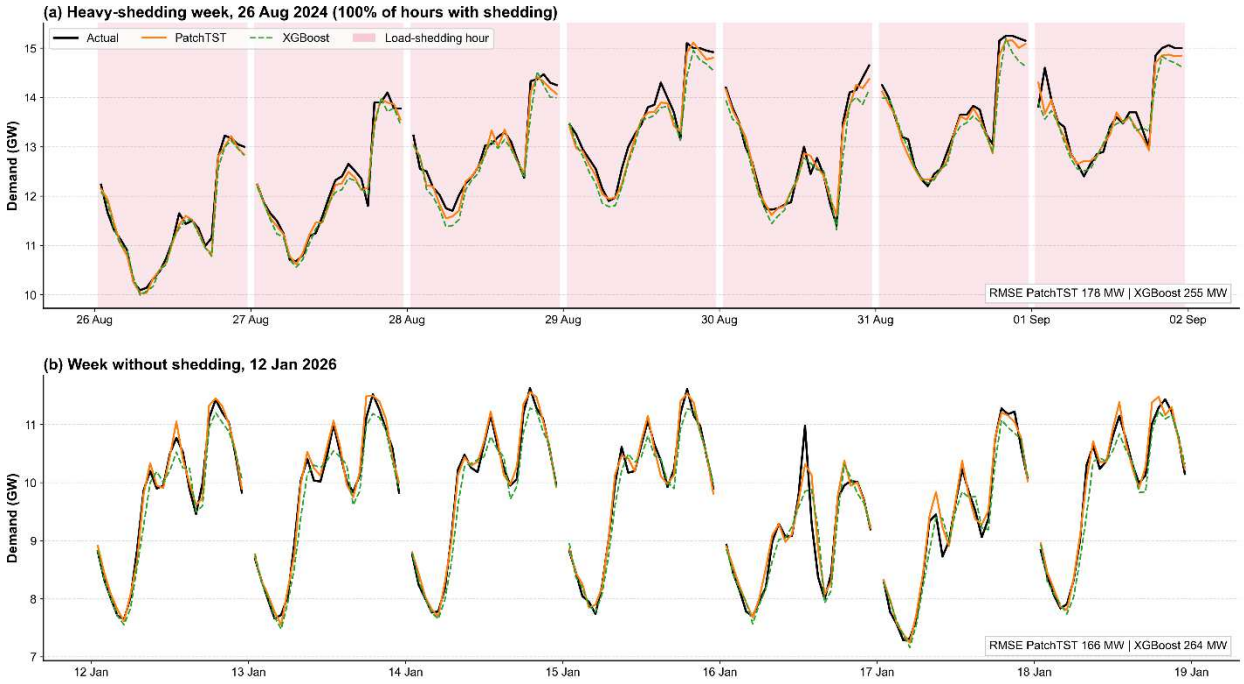



Figure 5 Actual and forecast demand in two test weeks. (a) A heavy-shedding week starting 26 August 2024, in which load was shed in every hour (shaded). (b) A week without load shedding starting 12 January 2026. Black lines show PGCB demand, orange lines PatchTST and dashed green lines XGBoost. The weekly RMSE of each model is given in each panel. Gaps are hours without an original PGCB reading.

### 4.3 Statistical significance

**Table 4** and **Figure 6** test whether the differences in **Table 3** are statistically significant. Each model was compared with PatchTST on the same 14,142 test hours. These tests use the representative seed of each deep learning model, so the RMSE values differ slightly from the seed means in **Table 3**.

PatchTST was significantly more accurate than every other single model. The Diebold-Mariano statistic was negative in every comparison, which means that PatchTST had the lower squared error. All p-values were below 0.001 after Holm correction. The paired block bootstrap gave the same result. The 95% confidence interval of the RMSE difference excluded zero for every model. The smallest gap was against iTransformer, at 12.4 MW (95% CI 8.6 to 15.7 MW). The gap against XGBoost was 105.2 MW (95% CI 88.7 to 123.7 MW). The daily comparison supports the hourly

tests. PatchTST had a lower daily mean absolute error than iTransformer on 68% of the 616 test days, and lower than XGBoost on 92% of days. The Wilcoxon test on these daily errors was significant for every model ($p < 0.001$). The conclusions also did not depend on the test settings. They held when the Newey-West lag was increased from 24 to 168 hours, and when absolute errors replaced squared errors.

The pairwise tests in **Figure 6 a** show a clear order among the remaining models. iTransformer was significantly better than CNN-BiLSTM, and CNN-BiLSTM was significantly better than BiGRU. All four deep learning models were significantly better than both tree models. XGBoost was only marginally better than Random Forest ($p = 0.03$). The advantage of PatchTST over XGBoost was significant in every subgroup. This held in shedding hours, in normal hours and in each test year (**Figure 6 c**). The advantage over iTransformer was significant overall and in 2025 and 2026, but not in 2024 ($p = 0.21$). The two transformers were therefore close in the first months of the test period, and PatchTST pulled ahead later.

An equal-weight average of PatchTST and iTransformer was also tested. It was not significantly different from PatchTST alone (RMSE difference 0.2 MW, $p = 0.78$). Combining the two transformers gave no further gain, so PatchTST alone is the recommended model. **Figure 6 d** shows the mean absolute error by hour of day. For most models, the largest errors occurred during the evening ramp between 18:00 and 20:00, when demand rises fastest. The two transformers also showed a second peak at 14:00. PatchTST and iTransformer stayed below XGBoost at almost every hour of the day.

Table 4 Comparison of each model with PatchTST on the 14,142 test hours, using the representative seed of each deep learning model. ΔRMSE is the model's RMSE minus PatchTST's, with a 95% paired block-bootstrap interval. DM is the Diebold-Mariano statistic on squared errors (Newey-West variance, 24

lags). Negative values favour PatchTST. The Wilcoxon test compares daily mean absolute errors. All p-values are Holm-adjusted.

| Model | RMSE (MW) | ΔRMSE [95% CI] (MW) | DM | p (DM) | p (Wilcoxon) | Days PatchTST better (%) |
|---|---|---|---|---|---|---|
| PatchTST (reference) | 203.0 | | | | | |
| Transformer ensemble | 203.2 | +0.2 [−1.5, +1.7] | −0.28 | 0.78 | 0.54 | 51.3 |
| iTransformer | 215.4 | +12.4 [+8.6, +15.7] | −8.42 | < 0.001 | < 0.001 | 67.9 |
| CNN-BiLSTM | 222.1 | +19.1 [+15.0, +23.1] | −10.06 | < 0.001 | < 0.001 | 71.4 |
| BiGRU | 228.2 | +25.2 [+19.4, +31.4] | −11.71 | < 0.001 | < 0.001 | 72.1 |
| XGBoost | 308.1 | +105.2 [+88.7, +123.7] | −16.49 | < 0.001 | < 0.001 | 92.2 |
| Random Forest | 318.6 | +115.6 [+105.1, +125.2] | −29.43 | < 0.001 | < 0.001 | 96.6 |

| | | | | | | |
|---|---|---|---|---|---|---|
| ElasticNet | 368.3 | +165.3 [+153.4, +177.4] | −31.71 | < 0.001 | < 0.001 | 98.2 |
| Persistence (t−1) | 483.9 | +280.9 [+269.9, +292.5] | −74.39 | < 0.001 | < 0.001 | 99.8 |
| Seasonal naive (t−24) | 903.6 | +700.6 [+623.1, +782.9] | −13.99 | < 0.001 | < 0.001 | 93.7 |

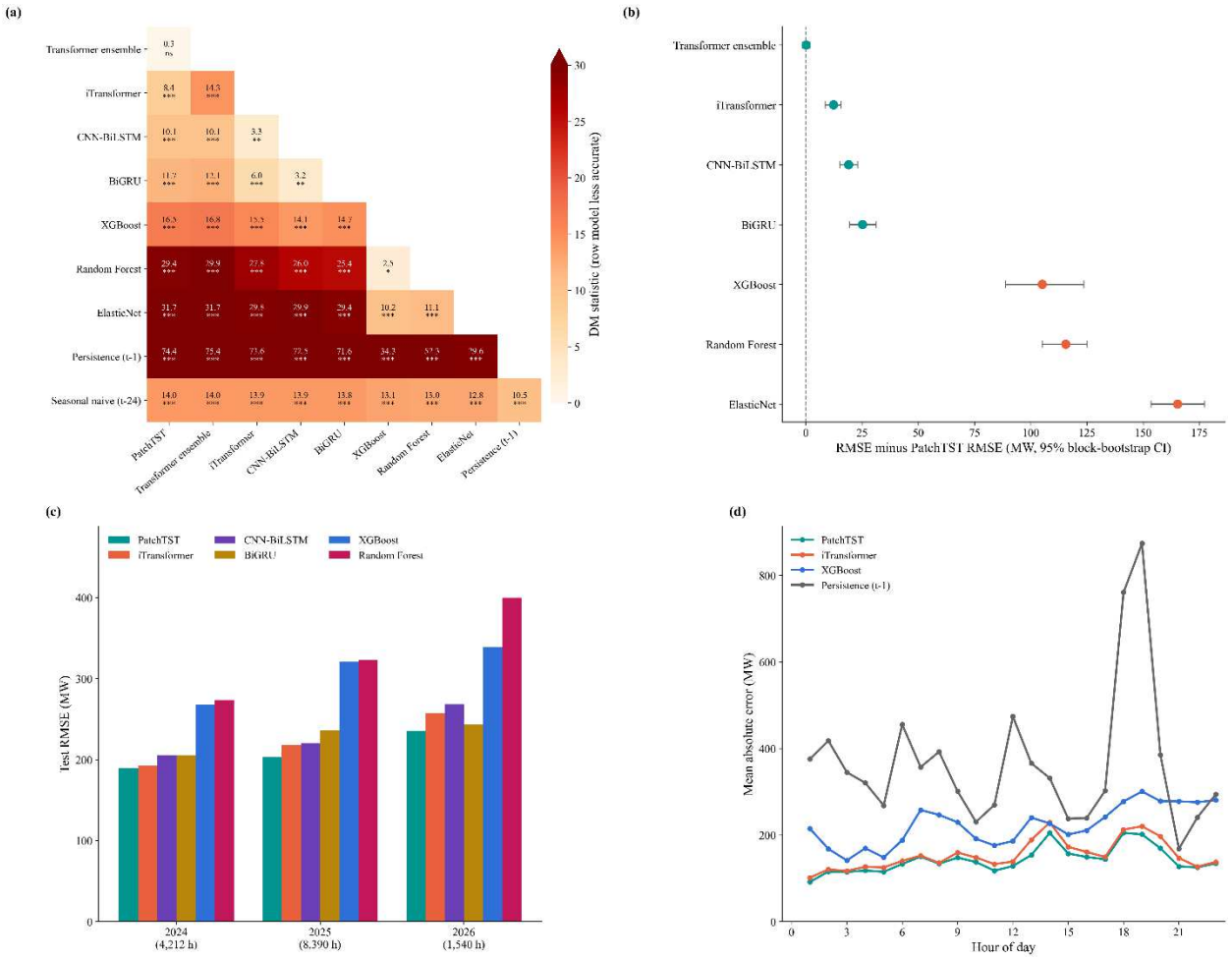



Figure 6 Statistical comparison of the models on the test period. (a) Pairwise Diebold-Mariano statistics on squared errors (row model minus column model). Positive values mean that the row model was less accurate. Stars give Holm-adjusted significance (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$, ns not significant). (b) RMSE difference between each model and PatchTST, with 95% paired block-bootstrap intervals (168-hour blocks, 2,000 resamples). (c) RMSE by test year. Hours of data per year are shown below each year. The 2026 data end on 8 March. (d) Mean absolute error by hour of day for PatchTST, iTransformer, XGBoost and persistence.

### 4.4 Normalization ablation

Each deep learning architecture was trained with LayerNorm and with BatchNorm. **Figure 7** compares the two variants, and **Table S4** gives the Diebold-Mariano tests. The effect of BatchNorm depended clearly on the architecture. BatchNorm helped the two transformers. The largest gain was for iTransformer. Its RMSE fell from 255.3 to 217.1 MW, a reduction of 38 MW ($p < 0.001$). The spread across seeds also fell from 13.8 to 5.6 MW, and the Overfitting Ratio fell

from 1.34 to 1.19. BatchNorm therefore made iTransformer both more accurate and more stable. PatchTST gained less, but the gain was still significant. Its RMSE fell from 207.8 to 203.8 MW ($p = 0.004$), and its Overfitting Ratio stayed at about 1.17. However, BatchNorm did not help the recurrent models. For CNN-BiLSTM, it raised the RMSE from 223.6 to 232.0 MW ($p < 0.001$). It also increased the spread across seeds from 2.2 to 11.4 MW. For BiGRU, the two variants were not significantly different (231.5 against 233.3 MW, $p = 0.99$).

These results agree with the design of the models. In the transformers, BatchNorm replaces LayerNorm inside every attention block. BatchNorm scales each embedding dimension using statistics from the whole batch. This makes the encoder less sensitive to individual extreme windows, which are common in the load-shedding period. The original PatchTST design also uses BatchNorm for this reason [20]. In the recurrent models, BatchNorm acts only on the final hidden state before the output layer. Its batch statistics then add noise to an already stable representation without improving it. This explains the higher variance of CNN-BiLSTM. The practical conclusion is that normalization should be chosen for each architecture rather than applied by default. BatchNorm is a good choice for transformer encoders on this type of data. It is not needed for recurrent models.

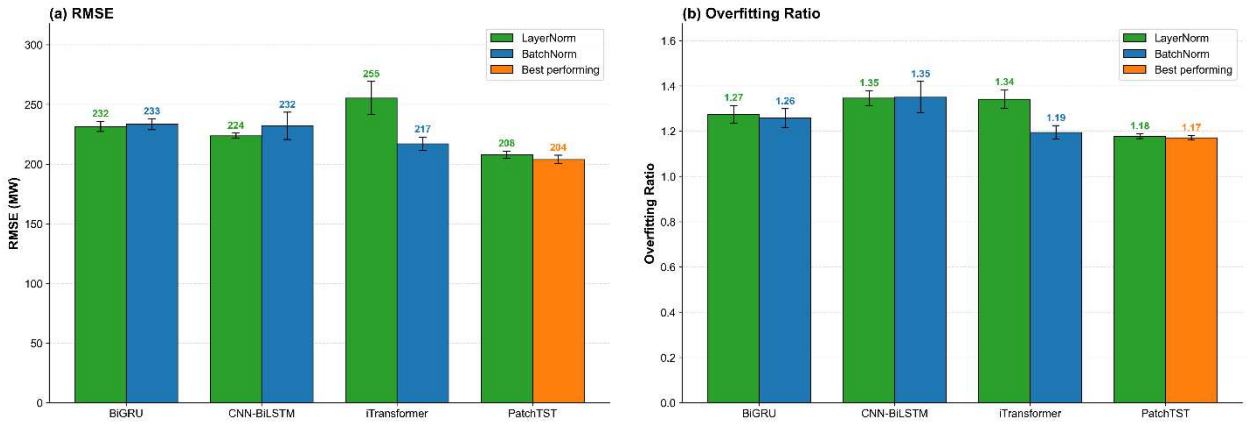



Figure 7 Normalization ablation for the four deep learning models. (a) Test RMSE and (b) Overfitting Ratio with LayerNorm and BatchNorm. Bars show the mean over seeds (five for iTransformer, three for the others), and error bars show the standard deviation. The best-performing configuration (PatchTST with BatchNorm) is shown in orange.

### 4.5 Model interpretability and feature attribution

**Figure 8** compares how the models use their inputs. Integrated Gradients was used for the deep learning models and TreeSHAP for XGBoost. Both methods passed their consistency checks. For PatchTST, iTransformer and BiGRU, the Integrated Gradients attributions added up to the model output with a median error below 0.1 MW. For CNN-BiLSTM, the error was 12.2 MW, or 0.4% of the explained output. The attributions also agreed with feature occlusion, a separate test that removes each feature in turn. The rank correlation between the two methods ranged from 0.56 for iTransformer to 0.84 for CNN-BiLSTM (**Figure S3b**). The explanations can therefore be trusted to reflect what the models actually use.

All models relied most on recent demand. Demand at t−1 was the most important feature for every model. The models differed in how much else they used (**Figure 8 a**). XGBoost placed 82% of its attribution on the demand lags. CNN-BiLSTM and BiGRU placed about 70% there. The two transformers spread their attention more widely. PatchTST placed only 41% on the demand lags. It placed 21% on rolling demand, 20% on calendar variables and 12% on the supply mix. The

transformers also used the load-shedding history more. It received 3.4 to 3.7% of their attribution, against 0.4% for XGBoost. Coal supply at t−1 was the fourth most important feature for PatchTST (**Figure 8 b**).

The clearest difference between the models was how far back they looked (**Figure 8 c**). CNN-BiLSTM and BiGRU placed more than 95% of their attribution on the last five hours of the window. The transformers used the whole week. PatchTST placed only 30% of its attribution on the last five hours and 50% on the last two days. Its cumulative attribution rose in steps at 24, 48, 72 and 96 hours. This shows that PatchTST compares the forecast hour with the same hour on previous days. Its daily patches are designed to capture exactly this structure. The ranking of the models in **Section 4.1** follows the same order. The models that used the full weekly cycle were the most accurate.

In load-shedding hours, every model shifted attribution towards the demand lags and away from calendar and supply variables (**Figure S3c**). The share of the load-shedding history also rose in shedding hours for all four deep learning models. In other words, the deep learning models paid more attention to recent shortfalls exactly when shortfalls mattered most. The SHAP plots in **Figure 9** explain why XGBoost struggled in shedding hours. Its contribution from demand at t−1 rose steadily up to about 15 GW and then flattened (**Figure 9 c**). Between 15 and 16 GW, the average contribution was about 5,250 MW. Above 16 GW, it was about 5,300 MW, almost the same. A tree model cannot predict values beyond the range it has seen during training. Only 130 training hours had demand at t−1 above 15 GW. In the test period, there were 727 such hours, and 97% of them were load-shedding hours. XGBoost therefore underestimated the highest-demand hours, which were mostly shedding hours. The deep learning models do not have this hard limit, because their output is a smooth function of the inputs. Temperature had a small but physically

sensible effect (**Figure 9 d**). Its contribution was negative below about 26 °C and near zero between 26 and 28 °C. Above 28 °C, it rose steadily and reached about 80 MW above 30 °C. This threshold matches the start of cooling demand. Weather received only about 2% of the attribution for every model. This is expected for a one-hour-ahead forecast, because demand at t−1 already reflects the current temperature.

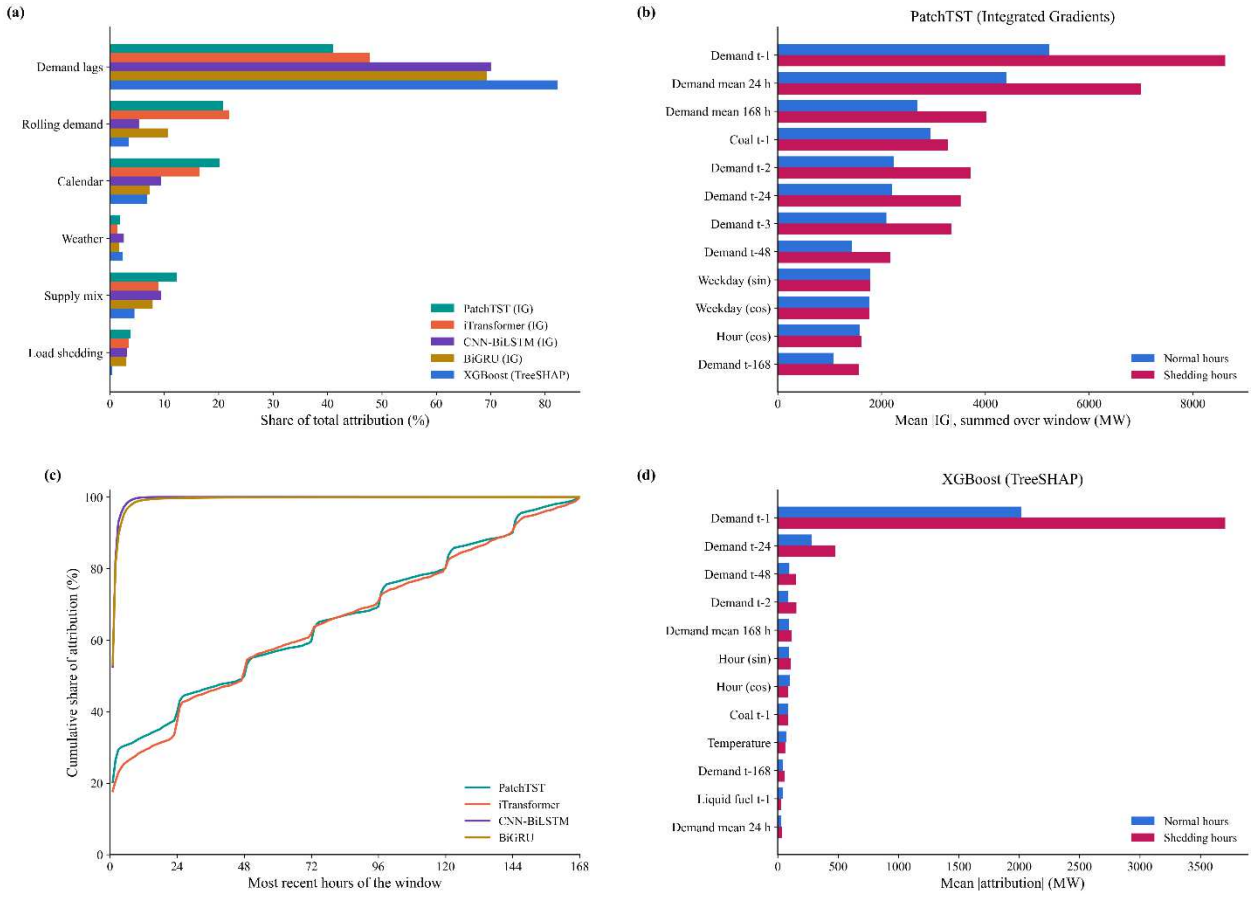



Figure 8 Attribution of the forecasts to the input features. (a) Share of total attribution by feature group for the four deep learning models (Integrated Gradients) and XGBoost (TreeSHAP). (b) Mean absolute Integrated Gradients attribution of the 12 most important features of PatchTST, in hours with and without load shedding. (c) Cumulative share of attribution over the 168-hour input window, starting from the most recent hour. (d) Mean absolute SHAP value of the 12 most important features of XGBoost, in hours with and without load shedding. Deep learning attributions are computed on 2,000 test hours (1,000 with and 1,000 without load shedding). XGBoost attributions use all 14,142 test hours.

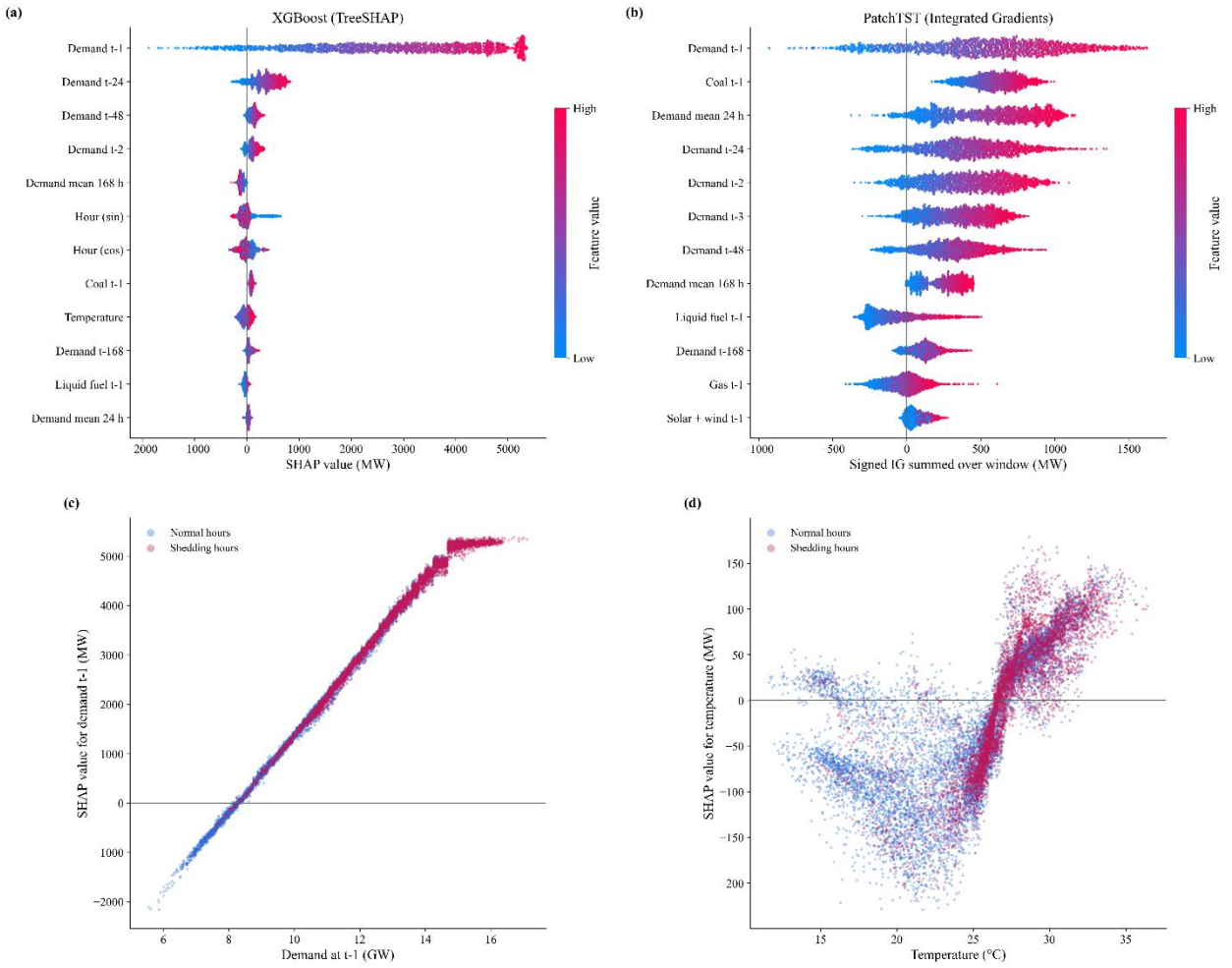



Figure 9 SHAP-style summary and dependence plots. (a) TreeSHAP values of XGBoost and (b) signed Integrated Gradients attributions of PatchTST, summed over the input window, for the 12 most important features. Each point is one test hour, coloured by the value of the feature at the forecast hour (blue low, red high). (c) SHAP value of demand at t−1 against demand at t−1 and (d) SHAP value of temperature against temperature for XGBoost, coloured by load-shedding regime.

### 4.6 Comparison with previous studies

The global studies reveal a consistent pattern: even on stable, well-resourced grids, the margin between transformer-based models and recurrent baselines is narrow under normal operating conditions. Huy et al. [14] demonstrate TFT's advantage over LSTM and ARIMA on Hanoi's city grid, while Wen et al. [15] achieve a 39% MAPE improvement on the GEFCom2014 international

benchmark using a GRU-TCN-Attention hybrid. Hannan et al. [59] push single-step MAPE below 1% on the AEP and ISONE US datasets. However, both are stable benchmark datasets with no supply-side disruptions in scope. **Table 5** places the present results alongside these studies and the previous work on the Bangladesh grid.

Table 5 Comparison with previous load forecasting studies. Horizons and test periods differ between studies, so the metrics are indicative only.

| Ref | Year Span/Dataset | Country / Grid | Test Period | Key Models | Key Metric |
|---|---|---|---|---|---|
| [14] | Not stated | Vietnam, Hanoi city | - | TFT, LSTM, ARIMA | MAE: 77.51 MW, MAPE: 3.38% for proposed LR+TFT model |
| [16] | Jan 2009-early 2024 | UK, National Grid ESO | Holdout within dataset (279,264 half-hourly records) | CNN-LSTM-DNN, CNN-GRU-DNN, LSTM | MAE: 0.03, MSE: 0.0018 for standalone LSTM; CNN-GRU-DNN most robust under volatile demand |
| [15] | GEFCom2014 | International benchmark | - | GRU-TCN-Attention | MAPE improved 39% vs baseline; FLOP reduced 48.8% |

| | | | | | |
|---|---|---|---|---|---|
| [59] | AEP + ISONE benchmark | USA (Ohio + New England ISO) | - | CNN-LSTM-Attention | AEP: RMSE=132.79, MAPE=0.68%; ISONE: RMSE=137.72, MAPE=0.51% |
| [18] | From January 1, 2017, to December 30, 2017 | China, actual power datasets | - | FMLP-iTransformer vs vanilla iTransformer, LSTM, CNN | 24 h RMSE: 1005.8, MAPE: 2.49%. |
| [19] | 1 January 2020 to 31 December 2023 | Belgium, Elia grid | - | iTransformer-TCN-MLP, Prophet+VMD preprocessing | TFE-iTransformer-MPTCN model MAPE: 1.862%, MAE: 196.59 KW, RMSE: 256.36 KW. |
| [60] | 2014 to 2019 | Bangladesh (Mymensingh) | 7 days | CNN-LSTM | MAPE: 2.8992% |
| [25] | January 2014 to April 2023 | Bangladesh, PGCB (9 zones) | 7 days | CNN-GRU | Lowest MAPE vs baselines |
| [4] | June 2, 2021, to June 30, 2023 | Bangladesh, PGCB | 24 hours | Bi-LSTM | MAE = 376.72 MW |

| | | | | | |
|---|---|---|---|---|---|
| [61] | September 2023 to February 2024 | Bangladesh, Western region | 6 months | ANN, ANFIS | MAPE = 1.72% |
| [26] | 2015-2023 | Bangladesh, PGCB national | 45 days (Jan-Feb 2024) | FB-Prophet, LSTM, RF | MAPE = 2.3% |
| [27] | Jul 2021-Jun 2025 | Bangladesh, PGCB | Rolling, Jul 2025 | GRU-TCN-Attention, BiLSTM, CNN-LSTM | RMSE = 709 MW, MAPE = 4.43% |
| **This study** | **Apr 2015 to Mar 2026** | **Bangladesh, PGCB national** | **14,142 h (Jul 2024 to Mar 2026), one hour ahead, 41% of hours with load shedding** | **PatchTST, iTransformer, CNN-BiLSTM, BiGRU, XGBoost, RF** | **RMSE = 203.8 MW, MAPE = 1.26% (PatchTST)** |

The studies differ in forecast horizon, test period and grid size, so the numbers in **Table 5** are not directly comparable. However, a few contrasts are still informative. Islam et al. [26] reported a MAPE of 2.3% with FB-Prophet, but they tested on 45 days in January and February 2024, before load shedding peaked. Shahiduzzaman et al. [27] tested on July 2025, a month inside the crisis period, and reported an RMSE of 709 MW and a MAPE of 4.43%. The present study evaluated 14,142 hours from July 2024 to March 2026, and 41% of these hours had load shedding. On this

harder test set, PatchTST reached an RMSE of 203.8 MW and a MAPE of 1.26%. This approaches the accuracy reported for stable benchmark grids such as AEP and ISONE (MAPE 0.51 to 0.68%) [59], even though 41% of the PGCB test hours had load shedding. Two features of the present evaluation also differ from most previous work. First, all models were scored on the same test hours, and the differences were tested for statistical significance. Most earlier studies reported point metrics only. Second, errors were reported separately for hours with and without load shedding. **Section 4.2** showed that this split changes the conclusion. XGBoost was close to the deep learning models in normal hours, but clearly worse in shedding hours. A single overall metric would hide this difference.

# 5. Conclusion

## 5.1 Summary of findings

This study compared eleven forecasting models on hourly demand from the Bangladesh national grid between 2015 and 2026. The period includes a severe supply crisis, in which load was shed in 44% of hours from 2022 to 2025. All models were trained under one leak-free protocol and scored on the same 14,142 test hours. The four deep learning models beat every naive, linear and tree-based baseline. PatchTST was the most accurate, with an RMSE of 203.8 MW and a MAPE of 1.26%. Its error was 34% lower than that of a tuned XGBoost and 58% lower than persistence. The gap was statistically significant against every other model, in hours with and without load shedding and in every test year. Most of the advantage appeared during load shedding. In those hours, the error of XGBoost rose by 36%, while the deep learning models held their accuracy. The explainability analysis points to two causes. The transformers drew on the full weekly cycle and on the recent shedding history. XGBoost, in contrast, could not predict demand above the levels it had seen in training, and nearly all of the highest-demand test hours fell in shedding periods. The

normalization ablation added a practical point. BatchNorm improved the transformers but not the recurrent models, so the choice of normalization should depend on the architecture.

### 5.2 Practical implications

For PGCB, the results suggest that forecasting models should be trained on data that include crisis years. The training set here included 2022 and 2023, and the deep learning models kept their accuracy when load shedding continued in the test period. Forecasts should also be checked separately for hours with and without load shedding. A model can look good on average and still fail when supply is short. The pipeline uses only public data and 30 standard features, which keeps deployment simple. Other grids that manage shortages through load shedding, such as those of Pakistan and Iraq, could apply the same protocol once local data are collected.

### 5.3 Limitations

These results come with clear limits. The forecasts are one hour ahead only. Grid operators also plan a day ahead, and the ranking of models may change at that horizon. Observed weather was treated as a perfect forecast, so real weather errors would lower accuracy somewhat. The target is PGCB's own estimate of demand. During load shedding, the unmet part is estimated by PGCB and cannot be checked, so the target may differ from true demand when shedding is heavy. The test set has no midnight hours, because PGCB rarely reports them, and the 2026 data cover just over two months. The study uses a single national grid and gives point forecasts without uncertainty. Finally, all models were evaluated offline on historical data. Real-time use, including the cost of regular retraining, has not been tested.

### 5.4 Future work

The most useful next step is day-ahead forecasting, to see whether the advantage of deep learning holds at the horizon used for unit commitment. Probabilistic forecasts with calibrated intervals

would give operators a basis for planning during crisis hours, when demand swings are largest. Automatic detection of structural breaks could also guide when models should be retrained. This study treats the grid as a single node. Graph neural networks have improved forecasts by modelling links between grid nodes [62], and extending the analysis to PGCB substations is a natural follow-up. Repeating the same protocol on other grids with frequent load shedding would show how far these findings generalize.

***Acknowledgments***. N/A

***Declaration of Competing Interest***. The authors declare no competing interests.

***Declaration of generative AI and AI-assisted technologies in the manuscript preparation process***. During the preparation of this work the authors used Claude (Anthropic) to assist with code development and language editing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

***Ethical approval.*** Not applicable. This article does not contain any studies with human participants or animals performed by any of the authors.

***Data availability.*** The PGCB dataset used in this study is openly available on Mendeley Data (doi: 10.17632/vpk8spw2mm.1).

***Code availability.*** All code, trained model weights and per-hour predictions are available at https://github.com/shekhibrahim433/pgbcforecast.

***CRediT authorship contribution statement.*** Md Ibrahim Shekh: Conceptualization, Methodology, Software, Formal analysis, Data curation, Visualization, Writing (original draft). Rakibul Hasan Rafi: Data curation, Validation, Writing (review and editing).

***Funding.*** This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## 6. References

[1] S. S. Arnob, A. I. M. S. Arefin, A. Y. Saber, and K. A. Mamun, "Energy Demand Forecasting and Optimizing Electric Systems for Developing Countries," *IEEE Access,* vol. 11, pp. 39751-39775, 2023, doi: 10.1109/ACCESS.2023.3250110.

[2] T. Bashir, C. Haoyong, M. F. Tahir, and Z. Liqiang, "Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN," *Energy Reports,* vol. 8, pp. 1678-1686, 2022/11/01/ 2022, doi: 10.1016/j.egyr.2021.12.067.

[3] M. R. Islam, A. A. Mamun, M. Sohel, M. L. Hossain, and M. M. Uddin, "LSTM-Based Electrical Load Forecasting for Chattogram City of Bangladesh," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 12-14 March 2020 2020, pp. 188-192, doi: 10.1109/ESCI48226.2020.9167536.

[4] M. S. U. Miah, M. I. Islam, S. Islam, A. Ahmed, M. M. Rahman, and M. Mahmud, "Sustainability-Driven Hourly Energy Demand Forecasting in Bangladesh Using Bi-LSTMs," *Procedia Computer Science,* vol. 236, pp. 41-50, 2024/01/01/ 2024, doi: 10.1016/j.procs.2024.05.002.

[5] E. A. H. Alkaldy, M. A. Albaqir, and M. S. Akhavan Hejazi, "A new load forecasting model considering planned load shedding effect," *International Journal of Energy Sector Management,* vol. 13, no. 1, pp. 149-165, 2019, doi: 10.1108/ijesm-03-2018-0008.

[6] R. Usman, P. Mirzania, S. W. Alnaser, P. Hart, and C. Long, "Systematic Review of Demand-Side Management Strategies in Power Systems of Developed and Developing Countries," *Energies,* vol. 15, no. 21, p. 7858, 2022, doi: 10.3390/en15217858.

[7] K. B. Debnath and M. Mourshed, "Why is Bangladesh's electricity generation heading towards a GHG emissions-intensive future?," *Carbon Management,* vol. 13, no. 1, pp. 216-237, 2022/01/02 2022, doi: 10.1080/17583004.2022.2068454.

[8] S. A. Supti, "Renewable Energy and Policy in Bangladesh," in *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 26-29 Oct. 2022 2022, pp. 0251-0256, doi: 10.1109/UEMCON54665.2022.9965627.

[9] S. Mohazzem Hossain, S. Biswas, and M. Raihan Uddin, "Sustainable energy transition in Bangladesh: Challenges and pathways for the future," *Engineering Reports,* vol. 6, no. 1, p. e12752, 2024, doi: 10.1002/eng2.12752.

[10] N. A. Mohammed, "Modelling of unsuppressed electrical demand forecasting in Iraq for long term," *Energy,* vol. 162, pp. 354-363, 2018/11/01/ 2018, doi: 10.1016/j.energy.2018.08.030.

[11] M. Aldarraji, B. Vega-Márquez, B. Pontes, B. Mahmood, and J. C. Riquelme, "Addressing energy challenges in Iraq: Forecasting power supply and demand using artificial intelligence models," *Heliyon,* vol. 10, no. 4, 2024, doi: 10.1016/j.heliyon.2024.e25821.

[12] U. Safder, T. N. Hai, J. Loy-Benitez, and C. Yoo, "Nationwide policymaking strategies to prevent future electricity crises in developing countries using data-driven forecasting and

fuzzy-SWOT analyses," *Energy,* vol. 259, p. 124962, 2022/11/15/ 2022, doi: 10.1016/j.energy.2022.124962.
[13] O. Oluwole, A. J. Collin, A. H. van der Weijde, A. E. Kiprakis, and G. P. Harrison, "A bottom-up weather-sensitive residential demand model for developing countries. A case study of Abuja, Nigeria," *Energy for Sustainable Development,* vol. 58, pp. 138-149, 2020/10/01/ 2020, doi: 10.1016/j.esd.2020.08.009.
[14] P. C. Huy, N. Q. Minh, N. D. Tien, and T. T. Q. Anh, "Short-Term Electricity Load Forecasting Based on Temporal Fusion Transformer Model," *IEEE Access,* vol. 10, pp. 106296-106304, 2022, doi: 10.1109/ACCESS.2022.3211941.
[15] X. Wen, J. Liao, Q. Niu, N. Shen, and Y. Bao, "Deep learning-driven hybrid model for short-term load forecasting and smart grid information management," *Scientific Reports,* vol. 14, no. 1, p. 13720, 2024/06/14 2024, doi: 10.1038/s41598-024-63262-x.
[16] Y. Özüpak and S. Mansurov, "A Hybrid Deep Learning Approach for Multi-Output Short-Term Electricity Demand Forecasting," *Concurrency and Computation: Practice and Experience,* vol. 37, no. 25-26, p. e70356, 2025, doi: 10.1002/cpe.70356.
[17] Y. Liu *et al.*, "itransformer: Inverted transformers are effective for time series forecasting," in *International conference on learning representations*, 2024, vol. 2024, pp. 11116-11140.
[18] B. Fang, L. Xu, Y. Luo, Z. Luo, and W. Li, "A method for short-term electric load forecasting based on the FMLP-iTransformer model," *Energy Reports,* vol. 12, pp. 3405-3411, 2024/12/01/ 2024, doi: 10.1016/j.egyr.2024.09.023.
[19] Z. Huang and Y. Yi, "Short-Term Load Forecasting for Regional Smart Energy Systems Based on Two-Stage Feature Extraction and Hybrid Inverted Transformer," *Sustainability,* vol. 16, no. 17, p. 7613, 2024, doi: 10.3390/su16177613.
[20] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730,* 2022, doi: 10.48550/arXiv.2211.14730.
[21] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," *Advances in neural information processing systems,* vol. 35, pp. 507-520, 2022, doi: 10.52202/068431-0037.
[22] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion,* vol. 81, pp. 84-90, 2022/05/01/ 2022, doi: 10.1016/j.inffus.2021.11.011.
[23] J. M. Aguiar-Pérez and M. Á. Pérez-Juárez, "An Insight of Deep Learning Based Demand Forecasting in Smart Grids," *Sensors,* vol. 23, no. 3, p. 1467, 2023, doi: 10.3390/s23031467.
[24] S. H. Rafi, M. Nahid Al, S. R. Deeba, and E. Hossain, "A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network," *IEEE Access,* vol. 9, pp. 32436-32448, 2021, doi: 10.1109/ACCESS.2021.3060654.
[25] A. Baul, G. C. Sarker, P. Sikder, U. Mozumder, and A. Abdelgawad, "Data-Driven Short-Term Load Forecasting for Multiple Locations: An Integrated Approach," *Big Data and Cognitive Computing,* vol. 8, no. 2, p. 12, 2024, doi: 10.3390/bdcc8020012.
[26] M. T. Islam, S. A. Turja, and A. Habib, "Enhanced power demand forecasting for Bangladesh: using feature engineering associated with environmental and economic impact," *Journal of Data, Information and Management,* vol. 7, no. 1, pp. 1-19, 2025/03/01 2025, doi: 10.1007/s42488-025-00140-9.

[27] M. Shahiduzzaman, M. G. Jami, S. Azad, M. A. Shahin, and I. Khan, "Enhancing Forecast Accuracy through Adaptive Deep Learning: A Comparative Analysis of Hybrid Neural Models," in *2026 5th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 29-31 Jan. 2026 2026, pp. 1-6, doi: 10.1109/ICECTE69292.2026.11429452.

[28] M. I. Shekh and R. H. Rafi. *Hourly Electricity Generation, Demand, Load Shedding, and Fuel Mix Dataset for Bangladesh (2015–2026)*, Mendeley Dataset, doi: 10.17632/vpk8spw2mm.1.

[29] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," presented at the Proceedings of the 34th International Conference on Machine Learning - Volume 70, Sydney, NSW, Australia, 2017.

[30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems,* vol. 30, 2017.

[31] S. M. Lundberg *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nature Machine Intelligence,* vol. 2, no. 1, pp. 56-67, 2020/01/01 2020, doi: 10.1038/s42256-019-0138-9.

[32] *Open-Meteo.com Weather API*. (2023). Zenodo.

[33] R. G. Steadman, "The Assessment of Sultriness. Part I: A Temperature-Humidity Index Based on Human Physiology and Clothing Science," (in English), *Journal of Applied Meteorology and Climatology,* vol. 18, no. 7, pp. 861-873, 01 Jul. 1979 1979, doi: 10.1175/1520-0450(1979)018<0861:TAOSPI>2.0.CO;2.

[34] L. P. Rothfusz and N. S. R. Headquarters, "The heat index equation (or, more than you ever wanted to know about heat index)," *Fort Worth, Texas: National Oceanic and Atmospheric Administration, National Weather Service, Office of Meteorology,* vol. 9023, p. 640, 1990.

[35] F. R. Hampel, "The Influence Curve and its Role in Robust Estimation," *Journal of the American Statistical Association,* vol. 69, no. 346, pp. 383-393, 1974/06/01 1974, doi: 10.1080/01621459.1974.10482962.

[36] W. K. Newey and K. D. West, "A simple, positive semi-definite, heteroskedasticity and autocorrelationconsistent covariance matrix," 1986.

[37] H. Zou and T. Hastie, "Regularization and Variable Selection Via the Elastic Net," *Journal of the Royal Statistical Society Series B: Statistical Methodology,* vol. 67, no. 2, pp. 301-320, 2005, doi: 10.1111/j.1467-9868.2005.00503.x.

[38] D. J. C. MacKay, "Bayesian Interpolation," *Neural Computation,* vol. 4, no. 3, pp. 415-447, 1992, doi: 10.1162/neco.1992.4.3.415.

[39] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016. [Online]. Available: https://doi.org/10.1145/2939672.2939785.

[40] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001/10/01 2001, doi: 10.1023/A:1010933404324.

[41] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," presented at the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 2019.

[42] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems,* vol. 24, 2011.
[43] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," Doha, Qatar, October 2014: Association for Computational Linguistics, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734, doi: 10.3115/v1/D14-1179.
[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.
[45] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[46] A. Vaswani *et al.*, "Attention is all you need," presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017.
[47] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101,* 2017, doi: 10.48550/arXiv.1711.05101.
[48] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics,* vol. 35, no. 1, pp. 73-101, 3/1 1964, doi: 10.1214/aoms/1177703732.
[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.,* vol. 15, no. 1, pp. 1929–1958, 2014.
[50] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," presented at the Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, 2015. [Online]. Available: https://proceedings.mlr.press/v37/ioffe15.html.
[51] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature," *Geosci. Model Dev.,* vol. 7, no. 3, pp. 1247-1250, 2014, doi: 10.5194/gmd-7-1247-2014.
[52] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting,* vol. 9, no. 4, pp. 527-529, 1993/12/01/ 1993, doi: 10.1016/0169-2070(93)90079-3.
[53] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *Peerj computer science,* vol. 7, p. e623, 2021, doi: 10.7717/peerj-cs.623.
[54] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," Cham, 2014: Springer International Publishing, in Computer Vision – ECCV 2014, pp. 818-833, doi: 10.1007/978-3-319-10590-1_53.
[55] F. X. Diebold and R. S. Mariano, "Comparing Predictive Accuracy," *Journal of Business & Economic Statistics,* vol. 13, no. 3, pp. 253-263, 1995/07/01 1995, doi: 10.1080/07350015.1995.10524599.
[56] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin,* vol. 1, no. 6, pp. 80-83, 1945, doi: 10.2307/3001968.
[57] H. R. Kunsch, "The Jackknife and the Bootstrap for General Stationary Observations," *The Annals of Statistics,* vol. 17, no. 3, pp. 1217-1241, 25, 1989, doi: 10.1214/aos/1176347265.

[58] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics,* pp. 65-70, 1979.

[59] A. Hannan, Y. Hussain, K. Ullah, M. S. Khan, H. Saleem, and B. Azeem, "Intelligent Short-Term Load Forecasting for Renewable Power Grids Using Attention-Based Hybrid Deep Learning," in *2026 International Conference on Integrated Intelligence and Cognitive Engineering (ICIICE)*, 18-19 April 2026 2026, pp. 1-6, doi: 10.1109/ICIICE69672.2026.11565331.

[60] S. M. A. Haque, G. C. Sarker, and K. M. Sadat, "Short-Term Electrical Load Prediction for Future Generation Using Hybrid Deep Learning Model," in *2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*, 24-26 Feb. 2022 2022, pp. 1-6, doi: 10.1109/ICAEEE54957.2022.9836359.

[61] F. Hussain, M. Hasanuzzaman, and N. A. Rahim, "Multivariate machine learning algorithms for energy demand forecasting and load behavior analysis," *Energy Conversion and Management: X,* vol. 26, p. 100903, 2025/04/01/ 2025, doi: 10.1016/j.ecmx.2025.100903.

[62] S. Park, K. Kwon, and Y. Kim, "Load Forecasting in Power Grids Using Spatio-Temporal Graph Neural Networks," in *2025 IEEE 11th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*, 21-24 July 2025 2025, pp. 170-171, doi: 10.1109/BigDataService65758.2025.00031.

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- 20261018PowerSupply.docx