

Please cite the Published Version

Shafiuzzaman, Md , Safayet Islam, Md, Rubaith Bashir, T.M, Munem, Mohammad, Nahiduzzaman, Md, Ahsan, Mominul and Haider, Julfikar  (2025) Enhanced very short-term load forecasting with multi-lag feature engineering and prophet-XGBoost-CatBoost architecture. Energy, 335. 137981 ISSN 0360-5442

DOI: <https://doi.org/10.1016/j.energy.2025.137981>

Publisher: Elsevier

Version: Published Version

Downloaded from: <https://e-space.mmu.ac.uk/641540/>

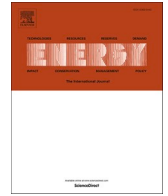
Usage rights:  [Creative Commons: Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)

Additional Information: This is an open access article published in Energy, by Elsevier.

Data Access Statement: Data will be made available on request.

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)



Enhanced very short-term load forecasting with multi-lag feature engineering and prophet-XGBoost-CatBoost architecture

Md Shafiuzzaman^a, Md Safayet Islam^a, T.M. Rubaith Bashar^b, Mohammad Munem^a,
Md Nahiduzzaman^a, Mominul Ahsan^c, Julfikar Haider^{d,*}

^a Department of Electrical & Computer Engineering, Rajshahi University of Engineering & Technology, Rajshahi, 6204, Bangladesh

^b Department of Data Science, New Jersey Institute of Technology, 154 Summit Street, Newark, 07102, NJ, USA

^c Department of Computer Science, University of York, Deramore Lane, Heslington, York, YO10 5GH, UK

^d Department of Engineering, Manchester Metropolitan University, Chester Street, Manchester, M1 5GD, UK

ARTICLE INFO

Keywords:

Extreme gradient boosting (XGBoost)
Feature engineering
Machine learning
Prophet
Renewable energy
Very short-term load forecasting (VSTLF)

ABSTRACT

An efficient very short-term load forecast (VSTLF) is essential for power plant unit scheduling because it reduces the planning uncertainty caused by variable renewable energy outputs, which in turn decreases the overall electricity production costs in a power production system. Even though this sector has seen a number of studies and applications in plant scheduling, improving projections with minimum errors is still necessary. In this paper, a novel machine learning framework for predicting very short-term electricity usage was developed. The data utilized in this study are publicly available and were accessed from Kaggle's platform. The framework involved two pivotal stages in its development: robust feature engineering and electric load forecasting via a Prophet-XGBoost-CatBoost (Pr-XGB-CB) model. Feature engineering involves the use of multiple sets of historical electricity consumption data with different time lags for analysis and enhancement. Prophet dissects forecasted data into understandable seasonal, trend, and holiday segments, whereas XGBoost stands out for its speed and effectiveness, especially when dealing with numerous features. To obtain the ensemble forecast, the CatBoost algorithm was utilized. The optimal hyperparameters for the model are evaluated by Optuna. The effectiveness of the suggested model was evaluated by contrasting its performance with that of five alternative machine learning (ML) and deep learning (DL) algorithms. In addition, a sensitivity analysis was conducted to assess the model's robustness under scenarios of missing or limited data, demonstrating its resilience in real-world applications. The proposed framework outperformed the state-of-the-art (SOTA) models, with a Mean Absolute Error (MAE) of 23.70, Root Mean Squared Error (RMSE) of 32.32, and an R^2 of 0.97; hence, the proposed framework performs an efficient guiding role in electrical load prediction. This research offered practical significance by enhancing power plant scheduling efficiency and reducing overall electricity production costs through superior predictive accuracy.

1. Introduction

Ensuring a balance between power supply and demand is crucial for enhancing the stability and energy efficiency of a power supply system. Accurate load forecasting underpins effective power system management by enabling optimal generation scheduling, enhancing grid reliability, and supporting market operations [1]. It is critical in integrating renewable energy sources such as solar and wind by planning resources,

meeting demand, keeping the grid stable, optimizing energy storage, and setting fair market prices [2]. Owing to the nature of intermittent power generation by renewable sources, precise forecasting must be aligned with demand to mitigate any supply variability. In industry-heavy regions, where manufacturing is the main source of high energy consumption [2], reliable load estimation is vital for balancing supply and demand. Moreover, robust forecasting shapes energy policies, drives investments in renewable infrastructure, and strengthens

* Corresponding author.

E-mail addresses: shafiuzzaman.ruet@gmail.com (M. Shafiuzzaman), ornob39@gmail.com (M. Safayet Islam), rubaithbasha@gmail.com (T.M. Rubaith Bashar), Munem.mh@gmail.com (M. Munem), nahiduzzaman@ece.ruet.ac.bd (M. Nahiduzzaman), mominul.ahsan2@gmail.com (M. Ahsan), j.haider@mmu.ac.uk (J. Haider).

<https://doi.org/10.1016/j.energy.2025.137981>

Received 29 February 2024; Received in revised form 15 June 2025; Accepted 8 August 2025

Available online 12 August 2025

0360-5442/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

grid resilience. As power systems evolve to meet ambitious sustainability goals, high-precision load forecasting serves as a backbone for a balanced and decarbonized energy future.

The planning span of a power system operation is often divided into four time frames: very short-term, short-term, mid-term, and long-term [3]. Each of these time frames focuses on some particular measures to take [4]. The very short-term period spans from a few minutes to a single day. It responds rapidly to fluctuations in load and renewable output, minimizes imbalance penalties, and supports intra-day energy trading and demand-response programs [5]. Short-term load forecasting (STLF), spanning 24 h to two weeks with hourly granularity, underpins day-ahead markets, unit commitment, and maintenance scheduling. On the other hand, mid-term forecasting, covering two weeks to three years, focuses on managing generation assets and preventing energy shortfalls. However, in order to specify the construction of new power plants or modifications to the transmission network, the long-term timescale concentrates on a few years to decades.

VSTLF is essential for energy management, helping utilities and grid operators make informed decisions on electricity generation and distribution over short time horizons. Achieving high accuracy in VSTLF is challenging because of the stochastic and dynamic nature of electricity demand, which is driven by multiple interdependent factors. Load patterns are influenced by weather variability, consumer behavior, economic conditions, and the growing integration of distributed energy resources such as rooftop solar and electric vehicles, which introduce bidirectional power flows and volatility [6]. These complexities require advanced modeling techniques, such as machine learning (ML) and hybrid statistical approaches, to handle non-linear relationships and high-dimensional data. Moreover, computational constraints and data quality issues further complicate real-time implementation. To address common time series forecasting challenges such as seasonality and trends, noise and irregular events, and missing data, various statistical approaches, such as autoregressive integrated moving average [6] and grey prediction [7], have been suggested. Nonetheless, these models often prove insufficient in capturing intricate patterns of power consumption, leading to inferior forecasting accuracy [8].

In recent years, several ML techniques, including artificial neural networks, fuzzy neural networks, and support vector regression (SVR), along with several hybrid models, such as region-based Convolutional Neural Networks (RCNNs) [9], Generalized Autoregressive Conditional Heteroskedasticity (GARCH), Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH), and Asymmetric Power GARCH (APGARCH), have demonstrated successful applications in addressing the nonlinear aspects of load forecasting [10]. Jiang et al. [11] demonstrated superior forecasting accuracy and computational efficiency of SVR-based hybrid models. Cecati et al. [12] provided a comprehensive Artificial Neural Network (ANN) for electric power system load prediction and evaluated newly designed algorithms, particularly the ErrCor method, for training Radial Basis Function networks in 24-h electric load forecasting.

A good number of approaches have also been reported with feature engineering to increase the feasibility of load forecasting. Tsalikidis et al. [13] developed a hybrid ensemble model combining multiple ML algorithms with a voting regressor, leveraging temporal features, lagged load values, and exogenous inputs such as temperature and PV generation. Feature selection was guided by permutation importance to capture key patterns, resulting in improved one-step-ahead energy load forecasts. Yamasaki et al. [14] employed feature engineering with weather, calendar, and lagged demand data alongside refined date features and proposed a hybrid ML framework that combines multiple regressors with signal decomposition and AutoML optimization for enhanced VSTLF.

However, traditional approaches often rely on a single forecasting model, which limits their ability to capture the complex, nonlinear relationships in electricity load time series. Additionally, although some studies introduce feature engineering techniques, they often fail to

address the full complexity of temporal relationships, and the impact of these techniques is not adequately demonstrated or quantified, limiting their effectiveness in improving forecast accuracy. Finally, other studies lack real-world applicability, as they do not test models under conditions of data disruption or inadequate datasets.

The major novel contributions of this study are as follows.

- This study introduces a novel hybrid Prophet (Pr), Extreme Gradient Boosting (XGB), and CatBoost (CB), Pr-XGB-CB specifically designed for hourly load forecasting. This hybrid approach combines the strengths of the Prophet, XGBoost, and CatBoost models, forming a powerful ensemble framework to predict very short-term electricity loads with enhanced accuracy and reliability.
- A robust statistical feature engineering methodology is presented, incorporating multilag features for various temporal dependencies. This method provides a more accurate and dynamic representation of historical data, enabling the model to effectively capture the complex patterns in electricity load forecasting, ultimately enhancing forecast precision.
- To assess the robustness and practical utility of the proposed framework, sensitivity analysis is conducted by simulating scenarios with missing or limited data. This analysis not only demonstrates the model's ability to maintain high performance under challenging real-world conditions but also highlights its potential to ensure reliable load forecasting and operational decision-making in power systems where data quality and availability are often imperfect.

This paper begins with an introduction, followed by a literature review on VSTLF in Section 2. Section 3 details the materials and methods used. A discussion of the results and sensitivity analysis are presented in Section 4. Section 5 includes conclusions and recommendations for future research.

2. Literature review

VSTLF represents an active area of research, with a wealth of literature exploring various forecasting approaches, which are typically categorized into four groups according to the employed learning algorithms: physical models, persistence methods, artificial intelligence (AI) techniques, and statistical approaches.

Bae et al. [15] proposed an XGBoost-based method to mitigate behind-the-meter (BTM) solar PV impact on load prediction. By analyzing demand deviation, estimating BTM capacity via grid search, and filtering data based on lighting loads and base temperature, the proposed method accurately restored distorted loads using projected solar PV power. Madrid et al. [16] suggested a group of models based on ML in order to precisely predict the hourly electric load for the next 168 h, where the XGBoost approach demonstrated the optimal results of hourly load forecasts, outperforming the performances of neural networks and five ML models constructed and evaluated in diverse load profile settings. Massaoudi et al. [17] proposed a tiered forecasting method combining LightGBM, XGBoost, and MLP in a stacked model, where meta-data from XGBoost and LightGBM were processed by MLP for load prediction. Their approach was validated on Malaysian and New England datasets, showing strong predictive performance.

More recently, Wan et al. [18] proposed Gated Recurrent Unit (GRU) model for STLF incorporating feature selection and data balancing via sample extension after clustering. This method employed error-focused training and error correction modeling to improve the prediction accuracy. Wang et al. [19] proposed Sample Entropy Variational Mode Decomposition (SVMD), combining VMD and SampEn to separate load data into trend and fluctuation components. A linear regression model was used to predict trends, whereas Bayesian-optimized XGBoost models handled fluctuations. Upon application to industrial users in China and Ireland, the method improved prediction accuracy by accounting for diverse consumption factors. Jiang et al. [20] introduced

Neural-based Similar Days Auto Regression (NSDAR), a neural network for identifying analogous days beyond conventional meteorological methods. NSDAR enables explicable load forecasts by a linear framework based on these patterns and adds load clustering and regional load analysis. Mo et al. [21] employed a Temporal Convolutional Network (TCN) tailored for time data, surpassing CNN limitations. Using the Prophet model, they partitioned power load data into trend, seasonal, and holiday components. By separately forecasting with TCN and Prophet, the models were merged using the least squares method. Shohan et al. [22] aimed to enhance electricity load prediction precision by developing a hybrid Long Short-Term Memory (LSTM)-Neural Prophet model, using historical data and statistical traits for forecasting both long and short-time spans.

A new method for STLTF was devised, emphasizing data refinement and a complex R-CNN combined with the ML-LSTM architecture [9]. The approach involved preprocessing the IHEC dataset with cleansing and Box-Cox transformation, extracting key features via deep R-CNN, and leveraging ML-LSTM for sequential learning while being validated on the PJM benchmark. Bashir et al. [23] introduced a hybrid load forecasting method integrating Prophet for linear and non-linear trends with LSTM for residual non-linear patterns. Using a Back Propagation Neural Network to combine predictions, their approach significantly improved accuracy when validated on Elia Grid's quarter-hourly load data from 2014 to 2021. Ran et al. [24] developed a Transformer model, Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN-SE-TR), which integrated sample entropy, and attention mechanisms to address long-term memory issues. The CEEMDAN-SE-TR model demonstrated significant superiority, particularly over conventional ML models, substantiating its advantages over Empirical Mode Decomposition (EMD) approaches.

The literature provides a various approaches to electricity load forecasting using ML and DL techniques but each method has its own limitations. For instance, the XGBoost approach [16] struggles with adaptability to dynamic load profiles, SVM [19] faces limitations in scenarios with rapidly changing industrial electricity consumption patterns, and the advanced GRU model [18] is sensitive to the choice of clustering methods and feature selection. While Bae et al.'s [15] XGBoost method effectively addresses BTM solar PV interference, its accuracy decreases under varying solar PV conditions. The CEEMDAN-SE-TR Transformer model has computational complexities due to the integration of multiple techniques [24]. NSDAR relies on neural networks for identifying similar days which is sensitive to the quality and representativeness of the training data [20]. The TCN-Prophet hybrid approach [21] struggles with capturing long-term dependencies and the hybrid LSTM-Neural Prophet model [22], while outperforming established methods, requires careful tuning of hyperparameters for optimal performance. The R-CNN combined with the ML-LSTM architecture introduced for STLTF is computationally intensive and sensitive to data preprocessing techniques [9], whereas the stacked XGB-LGBM-MLP models [17] face challenges in generalizing to diverse datasets. Finally, the hybrid method merging Prophet and LSTM models [23] is computationally demanding and requires careful consideration of the temporal dynamics of the electricity load data for optimal performance.

The existing load forecasting studies can benefit from incorporating a comprehensive feature engineering technique. Even in cases where some studies do account for feature engineering impact, it is uncommon to find a comprehensive assessment of feature importance and sensitivity analysis across the entire power system. To address these shortcomings, the proposed study aims to enhance load forecasting by embracing a robust approach with feature engineering.

3. Data and methods

3.1. Overview of proposed architecture

The proposed forecasting framework starts with dataset splitting into training (8 January 2015–31 December 2018) and testing (1 January 2019–27 June 2020) sets. Extensive feature engineering was incorporated to create new features based on several statistical methods. Stationarity and causal relationships were assessed using Augmented Dickey-Fuller (ADF) and Granger Causality tests, respectively. This study introduced a novel ML architecture (Pr-XGB-CB) (Fig. 1), which was designed to improve power system operations by enhancing the accuracy of electrical load predictions. Prophet and XGBoost were trained individually, and their outputs were combined in a CatBoost hybrid model. Prophet effectively addressed the inherent time series nature of the data by decomposing it into interpretable components such as seasonality, trend, and holiday effects, thereby enhancing model interpretability and incorporating temporal patterns for improved forecasting accuracy. XGBoost excels at efficiently handling high-dimensional data by constructing an ensemble of decision trees that progressively refine predictions, enabling the capture of complex, non-linear relationships. CatBoost played a dual role within the Pr-XGB-CB architecture, acting as both a meta-learner to exploit the individual strengths and weaknesses of Prophet and XGBoost for informed predictions and utilizing its robust performance in handling categorical features and ensemble construction to combine the model outputs and generate the final load forecast.

Meticulous hyperparameter optimization using Optuna ensured that each model operated at peak capacity, contributing to the overall accuracy of the ensemble architecture. By integrating the hyperparameter optimization technique, Pr-XGB-CB could emerge as a promising approach for achieving accurate and interpretable VSTLF. Model performance is evaluated and compared with that of other state-of-the-art models, using various metrics such as RMSE, MAE, Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), R-squared (R^2), Explained Variance, Max Error, and deviance measures. These metrics are crucial in power systems for accurately assessing prediction errors and model reliability, ensuring stable operation, efficient load forecasting, and effective risk management by quantifying both average and worst-case deviations.

3.2. Data collection and processing

This study utilizes a publicly available dataset from the National Dispatch Center, Panama, spanning from January 2015 to June 2020 [25]. Covering national electricity demand, the dataset facilitates training and evaluating ML forecasting models, enabling comparisons with official forecasts. Detailed variable descriptions and units are provided in Table 1. Fig. 2 graphically presents each variable of the dataset across different periods in the whole dataset. The dataset comprised 48,048 readings with 17 different features and was complete, with no missing values. A limited number of instances of low load values were observed, which were attributed to hourly blackouts and power grid damage. The data clearly exhibited a noticeable pattern that suggested the presence of seasonal variations except for holiday, holiday id and school, which are categorical.

Fig. 3 shows the power usage for an individual day of a week for four different seasons. The power usages on Friday, Monday, Thursday, Tuesday, and Wednesday are similar for different seasons. A noticeable decline in power consumption was observed on Saturdays, with Sundays recording the lowest usage across all days. Feature stationarity, which is crucial for reliable time series forecasting models [26], was assessed using the ADF test to ensure statistically consistent characteristics over time. Table 2 confirms feature stationarity through consistently low P values.

Fig. 4 illustrates the outcomes of the Granger causality test [27],

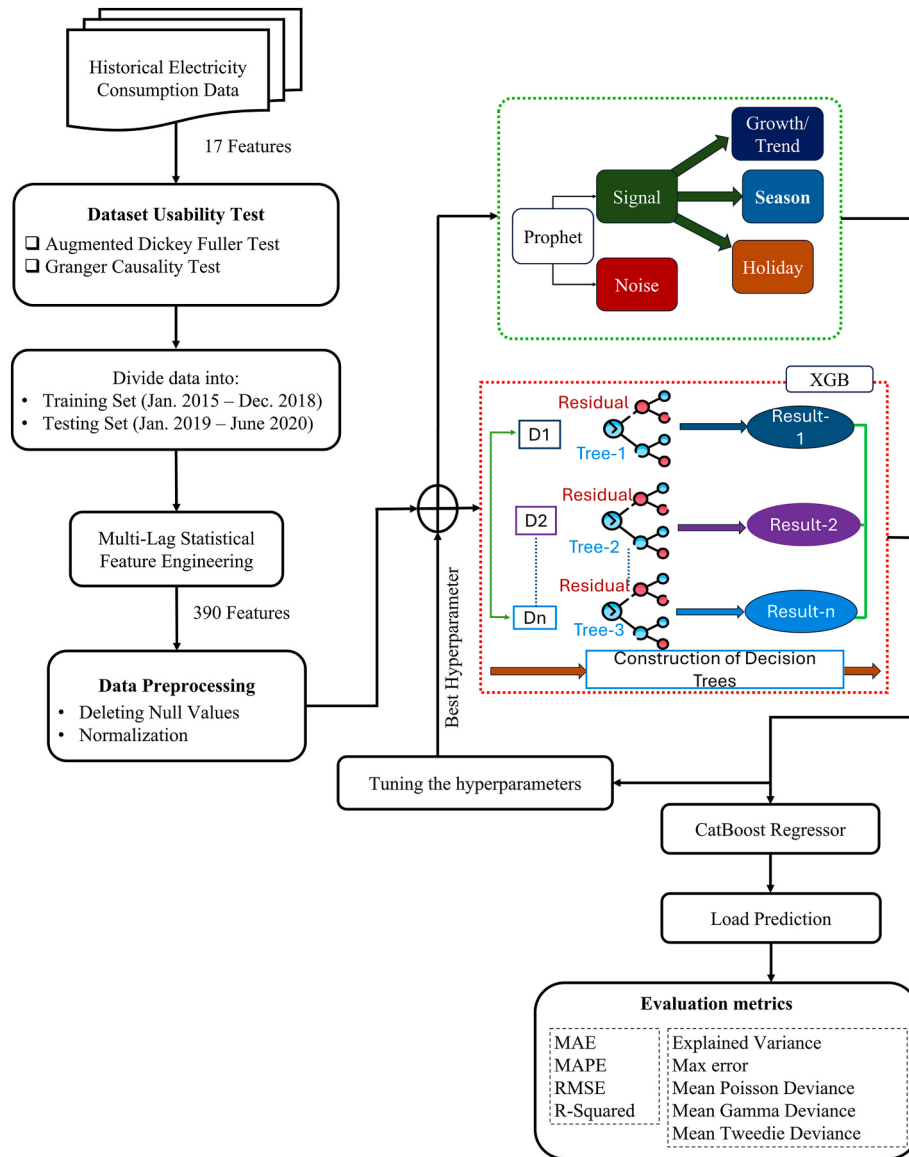


Fig. 1. Comprehensive flowchart illustrating the stages of the proposed forecasting methodology.

Table 1

Variables, description and units of measure from the electricity consumption dataset.

Variable	Description of variables	Unit of Measurement
datetime	Date with time	Hour
nat_demand	National electricity load	MWh
T2M_c	Temperature at 2 m	°C
QV2M_c	Relative humidity at 2 m	%
TQL_c	Liquid precipitation	l/m ²
W2M_c	Wind speed at 2 m	m/s
holiday	Holiday binary indicator (1 = holiday, 0 = regular day)	–
Holiday_ID	Unique identification number for holiday	–
school	School period binary indicator (1 = school, 0 = vacations)	–

Subindex c stands for different cities (David, Santiago and Tocumen, Panama City) such as T2M_dav represents Temperature at 2 m in David City.

highlighting both significant and non-significant temporal dependencies among features. While T2M_san showed a weaker association (p-value = 0.19), W2M_san displayed no statistically significant predictive power

(p-value = 0.5524). Other features—such as T2M_toc, QV2M_toc, TQL_toc, W2M_toc, QV2M_san, TQL_san, T2M_dav, QV2M_dav, TQL_dav, W2M_dav, Holiday_ID, and holiday—exhibited strong Granger-causal relationships ($p < 0.05$). These results underscore the relevance of meteorological and calendar-based features, while suggesting that further analysis of the non-significant variables may help refining forecasting accuracy.

3.3. Feature engineering

The feature engineering method proposed in Fig. 5 involves the use of ten different statistical measurements, including (i) Lag Feature (LAG), (ii) Rolling Mean (MEAN), (iii) Rolling Standard Deviation (STD), (iv) Exponentially Weighted Moving Mean (EWM MA), (v) Exponentially Weighted Moving Standard deviation (EWM STD), (vi) Min-max normalized features (MIN MAX), (vii) Median (MEDIAN), (viii) Skewness (SKEW), (ix) Kurtosis (KURT), and (x) 50th percentile (P50). These statistical components are computed within 12, 24, and 128 rolling windows to capture short-term, medium-term, and long-term dependencies in the data, allowing for the extraction of temporal patterns at multiple time scales and the recognition of both transient and

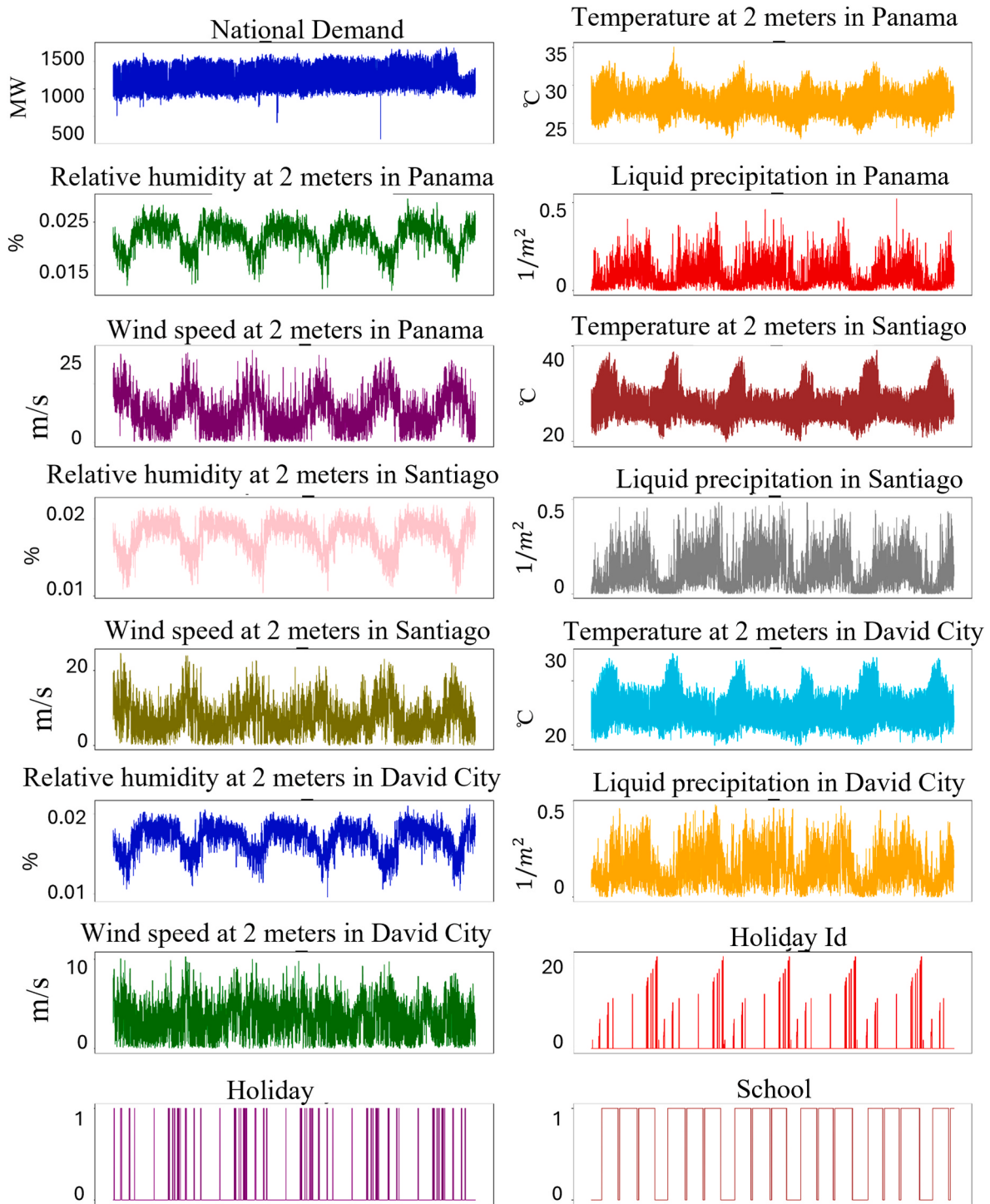


Fig. 2. Variables in the electricity consumption dataset across the time period.

sustained trends. This process expands the dataset from its initial 17 features to a comprehensive set of 390 features.

The first five statistical techniques, LAG, MEAN, STD, EWM MA, and EWM STD, were chosen for the target feature (nat_demand) and environmental features, namely, temperature (T2M), humidity (QV2M), rainfall (TQL), and wind speed (W2M) (across David, Santiago and Tocumen city), based on the specific characteristics of demand data and the forecasting requirements. Since these raw features are time-dependent, lag features and moving averages were used to capture temporal dependencies and trends where past demand and weather conditions can significantly impact present demand. Rolling mean and

exponentially weighted moving mean helped smooth out short-term fluctuations and emphasize long-term patterns. To address the inherent volatility in demand, the rolling standard deviation and exponentially weighted moving standard deviation were applied to capture recent changes over older data, ensuring that the model adapts quickly to fresh patterns such as abrupt temperature spikes or sudden humidity changes.

For temperature at 2 m (T2M), specific humidity at 2 m (QV2M), total cloud liquid water (TQL), and wind at 2 m (W2M)—across the Tocumen (toc), San Jose (san), and David (dav) stations—five additional techniques were applied to enhance their predictive power. Min-max

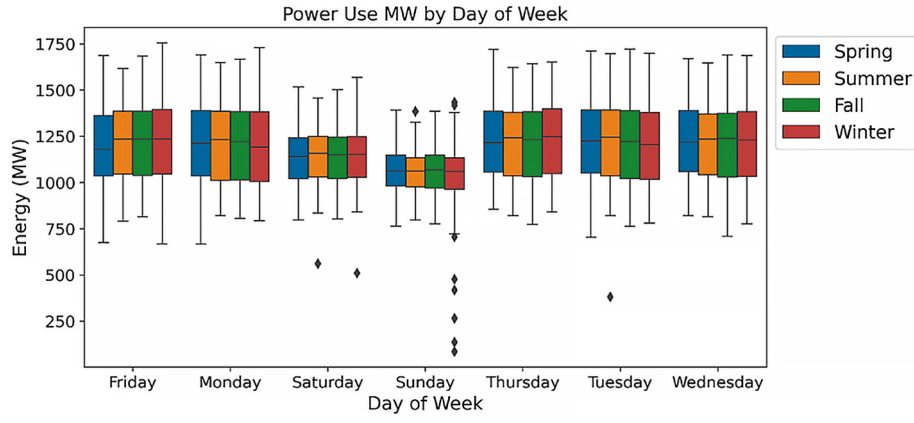


Fig. 3. Power use in MW every day for a week.

Table 2
Result of Augmented Dickey-Fuller (ADF) test for time series stationarity analysis.

Feature Name	P Value
nat_demand	0.0
T2M_toc	8.56×10^{-24}
QV2M_toc	1.53×10^{-12}
TQL_toc	1.01×10^{-26}
W2M_toc	3.27×10^{-20}
T2M_san	9.01×10^{-15}
QV2M_san	1.64×10^{-11}
TQL_san	3.85×10^{-25}
W2M_san	4.29×10^{-26}
T2M_dav	5.41×10^{-18}
QV2M_dav	4.05×10^{-15}
TQL_dav	9.7×10^{-27}
W2M_dav	4.06×10^{-30}
Holiday_ID	0.0
holiday	0.0
school	9.15×10^{-09}

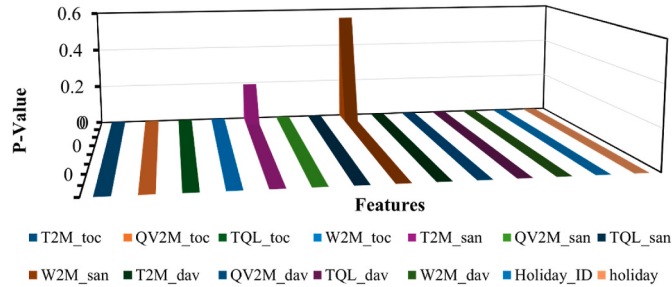


Fig. 4. Visualization of Granger causality test results for variable relationships.

normalization was necessary to bring different units (temperature, humidity, and wind speed) onto a comparable scale, especially for the ML models sensitive to feature magnitude. The median, skewness, kurtosis, and 50th percentile metrics were extracted to capture the distributional shape, asymmetry, presence of outliers, and central tendency of each feature, making the model more robust against extreme environmental events.

Additional feature engineering techniques such as min-max normalization, skewness, kurtosis, and percentile-based metrics were not applied to the nat_demand feature, as it is the target variable. Applying such transformations risks data leakage by requiring future data access, which is inappropriate for time series targets that are dependent on past values. Moreover, altering the target's scale may distort its structure and impair model performance. Similarly,

Holiday_ID, holiday, and school were excluded from these transformations as categorical features not suited to time-based processing.

The naming convention for the newly created features follows a defined specification: Original Name.Statistical Operation.Window Size. For instance, QV2M_san_ewm_mean128 represents a new feature where QV2M_san is the original feature, ewm_mean denotes the statistical operation, and 128 is the window size.

To check the stationarity after feature engineering, an ADF test was conducted, and the top 40 resulting p values are presented in Fig. 6. A lower p value, usually below 0.05, indicates stronger evidence against the null hypothesis of non-stationarity, suggesting the presence of stationarity within the time series. It was obvious that among the 390 features analyzed, 26 exhibited p-values exceeding 0.05, suggesting non-stationarity. T2M_san_ewm_std_128 emerged as the most non-stationary feature with a P-value of 0.74. The top 10 non-stationary features include various temperature and humidity-based engineered variables. However, most features after the 26th feature have P values less than 0.05, indicating stationarity post-feature engineering.

3.4. Machine learning models

This section provides a brief overview of the principles underlying the different models chosen for developing the hybrid model and comparison.

3.4.1. LSTM

LSTM models, a subset of recurrent neural networks (RNNs), have emerged as formidable tools for time series forecasting [28]. The mathematical expressions for the LSTM model demonstrated in Fig. 7 (a) are shown in equations (1)–(5) [5]:

$$\begin{bmatrix} F_t \\ I_t \\ C_t \\ O_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \\ \sigma \end{bmatrix} (W \cdot [X_t, h_{t-1}] + b) \quad (1)$$

$$C_t = f_t * C_{t-1} + I_t \odot \tilde{C}_t \quad (2)$$

$$h_t = O_t * \tanh(C_t) \quad (3)$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5)$$

where X_t is the input sequential; b denotes the bias weights; the input weights are W is the latest time step; $t-1$ is the previous time step; h_t denotes the output; C_t signifies the cell state; and f_t , I_t and O_t are the

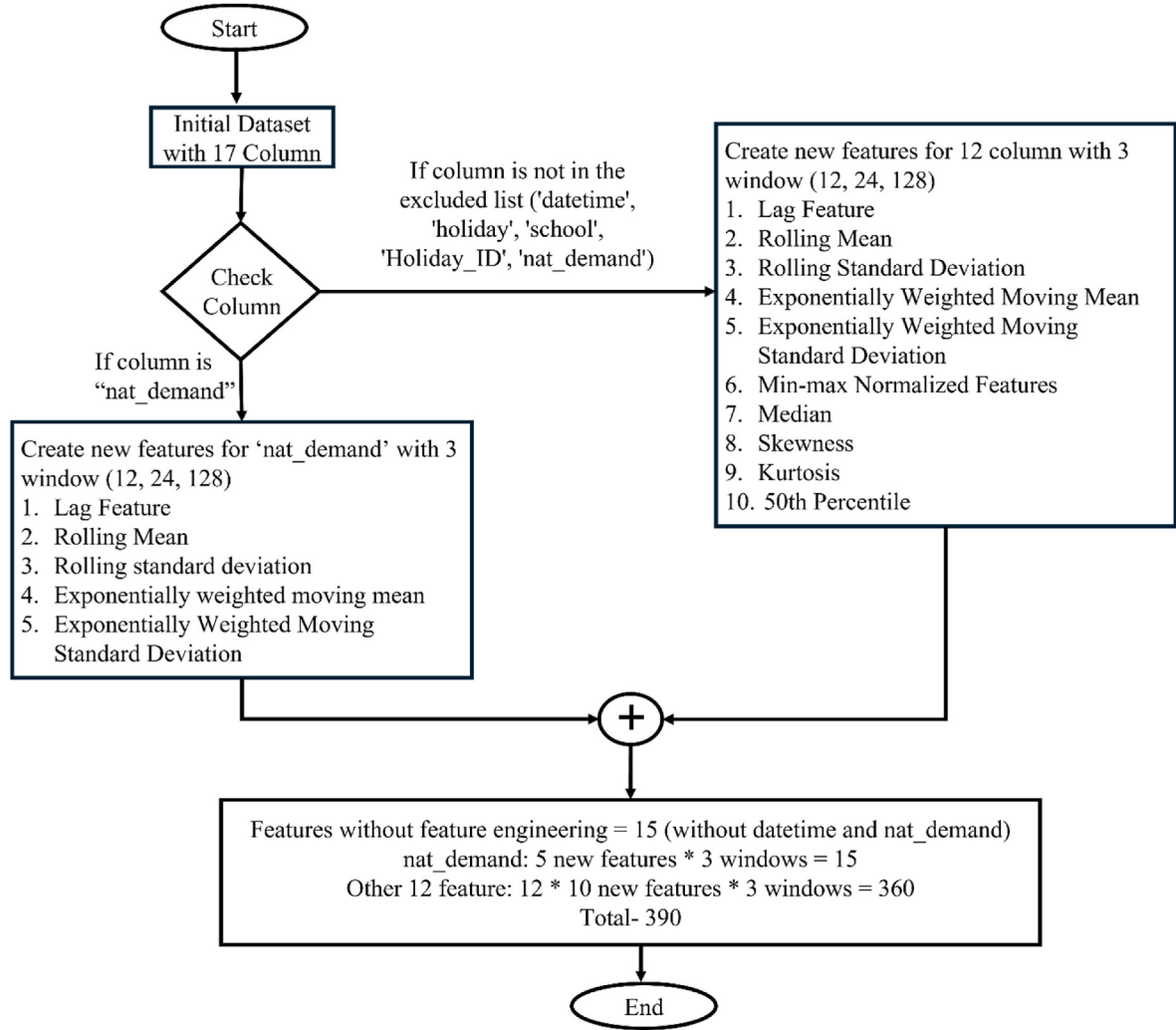


Fig. 5. Detailed flowchart of proposed feature engineering process.

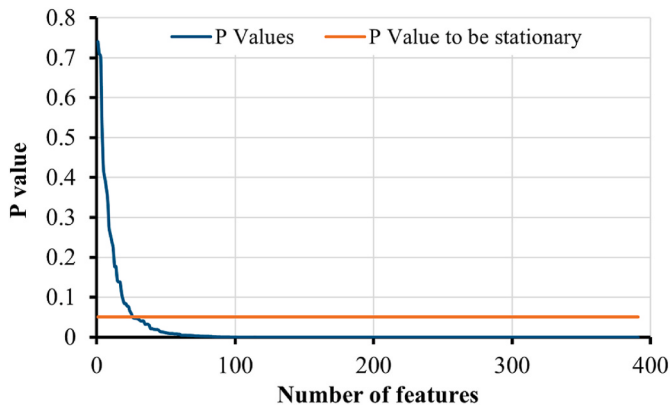


Fig. 6. Visualization of Augmented Dickey Fuller (ADF) test P-values and the 0.05 stationarity threshold after feature engineering.

forget, input, and output gates, respectively.

3.4.2. GRU

GRU (Gated Recurrent Unit) models, a variant of RNNs, are valuable tools for time series forecasting, including electric load prediction. The simplified architecture of GRUs demonstrated in Fig. 7 (b) is well known for enabling effective training and quicker convergence. The

mathematical expressions for the GRU model [29] are expressed in equations (6) and (7):

$$\begin{bmatrix} z_t \\ r_t \\ \tilde{h}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \end{bmatrix} (W \cdot [X_t, H_{t-1}] + b) \quad (6)$$

$$h_t = (1 - h_t) * h_{t-1} + z_t * \tilde{h}_t \quad (7)$$

where X_t represents the input at time step t , h_t is the output at time step t , z_t is the update gate, r_t is the reset gate, and $\tilde{h}(t)$ is the new candidate activation.

3.4.3. LightGBM

LightGBM, a gradient boosting framework, stands out for its efficiency in handling large datasets through histogram-based binning and leaf-wise tree growth [30]. The objective function optimizes the model by minimizing the prediction error over n training samples—where y_i is the true label and $\hat{y}_i$ is the predicted output—while incorporating a regularization term $\Omega(f_k)$ to penalize the complexity of each of the K trees f_k in the model. The basic architecture of the LightGBM is demonstrated in Fig. 8. The LightGBM model's loss function is defined by Eq. (8) [30]:

$$\text{Objective} = \sum_{i=1}^n \text{Loss}(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (8)$$

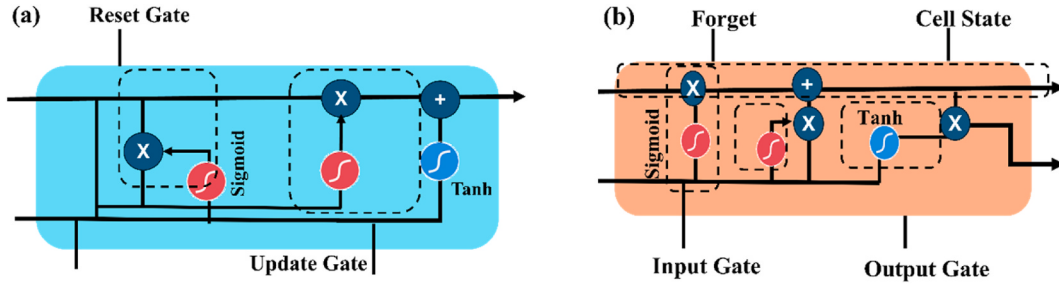


Fig. 7. Basic architecture of (a) Long Short-Term Memory (LSTM) and (b) Gated Recurrent Unit (GRU).

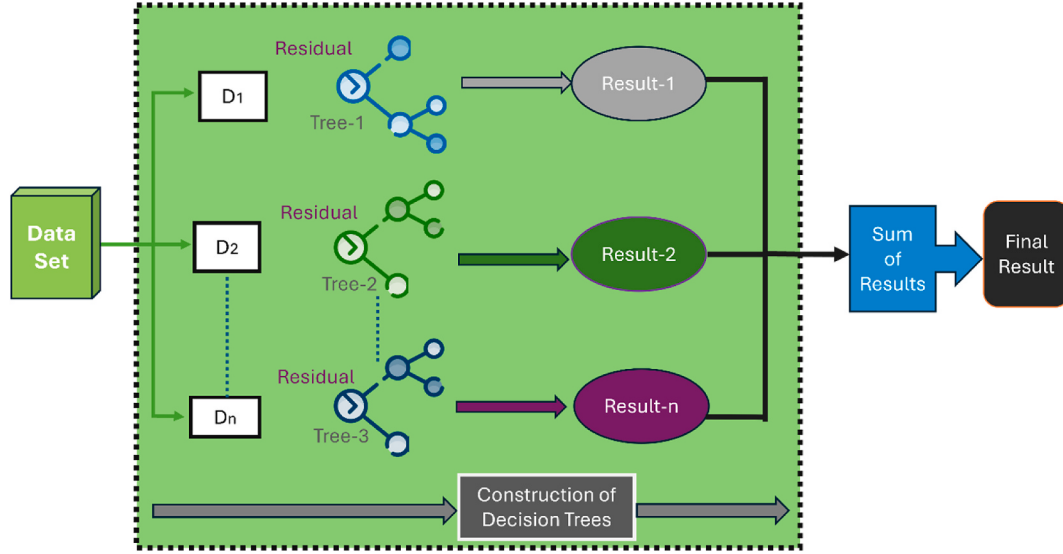


Fig. 8. Basic architecture of LightGBM model.

3.4.4. XGBoost

Extreme Gradient Boosting (XGBoost), is a popular ML method that integrates results from various decision trees to produce precise predictions for regression and classification issues (Fig. 9). The unique architecture of XGBoost can handle complex datasets and operates with extreme efficiency. The gradient boosting framework continuously improves the model by correcting errors generated in previous rounds [31].

Owing to the high dimensionality of the features involved in this research, handling them efficiently becomes a complex task. XGBoost presents itself as an optimal solution. The XGBoost model in the

proposed Pr-XGB-CB framework uses a carefully crafted objective function that balances prediction accuracy and model complexity via regularization. Equation (9)–11 [32] present the detailed formulation of both the objective and prediction functions employed by XGBoost.

Objective Function: The core objective function for XGBoost is a combination of the loss function (commonly the mean squared error for regression tasks) and the regularization term:

$$L(\theta) = \sum_{i=1}^N [l(y_i, \hat{y}_i)] + \Omega(f) \quad (9)$$

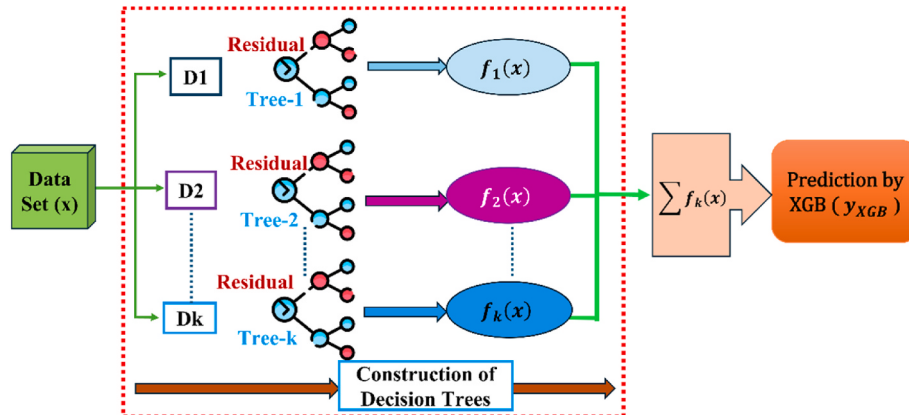


Fig. 9. Illustration of the XGBoost model architecture: from input features to optimized boosted trees.

where $l(y_i, \hat{y}_i)$ is the loss function and where $\Omega(f)$ is the regularization term for the tree parameters f to avoid loss.

Regularization Term: The regularization term penalizes model complexity by controlling the number of leaves (T) and the magnitude of the leaf weights, w_j .

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (10)$$

where γ is the regularization on the number of leaves and where λ is the regularization on the leaf weights.

Prediction: The output of the XGBoost model is the sum of the outputs from all the trees. The prediction of the k^{th} tree:

$$y_{XGB} = \sum_{k=1}^K f_k(x) \quad (11)$$

where $f_k(x)$ is the prediction of the k^{th} decision tree and where k is the total number of trees in the model.

3.4.5. Prophet

The Prophet model is a forecasting tool designed to address time series challenges in business, finance, and weather forecasting. This model has gained popularity for its ease of use and ability to capture complex patterns. Key features include automatic seasonality detection, holiday effects, flexible trend modeling, uncertainty estimation, and customizable parameters. A graphical representation of the model is shown in Fig. 10.

The Prophet model is selected for this research because of its robustness in handling complex, high-dimensional time series data. Its ability to automatically detect seasonality, model trends flexibly, and incorporate external factors which make it well suited for capturing dynamic and periodic patterns. Additionally, Prophet's uncertainty estimation ensures reliable forecasting, which is essential for optimizing power grid operations. The general equation for the Prophet model [21] can be written as:

$$y_t = g(t) + s(t) + h(t) + e_t \quad (12)$$

where y_t is the observed value at time t , $g(t)$ = growth/trend component (logistic or linear growth), $s(t)$ is the seasonal component, $h(t)$ is the holiday/event component, and e_t is the error term.

Trend component: The trend component $g(t)$ in the Prophet model can be modeled in two ways: as a linear function capturing gradual changes in the growth rate with changepoints or as a logistic growth function accounting for saturation effects where growth slows near a carrying capacity. Both approaches enable flexible modeling of long-term temporal trends.

$$\text{or, } g(t) = \text{linear}(t) = kt + m \quad (13)$$

$$g(t) = \text{logistic}(k, t) = \frac{C}{(1 + \exp(-k(t - t_0)))} \quad (14)$$

where C is the carrying capacity, k is the growth rate, t_0 is the time when the growth rate changes, k is the trend growth rate, and m is the starting value.

Seasonal component: To represent the daily, weekly and yearly seasonality of the electric load data, the seasonality component $s(t)$ is typically modeled using Fourier series expansions to capture periodic effects:

$$s(t) = \sum_{i=1}^M \left[a_i \cos\left(\frac{2\pi i t}{P}\right) + b_i \sin\left(\frac{2\pi i t}{P}\right) \right] \quad (15)$$

where M is the number of Fourier terms, P is the period (365 for yearly seasonality, 7 for weekly seasonality), and a_i and b_i represents the coefficients for the sine and cosine terms.

Holiday component: The holiday component, $h(t)$, models the effect of holidays on the time series by summing the individual holiday effects δ_j weighted by the indicator function $I(t; h_j)$, which determines whether a particular holiday h_j occurs at time t for holiday j .

$$h(t) = \sum_{j=1}^J [I(t; h_j) \delta_j] \quad (16)$$

Prediction: Combining the growth, seasonality and holiday components, the prophet model predicts the final demand represented by Eq. (17).

$$y_{\text{Prophet}} = \hat{y}_t = g(t) + s(t) + h(t) \quad (17)$$

3.4.6. CatBoost

CatBoost is a high-performance gradient boosting ML tool designed specifically to handle category features. It is well known for its ability to automatically handle categorical data, eliminating the need for laborious preparation. CatBoost offers exceptional prediction accuracy while minimizing overfitting through the use of an innovative ordered boosting algorithm [33].

The detailed formula of the CatBoost model is as follows:

$$y_{\text{meta}} = \text{CatBoost}(y_{\text{Prophet}}, y_{\text{XGB}}) \quad (18)$$

where y_{meta} is the final prediction from the meta-model.

The Cat Boost Regressor tries to learn a function $\hat{y}_{\text{meta}}$ that minimizes the loss function l :

$$\hat{y}_{\text{meta}} = \text{CatBoost}(y_{\text{Prophet}}, y_{\text{XGB}}) = \arg \min \sum_{i=1}^N l(y_i, \hat{y}_i) \quad (19)$$

where $l(y_i, \hat{y}_i)$ is the loss function (RMSE).

3.4.7. Proposed hybrid model

In this study, CatBoost is employed as an ensemble method to integrate the predictions from individual models, Prophet and XGBoost,. The ensemble model is a simple method to combine the predictions from the Prophet and XGBoost models. The predictions from both models are

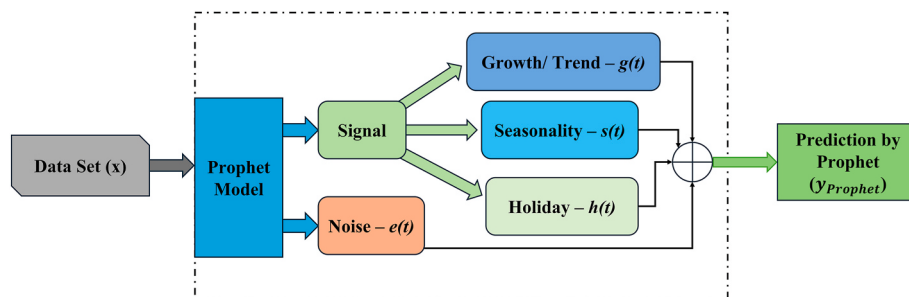


Fig. 10. Schematic overview of the Prophet model architecture.

combined using a weighted sum:

$$Y_{ensemble} = \alpha Y_{Prophet} + \beta Y_{XGB} \quad (20)$$

where $Y_{ensemble}$ is the combined prediction from the ensemble model and where α and β are the weights assigned to the predictions of Prophet and XGBoost, respectively.

The ensemble combines Prophet and XGBoost predictions with weights of 0.7 and 0.3, respectively, based on their validation performance. Prophet, showing lower error and better handling of seasonal patterns, is given greater emphasis. XGBoost complements it by capturing nonlinear relationships. This weighted approach balances temporal trends and feature interactions, enhancing overall forecast accuracy. The performance is measured using the RMSE on the test set.

3.5. Hyperparameter tuning

To address the high dimensionality of the dataset and ensure optimal model performance, a multifaceted approach to hyperparameter tuning was undertaken. Optuna [34] was used for efficient automated tuning of gradient boosting models (XGBoost, LightGBM, and CatBoost), with 100 trials minimizing the RMSE on the validation set. The best configurations were then used for final training and testing. For the DL models (LSTM, GRU) and Prophet, manual tuning was applied. The LSTM and GRU were trained with various hyperparameter sets, and those with the lowest validation RMSEs were selected. The selected hyperparameters for all the models are detailed in Table 3.

3.6. Evaluation criteria

In addition to common regression metrics such as the MSE, MAE, MAPE, and R-squared, this study employs advanced statistical metrics tailored to the unique challenges of power system operations. Mean Poisson Deviance (MPD) captures count-based events, Mean Gamma Deviance (MGD) addresses skewed peak demands, and Mean Tweedie Deviance (MTD) handles zero or low load periods common in power grids. Explained Variance (EV) reflects the model's ability to capture load variability, while Max Error (ME) highlights extreme forecasting errors critical for system reliability. These metrics collectively ensure robust and reliable forecasting performance across the complex and variable conditions inherent in power systems. Mathematical equations [3] for these metrics are provided below:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (21)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (22)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (23)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (24)$$

$$MPD = \frac{1}{n} \sum_{i=1}^n \left[2 \cdot \left(y_i \cdot \log \frac{y_i}{\hat{\lambda}_i} - (y_i - \hat{\lambda}_i) \right) \right] \quad (25)$$

$$MGD = \frac{1}{n} \sum_{i=1}^n \left[2 \cdot \left(\frac{y_i}{\hat{\alpha}_i} - \log \frac{y_i}{\hat{\alpha}_i} - 1 \right) \right] \quad (26)$$

$$MTD = \frac{1}{n} \sum_{i=1}^n \left[2\omega_i \left(\frac{(\hat{p}_i - \hat{\mu}_i) \cdot y_i}{\hat{\phi}} - b(\hat{p}_i, \hat{\phi}) \right) \right] \quad (27)$$

Table 3

Key hyperparameter configurations for the LSTM, GRU, LightGBM, Prophet, XGBoost, and CatBoost models.

Model Name	Hyper Parameter Name	Range	Selected Value	Reasons for selection
LSTM & GRU	Epochs	10–100	60	Controls training iterations.
	Batch Size	16–512	64	Impacts memory and computation.
	Learning Rate	0.001–0.1	0.01	Affects convergence speed.
	Number of LSTM/GRU Layers	1–5	3	Affects model depth.
	Input Shape	–	13031, 1, 390	Defines dataset dimensions.
LightGBM	Number of LSTM/GRU Units	50–512	112	Larger units capture more complexity, but may overfit.
	Number of Leaves	20–1000	61	Controls tree complexity.
	Learning Rate	0.001–0.1	0.00665	Determines step size.
	Feature Fraction	0.3–1	0.5134	Limits feature exposure to reduce overfitting.
	Bagging Fraction	0.5–1	0.6470	Reduces overfitting by subsampling data.
Prophet	Bagging Frequency	1–100	9	Increases model diversity to reduce overfitting.
	Change point prior scale	0.01–0.1	0.05	Controls model flexibility to adapt to changes.
	Seasonality prior scale	1–20	10.0	Adjusts seasonal influence.
	Holidays prior scale	1–20	10.0	Determines holiday.
	Interval width	0.5–1	0.8	Sets uncertainty interval width.
XGBoost	Uncertainty Samples	0–1000	0	Affects uncertainty estimation accuracy.
	Learning Rate	0.001–0.1	0.068	Controls learning speed.
	Max Depth	3–10	6	Affects tree depth.
	Minimum Child Weight	1–100	3.92	Reduces overfitting by controlling child node weight.
	Sub sample	0.5–1	0.945	Reduces overfitting by using a subset of data.
CatBoost	Col sample by tree	0.3–1	0.612	Limits feature usage to prevent overfitting.
	Gamma	0–10	3.14	Regularization to prevent overly complex models.
	L2 Regularization (Lamda)	0–10	1.207	Penalizes large weights to avoid overfitting.
	L1 Regularization (Alpha)	0–10	0.98	Encourages sparsity to reduce overfitting.
	Iterations	10–1000	100	Controls number of trees.
CatBoost	Depth	4–10	6	Affects tree complexity.
	Learning Rate	0.001–0.3	0.1	Step size for training.
	Loss Function	MAE, RMSE	RMSE	RMSE for regression tasks.

$$EV = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (28)$$

$$ME = \max_{i=1}^n |y_i - \hat{y}_i| \quad (29)$$

where n is the number of samples or observations, y_i represents the actual values, $\hat{y}_i$ represents the predicted values, $\hat{\lambda}_i$ represents the predicted values from the Poisson regression model, $\hat{\alpha}_i$ represents the predicted shape parameter from the gamma regression model, $\hat{p}$ represents the predicted mean from the tweedie regression model, $\hat{\mu}_i$ represents the predicted variance function from the tweedie regression model, $\hat{\phi}_i$ represents the estimated dispersion parameter and ω_i represents a weight associated with each observation.

4. Results and analysis

4.1. Experimental setup and plan

The experiments used a cloud infrastructure from the Kaggle platform, providing access to necessary GPUs and storage for efficient simulations. An NVIDIA Tesla P100 GPU with 16 GB GDDR5 memory and 3584 CUDA parallel processing cores was chosen. The CPU model was Intel(R) Xeon(R) @ 2.30 GHz with 2 physical cores clocked at 2.3 GHz. The total system RAM amount was 12 GB. All simulations were implemented using Python 3.10 for the ML workflow, with Pandas 2.1 NumPy 1.26 and Scikit-Learn 1.3.2 serving as the core data science libraries. The public Panama electricity demand dataset was loaded into this framework directly from Kaggle repositories [25]. The performance of the models was evaluated by training without feature engineering and with feature engineering.

4.2. Case-I: without feature engineering

The first case evaluates the models' performance using only the 16 (including the target feature) specified features, without any additional feature transformations. This approach allows us to assess how the inclusion of engineered features improves model accuracy and predictive power at later stages. As shown in Table 4, the Pr-XGB-CB, LSTM, GRU, and Prophet models outperformed the LightGBM and XGBoost models across all the evaluation metrics without integrating feature engineering. Specifically, the Pr-XGB-CB model demonstrated the best performance, with an RMSE of 110.20, an R^2 of 0.66, and an MPD of 9.77, underscoring its effectiveness and robustness. Significant gaps across these evaluation metrics between Pr-XGB-CB and other models suggest the superiority of the proposed model.

Fig. 11(a) shows the actual vs. predicted load in the whole test dataset. It is clear that the predicted graph cannot cover the actual load demand curve. In this particular case, the suggested framework lacks the ability to forecast seasonal patterns. It captures the daily pattern but not to an effective extent, consistently produces a nearly uniform output and fails to identify a sudden drop in the electricity load demand. This suggested that the proposed model demonstrated unsatisfactory performance for the entire dataset. Fig. 11(b) compares the actual load

demand (in MWh) with the predicted demand over the worst 24-h period.

The actual demand fluctuates between approximately 900 MWh and 1400 MWh, showing peaks in the early morning and evening hours. The gap between the actual demand and the predicted demand varies significantly throughout the day, with the minimum gap occurring at approximately 21:00 h and the maximum gap occurring between 00:00 and 02:00 h. Throughout the entire 24-h period, the predicted demand consistently exceeds the actual demand by a large margin. Fig. 11 (c) represents the best 24-h prediction, where both the actual and predicted demands follow a similar trend with close tracking throughout the 24-h period. The model prediction captures the peaks and dips in the actual demand, such as the rise in demand in the early morning and evening. Compared with the worst 24 h, there is a notable improvement, but there is still a discrepancy, particularly after midday and in the evening hours, where the predicted demand overestimates or underestimates the actual demand. The predicted curve is somewhat smoother than the actual demand curve, suggesting that the model oversmooths or does not capture the finer variations in the actual demand. According to the best scenario, predictions still need refinement, especially in accounting for midday dips and ensuring finer adjustments to capture demand variations more accurately.

The feature importance without feature engineering analysis shows a hierarchy of significant factors, as illustrated in Fig. 12. Metrics related to the atmospheric moisture content, such as QV2M_san (9.48 %), appeared essential in predicting the target variable. This was closely followed by TQL_san (9.16 %) and W2M_san (9.06 %), which highlighted the profound impact of temperature and wind speed at a height of 2 m above the surface. Additionally, variables capturing temperature and wind at different vertical levels, such as TQL_dav (8.20 %) and W2M_dav (8.71 %), present considerable importance. In contrast, features such as school (1.04 %) and holiday (0.12 %) exhibited marginal significance and had a weaker impact on the prediction.

Despite the fairly uniform importance of the primary meteorological predictors, the model's reliance on raw features can mask complex interactions, non-linear effects, and temporal dependencies inherent in the data. By applying feature engineering—such as generating lagged variables, rolling-window statistics, and interaction terms between temperature, humidity, and wind speed—we can expose latent patterns, reduce multicollinearity, and better capture seasonal and trend components. The derived features can improve the forecasting accuracy of the model, as evidenced by the performance gaps in Table 4 and the residual patterns in Fig. 11, which indicate the need for further enhancement.

4.3. Case II: with feature engineering

Considering the unsatisfactory performance of the proposed model without feature engineering across the entire test dataset, it became clear that advanced techniques were required to enhance its predictive capability. As a result, feature engineering was applied to improve the model's performance. Table 5 presents the evaluation metrics of various models after feature engineering. Among them, the proposed Pr-XGB-CB model consistently outperforms the other models across key metrics. It achieves the lowest MAPE (0.0203), indicating only a 2.03 % average

Table 4
Evaluation metrics of different models without feature engineering.

Model	RMSE	MAE	MAPE	R^2	EV	ME	MPD	MGD	MTD
LSTM	125.16	96.57	0.081	0.56	0.57	1376.2	12.56	0.01	15666.58
GRU	123.38	87.55	0.075	0.57	0.57	1410.2	11.98	0.0097	15222.09
Prophet	123.47	87.67	0.0757	0.57	0.59	530.98	12.08	0.0098	15142.56
XGBoost	143.64	114.845	0.0948	0.42	0.46	1209.94	16.75	0.01398	20634.41
LightGBM	156.96	126.36	0.102	0.30	0.36	1169.59	19.75	0.016	24636.41
Pr-XGB-CB	110.20	81.32	0.069	0.66	0.66	1153.76	9.77	0.00818	12145.60

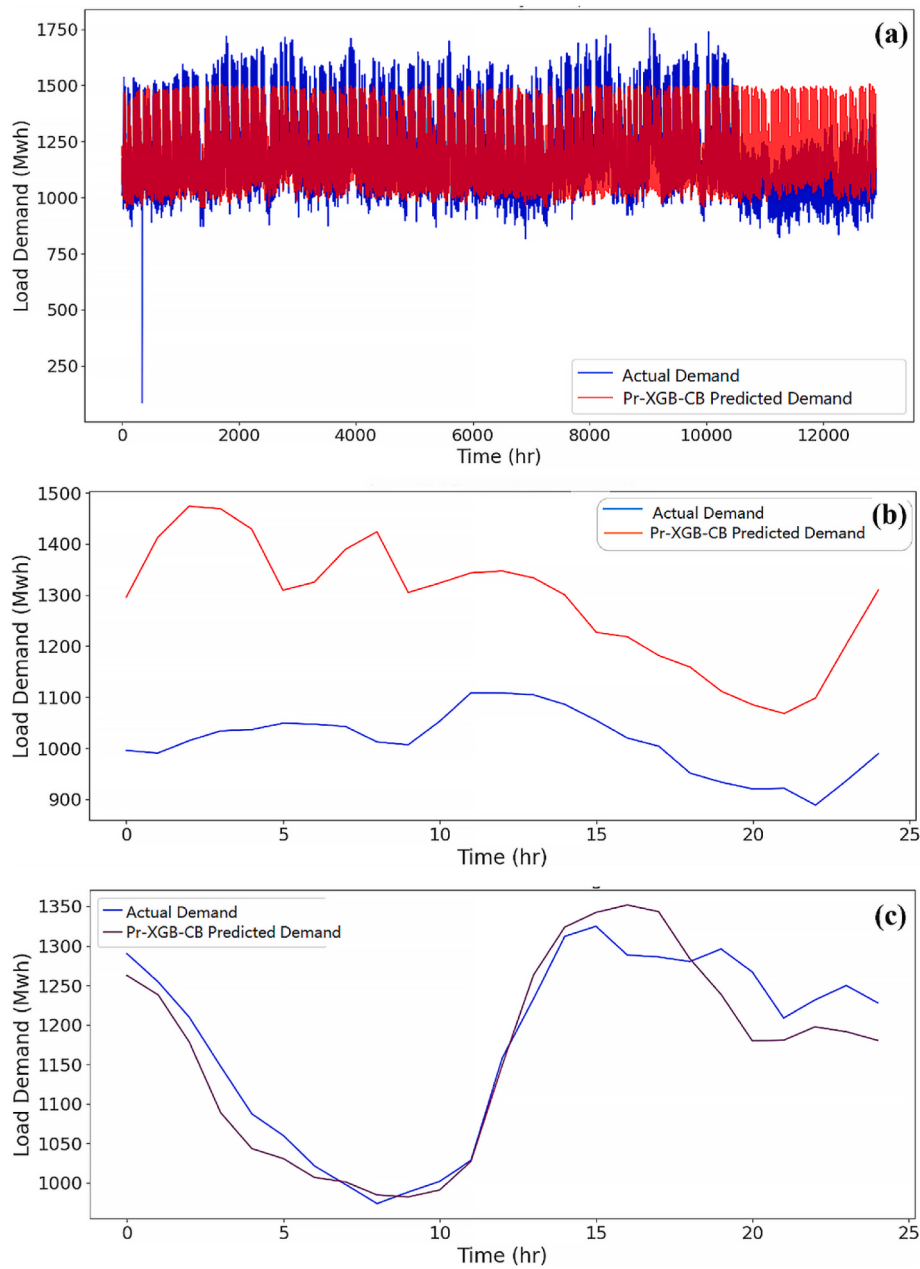


Fig. 11. Actual vs. prediction results of the proposed model without feature engineering: (a) full test dataset; (b) worst 24 h; and (c) best 24 h (worst and best are selected based on evaluation metrics).

prediction error, along with the lowest ME (223.83), suggesting minimal bias. It also records the highest R^2 and EV (0.9704), demonstrating its strong ability to capture data variance. Its RMSE (32.32) and MAE (23.70) are also the lowest, reflecting higher accuracy than the other models do. Compared with Prophet, Pr-XGB-CB reduces the RMSE by 22.6 %, the MAE by 18.9 %, and the MAPE by 21.9 %, with a 2.8 % increase in R^2 . Compared with XGBoost, XGBoost improves the RMSE by 36.4 %, the MAE by 26.2 %, the MAPE by 27.7 %, and the R^2 and EV by 4.7 % each. These results highlight the superior accuracy, consistency, and predictive performance of the Pr-XGB-CB model, making it the most effective for load forecasting.

The results presented in Tables 4 and 5 highlight the significant impact of the feature engineering technique on improving the performance of the proposed model. Specifically, for the proposed Pr-XGB-CB framework, feature engineering led to a remarkable reduction in the RMSE by approximately 70.67 % (from 110.20 to 32.32) and the MAE

by 70.85 % (from 81.32 to 23.70), along with a 46.97 % increase in the R^2 score (from 0.66 to 0.97). These improvements clearly demonstrate the effectiveness of the feature engineering strategy in enhancing the model's predictive accuracy and robustness, reinforcing the suitability of the hybrid Pr-XGB-CB architecture for high-performance VSTLF.

Fig. 13 (a) shows a comparison between the actual and predicted loads across the entire test dataset. The graph shows a strong alignment between the predicted load and the actual load demand. Notably, the proposed framework effectively captures seasonal patterns, consistently producing predictions that closely match actual demand, even during sudden drops. At the beginning of the graph, a notable drop in demand occurred, which was precisely captured by the proposed model. Similarly, in the latter part of the graph, the model accurately tracked the overall decrease in energy demand. Fig. 13(b), (c), and 13(d) compare the actual and predicted demand across three distinct segments, i.e., worst, average, and best, as derived from Table A1 (see the appendix).

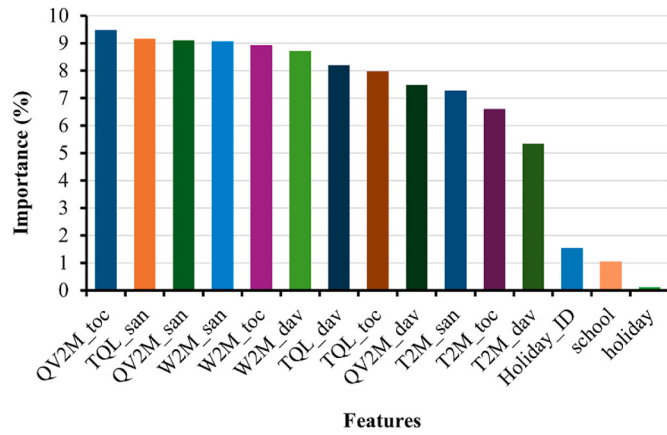


Fig. 12. Feature importance analysis without feature engineering.

The least favorable performance, observed in the first week of June 2020, presented an RMSE of 48.69, an MAE of 35.08, a MAPE of 0.03, and an R^2 of 0.74. This worst-case scenario reveals a slight discrepancy, where the predicted graph marginally fails to capture key change points. Fig. 13(b) highlights the model's average performance, recorded in December 2019, with an RMSE of 30.24, an MAE of 23.13, a MAPE of 0.017, and an R^2 of 0.97. While there is some misalignment between the predicted and actual curves, the performance remains commendable. Fig. 13(c) shows the exceptional performance of the model during the first week of February 2019, with an RMSE of 17.9, an MAE of 14.31, a MAPE of 0.012, and an R^2 of 0.99. This ideal scenario demonstrates a near-perfect alignment of the predicted graph with the actual demand curve across the majority of data points. The proposed model effectively captures both seasonal trends and sudden fluctuations in demand, as evidenced by its strong alignment with the actual load across various segments. Notably, Fig. 13 shows a significant improvement in prediction performance after applying feature engineering not only for the best case but also for both the whole test dataset and the worst case.

To further validate the reliability and generalizability of the proposed model, short-term load forecasting was performed on unseen data across different time frames. Following model training, predictions for load forecasting 1 h ahead were conducted at various time points spanning from January 2019 to June 2020. The training and testing processes were conducted within a consistent environment. Fig. 14 shows that the proposed Pr-XGB-CB model outperforms all the other models in terms of load prediction for Panama City during the initial 100 h of January 2019. The forecast values closely followed the actual load values, showing very minimal deviations and outperforming the individual models.

To understand the role of individual features and the importance of derived features in model predictions, Table 6 presents the percentage of feature importance across distinct features utilized in this study from multiple models. The feature importance analysis revealed several interesting patterns. LSTM and GRU exhibited balanced importance across features, relying more on temporal demand indicators such as `nat_demand_ma_mean24` and `nat_demand_ewm_mean12`, which reflect historical trends and smoothing. In contrast, XGBoost and LightGBM

emphasized features such as `nat_demand_lag_24` and `T2M_san_min-max24`, indicating reliance on past demand and temperature extremes.

Features such as `nat_demand_lag_12`, `T2M_san_skew12`, and `T2M_toc_skew12` appeared to have relatively low importance across all the models, indicating that they contributed to a lesser extent to the predictive performance, possibly due to weaker relationships with the target variable. The XGBoost model showed near-constant and consistent importance across a variety of features (ranging from 2.46 % to 0.12 %), indicating that it was predicted based on a comprehensive set of inputs rather than relying heavily on a few key variables, resulting in more robust predictions.

Since the Prophet model does not directly provide feature importance, the Mean Absolute Effect was used to assess the regressor impact. This is valuable because it quantifies the average magnitude of a feature's effect on the forecast, offering an interpretable measure of feature influence regardless of direction. The equation for the Mean Absolute Effect of each regressor feature in a Prophet forecast is as follows:

$$Effect_j = \frac{1}{n} \sum_{i=1}^n |y_{ij}| \quad (30)$$

where j refers to the j^{th} is the regressor feature, n is the number of forecasted time points, y_{ij} is the predicted effect (or contribution) of the j^{th} regressor at the i^{th} time point, which is taken from the prophet forecast data frame, and $|y_{ij}|$ is the absolute value of the predicted effect for each time point i .

Fig. 15 shows the top 10 influential features, with `nat_demand_ewm_mean12` (1093.4) having the greatest effect. This feature influences the predicted electricity demand by ± 1093.4 MW on average. Other key contributors include `nat_demand_ewm_mean24` (539.31) and `nat_demand_ma_mean12` (351.00), reflecting the importance of recent 12–24-h load demand trends in capturing short-term fluctuations. Environmental features such as `QV2M_san_ewm_mean12` (130.60), `QV2M_toc_ewm_mean12` (70.06), and `T2M_san_ewm_mean12` (65.25) also contribute meaningfully. These derived features enable the model to adapt to recent changes in demand and weather, enhancing prediction accuracy. Additionally, Fig. A1 (Appendix) shows the decomposition of the time series isolating trend, weekly, yearly, and hourly seasonality, and noise. The stable trend (1160–1180) and pronounced seasonal peaks—weekly (+20 on Wednesday, −20 on Sunday), yearly (0–5 in March, June, September), and hourly (up to 100 at 18:00)—highlight the temporal features most influential to the model. Noise (−500 to +500) captures randomness with limited predictive value. This decomposition reveals the driving features for the model's predictions, guiding refinement by emphasizing impactful components and reducing overfitting. For both XGBoost and Prophet, all the top features are derived features from the proposed feature engineering, which demonstrates the effectiveness of the feature transformation process in capturing important patterns and relationships within the data.

To assess the robustness of the proposed model, the test set was divided into three parts, where each part consisted of data from six consecutive months. As depicted in Table 7, although the proposed model performed identically in terms of the measured metrics in the first two parts, the performance declined in the last six months due to a sudden drop in electric load, as shown in Fig. 13(a), leading to slight

Table 5
Evaluation metrics by different models with feature engineering for whole test dataset.

Model	RMSE	MAE	MAPE	R^2	EV	ME	MPD	MGD	MTD
LSTM	44.53	32.26	0.027	0.94	0.95	788.27	1.74	0.002	1983.23
GRU	79.21	63.26	0.053	0.82	0.86	924.1	5.23	0.004	6273.79
LightGBM	52.42	34.197	0.0303	0.922	0.92	1237.43	2.42	0.002	2748.25
XGBoost	50.74	32.18	0.028	0.927	0.93	1208.75	2.28	0.002	2574.66
Prophet	41.72	29.99	0.026	0.95	0.95	622.23	1.61	0.0017	1746.91
Pr-XGB-CB	32.32	23.70	0.0203	0.97	0.97	223.83	0.905	0.00084	1044.99

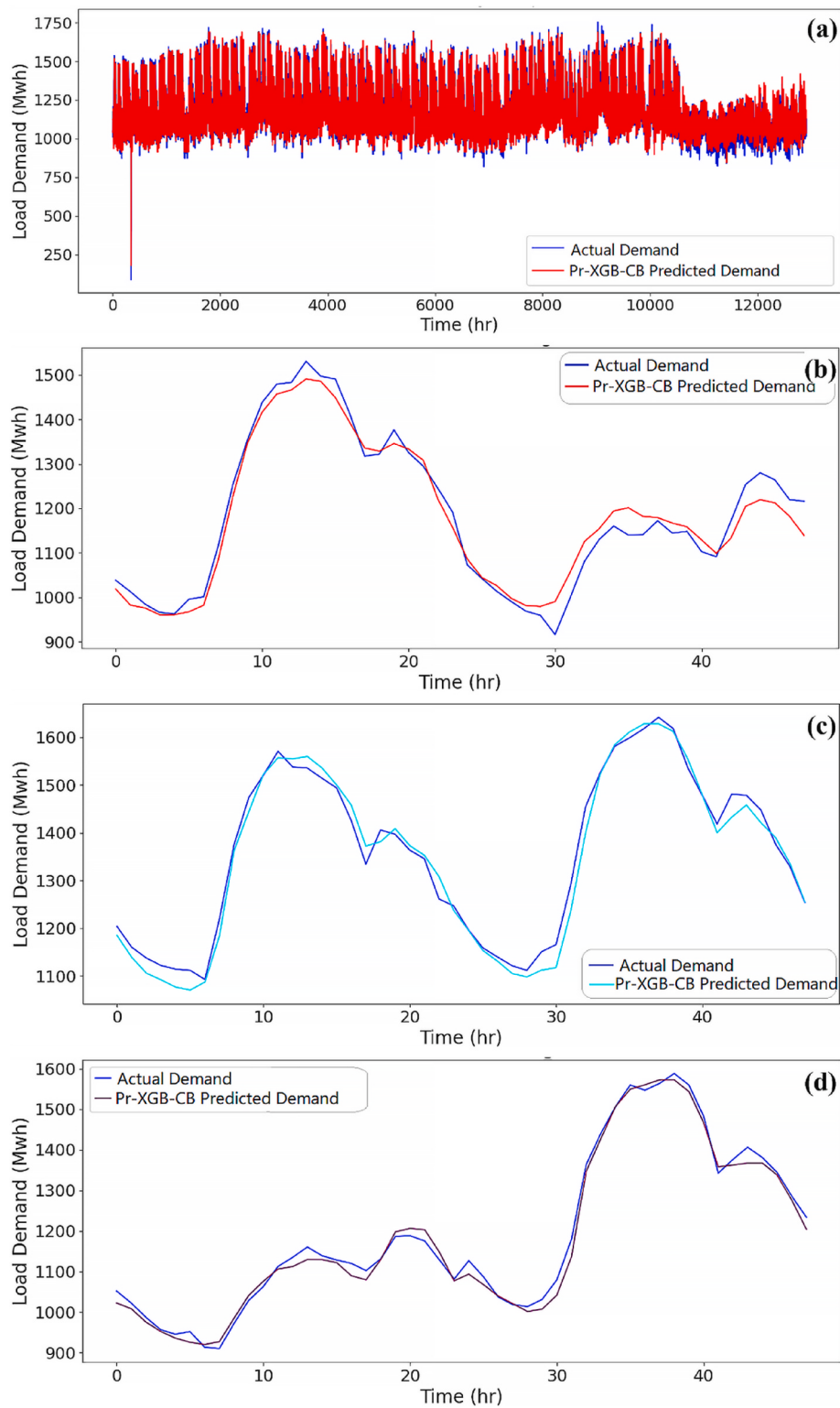


Fig. 13. Actual vs. predicted outputs of the proposed model after feature engineering — showcasing the enhanced accuracy and alignment achieved through the derived features. (a) Full test dataset, (b) Worst 48 h (6 June 2020 to 7 June 2020) (c) Average 48 h (4 December 2019 to 5 December 2019) and (d) Best 48 h (3 February 2019 to 4 February 2019).

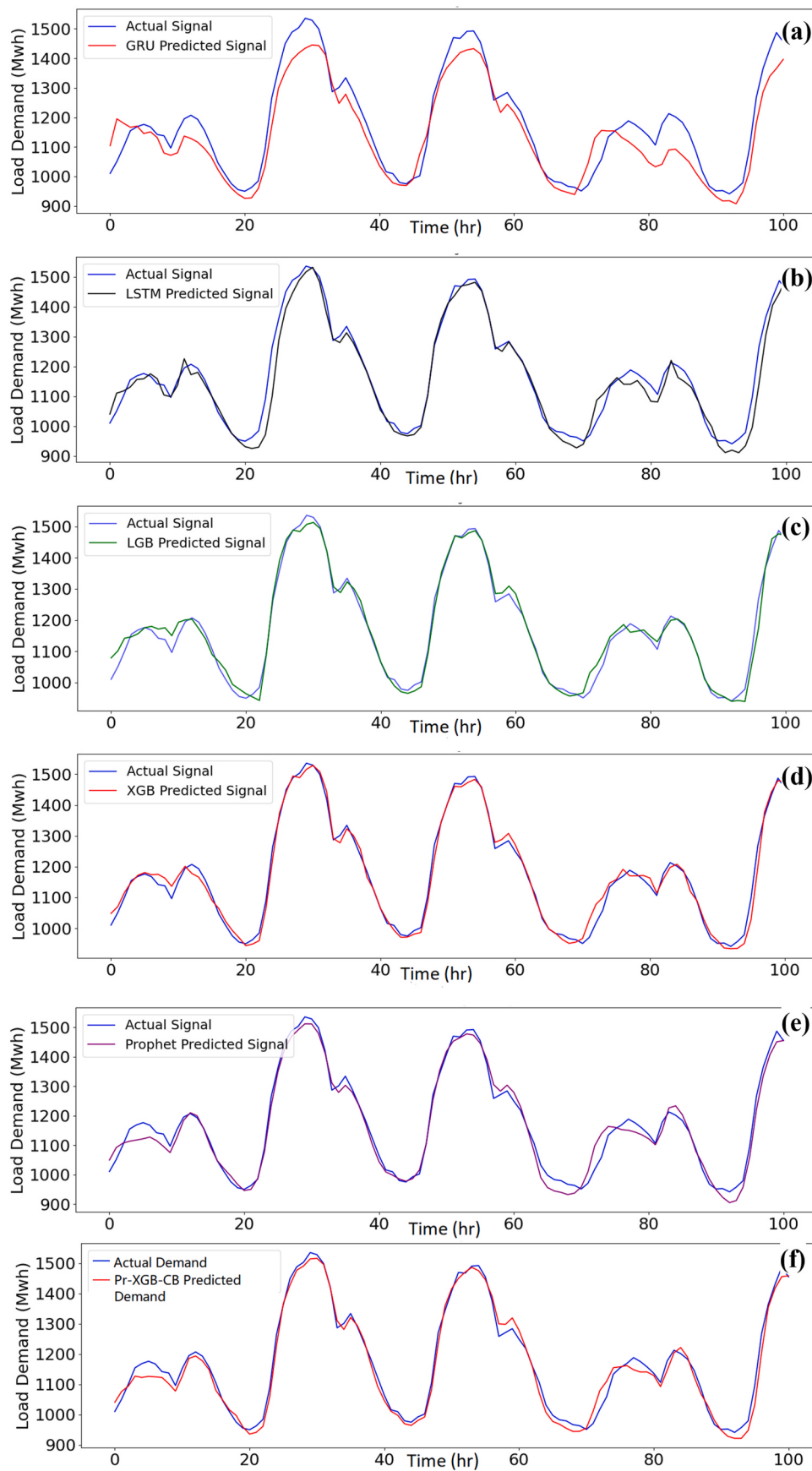


Fig. 14. Comparison of actual vs. predicted demand curves from January 6, 2019, 08:00 to January 10, 2019, at 11:00 (100 h) using six models: (a) GRU, (b) LSTM, (c) LightGBM, (d) XGBoost, (e) Prophet, and (f) the proposed Pr-XGB-CB model—highlighting enhanced accuracy and trend capture by the proposed approach.

Table 6

Feature importance of different models after feature engineering: highlighting the impact of derived features.

Feature Name	LSTM	GRU	XGBoost	LightGBM
T2M_dav_min_max24	0.56 %	0.2 %	0.56 %	28.02 %
nat_demand_std_std12	2.78 %	2.17 %	2.63 %	20.74 %
nat_demand_lag_24	0.28 %	0.37 %	1.34 %	12.62 %
T2M_san_min_max24	0.42 %	0.17 %	0.43 %	11.27 %
T2M_toc_min_max24	0.42 %	0.2 %	0.48 %	9.35 %
nat_demand_ma_mean24	3.28 %	2.46 %	1.1 %	3.77 %
nat_demand_ewm_mean12	1.28 %	1.28 %	0.53 %	3.28 %
nat_demand_ma_mean12	3.03 %	2.16 %	1.93 %	2.81 %
nat_demand_lag_12	0.56 %	0.21 %	1.73 %	1.26 %
nat_demand_ewm_std12	0.62 %	0.27 %	1.31 %	1.08 %
T2M_dav_min_max12	1.2 %	0.19 %	0.37 %	1.32 %
nat_demand_std_std24	1.1 %	0.56 %	1.33 %	0.39 %
T2M_toc_skew12	0.31 %	0.45 %	0.6 %	0.65 %
T2M_san_skew12	0.46 %	0.39 %	0.55 %	0.43 %
nat_demand_ma_mean128	1.01 %	1.18 %	0.79 %	0.174 %

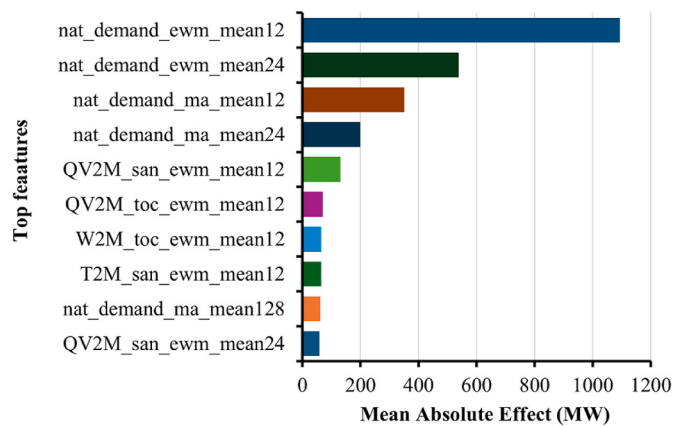


Fig. 15. Top 10 effective regressors of prophet model using Mean Absolute Effect.

deterioration during this period. The average error metrics for these periods closely align with the values for the entire dataset (Table 5), with the RMSE showing a change of 0.28 %, the MAE showing a change of −0.59 %, the MAPE showing a change of 1.48 %, and the R^2 remaining unchanged, which supported the robustness of the proposed model.

4.4. Sensitivity analysis

In practical power system operations, missing data and inaccuracies frequently occur due to sensor faults, communication errors, or measurement delays, and these issues can critically affect the reliability of forecasting. Thus, the accuracy of forecasting is expected to be influenced by the reliability and accuracy of the input data [35]. To prevent unexpected or unintentional outcomes, a simulation was conducted repeatedly by removing 0 %, 10 %, 20 %, 30 %, 40 %, and 50 % of the dataset. All the comparative forecasting models were employed to generate predictions, allowing for a comparison of the prediction errors. As indicated in Table 8, the proposed model consistently demonstrated

Table 7

Evaluation metrics by the proposed model for each testing specific timestamp.

Time	RMSE	MAE	R^2	EV	ME	MPD	MGD	MTD
01/01/2019 - 30/06/2019	28.17	20.78	0.98	0.978	210.65	0.682	0.00078	793.46
01/07/2019 - 31/12/2019	29.52	22.47	0.98	0.976	193.49	0.714	0.0005	871.39
01/01/2020 - 27/06/2020	38.99	28.26	0.95	0.953	189.75	1.366	0.0012	1520.88
Average	32.23	23.84	0.97	0.969	197.96	0.92067	0.000827	1061.91

Table 8

Model performance under data removal: RMSE score analysis.

Data Removal (%)	RMSE score for different models					
	GRU	LSTM	LightGBM	XGBoost	Prophet	Pr-XGB-CB
10	67.87	90.51	53.42	49.44	42.11	32.19
20	65.70	82.34	52.90	50.89	40.74	32.33
30	85.33	96.48	53.90	51.51	41.02	32.89
40	81.78	86.54	53.81	48.55	43.48	32.92
50	86.69	92.24	49.15	55.35	40.33	33.06
0	79.21	44.53	52.42	50.74	41.72	32.32
Standard deviation	8.93	19.03	1.78	2.35	1.14	0.38

the smallest RMSE and the narrowest range of variation (only 0.38 standard deviation), regardless of the dataset's completeness. Therefore, the proposed model exhibited great robustness, and its forecasting accuracy was not significantly affected by variations in data quality. These results affirmed that the proposed model not only achieved superior accuracy but also maintained consistent performance under challenging data conditions, making it a reliable choice for deployment in real-life operational environments.

4.5. Discussion

4.5.1. Comparison with literature

A comparison of the proposed Pr-XGB-CB model with SOTA models on the same dataset is presented in Table 9. An existing XGBoost model by Madrid et al. [16] reported an RMSE of 44.52 and a MAPE of 3.66. In contrast, the proposed model attained an RMSE of 32.32 and a MAPE of 0.0203, indicating improvements of 27 % and 99.44 %, respectively. Although they employed feature engineering, the multi-lag feature approach and extensive statistical analysis in the current study contributed significantly to its superior performance. The proposed Pr-XGB-CB model also demonstrated substantial performance improvements, such as a reduction in the RMSE by 27.59 % compared with the feature extraction-based secondary VMD model [36] and by 27.61 % compared with the EMD hybrid model [37]. Furthermore, it achieved the lowest MAE (23.70), outperforming the 1D-CNN-GRU (82.44) [38] and DIFM (43.12) models by 71.26 % and 45.02 %, respectively. In terms of the MAPE, the current model significantly lowered the error to 0.0203, compared with 2.88 for VMD and 3.84 for DIFM [39]. Additionally, the proposed model achieved a high R^2 score of 0.97, matching

Table 9

Comparison of the proposed Pr-XGB-CB model with SOTA forecasting methods.

Models	Evaluation Metrics				Reference
	RMSE	MAE	MAPE	R^2	
XGB (2021)	44.52	–	3.66	–	[16]
1D-CNN-GRU (2024)	177.88	82.44	–	–	[38]
EMD hybrid model (2024)	39.22	32.742	–	0.96	[37]
Diffusion-based Inpainting	–	43.12	3.84	–	[39]
Forecasting Method (DIFM) (2025)					
Feature extraction based	59.39	32.29	2.88	–	[36]
Secondary VMD (2025)					
Proposed Pr-XGB-CB	32.32	23.70	0.0203	0.97	–

or exceeding the best-performing models, highlighting its superior predictive capability [36,37].

In addition to standard evaluation metrics such as the MAE, MSE, and R^2 , this study introduces additional practical statistical indicators—Mean Poisson Deviance, Mean Gamma Deviance, and Mean Tweedie Deviance, Explained Variance, and Max Error as practical and useful tool for power system monitoring and decision-making.

4.5.2. Strength of the proposed approach

This research overcomes the key limitations of prior VSTLF models by introducing a hybrid Pr-XGB-CB framework that integrates multi-lag features to improve adaptability beyond XGBoost's constraints [11]. It employs robust statistical feature engineering to better manage temporal dependencies, addressing the sensitivities seen in the GRU [12] and R-CNN-ML-LSTM [22]. Additionally, its advanced feature selection surpasses the TCN-Prophet model's challenges in capturing long-term dependencies [21], whereas sensitivity analysis confirms resilience to missing data, mitigating computational issues faced by the CEEMDAN-SE-TR transformer [15].

The key strength of the Pr-XGB-CB model lies in its hybrid structure, which effectively leverages the complementary capabilities of Prophet, XGBoost, and CatBoost. Prophet efficiently captures seasonal and trend patterns, XGBoost excels in identifying complex nonlinear relationships, and CatBoost strategically integrates these diverse predictions to optimize overall accuracy.

The comprehensive feature engineering strategy significantly enhances the model's predictive capability by employing various statistical methods, including rolling averages, exponential moving statistics, normalization, and skewness calculations across different temporal windows. This robust feature engineering not only improves accuracy but also increases the model's adaptability to temporal fluctuations and sudden changes in electricity consumption patterns.

Additionally, the proposed model also exhibits impressive resilience in scenarios involving incomplete or disrupted data, as validated by comprehensive sensitivity analyses. These features position the Pr-XGB-CB model as an innovative, efficient, adaptable, accurate and practical VSTLF solution for real-world power system forecasting applications.

4.5.3. Practical significance in power system operations

The proposed forecasting framework enhances power system operations by delivering accurate, adaptive, and efficient load predictions. By employing lightweight, modular models, the proposed approach enables real-time scalability. Validated on public datasets, the framework offered a practical and resilient solution for modern smart grids.

By enabling more accurate and timely forecasts of load and generation patterns, the feature engineering approach facilitates optimized resource scheduling and reduces reliance on expensive reserve capacity, thereby lowering operational costs and improving overall economic efficiency in the power system [40]. The sensitivity analysis validated the framework's robustness against data sparsity and quality degradation, conditions frequently encountered in practical power system deployments such as sensor failures or communication outages. This resilience ensures sustained forecasting accuracy for critical applications such as real-time load balancing and contingency management, thereby enhancing system stability and operational reliability under real-world uncertainties [41]. Collectively, these attributes contributed to improved grid stability, efficient economic dispatch, and effective integration of variable renewable energy sources, aligning closely with the operational and economic challenges faced by contemporary power systems.

4.5.4. Limitations and future work

The proposed framework performed well in the VSTLF scenario but still has room for further improvement. Although it performs strongly on the Panama-specific dataset, its representativeness is limited by geographic, climatic, and demographic factors, potentially affecting its generalizability to other regions. Future work should focus on validating and retraining the model via diverse datasets from various regions and conditions, along with exploring transfer learning to increase its applicability. Additionally, expanding evaluations beyond very short-term horizons to capture seasonal and long-term patterns will provide a more comprehensive understanding of the model's capabilities. Beyond traditional approaches, delving into the potential of advanced architectures such as transformers or large language models could unlock further accuracy gains and potentially capture intricate temporal dependencies within the data. Prioritizing real-time deployment through cloud-based implementations and explainable AI techniques such as SHAP and LIME is crucial for practical application.

By addressing these limitations and integrating advanced techniques, future work can enhance the reliability, interpretability, and operational efficiency of VSTLF systems. The current framework has established a strong foundation, positioning it as a promising step toward smarter, more efficient, and sustainable energy management solutions.

5. Conclusions

This paper introduces a novel Pr-XGB-CB model that combines the Prophet, XGBoost and CatBoost models to predict very short-term electricity load consumption. This approach utilizes the strengths of each model, incorporating robust feature engineering to increase the forecasting accuracy. The model's effectiveness is tested by applying it to historical electricity load data from Panama City, validating its performance in load forecasting. According to the findings of this study, the following conclusions can be drawn.

1. The number of input features was expanded from 17 to 390 by employing multi-lagged statistical feature engineering for performance enhancement in electricity load forecasting, which results in superior model performance.
2. The proposed hybrid model produced minimum error metric values with RMSE = 32.32, MAE = 23.70, MAPE = 0.0203, and $R^2 = 0.97$. In addition, various statistical measurements, including Explained Variance = 0.9704, Max Error = 223.83, Mean Poisson Deviance = 0.905, Mean Gamma Deviance = 0.00084, and Mean Tweedie Deviance = 1044.99 were performed. The model showed 26.85 % and 99.44 % better performance in terms of the RMSE and MAE, respectively, with respect to the state-of-the-art model with the same dataset.
3. A thorough analysis was conducted on the hourly forecasting results, the initial week of each testing month, and the 6-month data to increase the robustness of the model. RMSE values of 32.32, 30.03, and 32.23 were found for the hourly, weekly and semiannual load predictions, respectively. The small fluctuations in the RMSE value denoted the robustness of the proposed model.
4. An evaluation of the sensitivity analysis was conducted by randomly including or excluding test data. The highest percentage change from the original RMSE value of 32.32 was 2.281 % (corresponding to 33.06), and the lowest percentage change was -0.402 % (corresponding to 32.19).

Future research will aim to broaden the applicability of the framework by incorporating adaptive learning strategies and scalable model architectures. Emphasis will be placed on enhancing system robustness, fostering interpretability, and aligning model development with practical deployment requirements in dynamic energy environments.

CRediT authorship contribution statement

Md Shafiuzzaman: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Md Safayet Islam:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **T.M. Rubaith Bashar:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Mohammad Munem:** Writing –

review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Md Nahiduzzaman:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Mominul Ahsan:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Julfikar Haider:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

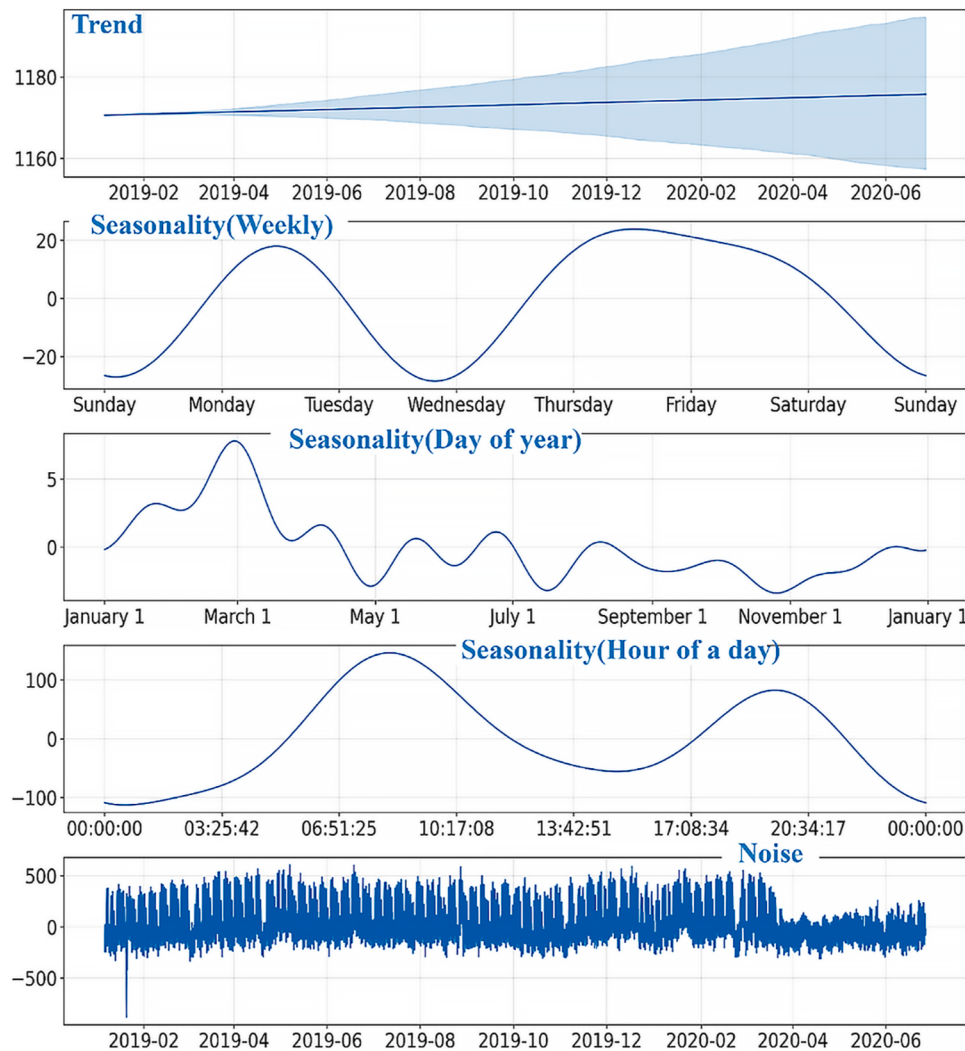


Fig. A1. Decomposition of the dataset into different components with the Prophet model

Table A1
Evaluation metrics of different models for every testing week

Model	Year	2019												2020						Average
		Month	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.	May	June
LSTM	RMSE	41.83	28.44	28.39	34.98	68.01	34.6	27.67	39.63	31.36	27.06	26.94	53.71	31.74	32.99	53.76	51.29	51.37	38.35	39.01
	MAE	27.19	21.26	21.62	26.5	55.14	26.42	21.83	30.11	22.79	20.65	20.33	42.1	22.46	25.5	42.37	37.83	38.74	29.81	29.59
	MAPE	0.022	0.018	0.019	0.2	0.04	0.021	0.017	0.023	0.018	0.017	0.017	0.032	0.019	0.02	0.033	0.036	0.037	0.027	0.034
	R ²	0.94	0.97	0.97	0.97	0.87	0.96	0.98	0.95	0.97	0.98	0.98	0.91	0.97	0.97	0.93	0.68	0.71	0.86	0.92
GRU	RMSE	61.19	46.89	55.0	75.42	106.6	81.07	71.01	83.37	78.41	58.98	86.17	80.98	61.77	66.57	85.53	100.9	96.98	65.39	75.68
	MAE	52.16	34.56	46.11	67.27	96.2	68.9	59.05	70.95	65.65	45.2	66.95	65.17	49.08	55.81	71.71	76.42	76.76	51.28	62.18
	MAPE	0.04	0.03	0.04	0.05	0.07	0.05	0.04	0.05	0.05	0.04	0.05	0.05	0.04	0.04	0.05	0.72	0.072	0.04	0.08
	R ²	0.87	0.93	0.89	0.85	0.69	0.81	0.85	0.79	0.81	0.91	0.77	0.79	0.88	0.87	0.81	−0.24	−0.01	0.59	0.71
Light GBM	RMSE	25.3	20.8	36.4	28.5	89.6	33.7	37.5	32.3	25.5	33.8	32.6	59.04	34.41	28.44	62.71	56.89	69.87	58.48	42.55
	MAE	19.8	15.35	27.89	22.32	69.59	27.46	26.83	24.37	20.31	24.63	25.03	46.18	25.68	22.51	43.52	38.77	46.9	44.22	31.74
	MAPE	0.01	0.01	0.02	0.02	0.05	0.02	0.02	0.02	0.016	0.02	0.02	0.03	0.022	0.018	0.034	0.037	0.04	0.04	0.02
	R ²	0.97	0.98	0.95	0.97	0.77	0.96	0.95	0.97	0.98	0.97	0.96	0.89	0.96	0.98	0.89	0.61	0.47	0.67	0.88
XGBoost	RMSE	19.46	18.60	35.92	24.66	84.56	31.25	29.73	31.33	25.08	29.03	38.87	59.94	31.68	25.01	58.18	54.47	69.64	57.67	40.28
	MAE	16.52	13.86	26.93	19.92	66.77	24.69	22.18	22.8	19.06	21.40	25.36	47.02	23.42	20.17	39.51	36.51	45.32	43.18	29.70
	MAPE	0.014	0.012	0.02	0.015	0.048	0.019	0.018	0.018	0.015	0.018	0.021	0.034	0.02	0.016	0.03	0.035	0.043	0.039	0.024
	R ²	0.98	0.99	0.95	0.98	0.80	0.97	0.97	0.97	0.98	0.97	0.95	0.88	0.97	0.98	0.91	0.64	0.48	0.68	0.892
Prophet	RMSE	24.08	22.39	36.73	24.41	41.15	28.07	37.96	29.4	27.75	28.19	39.19	26.7	40.48	25.73	34.62	67.9	73.53	53.76	36.78
	MAE	19.55	16.97	27.10	18.92	34.32	22.28	30.32	23.32	21.46	23.65	30.33	20.83	32.91	19.29	25.99	53.6	61.04	45.14	29.28
	MAPE	0.016	0.014	0.024	0.015	0.026	0.019	0.025	0.018	0.018	0.02	0.027	0.016	0.029	0.016	0.02	0.05	0.057	0.04	0.025
	R ²	0.98	0.98	0.95	0.98	0.95	0.98	0.95	0.97	0.97	0.98	0.95	0.97	0.95	0.98	0.97	0.43	0.42	0.72	0.89
Pr- XGB-CB	RMSE	22.47	17.9	28.22	20.05	42.79	23.65	28.41	25.98	24.0	24.95	27.73	30.24	31.59	20.55	40.94	43.35	48.69	39.02	30.03
	MAE	18.91	14.31	22.21	15.95	34.38	18.62	21.99	19.86	19.2	20.13	21.38	23.13	24.29	15.44	28.22	30.69	35.08	30.81	23.03
	MAPE	0.016	0.012	0.019	0.012	0.026	0.015	0.018	0.016	0.015	0.017	0.019	0.017	0.021	0.012	0.02	0.03	0.03	0.03	0.019
	R ²	0.98	0.99	0.97	0.99	0.95	0.98	0.97	0.98	0.98	0.98	0.97	0.97	0.97	0.99	0.96	0.77	0.74	0.85	0.94

Data availability

Data will be made available on request.

References

[1] Diamantoulakis PD, Kapinas VM, Karagiannidis GK. Big data analytics for dynamic energy management in smart grids. *Big Data Research Sep.* 2015;2(3):94–101. <https://doi.org/10.1016/J.BDR.2015.03.003>.

[2] Gomez W, Wang FK, Lo SC. A hybrid approach based machine learning models in electricity markets. *Energy Feb.* 2024;289:129988. <https://doi.org/10.1016/J.ENERGY.2023.129988>.

[3] Ghasempour A. Support vector regression to predict power consumption [Online]. Available: https://digitalrepository.unm.edu/ece_rpts. [Accessed 31 May 2025].

[4] Seifi H, Sepasian MS. Electric power system planning: issues, algorithms and solutions. *Power Systems* 2011;49. <https://doi.org/10.1007/978-3-642-17989-1/COVER>.

[5] Munem M, Rubaith Bashar TM, Roni MH, Shahriar M, Shawkat TB, Rahaman H. Electric power load forecasting based on multivariate LSTM neural network using bayesian optimization. In: 2020 IEEE electric power and energy conference, EPEC 2020; Nov. 2020. <https://doi.org/10.1109/EPEC48502.2020.9320123>.

[6] Pappas SS, Ekonomou I, Karamousantas DC, Chatzarakis GE, Katsikas SK, Liatsis P. Electricity demand loads modeling using Autoregressive moving average (ARMA) models. *Energy Sep.* 2008;33(9):1353–60. <https://doi.org/10.1016/J.ENERGY.2008.05.008>.

[7] Hsu CC, Chen CY. Applications of improved gray prediction model for power demand forecasting. *Energy Convers Manag Aug.* 2003;44(14):2241–9. [https://doi.org/10.1016/S0196-8904\(02\)00248-0](https://doi.org/10.1016/S0196-8904(02)00248-0).

[8] Cerne G, Dovzan D, Skrljanc I. Short-term load forecasting by separating daily profiles and using a single fuzzy model across the entire domain. *IEEE Trans Ind Electron Sep.* 2018;65(9):7406–15. <https://doi.org/10.1109/TIE.2018.2795555>.

[9] Alsharekh MF, Habib S, Dewi DA, Albattah W, Islam M, Albahli S. Improving the efficiency of multistep short-term electricity load forecasting via R-CNN with ML-LSTM. *Sensors Sep.* 2022;22(18):6913. <https://doi.org/10.3390/S22186913>.

[10] Ling SH, Leung FHF, Lam HK, Tam PKS. Short-term electric load forecasting based on a neural fuzzy network. *IEEE Trans Ind Electron Dec.* 2003;50(6):1305–16. <https://doi.org/10.1109/TIE.2003.819572>.

[11] Jiang H, Zhang Y, Muljadi E, Zhang JJ, Gao DW. A short-term and high-resolution distribution system load forecasting approach using support vector regression with hybrid parameters optimization. *IEEE Trans Smart Grid Jul.* 2018;9(4):3331–50. <https://doi.org/10.1109/TSG.2016.2628061>.

[12] Cecati C, Kolbusz J, Rózycki P, Siano P, Wilamowski BM. A novel RBF training algorithm for short-term electric load forecasting and comparative studies. *IEEE Trans Ind Electron Oct.* 2015;62(10):6519–29. <https://doi.org/10.1109/TIE.2015.2424399>.

[13] Tsalikidis N, Mystakidis A, Tjortjis C, Koukaras P, Ioannidis D. Energy load forecasting: one-step ahead hybrid model utilizing ensembling. *Computing Jan.* 2024;106(1):241–73. <https://doi.org/10.1007/s00607-023-01217-2>.

[14] Yamasaki M, Freire RZ, Seman LO, Stefenon SF, Mariani VC, dos Santos Coelho L. Optimized hybrid ensemble learning approaches applied to very short-term load forecasting. *Int J Electr Power Energy Syst Jan.* 2024;155. <https://doi.org/10.1016/j.ijepes.2023.109579>.

[15] Bae DJ, Kwon BS, Bin Song K. XGBoost-Based day-ahead load forecasting algorithm considering behind-the-meter solar PV generation. *Energies Dec.* 2021;15(1):128. <https://doi.org/10.3390/EN15010128>. 2022, Vol. 15, Page 128.

[16] Madrid EA, Antonio N. Short-term electricity load forecasting with machine learning. *Information Jan.* 2021;12(2):50. <https://doi.org/10.3390/INFO12020050>. 2021, Vol. 12, Page 50.

[17] Massaoudi M, Refaat SS, Chihi I, Trabelsi M, Oueslati FS, Abu-Rub H. A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting. *Energy Jan.* 2021;214:118874. <https://doi.org/10.1016/J.ENERGY.2020.118874>.

[18] Wan Z, Li H. Short-term power load forecasting based on feature filtering and error compensation under imbalanced samples. *Energies May* 2023;16(10):4130. <https://doi.org/10.3390/EN16104130>. 2023, Vol. 16, Page 4130.

[19] Wang Y, et al. Short-term load forecasting of industrial customers based on SVM and XGBoost. *Int J Electr Power Energy Syst Jul.* 2021;129:106830. <https://doi.org/10.1016/J.IJEPES.2021.106830>.

[20] Jiang Z, Zhang L, Ji T. NSDAR: a neural network-based model for similar day screening and electric load forecasting. *Appl Energy Nov.* 2023;349:121647. <https://doi.org/10.1016/J.APENERGY.2023.121647>.

[21] Mo J, Wang R, Cao M, Yang K, Yang X, Zhang T. A hybrid temporal convolutional network and prophet model for power load forecasting. *Complex and Intelligent Systems Aug.* 2023;9(4):4249–61. <https://doi.org/10.1007/S40747-022-00952-X/FIGURES/7>.

[22] Shohan MJA, Faruque MO, Foo SY. Forecasting of electric load using a hybrid LSTM-neural prophet model. *Energies Mar.* 2022;15(6):2158. <https://doi.org/10.3390/EN15062158>. 2022, Vol. 15, Page 2158.

[23] Bashir T, Haoyong C, Tahir MF, Liqiang Z. Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN. *Energy Rep Nov.* 2022;8: 1678–86. <https://doi.org/10.1016/J.EGYR.2021.12.067>.

[24] Ran P, Dong K, Liu X, Wang J. Short-term load forecasting based on CEEMDAN and Transformer. *Elec Power Syst Res Jan.* 2023;214:108885. <https://doi.org/10.1016/J.EPSR.2022.108885>.

[25] Electricity load forecasting [Online]. Available: <https://www.kaggle.com/dataset/saurabhshahane/electricity-load-forecasting>. [Accessed 25 November 2023].

[26] Mathelinea D, Chandrashekar R, Mawengkang H. Stationarity test for medicine time series data. *AIP Conf Proc May* 2023;2714(1). <https://doi.org/10.1063/5.0128444/2889789>.

[27] Rosol M, Młyńczak M, Cybulski G. Granger causality test with neural-network-based methods: python package and simulation study. *Comput Methods Progr Biomed Apr.* 2022;216:106669. <https://doi.org/10.1016/J.CMPB.2022.106669>.

- [28] Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput Jul.* 2019;31(7):1235–70. https://doi.org/10.1162/NECO_A_01199.
- [29] Dey R, Salemt FM. Gate-variants of gated recurrent unit (GRU) neural networks. In: *Midwest symposium on circuits and systems*; Sep. 2017. p. 1597–600. <https://doi.org/10.1109/MWSCAS.2017.8053243>. vol. 2017-August.
- [30] Ke G, et al. LightGBM: a highly efficient gradient boosting decision tree. *Adv Neural Inf Process Syst* 2017;30 [Online]. Available: <https://github.com/Microsoft/LightGBM>. [Accessed 25 November 2023].
- [31] Sagi O, Rokach L. Approximating XGBoost with an interpretable decision tree. *Inf Sci Sep.* 2021;572:522–42. <https://doi.org/10.1016/j.ins.2021.05.055>.
- [32] G R H, Sreedharan S, N BC. Advanced short-term load forecasting for residential demand response: an XGBoost-ANN ensemble approach. *Elec Power Syst Res May* 2025;242:111476. <https://doi.org/10.1016/j.epsr.2025.111476>.
- [33] Hancock JT, Khoshgoftaar TM. CatBoost for big data: an interdisciplinary review. *J Big Data Dec.* 2020;7(1):1–45. <https://doi.org/10.1186/S40537-020-00369-8/FIGURES/9>.
- [34] Srinivas P, Katarya R. hyOPTXg: OPTUNA hyperparameter optimization framework for predicting cardiovascular disease using XGBoost. *Biomed Signal Process Control Mar.* 2022;73:103456. <https://doi.org/10.1016/j.bspc.2021.103456>.
- [35] Shakeel A, Chong D, Wang J. District heating load forecasting with a hybrid model based on LightGBM and FB-prophet. *J Clean Prod Jul.* 2023;409:137130. <https://doi.org/10.1016/j.jclepro.2023.137130>.
- [36] Zhang M, Xiao G, Lu J, Liu Y, Chen H, Yang N. Robust load feature extraction based secondary VMD novel short-term load demand forecasting framework. *Elec Power Syst Res Feb.* 2025;239:111198. <https://doi.org/10.1016/j.epsr.2024.111198>.
- [37] Yin C, Wei N, Wu J, Ruan C, Luo X, Zeng F. An empirical mode decomposition-based hybrid model for sub-hourly load forecasting. *Energies Jan.* 2024;17(2):307. <https://doi.org/10.3390/EN17020307>. 2024, Vol. 17, Page 307.
- [38] Sarker MAA, Shanmugam B, Azam S, Thennadil S. Enhancing smart grid load forecasting: an attention-based deep learning model integrated with federated learning and XAI for security and interpretability. *Intelligent Systems with Applications Sep.* 2024;23:200422. <https://doi.org/10.1016/j.iswa.2024.200422>.
- [39] Zhang L, Jiang Z, Ji T, Chen Z. Diffusion-based inpainting approach for multifunctional short-term load forecasting. *Appl Energy Jan.* 2025;377:124442. <https://doi.org/10.1016/j.apenergy.2024.124442>.
- [40] Shobeiry SM. AI-Enabled modern power systems: challenges, solutions, and recommendations. In: *Power systems*; 2024. p. 19–67. https://doi.org/10.1007/978-3-031-69358-8_2. vol. Part F3666.
- [41] Psomopoulos S, et al. Deterioration of electrical load forecasting models in a smart grid environment. *Sensors Jun.* 2022;22(12):4363. <https://doi.org/10.3390/S22124363>. 2022, Vol. 22, Page 4363.