Enhanced Short-Term Load Forecasting with Multi-Lag Feature Engineering and Prophet-XGB-CatBoost Architecture

Md Shafiuzzaman[1], Md Safayet Islam[1], T. M. Rubayet Bashar[1], Mohammad Munem[1], Md Nahiduzzaman[1], Mominul Ahsan[2], Julfikar Haider[3,*]

[1]*Department of Electrical & Computer Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh, e-mail: shafiuzzaman.ruet@gmail.com; ornob39@gmail.com; rubaithbashar@gmail.com; Munem.mh@gmail.com; nahiduzzaman@ece.ruet.ac.bd*

[2]*Department of Computer Science, University of York, Deramore Lane, Heslington, York YO10 5GH, UK, e-mail: mominul.ahsan2@gmail.com*

[3]*Department of Engineering, Manchester Metropolitan University, Chester Street, Manchester M1 5GD, UK, e-mail: j.haider@mmu.ac.uk*

***Corresponding author:**

Julfikar Haider

Department of Engineering,

Manchester Metropolitan University,

Chester Street, Manchester

M1 5GD, UK

e-mail: j.haider@mmu.ac.uk

Abstract

An efficient short-term load forecast (STLF) is essential for power plant unit scheduling because it reduces planning uncertainty brought on by variable renewable energy output, which in turn lowers 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 is still necessary. In this paper, a novel machine learning framework for predicting short-term electricity usage has been proposed. The data utilized in this study is publicly available and was accessed from Kaggle's platform. The framework involved two pivotal stages in its development: robust feature engineering and forecasting electric load by a meta-Face Book Prophet-XGB model. Feature engineering involves utilizing multiple sets of historical electricity consumption data with different time lags for analysis and enhancement. Facebook Prophet dissects observed or forecasted data into understandable seasonal, trend, and holiday segments, while XGBoost stands out for its speed and effectiveness, especially when dealing with numerous features in a given scenario. To get the ensembled forecast Catboost algorithm is utilized. The optimal hyper-parameters for the model are evaluated by Optuna. The effectiveness of the suggested model is evaluated by contrasting its performance with that of 5 alternative machine learning and deep learning algorithms. The proposed framework surpasses the state-of-the-art (SOTA) models with MAE of 23.70, RMSE of 32.32, and R^2 of 0.97; therefore, it can perform an efficient guiding role in electrical load prediction. The research offering practical significance by enhancing power plant scheduling efficiency and reducing overall electricity production costs through superior predictive accuracy.

Keywords: Renewable Energy, short-term load forecasting; machine learning; FB Prophet; XGB; feature engineering.

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 forecasting of power load demand serves as the foundation for effective power management [1]. This is especially significant in many developing nations, where industrial users represent a significant portion of energy consumers. For instance, in China, approximately 70% of energy consumption is attributed to the industrial users [2].. Electricity load forecasting is important for combining renewable power systems by allowing better resource planning, anticipating demand for effective generation, and maintaining grid stability. Precise forecasting of electric load consumption aids in optimizing energy storage, balancing intermittent renewable sources, and guiding market operations for pricing strategies [3]. It influences policy decisions and investments in renewable technologies and infrastructure. With renewables like solar and wind being weather-dependent, load forecasting becomes crucial in predicting when these sources will generate power. Grid managers can anticipate and control the short-term availability of renewable energy sources due to this prediction, which ensures a balanced supply and demand for electricity. Accurate load forecasting improves the integration of renewable energy into the overall energy system by allowing the grid to adapt and use it efficiently.

The planning time span of a power system operation is often divided into three-time frames: short-term, mid-term, and long-term. Each of these time frames focuses on some particular activities [4]. The short-term period spans from a single day to a week and is particularly concerned with the safety and operational aspects of the electricity system [5]. On the other hand, the mid-term time scale generally takes a few weeks to a few months into consideration, for a greater emphasis on controlling the generation assets and preventing energy shortfalls with the current power plants. 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.

Short-term load forecasting plays a pivotal role in efficient and reliable operation of electrical power systems. It is a critical aspect of energy management, enabling utilities, grid operators, and energy providers to make informed decisions regarding the future generation, distribution, and allocation of electricity over a short time horizon, typically ranging from a few hours to a few days [6]. Electric power load varies with time, and it is associated with multiple features [3]. Normally several features are fed into a model to predict electric power consumption. Electric power load a subject of uncertainty and irregular seasonal trends. In addressing the common time series forecasting challenge, various statistical approaches like auto-regressive integrated moving average [7] and grey prediction [8] have been suggested. Nonetheless, these models often prove insufficient in capturing the intricate patterns of power consumption, leading to subpar forecasting accuracy [8, 9]. In recent years, several machine learning (ML) techniques, including artificial neural networks, fuzzy neural networks, and support vector regression, along with some hybrid models [9-12] such as region-based Convolutional Neural Networks (RCNNs), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH), and Asymmetric Power GARCH (APGARCH) [10], have demonstrated successful applications in dealing with the nonlinear aspects of load forecasting. Jiang demonstrated superior forecasting accuracy and computational efficiency of the proposed SVR-based hybrid models [11]. 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-hour electric load forecasting. The ErrCor algorithm demonstrates superior efficiency, achieving a lower MAPE outperforming SVR and ELM methods and contributing valuable insights to short-term load forecasting, crucial for microgrid and smart grid development. As a result, the precision of prediction and model complexity are both continually rising[11]. Accurate STLF aids in mitigating planning uncertainties stemming from intermittent renewable energy production. It subsequently determines optimal costs for hydroelectric plants with reservoirs and facilitates efficient dispatching of thermal power plants by minimizing production-transmission costs [12]. Ultimately, it aims to reduce operational expenses associated with real-time dispatch by optimizing the selection of power plants.

This study primarily concentrates on predicting short-term electricity demand, commonly referred to in research as short-term load forecasting (STLF). Specifically, it addresses the STLF issue concerning the Panama power system, forecasting for a one-week horizon in hourly increments, totaling 168 hours. In this study, a combination of Facebook Prophet and Extreme Gradient Boosting Regressor (XGBoost) with CatBoost has been proposed to predict short-term load forecasting and its performance is compared to various other machine learning algorithms. This paper begins with an introduction, followed by a literature review on short-term load forecasting

(STLF), a section detailing the materials and methods used, results with sensitivity analysis and discussion of the results in the subsequent section. The final section highlights the key conclusions resulted from this study.

2. Literature Review

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

Madrid et. al. suggested a group of models based on machine learning (ML) in order to precisely predict the hourly electric load for the next 168 hours, where the Extreme Gradient Boosting Regressor (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 [13]. In a recent study by Wan et al. (2023), an advanced Gated Recurrent Unit (GRU) model was proposed for short-term load estimation. They integrated feature selection and balanced data through sample extension after load data clustering. Their approach involved error-focused training, leveraging mistake-derived data, and implementing error correction modeling to enhance prediction accuracy [14]. In 2021, a method utilizing XGBoost was developed to counter the impact of behind-the-meter (BTM) solar PV on load prediction. This techniques involved analyzing demand deviation, predicting BTM solar PV capacity through grid search, and factoring in lighting appliance power variation and base temperature for data filtering. The method accurately assessed BTM solar PV generator capacity and restored load, eliminating load distortion by incorporating projected solar PV power using the XGBoost model for load estimation [15]. Wang et al. introduced Sample Entropy Variational mode decomposition (SVMD), an adaptive decomposition method merging VMD and SampEn, dividing raw load data into a trend series and fluctuation sub-series in 2021. Distinct prediction models for each—a linear regression model for trends and XGBoost models for fluctuations, fine-tuned using Bayesian optimization was designed. Considering diverse factors influencing industrial electricity consumption, their approach aimed to boost accuracy. Their method's performance was assessed across various scenarios involving industrial customers in China and Ireland [16]. In 2023, a Transformer model, Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN-SE-TR), integrated sample entropy, and attention mechanisms to tackle long-term memory issues. Comparing various machine learning methods, the CEEMDAN-SE-TR model demonstrated significant superiority, particularly against the conventional models, substantiating its advantages over Empirical Mode Decomposition (EMD) approaches [17].

Jiang et al. introduced Neural-based Similar Days Auto Regression (NSDAR) in 2023, a unique technique employing neural networks for identifying analogous days, departing from conventional methods dependent on meteorological factors or basic guidelines. NSDAR's neural networks perceived numerous unusual similar days, enabling explicable load forecasts by a linear framework based on these identifications. The model's outputs extended utility beyond forecasting, supporting load clustering and regional load analysis [18]. Mo et al. employed a Temporal Convolutional Network (TCN) tailored for time data, surpassing CNN limitations. Leveraging the Prophet model, they partitioned power load data into trend, seasonal, and holiday components. By separately forecasting with TCN and Prophet, then merging the models using the least squares method, they harnessed the strengths of each architecture to enhance prediction accuracy [19]. In 2022, Shohan et al. aimed to enhance electricity load prediction precision. A hybrid LSTM-Neural Prophet model, leveraging historical data and statistical traits for forecasting across various time spans was introduced. Comparisons with established methods confirmed its superior performance [20]. A new method for short-term power forecasting was devised, emphasizing data refinement and a complex Region-Based Convolutional Neural Network or R-CNN combined with ML-LSTM architecture. They preprocessed the IHEC Dataset using data cleansing techniques and applied Box-Cox transformation for power distribution analysis. The subsequent deep R-CNN extracted crucial patterns, and the ML-LSTM network learned sequential information for accurate prediction, assessed against the PJM benchmark dataset [21]. In 2021, Massaoudi et al. proposed a technique that employs a tiered approach to handle load demand fluctuations. By combining Light Gradient Boosting Machine (LGBM), eXtreme Gradient Boosting machine (XGB), and Multi-Layer Perceptron (MLP)models, their Stacked XGB-LGBM-MLP model generated predictions by utilizing meta-data from XGB and LGBM, processed through an MLP network. They validated this method using datasets from Malaysia and New England to demonstrate its efficacy [22]. Bashir et al. introduced a hybrid method integration Prophet and LSTM models to improve load forecasting accuracy in 2022. Whereas Prophet tackled linear and non-linear data, residual non-linear data was addressed using LSTM. Employing Back Propagation Neural Network (BPNN), both the predicted data from Prophet and LSTM were superior, significantly boosting the accuracy. Moreover, their method was validated using Elia Grid's real-time quarter-hour electricity load data from 2014 to 2021 to demonstrate its effectiveness [23].The literature review provides a comprehensive overview of various approaches to electricity load forecasting using machine learning and deep learning techniques. However, each method has its limitations. For instance, The XGBoost approach suggested by Madrid et al. [11] struggles with adaptability to dynamic load profiles. Advanced GRU model by Wan et al., incorporating error-focused training, is sensitive to the choice of clustering methods and feature selection. The XGBoost method for countering the impact of behind-the-meter solar PV on load prediction, while effective, encounters challenges in accurately predicting solar PV capacity under diverse conditions. Sample Entropy Variational mode decomposition suggested by Wang et al. [12] faces limitations in scenarios with rapidly changing industrial electricity consumption patterns. The CEEMDAN-SE-TR Transformer model has computational complexities due to the integration of multiple techniques [15]. NSDAR relies on neural networks for identifying similar days, which is sensitive to the quality and representativeness of the training data [16]. TCN-Prophet hybrid approach by Mo et al. [17]. struggles with capturing long-term dependencies. Shohan et al.'s hybrid LSTM-Neural Prophet model, while outperforming established methods, requires careful tuning of hyperparameters for optimal performance. The R-CNN combined with ML-LSTM architecture introduced for short-term power forecasting is computationally intensive and sensitive to data preprocessing techniques [18]. Tiered approach using stacked XGB-LGBM-MLP models Massaoudi et al. [20] faces challenges in generalizing to diverse datasets. Finally, a hybrid method merging Prophet and LSTM models proposed by Tasarruf Bashir et al. is computationally demanding and requires careful consideration of the temporal dynamics of the electricity load data for optimal performance [21].

Numerous load forecasting studies face limitations as they often neglect to systematically incorporate the effects of feature engineering. 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 algorithm aims to enhance load forecasting by embracing a robust approach with feature engineering.

3. Data and Methods

3.1. Proposed Architecture

The proposed forecasting framework begins with data pre-processing. The dataset is split into training (2015-01-08 to 2018-12-31) and testing (2019-01-06 to 2020-06-27) sets. Feature engineering involves creating new features and handling missing values. Stationarity and causal relationships are assessed using Augmented Dickey-Fuller (ADF) and Granger Causality tests, respectively. In this study, a novel machine learning architecture is introduced, Facebook Prophet-XGB-CatBoost (FPXGB-CatBoost), designed for short-term electrical load forecasting (STLF). Two individual models (Facebook Prophet and XGBoost) were trained, and their outputs were combined in a CatBoost hybrid model. The complementary strengths of three powerful algorithms—Facebook Prophet, XGBoost, and CatBoost—are combined to utilize their collective potential (Figure 1). Facebook Prophet effectively addresses 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 (Extreme Gradient Boosting) 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 plays a dual role within the FPXGB-CatBoost architecture, acting as both a meta-learner to exploit the individual strengths and weaknesses of Facebook Prophet and XGBoost for informed predictions and leveraging its robust performance in handling categorical features and ensemble construction to combine the model outputs and generate the final load forecast. For comparison, GRU, LSTM, and LightGBM models were also trained on the same data.

Meticulous hyperparameter optimization using Optuna ensures each model operates at peak capacity, contributing to the overall accuracy of the ensemble. The FPXGB-CatBoost architecture offers several advantages, including the exploitation of individual model strengths, enhancement of interpretability through Prophet's decomposition, and optimization of model performance through hyperparameter tuning. By combining these techniques, FPXGB-CatBoost emerges as a promising approach for achieving accurate and interpretable STLF, with potential benefits for power system management and cost reduction.

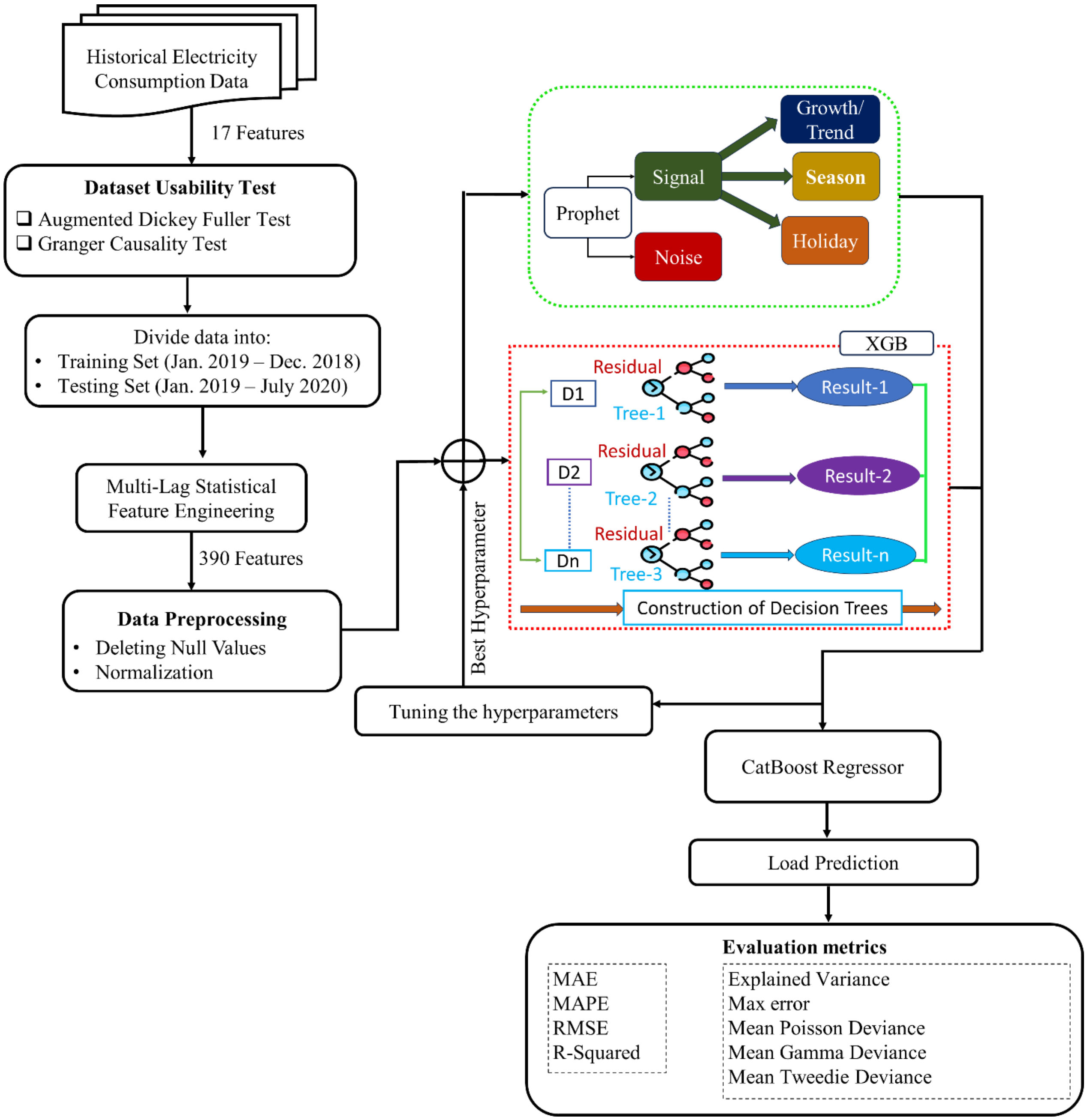


Figure 1: Flowchart of the proposed model

Model performance is evaluated using various metrics (MAE, Explained Variance, MAPE, Max Error, RMSE, deviance measures, R-squared). The hybrid model's results are compared against individual predictions from other models.

3.2. Data Collection and Processing

This study utilizes a publicly available dataset from the National Dispatch Center, Panama, spanning January 2015 to June 2020 [24]. Covering national electricity demand, the dataset facilitates training and evaluating machine learning forecasting models, enabling comparisons with official forecasts. Fairness in comparison necessitates adherence to the dataset's weekly forecasting cycle (Saturday-Friday) and a 72-hour data gap before predictions (forecasts based on data up to Tuesday).

Data components include:

1. Historical electricity consumption (CND, Panama)
2. Historical weekly predictions (CND, Panama)
3. School schedule data (Panama's Ministry of Education)
4. Holiday data ("When on Earth?")
5. Historical weather data (temperature, humidity, precipitation, wind speed) for three Panamanian cities (Earth data)

Detailed variable descriptions and units are provided in Table 1.

Table 1: Variables, description and units of measure from the electricity consumption dataset

Variable	Description	Unit of Measurement
datetime	Date with time	Hour
nat_demand	National electricity load	MWh
T2M_c	Temperature at 2 meters	℃
QV2M_c	Relative humidity at 2 meters	%
TQL_c	Liquid precipitation	l/m^2
W2M_c	Wind speed at 2 meters	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)	-

Sub-index c stands for different cities (David, Santiago and Tocumen, Panama City) such as T2M_dav represents Temperature at 2 meters in David City

Figure 2 graphically presents each variable of the dataset across the date. The data set comprises 48,048 readings with 17 different features. The dataset is complete, having 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 exhibits a noticeable pattern that suggests the presence of seasonal variations.

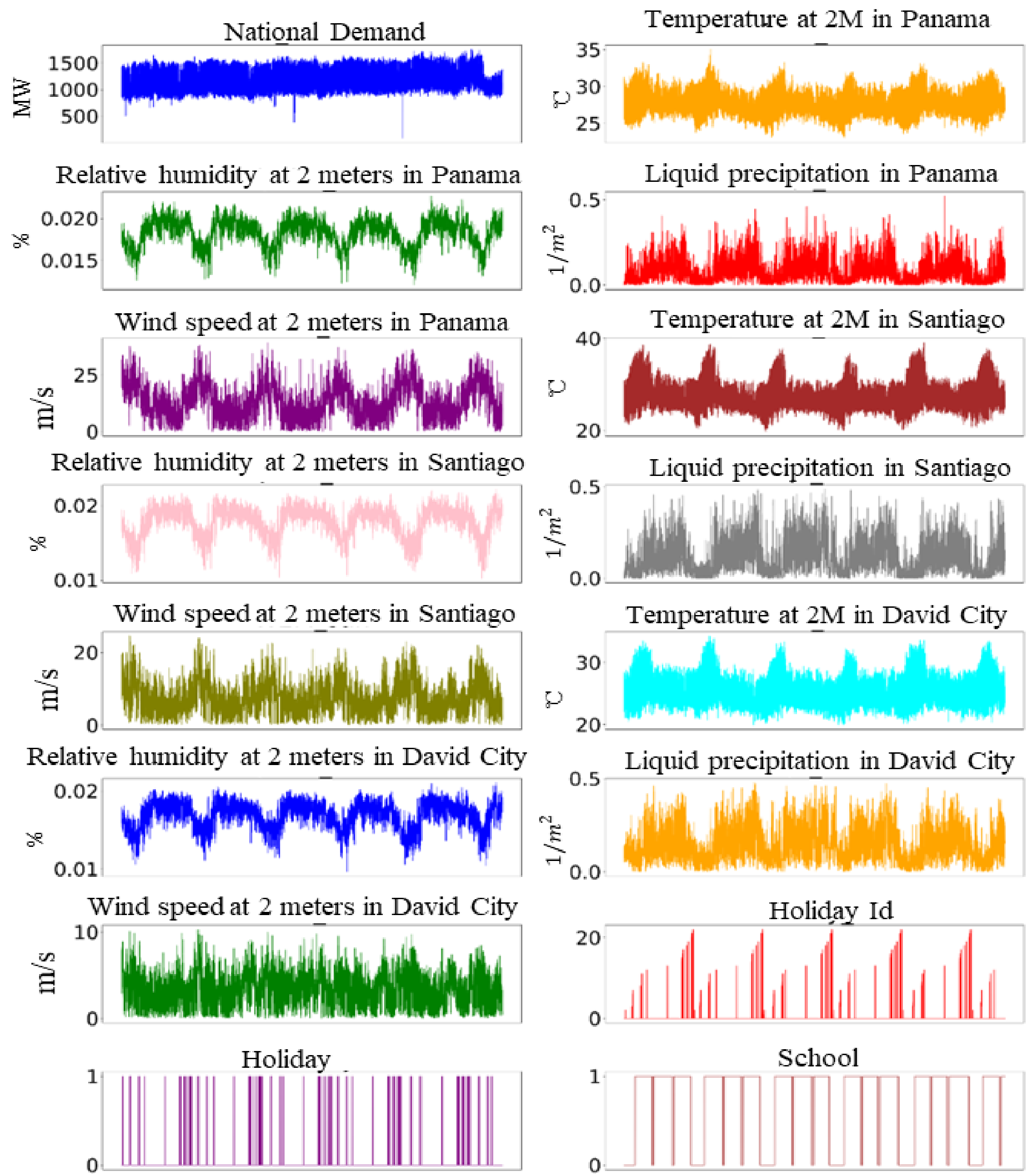


Figure 2: Variables in the electricity consumption dataset across the time period

Figure 3 demonstrates the power usage for individual day of a week for four different seasons. The power usage on Friday, Monday, Thursday, Tuesday, and Wednesday are similar for different seasons. On Saturdays, there is lower power consumption, and on Sundays, it's even less compared to other days.

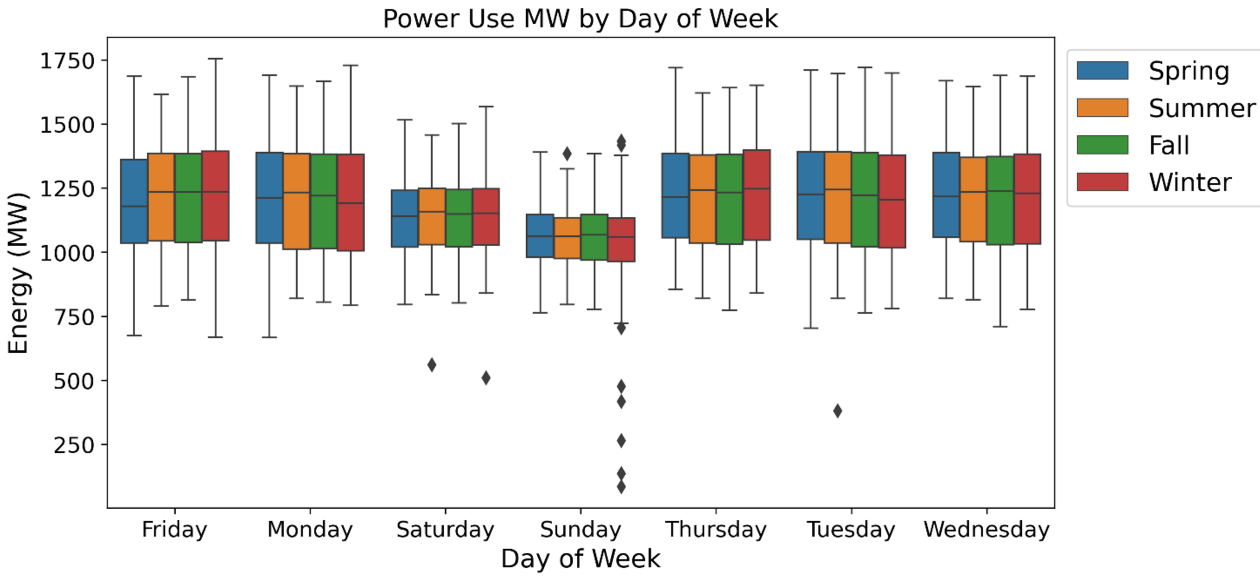


Figure 3: Power Use in MW in every day of a week

Feature stationarity, crucial for reliable time series forecasting models, is assessed using the Augmented Dickey-Fuller (ADF) test [25]. This ensures statistically consistent characteristics over time. Low p-values, indicative of stationarity, support using these features for forecasting. Table 2 confirms feature stationarity through consistent low p-values. These stationary features provide a robust foundation for accurate load forecasting [26]. This supports the inclusion of these features in forecasting models. While T2M_san showed a weaker association (p-value = 0.19), W2M_san displayed no statistically significant predictive power (p-value = 0.5524) as shown in Figure 4. Further investigation into the reasons behind these exceptions and their potential impact on forecasting accuracy is recommended.

Table 2: Augmented Dickey-Fuller test output

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}

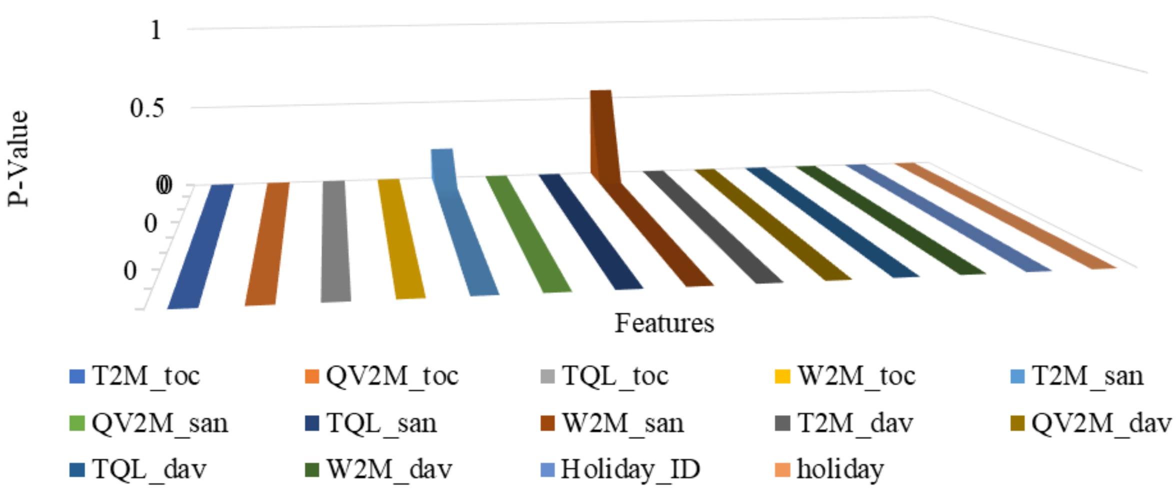


Figure 4: Granger causality test

3.3. Feature Engineering

The process of choosing, modifying, or developing new features from raw data to enhance the effectiveness of machine learning models is known as feature engineering. It entails finding and encoding pertinent data in a more detailed and organized manner. More accurate, general, and efficient models can result from effective feature engineering. It is essential to data preparation and has a big influence on how well a machine learning project turns out.

The feature engineering method proposed in this paper involves working with the dataset to generate a collection of novel attributes extracted from the pre-existing variables. The process enhances the dataset's predictive power by capturing temporal patterns and statistical characteristics, making it an integral step in building accurate electric load forecasting models. This process is designed to create a rich feature set by considering different time windows, ensuring the model's capacity to understand historical load behavior and associated variables. The feature engineering process, depicted in Figure 5, involves leveraging ten different Lag Feature (LAG), Rolling Mean (MEAN), Rolling Standard Deviation (STD), Exponentially Weighted Moving Mean (EWM MA), Exponentially Weighted Moving Standard deviation (EWM STD), Min-max normalized features (MIN MAX), Median (MEDIAN), Skewness (SKEW), Kurtosis (KURT), and 50th percentile (P50) , all computed within 12, 24, and 128 rolling windows. , Among the ten statistical component, LAG, MEAN, STD, EWM MA, and EWM STD are implemented in total demand (nat_demand) column. For other columns all of the ten statistical components are applied except the date time, holiday, holiday id, and school column. This process expands the dataset from its initial 17 features to a comprehensive set of 390 features.

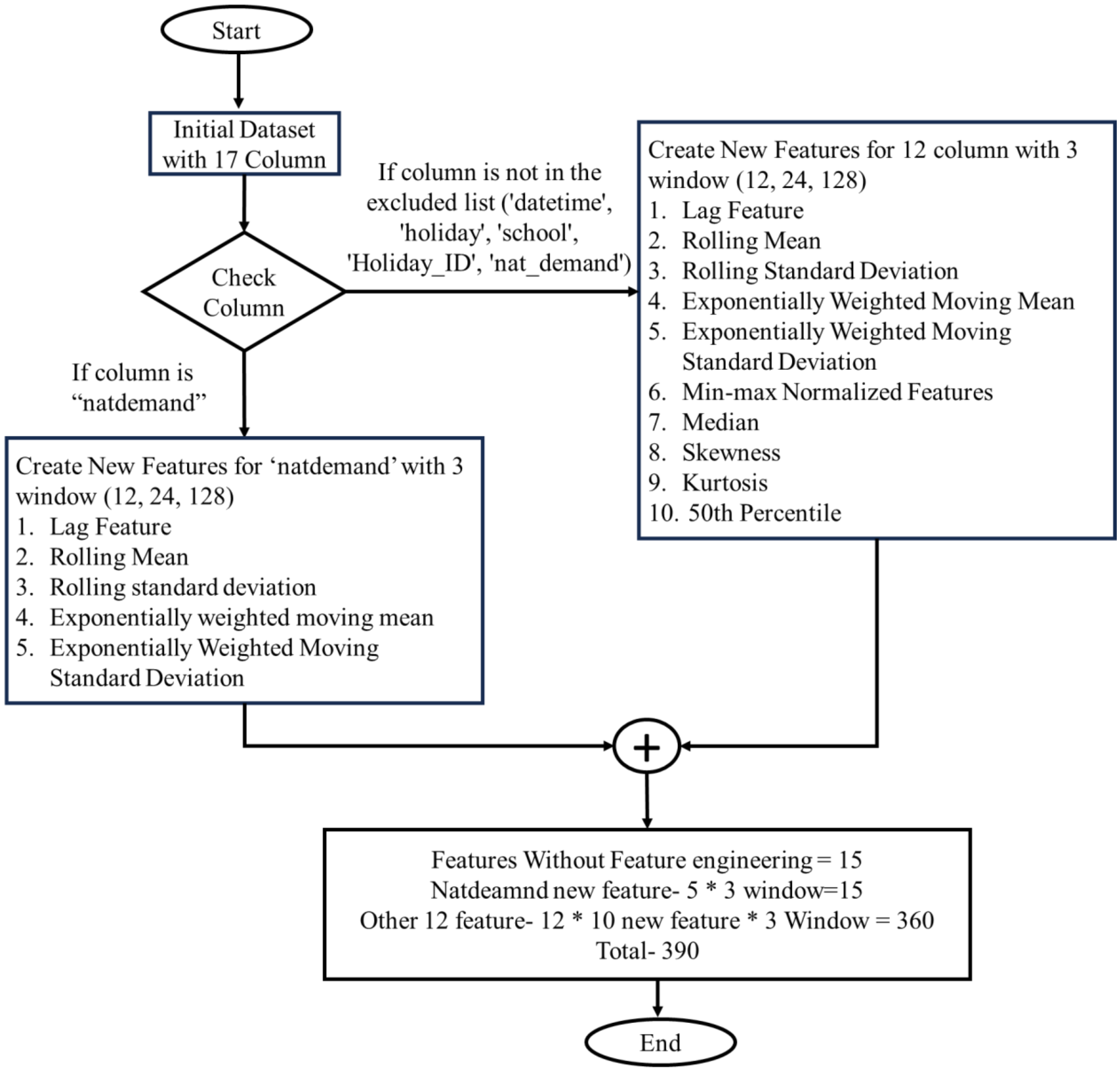


Figure 5: Flow Chart of Feature Engineering

The naming convention for the newly created features follows a pattern: 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 signifies the window size. In Figure 6, the p-values resulting from an augmented Dickey-Fuller test post feature engineering are presented. The yellow line represents the standard p-value which is 0.05. Below this value, a dataset or feature will be non-stationary. 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 is obvious that among the 390 features analyzed, 26 exhibit p-values exceeding 0.05, suggesting non-stationarity.

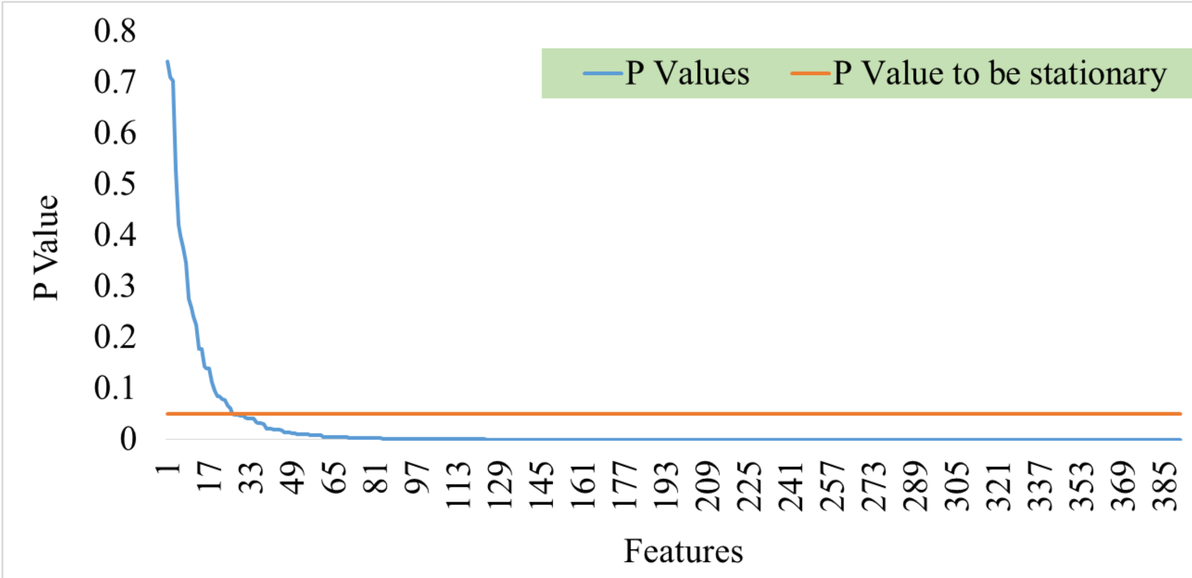


Figure 6: Augmented Dickey Fuller test after feature engineering

Notably, "T2M_san_ewm_std128" stands out as the most non-stationary feature with a p-value of 0.74. Additionally, the top 10 non-stationary features include "T2M_toc_ewm_std128," "T2M_dav_ewm_std128," "QV2M_san_ewm_mean128," "QV2M_dav_ewm_mean128," "nat_demand_ewm_std128," "QV2M_toc_ewm_mean128," "W2M_toc_ewm_mean128," "T2M_toc_std_std128," "T2M_san_std_std128," "T2M_dav_std_std128," and "W2M_san_ewm_mean128." The majority of the features exhibit p-values below 0.05, indicating stationarity within the dataset after the feature engineering process.

3.4. Machine Learning models employed

This section provides a brief overview of the principles underlying the different models, chosen for developing the hybrid model and comparison. Furthermore, Section 3.7 explores into the optimization techniques employed for each algorithm, aimed at enhancing the accuracy of the proposed hybrid model.

3.6.1. LSTM

LSTM (Long Short-Term Memory) models, a subset of recurrent neural networks (RNNs), have emerged as a formidable tool for time series forecasting in the context of electric load prediction. What sets LSTMs apart is their unique ability to capture and model complex temporal dependencies. LSTMs require minimal manual feature engineering, as they autonomously extract relevant features from the data, reducing the burden of preprocessing [27]. The mathematical expression for LSTM model that is demonstrated in Figure 7 are stated in equation (1) to (8) [5]:

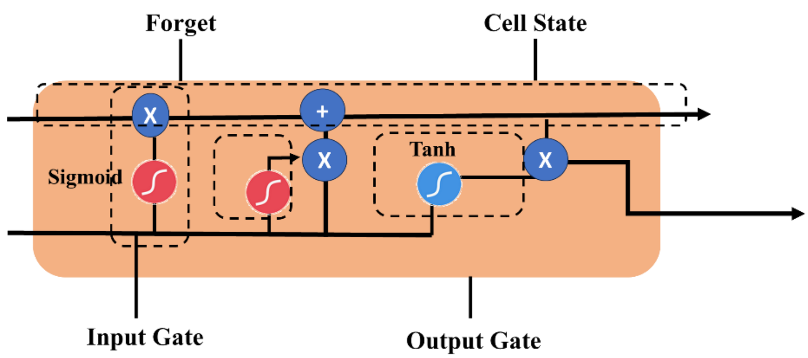


Figure 7: Basic Long Short-Term Memory (LSTM) unit

$$F(t)= \sigma(W_f.[H_{t-1}, X_t]+b_f) \tag{1}$$

$$I(t)= \sigma(W_i.[H_{t-1}, X_t] +b_i) \tag{2}$$

$$C\tilde{}(t)=\tanh(W_c.[H_{t-1}, X_t]+b_c) \tag{3}$$

$$C(t)= f_i* (C_{t-1}+I_t* C\tilde{}_t) \tag{4}$$

$$O(t) = \sigma (Wo \cdot [H_{t-1}, X_t] + b_o) \tag{5}$$

$$H(t) = Ot * \tanh(Ct) \tag{6}$$

$$\text{sigmoid } (x) = \frac{1}{1 + e^{-x}} \tag{7}$$

$$\tanh (x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

In equation (1) to (8) the sequential input is represented by X(t), b_f , b_i, b_c and b_o denotes the bias weights, the input weights are represented by W_f , W_i, W_c and Wo, t is the latest time step and t−1 is the previous time step; H denotes the output , C signifies the cell state and F_t, i_t and o_t are the forget, input, and output gates respectively, controlling the flow of information.

3.6.2. GRU

GRU (Gated Recurrent Unit) models, a variant of recurrent neural networks (RNNs), are valuable tools for time series forecasting, including electric load prediction. The simplified architecture of

GRUs is well known for enabling effective training and quicker convergence. Because of their superior ability to capture short- and medium-term relationships in time series data, load forecasting—where patterns can be complex and dynamic benefits from their use [28]. The mathematical expressions for the GRU model demonstrated in Figure 8 are stated in equations [28] (9) to (14):

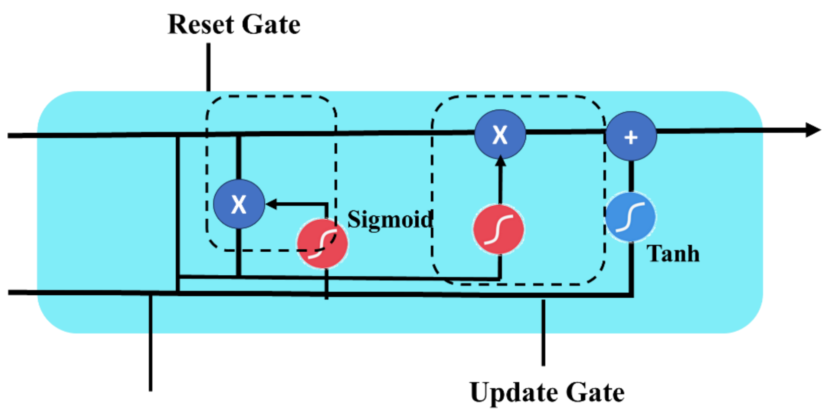


Figure 8: Basic Gated Recurrent Unit (GRU) unit

$z(t)= \sigma(W_Z.[H_{t-1}, X_t])$ (9)

$r(t)= \sigma(W_r.[H_{t-1}, X_t])$ (10)

$h\tilde{}(t)=\tanh(W_c.[r*\{H_{t-1}, X_t\}])$ (11)

$h(t)= (1-z_t)*h_{t-1}+z_t* h\tilde{}(t)$ (12)

$\text{Sigmoid }(x) = \frac{1}{1+e^{-x}}$ (13)

$\tanh (x) = \frac{e^x-e^{-x}}{e^x+e^{-x}}$ (14)

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, determining how much of the previous state to keep, r_t is the reset gate, deciding how much of the previous state to forget and $h\tilde{}(t)$ is the new candidate activation.

3.6.3. LightGBM

LightGBM, a gradient boosting framework, stands out for its efficiency in handling large datasets through histogram-based binning and leaf-wise tree growth. Leveraging techniques like Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), it accelerates training by optimizing instance sampling and feature selection (Figure 9). Its support for various objectives, evaluation metrics, and integration across multiple programming languages makes it a versatile and powerful tool for machine learning tasks [29].

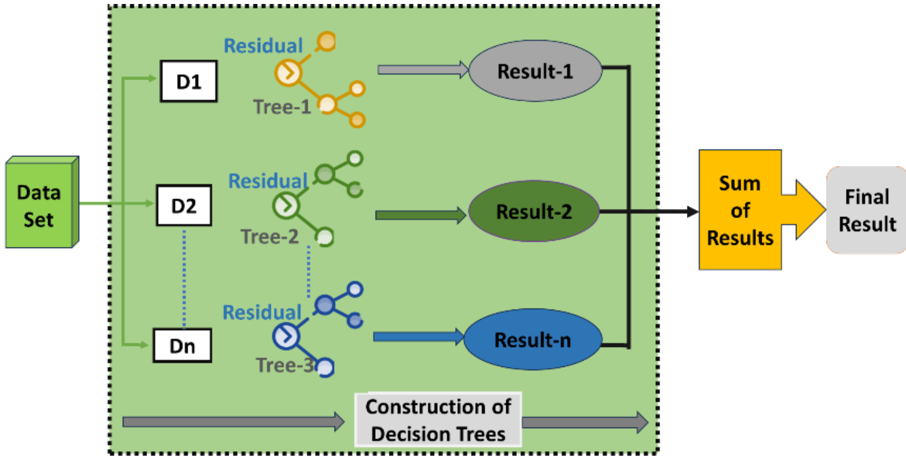


Figure 9: LightGBM model

The LightGBM model's loss function is defined as follows [30]:

$$\text{Objective} = \sum_{i=1}^{n} Loss(y_{i,}\hat{y}_i) + \sum_{k=1}^{n} \Omega(f_k) \quad (18)$$

Here in equation (18) Objective represents the overall objective function to be optimized. n is the number of training samples, y_i is the true label of the ith sample, $\hat{y}_i$ is the predicted output of the ith sample, K is the number of trees in the model, f_k represents the k^{th} tree, $\Omega(f_k)$ is the regularization term that penalizes the complexity of the model.

3.6.4. XGBoost

XGBoost, or Extreme Gradient Boosting, is a popular and effective machine learning method. It belongs to the ensemble learning subcategory and integrates results from various decision trees to produce precise predictions for the regression as well as classification issues (Figure 10). XGBoost is unique in that it can handle complex data sets and operates with extreme efficiency. The gradient boosting framework, which continuously improves the model by correcting errors made in previous rounds, is used to do this. For many data science applications, XGBoost is a popular choice due to its speed, durability, and ability to handle missing data [31][32]

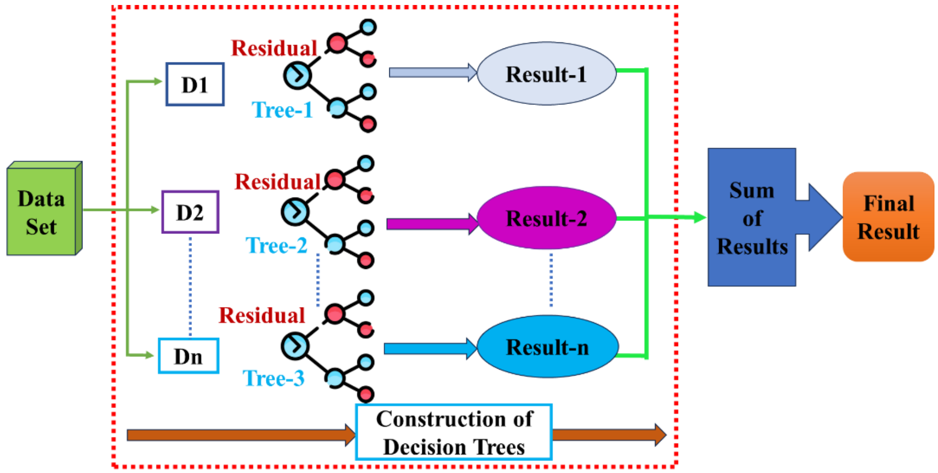


Figure 10: XGBoost Model

A simple equation for XGBoost model, $Y = F(X_1, X_2, X_3, ..., X_n)$ (16)

Where, Y represents the predicted feature (the dependent variable), $X_1, X_2, X_3, ..., X_n$ represent the multiple features (independent variables) used for prediction, F is the ensemble of decision trees generated by the XGBoost algorithm, where each tree contributes to the prediction. The final prediction is a combination of the individual decision tree predictions.

The XGBoost model's loss function is defined as follows [31][30]:

$$\text{Objective}=\sum_{i=1}^{n} Loss(y_{i,}\hat{y}_i) + \sum_{k=1}^{n} \Omega(f_k) \quad (17)$$

Here objective represents the overall objective function to be optimized. n is the number of training samples, y_i is the true label of the ith sample, $\hat{y}_i$ is the predicted output of the ith sample, K is the number of trees in the model, f_k represents the k^{th} tree and $\Omega(f_k)$ is the regularization term that penalizes the complexity of the model.

3.6.5. Fb Prophet

The Core Data Science team at Facebook created the Prophet model as a forecasting tool to tackle the difficulties associated with time series forecasting in a variety of industries, such as business, finance, and weather forecasting. Since its public release in 2017 as an open-source project, it has grown in popularity due to its simplicity of use and efficiency in capturing intricate time series patterns. Here are some key aspects of the Prophet model: Automatic Seasonality Detection, Holiday Effects, Flexible Trend Modeling, Uncertainty Estimation, Customizable Parameters, Scalability, Open-Source and Community Support, Prophet's Limitations [32][33]. The graphical representation of prophet model is shown in Figure 11.

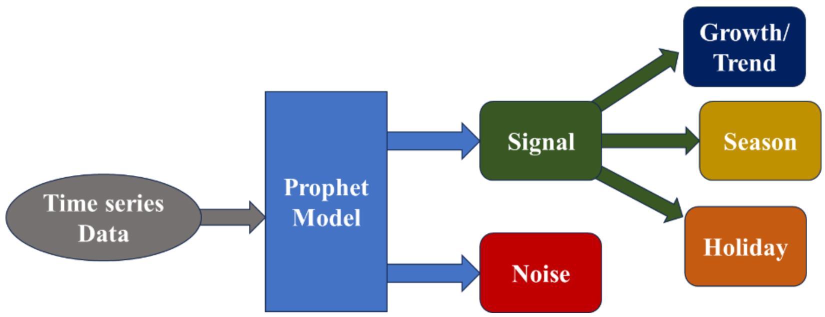


Figure 11: Facebook Prophet Model

Conceptually, the Facebook Prophet model [32] [34] can be represented as follows:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon(t) \tag{15}$$

Where g(t) is the trend component, which may be represented as a linear or logistic function of time, s(t) captures seasonality and can be represented as the sum of Fourier series components, which account for yearly, weekly, and any other periodic patterns, h(t) accounts for holiday effects and is a function of the dates associated with holidays or special events, ε(t) represents the error term, which accounts for any unexplained variation or noise in the data.

The FB-Prophet model, a recent development, acts as an automated parameter adjuster without requiring prior knowledge of the core model. It delivers consistent results when applied to electric load data. Training the FB-Prophet involves splitting data into three sections: a timestamp column (ds) holding time and date details, a column for logged load values (y), and another additional variable incorporated into the model's regressor layer. Figure 12 depicts the methodology of FB-Prophet utilized in this study. Additionally, due to the regular interval nature of electric load consumption data, the FB-Prophet model proves suitable for accurately forecasting electric load consumption patterns.

The dataset is divided into three distinct parts to represent visually three essential components: trend, seasonality, and noise by Facebook Prophet, which is depicted in Figure 12. The entire dataset is used to manifest how these elements are decomposed within the dataset.

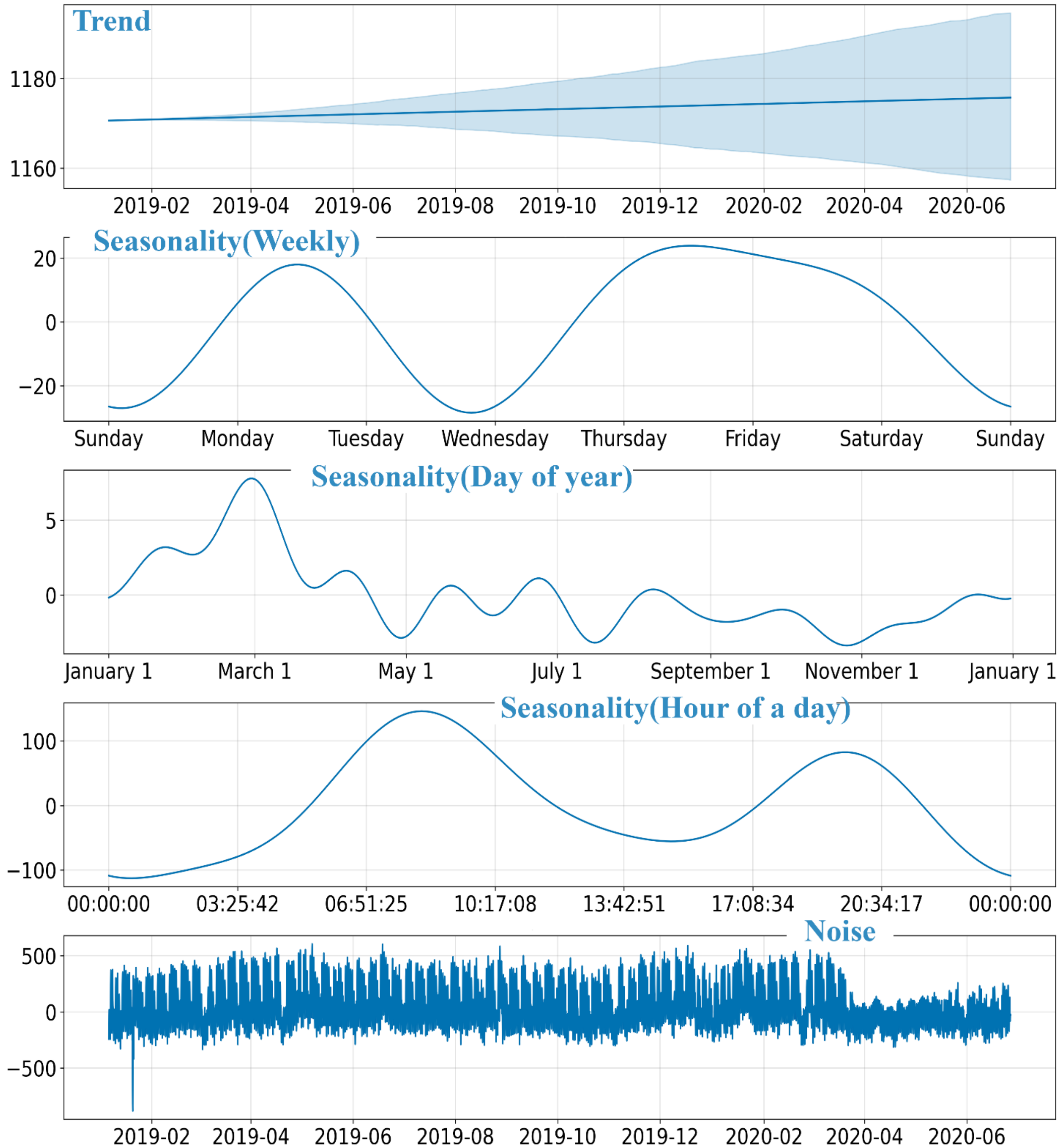


Figure 12: Decomposition of Dataset into different components with Prophet Model

3.6.6. CatBoost

CatBoost is a high-performance gradient boosting machine learning tool designed specifically to handle category features. It is well known for its ability to automatically handle category data, eliminating the need for laborious preparation. CatBoost offers exceptional prediction accuracy while minimizing overfitting through the use of an innovative ordered boosting algorithm. It is open-source, works with several computer languages, and may be used for tasks like classification and regression. Because of its widespread use, short training times, and competitive performance in a variety of scenarios, CatBoost is a useful tool for machine learning and data science professionals [35].

The CatBoost Regressor, known for its adeptness with categorical features and strong performance in regression tasks, stands out due to its resilience against overfitting, capability to handle missing data, and efficient training speed. In this context, it serves as a meta-model, learning from predictions made by both the FB-Prophet and XGBoost models, potentially enhancing overall predictive accuracy. The ensemble predictions result from a weighted combination of these models, assigning weights of 0.7 to FB-Prophet predictions and 0.3 to XGBoost predictions. The ensemble's accuracy is assessed using Root Mean Squared Error (RMSE) against the y test dataset.

3.5. Hyperparameter Tuning

To ensure optimal model performance, a multifaceted approach to hyperparameter tuning was undertaken, catering to the unique characteristics of each model type employed in this study.

For gradient boosting models (XGBoost, LightGBM, CatBoost), a hyperparameter search was conducted using Optuna, a well-established library renowned for its efficiency and automation capabilities. Within Optuna, a custom objective function was defined, minimizing the root mean squared error (RMSE) on the validation set as the primary optimization criterion. This process involved exploring the hyperparameter space through 100 trials, ultimately identifying the configuration that minimized RMSE. This optimized hyperparameter configuration was then utilized to train the final model on the entire training data, which subsequently generated predictions on the unseen test set.

For deep learning models (LSTM, GRU), a manual hyperparameter selection approach was adopted, similar to the one employed for Facebook Prophet. This involved iteratively training the models with various hyperparameter combinations while meticulously monitoring the RMSE metric on the validation set. The hyperparameter selections that yielded the best performance were then employed in the final models. For transparency, the specific hyperparameters chosen for these models are presented in Table 3.

Finally, Facebook Prophet, a specialized time series forecasting model, also underwent manual hyperparameter tuning. The focus was on parameters influencing the model's ability to capture temporal patterns, including changepoint_prior_scale, seasonality_prior_scale, and holidays_prior_scale. These parameters impact the model's flexibility in detecting changes, the strength of its prior belief about seasonality, and the influence of holidays, respectively. Additionally, parameters like interval_width and uncertainty_samples were adjusted to define the confidence interval width and quantify the associated uncertainty. Details of the chosen hyperparameters for Facebook Prophet are also included in Table 3 for comprehensive reference.

Table 3: Selected Hyper Parameters

Model Name	**Hyper Parameter Name**	**Value**
LSTM & GRU	Epochs	60
	Batch Size	64
	Learning Rate	0.01
	Number of LSTM Layers	3
	Input Shape	13031, 1, 390
	Number of LSTM Units	112
LightGBM	num_leaves	61
	learning_rate	0.00665
	feature_fraction	0.513417
	bagging_fraction	0.647086
	bagging_freq	9
Facebook Prophet	changepoint_prior_scale	0.05
	seasonality_prior_scale	10.0
	holidays_prior_scale	10.0
	interval_width	0.8
	uncertainty_samples	0
XGB	eta	0.06881
	max_depth	6
	min_child_weight	3.927
	subsample	0.94549
	colsample_bytree	0.61289
	gamma	3.1457
	lambda	1.2076
	alpha	0.982
CatBoost	Iterations	100
	Depth	6
	Learning Rate	0.1
	Loss Function	RMSE

4. Results and Analysis

4.1. Evaluation criteria

In the context of machine learning, evaluation criteria are numerical measurements that are used to evaluate a prediction model's efficacy and performance. These standards shed light on the model's effectiveness in carrying out its intended function. Different regression metrics like mean squared error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) or R-squared are examples of common evaluation criteria. The selection of criteria is contingent upon the nature of the problem (classification versus regression) and particular objectives (e.g., fraud detection: minimizing false positives). For a thorough assessment of the model's performance, it is imperative to choose relevant criteria that match the demands of the problem and, where needed, to combine several metrics. The paper includes several other statistical assessment metrics such as Mean Poisson Deviance, Mean Gamma Deviance, Mean Tweedie Deviance, Explained Variance, and Max Error for model evaluation. In the field of load forecasting models, the evaluation metrics, including Mean Poisson Deviance, Mean Gamma Deviance, Mean Tweedie Deviance, Explained Variance, and Max Error, serve crucial roles in assessing the performance and accuracy of the models. Mean Poisson Deviance is relevant for count data, such as predicting the number of units during a specific period, while Mean Gamma Deviance is applicable for capturing right-skewed distributions, common in load forecasting during peak demand. Mean Tweedie Deviance is useful for addressing excess zeros in the data, often seen in periods of low or no demand. Explained Variance provides an integrated measure of how well the model explains the total variance in load data, while Max Error helps identify instances of significant prediction errors by guiding improvements. These metrics help assess load forecasting models by looking at distribution, variability, explanatory power, and worst-case scenarios, assisting in refining and selecting effective models.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid \quad (19)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \text{ x } 100\% \quad (20)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mid y_i - \hat{y}_i \mid)^2} \quad (21)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(|y_i - \hat{y}_i|)^2}{\sum_{i=1}^{n}(|y_i - \hat{y}_i|)^2} \quad (22)$$

$$Mean\ Poisson\ Deviance = \frac{1}{n}\sum_{i=1}^{n}[2.(y_i . log\frac{y_i}{\hat{\lambda}_i} - (y_i - \hat{\lambda}_i))] \quad (23)$$

$$Mean\ Gamma\ Deviance = \frac{1}{n}\sum_{i=1}^{n}[2.(\frac{y_i}{\hat{\alpha}_i} - log\frac{y_i}{\hat{\alpha}_i} - 1)] \quad (24)$$

$$Mean\ Tweedie\ Deviance = \frac{1}{n}\sum_{i=1}^{n}\left[2\omega_i\left(\frac{(\hat{p}_i - \hat{\mu}_i).y_i}{\phi} - b(\hat{p}_i, \hat{\phi})\right)\right] \quad (25)$$

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

$$Max\ Error\ =\ MAX_{i=1}{}^{n}\ |\ y_i - \hat{y}_i\ |$$

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.

4.2. Experimental set-up and plan

The experiments used cloud infrastructure from the Kaggle platform, providing access to necessary GPUs and storage for efficient simulations. An NVIDIA Tesla P100 GPU was chosen with 16GB GDDR5 memory and 3584 CUDA parallel processing cores. The CPU model was Intel(R) Xeon(R) @ 2.30GHz with 2 physical cores clocked at 2.3 GHz. Total system RAM amounted to 12GB.

All simulations were implemented using Python 3.10 for the machine learning workflow, with Pandas 2.1 NumPy 1.26, and Scikit-Learn 1.3.2 serving as the core data science libraries.

Python 3.10 was the programming language used based on its widespread support for machine learning libraries. The public Panama electricity demand dataset was loaded into this framework directly from Kaggle repositories. The performance analysis is divided into two sections, without feature engineering and with feature engineering.

4.3. Case-I: Without Feature Engineering

In the first case, all testing models were trained and tested using solely the 16 specified features. Feature importance without feature engineering analysis showed a hierarchy of significant factors illustrated in Figure 13. Metrics related to atmospheric moisture content like 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 highlighting the profound impact of temperature and wind speed at 2-meter height above the surface. Additionally, variables capturing temperature and wind at different vertical levels such as TQL_dav (8.20%) and W2M_dav (8.71%) presents considerable importance. In contrast, features like Holiday_ID (1.55%), school (1.04%), and holiday (0.12%) exhibited marginal significance and indicated a less strong impact on prediction.

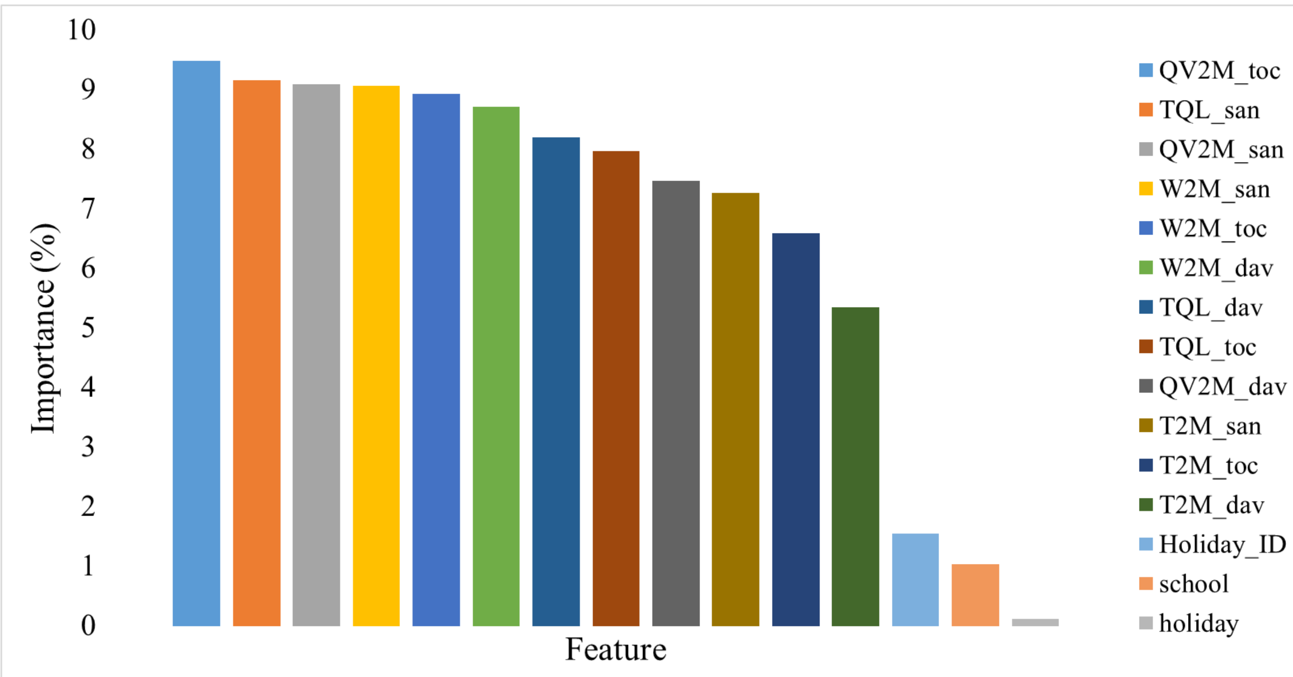


Figure 13: Feature importance before feature engineering

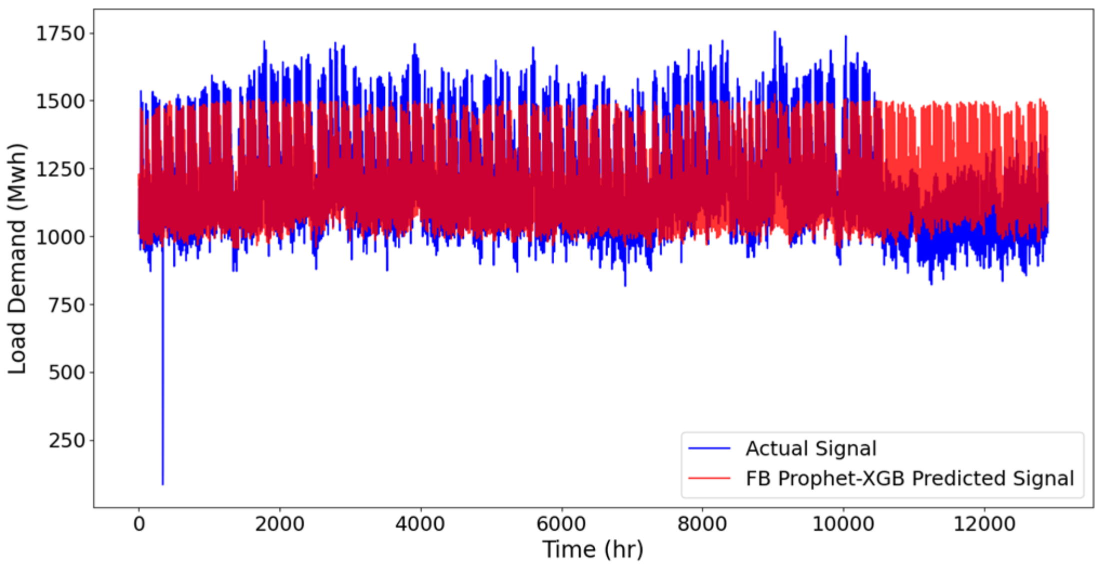
It is demonstrated in Table 4 that the proposed framework outperformed the state-of-the-art architectures without feature engineering. Overall, it was apparent that LSTM, GRU, FB Prophet, and Prophet-XGB performed better than LBG and XGB architectures. In terms of all nine metrics, the Prophet-XGB architectures manifested the best performance values of 110.20, 0.66, and 9.77 for RMSE, R^2 and mean position deviance respectively.

Table 4: Evaluation metrics by different models without feature engineering on the same dataset

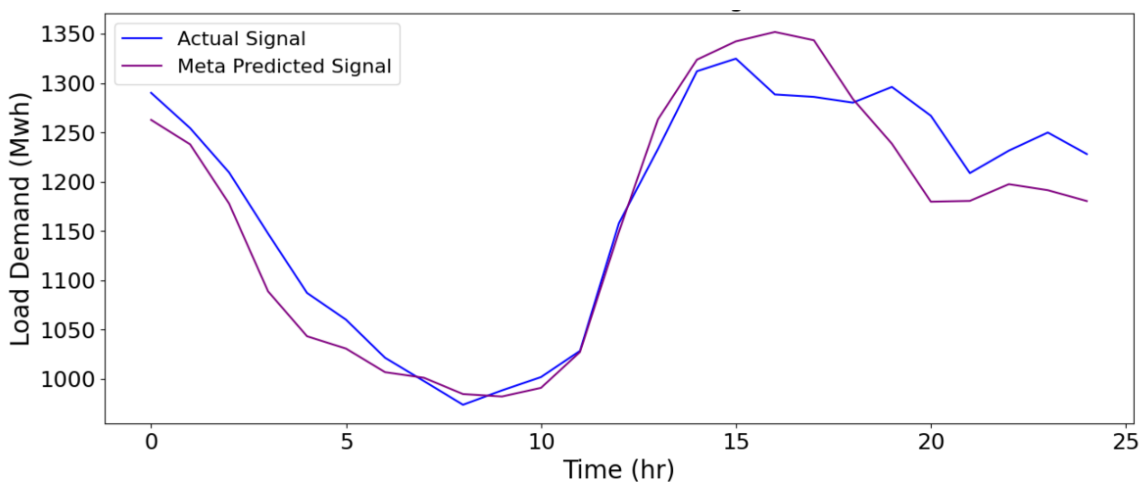
Model	**RMSE**	**MAE**	**MAPE**	**R^2**	**Explained Variance**	**Max Error**	**Mean Poisson Deviance**	**Mean Gamma Deviance**	**Mean Tweedie Deviance**
LSTM	125.16	96.57	0.081	0.56	0.573	1376.2	12.56	0.01	15666.58
GRU	123.38	87.55	0.075	0.57	0.57	1410.2	11.98	0.0097	15222.09

Fb Prophet	123.47	87.67	0.0757	0.57	0.59	530.98	12.08	0.0098	15142.56
XGB	143.64	114.845	0.0948	0.42	0.463	1209.94	16.75	0.01398	20634.41
LGB	156.96	126.36	0.102	0.30	0.361	1169.59	19.75	0.016	24636.41
Prophet-XGB	**110.20**	**81.32**	**0.069**	**0.66**	**0.6568**	**1153.76**	**9.77**	**0.00818**	**12145.60**

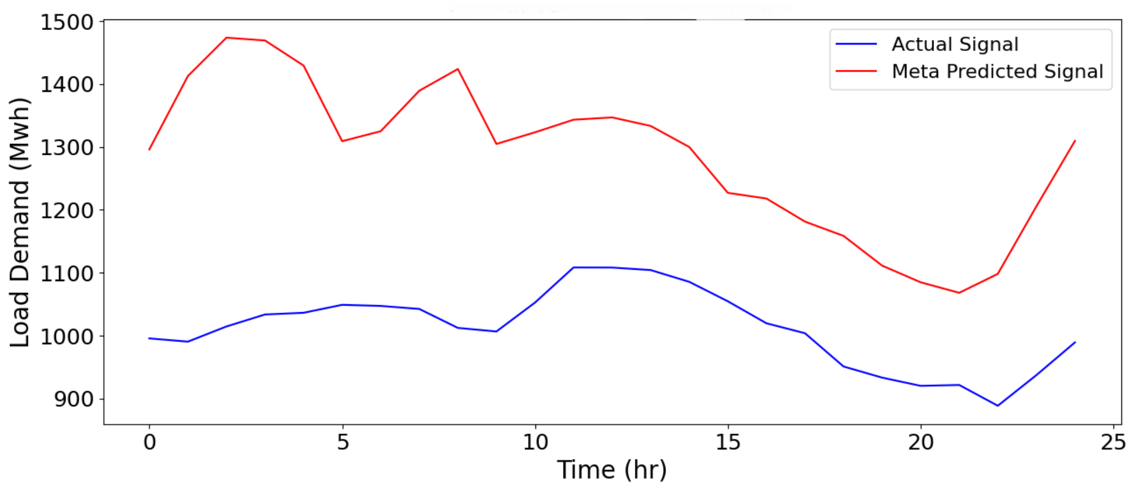
Figure 14(a) demonstrates the actual vs predicted load in the whole test data set without feature engineering. It is clear that the predicted graph cannot cover the actual load demand. In this particular case, the suggested framework lacks the ability to forecast seasonal patterns. It consistently produces a nearly uniform output and fails to identify sudden drop in the electricity load demand. In Figure 14(b), the optimal 24-hour prediction scenario is observed, characterized by a remarkable alignment between the predicted and actual data. Conversely, Figure 14(c) portrays the worst 24-hour prediction case, characterized by a substantial difference between the predicted and actual data.



(a)



(b)



(c)

Figure 14: Actual vs prediction by the proposed model without feature engineering of: (a) Full Test Data set; (b) Best 24 hours; and (c) Worst 24 hours

4.4. Case II: With Feature Engineering

After accomplishing feature engineering it was apparent from Table 5 that performance of all seven models increased by a great margin. The proposed model outclassed the other model. It was also evident that Prophet-XGB, GRU, FB Prophet, and LSTM performed in a superior way, which suggested that these four models were able to catch the trend of electric load forecasting. On the other side, XGB, and LGB lagged behind compared to the other models.

Table 5: Evaluation metrics by different models with Feature Engineering on the same dataset

Model Name	RMSE	MAE	MAPE	R^2	Explained Variance	Max Error	Mean Poisson	Mean Gamma	Mean Tweedie

							Deviance	**Deviance**	**Deviance**
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	0.863	924.1	5.23	0.004	6273.79
LGB	52.42	34.197	0.0303	0.922	0.9246	1237.43	2.42	0.002	2748.25
XGB	50.74	32.18	0.028	0.927	0.9303	1208.75	2.28	0.002	2574.669
FB Prophet	41.72	29.99	0.026	0.95	0.954	622.23	1.61	0.0017	1746.91
FB Prophet-XGB-CatBoost	**32.32**	**23.70**	**0.0203**	**0.97**	**0.9704**	**223.83**	**0.905**	**0.00084**	**1044.99**

Table 5 showcases the performance metrics of various models employed in load forecasting. Metrics such as RMSE, MAE, MAPE, and R^2, explained variance, max error, and deviance measures are reported. The LSTM and FB Prophet models exhibited strong performance with lower RMSE and higher R^2 values, indicating superior predictive accuracy. Fb Prophet-XGB-CatBoost stands out with the lowest RMSE and MAE, suggesting enhanced precision in load forecasting compared to the other models like GRU, LGB, and XGB. Based on the results, the "Prophet-XGB" hybrid model emerged as the top performer, boasting the lowest RMSE, MAE, and MAPE. Furthermore, it achieved a notably high R^2 and Explained Variance, signifying its superior accuracy and capability to capture variations in electric load data. This model presents a compelling option for electric load forecasting applications.

Table 6 illustrates feature relevance ratings across distinct features utilized in this study from multiple models: LSTM, GRU, XGBoost (XGB), LightGBM (LGBM), and Meta Model. The Meta Model, in contrast, gives greater weight to features such as 'T2M_dav_min_max24' (19.784%) and 'nat_demand_std_std12' (15.308%), but XGB and LGBM show significant preferences for 'T2M_dav_min_max24' (28.02%) and 'nat_demand_std_std12' (20.74%), demonstrating their divergent emphasis. The meta model relies heavily on temperature for predicting electricity demand. This makes sense as it depends on a feature called 'nat_demand_std_std12,' which is a past record of national demand, highlighting its importance in making predictions.

The results shed light on each model's interpretation of feature importance, pointing out possible major predictors for the target variable and enabling a more sophisticated comprehension of feature interpretations unique to the research area. Figure 15 graphically presents importance of top 15 features of the proposed meta model.

Table 6: Feature importance of different models

Feature Name	LSTM	GRU	XGB	LightGBM	Meta Model
T2M_dav_min_max24	0.56%	0.2%	0.56%	28.02%	19.784%
nat_demand_std_std12	2.78%	2.17%	2.63%	20.74%	15.308%
nat_demand_lag_24	0.28%	0.37%	1.34%	12.62%	9.23%
T2M_san_min_max24	0.42%	0.17%	0.43%	11.27%	8.017%
T2M_toc_min_max24	0.42%	0.2%	0.48%	9.35%	6.69%
nat_demand_ma_mean24	3.28%	2.46%	1.1%	3.77%	2.97%
nat_demand_ewm_mean12	1.28%	1.28%	0.53%	3.28%	2.64%
nat_demand_ma_mean12	3.03%	2.16%	1.93%	2.81%	2.55%
nat_demand_lag_12	0.56%	0.21%	1.73%	1.26%	1.41%
nat_demand_ewm_std12	0.62%	0.27%	1.31%	1.08%	1.15%
T2M_dav_min_max12	1.2%	0.19%	0.37%	1.32%	1.04%
nat_demand_std_std24	1.1%	0.56%	1.33%	0.39%	0.68%
T2M_toc_skew12	0.31%	0.45%	0.6%	0.65%	0.64%
T2M_san_skew12	0.46%	0.39%	0.55%	0.43%	0.465%
nat_demand_ma_mean128	1.01%	1.18%	0.79%	0.174%	0.365%

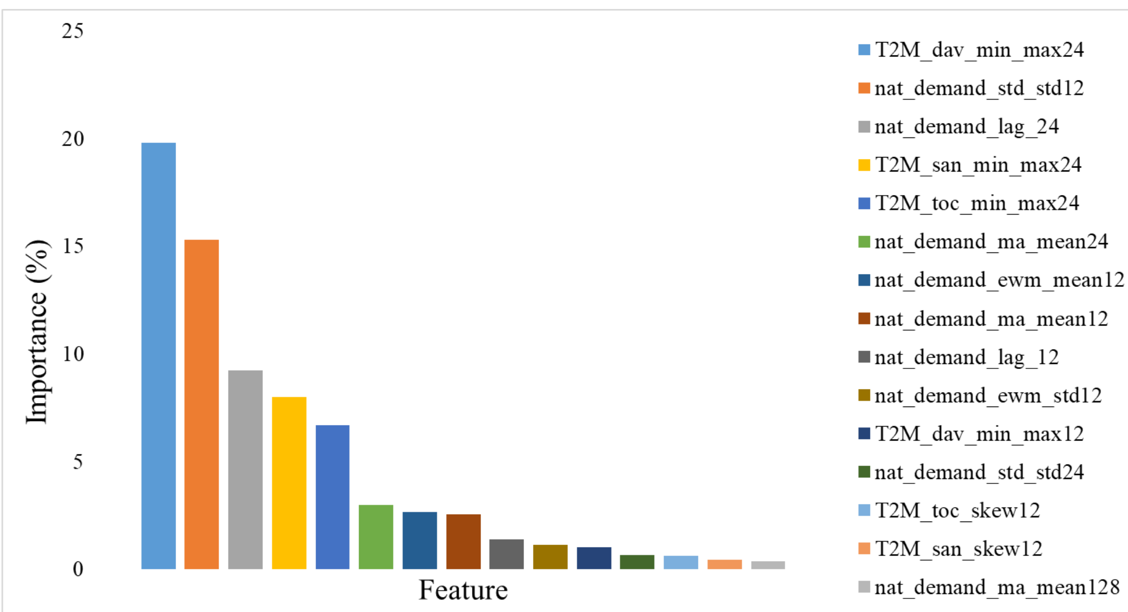


Figure 15: Top 15 important features of meta model

Figure 16 exhibits the comparison between the actual and predicted load across the entire test dataset following feature engineering. The graph shows a strong alignment between the predicted load and the actual load demand. Notably, the proposed framework effectively captures seasonal patterns which consistently generating a graph that closely matches the actual data, even during sudden drops in demand.

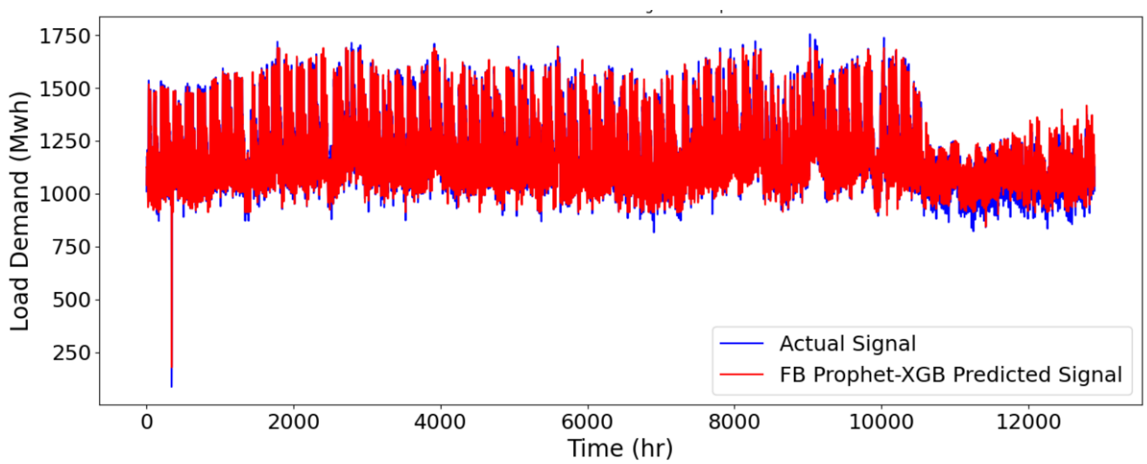
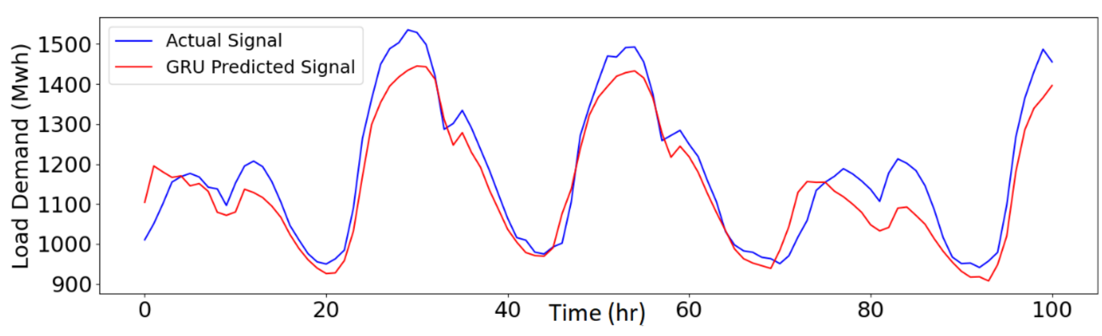
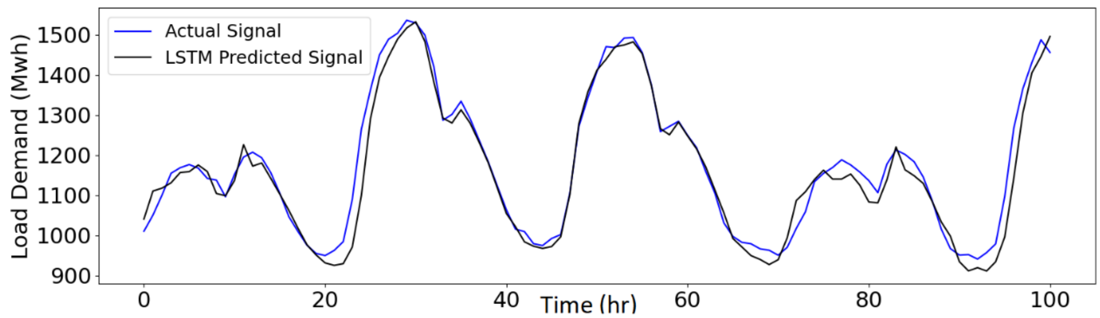


Figure 16: Actual vs prediction of proposed model after feature engineering

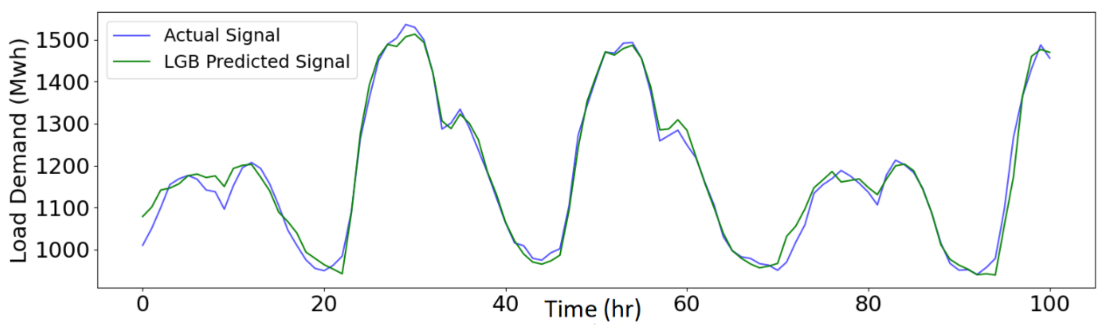
Following the model training, predictions for load forecasting one hour ahead were conducted at various time points spanning from January 2019 to July 2020. The training and testing processes were conducted within a consistent environment. Figure 17 demonstrates the performances of distinct models in predicting the load for Panama City during the initial 100 hours of January 2019. The figures evidently indicated that the proposed hybrid model exhibited superior performance compared to all other models. As the testing dataset spans 18 consecutive months, it can confidently affirm that the model has been trained to account for every season throughout the year.



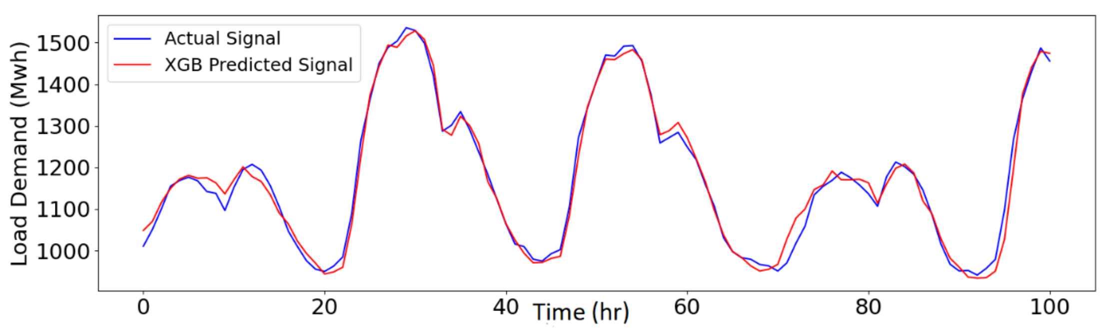
(a)


Actual Signal
LSTM Predicted Signal
Load Demand (Mwh)
Time (hr)

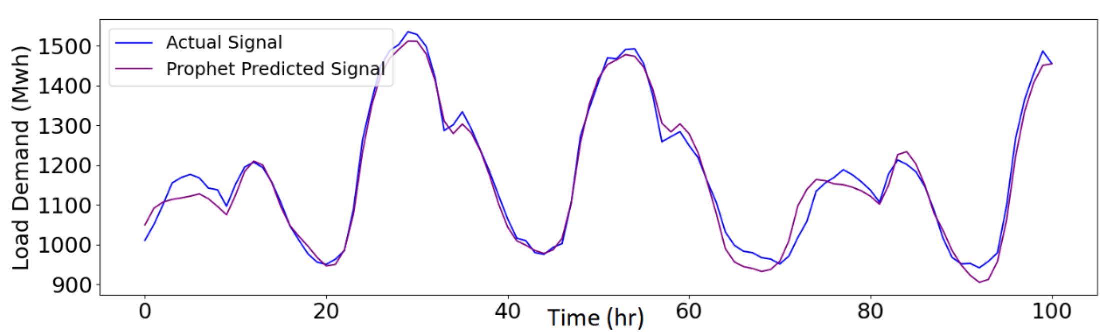
(b)


Actual Signal
LGB Predicted Signal
Load Demand (Mwh)
Time (hr)

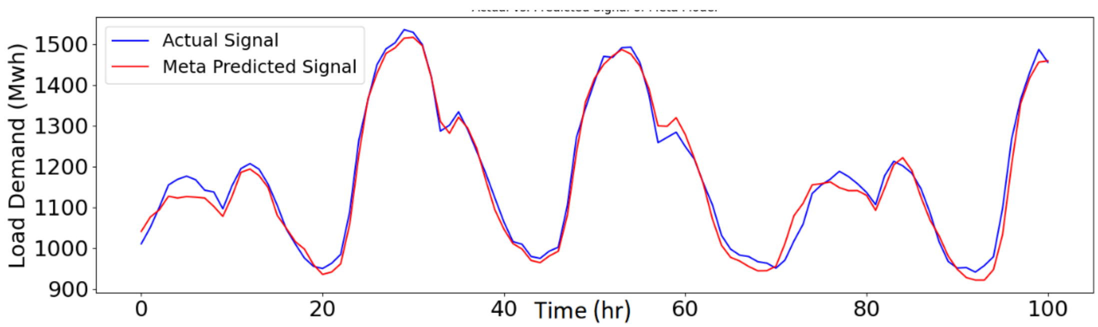
(c)


Actual Signal
XGB Predicted Signal
Load Demand (Mwh)
Time (hr)

(d)


Actual Signal
Prophet Predicted Signal
Load Demand (Mwh)
Time (hr)

(e)



(f)

Figure 17: Actual vs predicted load of: (a) GRU (b) LSTM (c) LightGB (d) XGB (e) FB Prophet (f) Proposed Meta Model for 1/6/2019 8:00 to 1/10/2019 11:00 (100 hour)

In Figure 18, the efficacy of various techniques including the hybrid model is depicted for the first 48 hours in January 2019, to provide clear understanding. The outcome aligned with the earlier findings, as the suggested hybrid model continues to surpass the performance of other models.

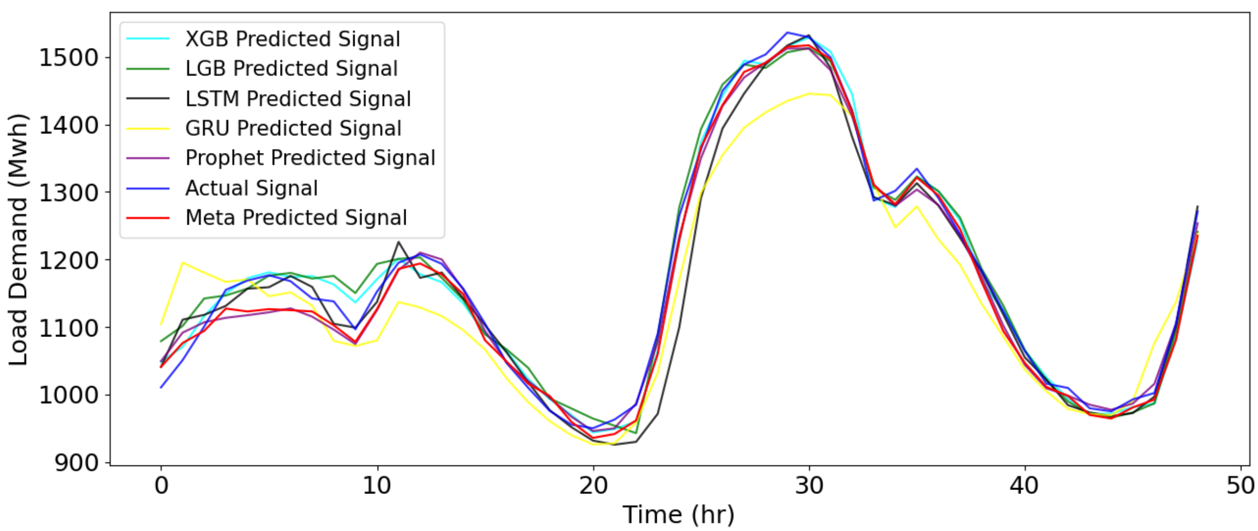


Figure 18: Actual vs prediction at different times with the proposed model compared to other models

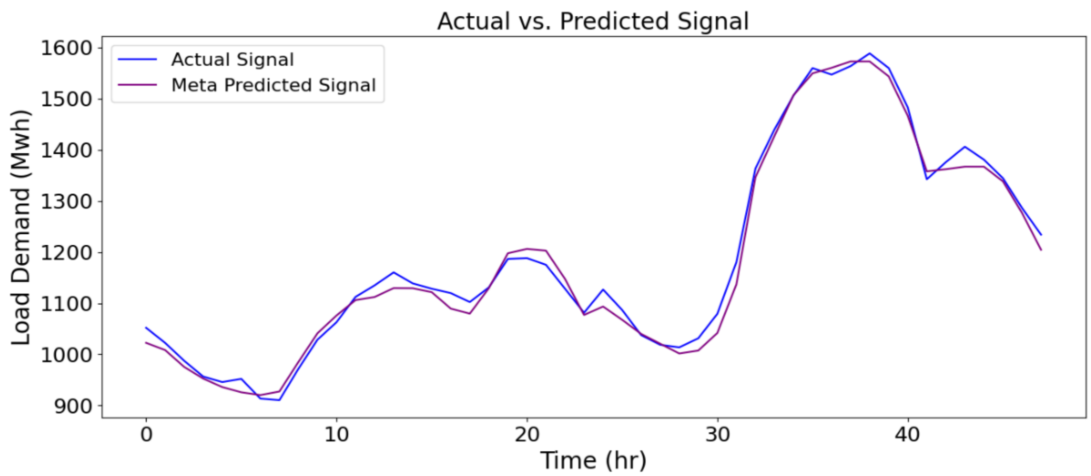
The test set was divided into three parts, where each part consisted of data of six consecutive months. Then, the corresponding performance metrics of each part are depicted in Table 7. It is obvious from the table that though the proposed model performed identically in terms of the metrics in the first two parts, the performance declined in the last six months. It is demonstrated from Figure 2 that the electric loads in the last six months experienced a sudden drop compared to the other sections of the dataset. Consequently, the performance during this specific period has slightly deteriorated. The average error metrics in this section is close to the values depicted in Table 5 which demonstrates the robustness of the proposed model.

Table 7: Evaluation metrics by the proposed model, for each testing specific timestamp

Time	RMSE	MAE	MAPE	R^2	Explained Variance	Max Error	Mean Poisson Deviance	Mean Gamma Deviance	Mean Tweedie Deviance
01/01/2019 - 30/06/2019	28.17	20.78	0.017	0.98	0.978	210.65	0.682	0.00078	793.46
01/07/2019 - 31/12/2019	29.52	22.47	0.0184	0.98	0.976	193.49	0.714	0.0005	871.39
01/01/2020 - 30/06/2020	38.99	28.26	0.025	0.95	0.953	189.75	1.366	0.0012	1520.88
Average	**32.23**	**23.84**	**0.02**	**0.97**	**0.969**	**197.96**	**0.92067**	**0.000827**	**1061.91**

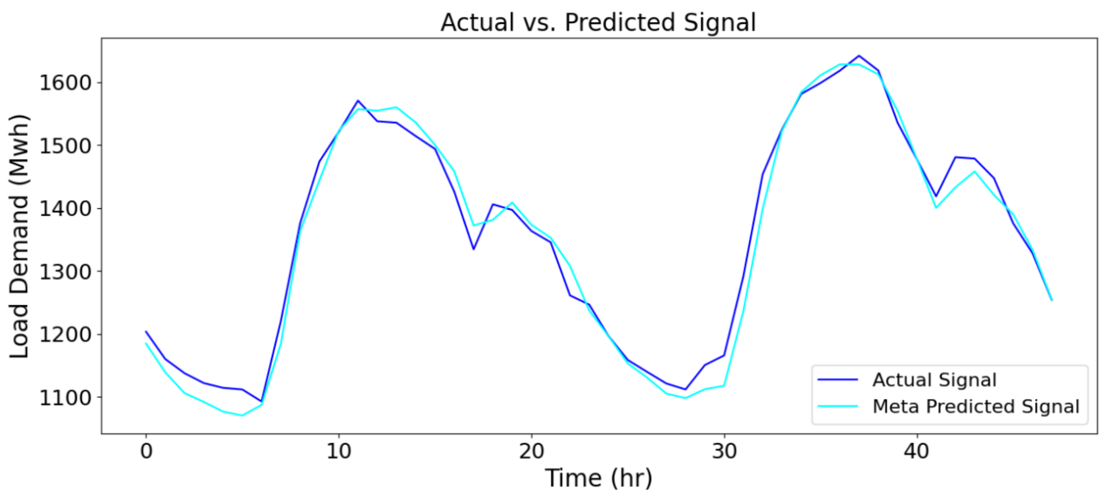
The simulation results were contrasted with 6 methods (LSTM, GRU, FB Prophet, XGB, LGB, and Prophet-XGB) that were suggested under identical testing conditions and configurations.

Figure 19 illustrates the comparison between projected and actual values within distinct segments (best, average and worst) of the whole testing dataset. In Figure 19 (a), the most splendid performance is showcased, demonstrating an RMSE of 17.9, MAE of 14.31, MAPE of 0.012, and an R^2 of 0.99 which is observed during the initial week of February 2019. This represents an ideal scenario where the predicted graph aligns closely with the actual demand curve across most data points. Subsequently, Figure 19 (b) represents the model's average performance, situated around December 2019, where the RMSE stands at 30.24, MAE at 23.13, MAPE at 0.017, and an R^2 of 0.97. In this case, the predicted graph may not fully capture the changes in the actual curve. Conversely, the least favorable performance emerges in the initial week of July 2020, revealing an RMSE of 48.69, MAE of 35.08, MAPE of 0.03, and an R2 of 0.74 for this specific analysis. This shows the worst-case scenario where the predicted graph fails to cover both the changing points with some other sections of the actual curve.



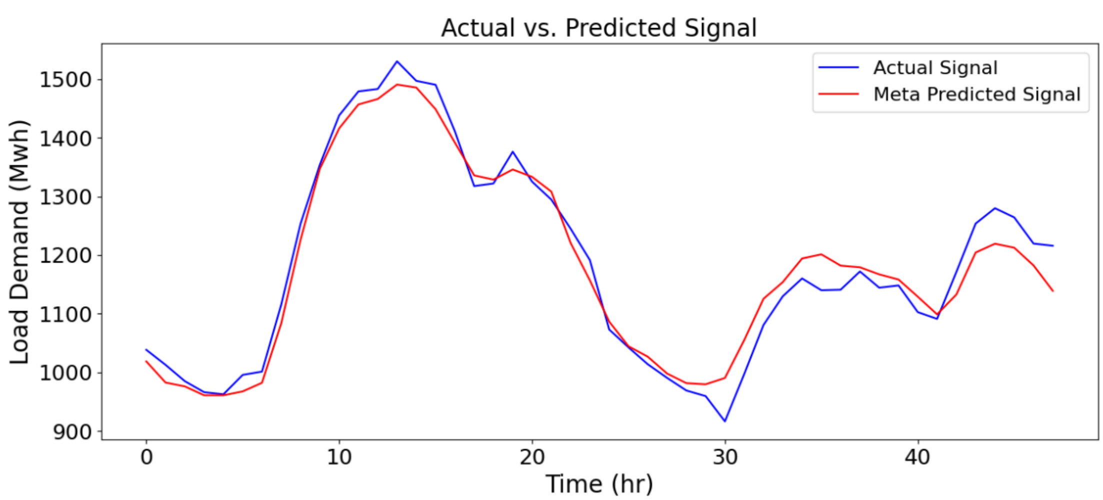
Actual vs. Predicted Signal
Actual Signal
Meta Predicted Signal
Load Demand (Mwh)
Time (hr)

(a)



Actual vs. Predicted Signal
Actual Signal
Meta Predicted Signal
Load Demand (Mwh)
Time (hr)

(b)



Actual vs. Predicted Signal
Actual Signal
Meta Predicted Signal
Load Demand (Mwh)
Time (hr)

(c)

Figure 19: Actual vs Prediction for of (a) Best 48 hours (3 February 2019 to 4 February 2019), (b) Average 48 hours (4 December 2019 to 5 December 2019) and (c) Worst 48 hours (6 June 2020 to 7 June 2020) of the whole test dataset.

Notable observation from Figure 20 is that the proposed approach outperformed the other methods significantly in terms of MAPE, RMSE, MAE, and R^2 values. The hybrid FB Prophet-XGB model effectively eliminates the shortcomings in individual models' predictions. The proposed method yields the lowest MAPE value, confirming its superiority in Short-Term Load Forecasting (STLF).

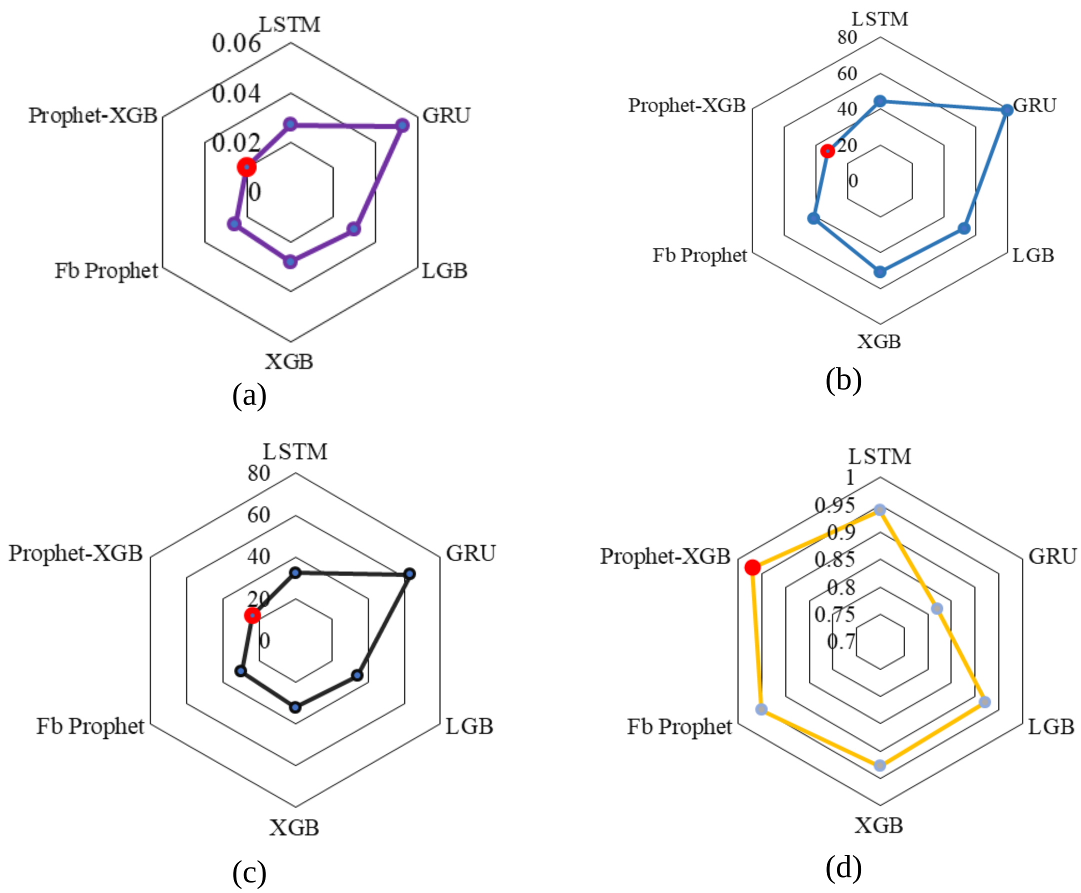


Figure 20: (a) MAPE, (b) RMSE, (c) MAE, and (d) R2 Error of Prophet-XGB Model

The findings, as shown in Table 8, indicate that effective prediction is attainable under a smaller dataset. Based on Table 5, the effectiveness of the suggested approach is evident in the form of a 9.15 and 11.96 reduction in RMSE when compared to the Prophet and LSTM model. Around 97%

of the variance in the dependent variable can be explained by the independent variable(s) in the model, as evidenced by an R-squared (R^2) value of 0.97.

Table 8: Evaluation metrics by different models, for every testing week, and average

Model	Year	2019												2020						Average
	Month	**Jan.**	**Feb.**	**Mar.**	**Apr.**	**May**	**June**	**July**	**Aug.**	**Sept.**	**Oct.**	**Nov.**	**Dec.**	**Jan.**	**Feb.**	**Mar.**	**Apr.**	**June**	**July**	
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
LGB	**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
XGB	**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
Proposed	**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

4.5. Sensitivity Analysis

In practical scenarios, missing data and inaccuracies are common. The accuracy of forecasting is expected to be influenced by the reliability and accuracy of the input data [36]. This study explores into the impact of disruptions to data integrity on the accuracy of forecasting models. Simulating data integrity involves randomly eliminating 10%, 20%, 30%, 40%, and 50% data from the complete dataset. The process is iterated five times to mitigate any negative impacts on the experiments. To prevent unexpected or unintentional outcomes, the simulation is conducted repeatedly by removing 0%, 10%, 20%, 30%, 40%, and 50% of datasets from distinct locations each time. Various forecasting models are employed to generate predictions, allowing for a comparison of prediction errors.

As indicated in Table 9, the proposed model consistently demonstrates the smallest RMSE and the narrowest range of variation, regardless of the dataset's completeness. Figure 21 illustrates that the RMSE of the proposed model remains consistently the lowest, with a small fluctuation range. Therefore, it can be concluded that the proposed model exhibits high robustness, and its forecasting accuracy is not significantly affected by variations in data quality.

Table 9: Results of sensitivity analysis

Data Removal (%)	**RMSE scores of different models**					
	GRU	**LSTM**	**LGB**	**XGB**	**FB Prophet**	**Fb Prophet-XGBoost-CatBoost**
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

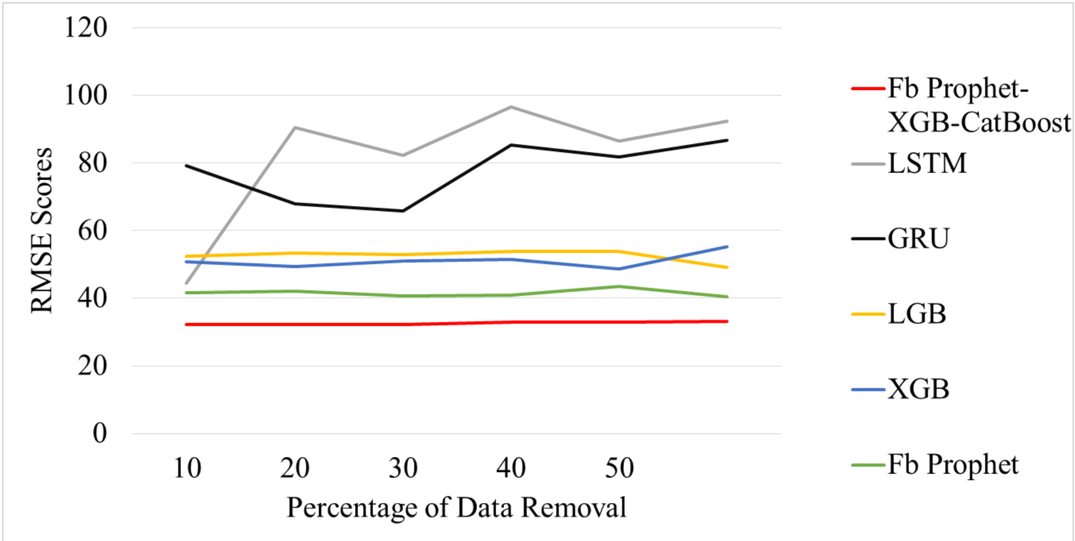


Figure 21: Results of sensitivity tests

4.6. Discussion

4.6.1. Comparison with literature

From table 4 and table 5 it is shown that the feature engineering technique depicts approximately a 70.67% improvement from 110.20 to 32.32 in term of RMSE and 70.85% improvement from 81.32 to 23.70 in terms of MAE error.

From table 5 the proposed hybrid model demonstrated an increase in accuracy ranging from 22% to 59% for RMSE values, from 21% to 62% for MAE values, and from 22% to 62% for MAPE values when compared with the performance of the other 5 machine learning algorithms trained in this study. Moreover, the R2 value of 0.97 for the proposed hybrid model guarantees the optimal fitting of the curve to the historical electricity load data. Overall, the proposed Fb Prophet-XGB-CatBoost hybrid model showed the most accurate results. The suggested hybrid model offers precise predictions of real statistical patterns without any time lag effects, overfitting, or underfitting. Its stability presents a clear advantage, making it a valuable tool for guiding electric load forecasting.

The proposed framework with the feature engineering offers a 26.84% less RMSE than other [15] existing frameworks. Madrid et. al. [15] introduced an XGBoost model for the same dataset employed in this study. Their model showed an average RMSE of 44.52 across the entire test dataset. In contrast, the proposed model achieved a notably lower RMSE of 32.57. Their model exhibits a 3.66 MAPE, while the proposed model demonstrates a 0.0203 MAPE, signifying a 99.44% superior performance in MAPE terms. Although Madrid et al. engaged in feature engineering, this research distinguishes itself through its emphasis on multi-lag feature engineering coupled with extensive statistical analysis. This distinct methodology plays a crucial role in the success of the study.

Table 11: Comparison with other proposed models with evaluation metric

Proposed Model	Cross-Validation	Test Set	Error				Paper ID
			MAE	RMSE	MAPE	R^2	
XGB	2 folds	20%	….	44.52	3.66	…	[13]
Whale Optimization Algorithm (WOA) - Variational mode decomposition (VMD)- Temporal Convolutional Networks (TCN)- improved GRU	No	2%	50.561	67.51 /kW	0.96	…	[14]
SVMD-XGBoost with Bayesian Optimization Algorithm	No	60 days	0.024	0.0316/kW	4.96%	……	[16]
Temporal Convolutional Networks (TCN)-Prophet	No	20%	1.40025	0.67	0.02187	……	[19]
LSTM-Neural Prophet	No	405 days	….	85.181	9.7606	0.83	[20]
CNN-LSTM	No	Not Mentioned	0.0144	0.0325	0.033	0.98	[21]
Complete Ensemble Empirical Mode Decomposition with Adaptive Noise- Sample Entropy-Transformer (CEEMDAN-SE-TR)	No	25%	0.95	1.26	4.80%	0.80	[17]
Neural-based Similar Days Auto Regression (NSDAR)	No	Not Mentioned	224.22	…	0.03448	0.97	[18]
Proposed model in this paper	**No**	**18 Month**	**23.84**	**32.57**	**0.0203**	**0.97**	

4.6.2. Strength of the proposed approach

This study proposes a novel ensemble machine learning framework for short-term electricity load forecasting, integrating Facebook Prophet, XGBoost, and CatBoost models empowered by comprehensive feature engineering. This approach significantly outperforms existing methods, demonstrating its capability and potential for real-world applications. Leveraging intricate feature engineering techniques, the model captures complex temporal patterns and dependencies, leading to superior prediction accuracy. Rigorous evaluation using diverse metrics comprehensively assesses its effectiveness, while surpassing established methods highlights its generalizability. The model's resilience to data integrity disruptions further underscores its robustness and practical applicability, further validated by its efficacy on an extensive real-world dataset. This novel framework represents a significant step forward in accurate short-term electricity load forecasting, offering valuable insights for grid management and resource optimization.

4.6.3. Practical Significance

The presented framework establishes a comprehensive and user-friendly end-to-end system, enabling seamless implementation in power grids. This work encompasses crucial aspects such as feature engineering and machine learning model parameter selection, making it accessible for practitioners in the field. The adaptability of the proposed system is highlighted by its applicability across diverse regions and usage profiles, ensuring accurate load forecasts through meticulous hyperparameter optimization.

Recognizing the regional and usage-based variations in load forecasts, the proposed approach places emphasis on hyperparameter optimization. This ensures that the model adapts optimally to the characteristics of the input data, enhancing its predictive capabilities across different grids. The incorporation of various statistical feature engineering techniques further enhances the model's adaptability, allowing it to generalize effectively for out-of-sample forecasts.

A noteworthy aspect of the methodology is the preference for machine learning models over contemporary deep learning alternatives. This choice not only contributes to shorter training times but also reduces memory requirements, facilitating deployment on distributed computing frameworks. This scalability is particularly advantageous, enabling utility providers to leverage cloud resources for rapid calibration of new data. The simplicity of the model also enhances transparency, making it easier for stakeholders to interpret and trust the forecasting results.

This self-contained approach empowers energy system stakeholders, particularly utility providers, to make informed decisions for smarter management. From dynamic optimization of generation scheduling and demand-response efficiency to seamlessly integrating renewable sources, the proposed architecture minimizes costs and environmental impacts. The ability to forecast load weeks ahead facilitates strategic planning for upgrades, maintenance, and the integration of new power plants. Moreover, this work extends beyond utility providers, benefiting industries and microgrids by enabling intelligent resource allocation. The comprehensive evaluation of the methods on public data invites collaboration and further research from fellow professionals in the field. As the move is towards global electrification, the precision and practicality of short-term load forecasting become paramount for the transition to stable and smart grids. This work serves as a significant step in this direction, providing an implementable solution for a wide array of stakeholders in the energy systems domain.

4.6.4. Limitations and future work

The proposed framework shines in short-term load forecasting but acknowledges room for growth. While the Panama-specific dataset demonstrates effectiveness, concerns regarding generalizability necessitate diversifying training data and exploring strategies like transfer learning for broader applicability. Furthermore, expanding evaluation beyond short-term horizons to capture seasonal trends and long-term patterns will provide a more holistic understanding of the model's capabilities.

Beyond traditional approaches, delving into the potential of advanced architectures like transformers or large language models could unlock further accuracy gains and potentially capture intricate temporal dependencies within the data. Additionally, prioritizing real-time deployment through cloud-based implementations and explainable AI techniques like SHAP and LIME is crucial for practical application and fostering trust among stakeholders.

By addressing these limitations and actively pursuing these promising avenues, we can unlock significant advancements in short-term load forecasting. This, in turn, paves the way for more efficient and sustainable energy management systems by enabling seamless integration of renewable energy sources through precise load predictions. The versatile framework presented here serves as a springboard for further exploration, accelerating the journey towards a smarter and greener energy future.

5. Conclusions

The paper introduces a hybrid model that combines the FB-Prophet XGBoost and CatBoost models to predict short-term electricity load consumption. This approach leverages the strengths of each model, incorporating a robust feature engineering to enhance forecasting accuracy. The model's effectiveness is tested by applying to historical electricity load data from Panama City, validating its performance in load forecasting. According to the paper's findings, the following conclusions can be drawn:

1. The number of input features were expanded from 17 to 390 by employing multi-lagged statistical feature engineering for performance enhancement in electricity load forecasting, which has great positive impact on model performance.

2. The suggested model's effectiveness was assessed under two conditions: with and without feature engineering. The proposed hybrid model has the most accurate values of RMSE= 32.32, MAE=23.70, MAPE=0.0203, R2=0.97 error, alongside 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. The proposed model shows 26.85% and 99.44% better performance in terms of RMSE and MAE error with respect to other literature with the same data set.

3. A thorough analysis was conducted on the forecasting results for hourly, the initial week of each testing month, and 6-month data to enhance the assure of the model's robustness. 32.32, 30.03, and 32.23 are the RMSE values found for hourly, weekly and 6-month load prediction. The small fluctuations in the RMSE value are denoting the robustness of the proposed model.

4. An evaluation of sensitivity analysis was conducted by randomly including or excluding test data. The highest percentage change from the original value of 32.32 is 2.281% (corresponding to 33.06), and the lowest percentage change is −0.402% (corresponding to 32.19).

References:

[1] P. D. Diamantoulakis, V. M. Kapinas, and G. K. Karagiannidis, "Big Data Analytics for Dynamic Energy Management in Smart Grids," *Big Data Research*, vol. 2, no. 3, pp. 94–101, Sep. 2015, doi: 10.1016/J.BDR.2015.03.003.

[2] Lam and Pun-Lee, "Energy in China: Development and Prospects," *http://journals.openedition.org/chinaperspectives*, vol. 2005, no. 59, Jun. 2005, doi: 10.4000/CHINAPERSPECTIVES.2783.

[3] W. Gomez, F. K. Wang, and S. C. Lo, "A hybrid approach-based machine learning models in electricity markets," *Energy*, vol. 289, p. 129988, Feb. 2024, doi: 10.1016/J.ENERGY.2023.129988.

[4] H. Seifi and M. S. Sepasian, "Electric Power System Planning: Issues, Algorithms and Solutions," *Power Systems*, vol. 49, 2011, doi: 10.1007/978-3-642-17989-1/COVER.

[5] M. Munem, T. M. Rubaith Bashar, M. H. Roni, M. Shahriar, T. B. Shawkat, and H. Rahaman, "Electric power load forecasting based on multivariate LSTM neural network using bayesian optimization," *2020 IEEE Electric Power and Energy Conference, EPEC 2020*, Nov. 2020, doi: 10.1109/EPEC48502.2020.9320123.

[6] M. Q. Raza and A. Khosravi, "A review on artificial intelligence-based load demand forecasting techniques for smart grid and buildings," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352–1372, Oct. 2015, doi: 10.1016/J.RSER.2015.04.065.

[7] S. S. Pappas, L. Ekonomou, D. C. Karamousantas, G. E. Chatzarakis, S. K. Katsikas, and P. Liatsis, "Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models," *Energy*, vol. 33, no. 9, pp. 1353–1360, Sep. 2008, doi: 10.1016/J.ENERGY.2008.05.008.

[8] C. C. Hsu and C. Y. Chen, "Applications of improved grey prediction model for power demand forecasting," *Energy Convers Manag*, vol. 44, no. 14, pp. 2241–2249, Aug. 2003, doi: 10.1016/S0196-8904(02)00248-0.

[9] G. Cerne, D. Dovzan, and I. Skrjanc, "Short-Term Load Forecasting by Separating Daily Profiles and Using a Single Fuzzy Model Across the Entire Domain," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 7406–7415, Sep. 2018, doi: 10.1109/TIE.2018.2795555.

[10] S. H. Ling, F. H. F. Leung, H. K. Lam, and P. K. S. Tam, "Short-Term Electric Load Forecasting Based on a Neural Fuzzy Network," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 6, pp. 1305–1316, Dec. 2003, doi: 10.1109/TIE.2003.819572.

[11] M. Karimi, H. Karami, M. Gholami, H. Khatibzadehazad, and N. Moslemi, "Priority index considering temperature and date proximity for selection of similar days in knowledge-based short term load forecasting method," *Energy*, vol. 144, pp. 928–940, Feb. 2018, doi: 10.1016/J.ENERGY.2017.12.083.

[12] "Impact of the entrance of wind and photovoltaic generation in Panama Technological R&D." Accessed: Nov. 26, 2023. [Online]. Available: https://revistas.utp.ac.pa/index.php/id-tecnologico/article/view/1440

[13] E. A. Madrid and N. Antonio, "Short-Term Electricity Load Forecasting with Machine Learning," *Information 2021, Vol. 12, Page 50*, vol. 12, no. 2, p. 50, Jan. 2021, doi: 10.3390/INFO12020050.

[14] Z. Wan and H. Li, "Short-Term Power Load Forecasting Based on Feature Filtering and Error Compensation under Imbalanced Samples," *Energies 2023, Vol. 16, Page 4130*, vol. 16, no. 10, p. 4130, May 2023, doi: 10.3390/EN16104130.

[15] D. J. Bae, B. S. Kwon, and K. Bin Song, "XGBoost-Based Day-Ahead Load Forecasting Algorithm Considering Behind-the-Meter Solar PV Generation," *Energies 2022, Vol. 15, Page 128*, vol. 15, no. 1, p. 128, Dec. 2021, doi: 10.3390/EN15010128.

[16] Y. Wang *et al.*, "Short-term load forecasting of industrial customers based on SVMD and XGBoost," *International Journal of Electrical Power & Energy Systems*, vol. 129, p. 106830, Jul. 2021, doi: 10.1016/J.IJEPES.2021.106830.

[17] P. Ran, K. Dong, X. Liu, and J. Wang, "Short-term load forecasting based on CEEMDAN and Transformer," *Electric Power Systems Research*, vol. 214, p. 108885, Jan. 2023, doi: 10.1016/J.EPSR.2022.108885.

[18] Z. Jiang, L. Zhang, and T. Ji, "NSDAR: A neural network-based model for similar day screening and electric load forecasting," *Appl Energy*, vol. 349, p. 121647, Nov. 2023, doi: 10.1016/J.APENERGY.2023.121647.

[19] J. Mo, R. Wang, M. Cao, K. Yang, X. Yang, and T. Zhang, "A hybrid temporal convolutional network and Prophet model for power load forecasting," *Complex and Intelligent Systems*, vol. 9, no. 4, pp. 4249–4261, Aug. 2023, doi: 10.1007/S40747-022-00952-X/FIGURES/7.

[20] M. J. A. Shohan, M. O. Faruque, and S. Y. Foo, "Forecasting of Electric Load Using a Hybrid LSTM-Neural Prophet Model," *Energies 2022, Vol. 15, Page 2158*, vol. 15, no. 6, p. 2158, Mar. 2022, doi: 10.3390/EN15062158.

[21] M. F. Alsharekh, S. Habib, D. A. Dewi, W. Albattah, M. Islam, and S. Albahli, "Improving the Efficiency of Multistep Short-Term Electricity Load Forecasting via R-CNN with ML-LSTM," *Sensors 2022, Vol. 22, Page 6913*, vol. 22, no. 18, p. 6913, Sep. 2022, doi: 10.3390/S22186913.

[22] M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, F. S. Oueslati, and H. Abu-Rub, "A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting," *Energy*, vol. 214, p. 118874, Jan. 2021, doi: 10.1016/J.ENERGY.2020.118874.

[23] 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, Nov. 2022, doi: 10.1016/J.EGYR.2021.12.067.

[24] “Electricity Load Forecasting.” Accessed: Nov. 25, 2023. [Online]. Available: https://www.kaggle.com/datasets/saurabhshahane/electricity-load-forecasting

[25] D. Mathelinea, R. Chandrashekar, and H. Mawengkang, “Stationarity test for medicine time series data,” *AIP Conf Proc*, vol. 2714, no. 1, May 2023, doi: 10.1063/5.0128444/2889789.

[26] M. Rosoł, M. Młyńczak, and G. Cybulski, “Granger causality test with nonlinear neural-network-based methods: Python package and simulation study,” *Comput Methods Programs Biomed*, vol. 216, p. 106669, Apr. 2022, doi: 10.1016/J.CMPB.2022.106669.

[27] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Comput*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/NECO_A_01199.

[28] R. Dey and F. M. Salemt, “Gate-variants of Gated Recurrent Unit (GRU) neural networks,” *Midwest Symposium on Circuits and Systems*, vol. 2017-August, pp. 1597–1600, Sep. 2017, doi: 10.1109/MWSCAS.2017.8053243.

[29] G. Ke *et al.*, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” *Adv Neural Inf Process Syst*, vol. 30, 2017, Accessed: Nov. 25, 2023. [Online]. Available: https://github.com/Microsoft/LightGBM.

[30] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, doi: 10.1145/2939672.

[31] O. Sagi and L. Rokach, “Approximating XGBoost with an interpretable decision tree,” *Inf Sci (N Y)*, vol. 572, pp. 522–542, Sep. 2021, doi: 10.1016/J.INS.2021.05.055.

[32] C. B. Aditya Satrio, W. Darmawan, B. U. Nadia, and N. Hanafiah, “Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET,” *Procedia Comput Sci*, vol. 179, pp. 524–532, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.036.

[33] B. Kumar Jha and S. Pande, “Time Series Forecasting Model for Supermarket Sales using FB-Prophet,” *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, pp. 547–554, Apr. 2021, doi: 10.1109/ICCMC51019.2021.9418033.

[34] J. Mo, R. Wang, M. Cao, K. Yang, X. Yang, and T. Zhang, “A hybrid temporal convolutional network and Prophet model for power load forecasting,” *Complex and Intelligent Systems*, vol. 9, no. 4, pp. 4249–4261, Aug. 2023, doi: 10.1007/S40747-022-00952-X/FIGURES/7.

[35] J. T. Hancock and T. M. Khoshgoftaar, “CatBoost for big data: an interdisciplinary review,” *J Big Data*, vol. 7, no. 1, pp. 1–45, Dec. 2020, doi: 10.1186/S40537-020-00369-8/FIGURES/9.

[36] A. Shakeel, D. Chong, and J. Wang, “District heating load forecasting with a hybrid model based on LightGBM and FB-prophet,” *J Clean Prod*, vol. 409, p. 137130, Jul. 2023, doi: 10.1016/J.JCLEPRO.2023.137130.