

Article

PredXGBR: A Machine Learning Framework for Short-Term Electrical Load Prediction [†]

Rifat Zabin ¹, Khandaker Foysal Haque ^{2,*} and Ahmed Abdelgawad ³

¹ Department of Computer Science and Engineering, Brac University, Dhaka 1212, Bangladesh; rifat.zabin@g.bracu.ac.bd

² Institute for the Wireless Internet of Things, Northeastern University, Boston, MA 02115, USA

³ College of Science and Engineering, Central Michigan University, Mount Pleasant, ME 48849, USA; abdel1a@cmich.edu

* Correspondence: haque.k@northeastern.edu

[†] This paper is an extended version of our paper published in International Conference on Information and Communication Technology for Development (ICICTD 2022).

Abstract: The growing demand for consumer-end electrical load is driving the need for smarter management of power sector utilities. In today's technologically advanced society, efficient energy usage is critical, leaving no room for waste. To prevent both electricity shortage and wastage, electrical load forecasting becomes the most convenient way out. However, the conventional and probabilistic methods are less adaptive to the acute, micro, and unusual changes in the demand trend. With the recent development of artificial intelligence (AI), machine learning (ML) has become the most popular choice due to its higher accuracy based on time-, demand-, and trend-based feature extractions. Thus, we propose an Extreme Gradient Boosting (XGBoost) regression-based model—PredXGBR-1, which employs short-term lag features to predict hourly load demand. The novelty of PredXGBR-1 lies in its focus on short-term lag autocorrelations to enhance adaptability to micro-trends and demand fluctuations. Validation across five datasets, representing electrical load in the eastern and western USA over a 20-year period, shows that PredXGBR-1 outperforms a long-term feature-based XGBoost model, PredXGBR-2, and state-of-the-art recurrent neural network (RNN) and long short-term memory (LSTM) models. Specifically, PredXGBR-1 achieves an mean absolute percentage error (MAPE) between 0.98 and 1.2% and an R^2 value of 0.99, significantly surpassing PredXGBR-2's R^2 of 0.61 and delivering up to 86.8% improvement in MAPE compared to LSTM models. These results confirm the superior performance of PredXGBR-1 in accurately forecasting short-term load demand.

Keywords: electrical load forecasting; load prediction; XGBoost; regression; ML-based load prediction



Citation: Zabin, R.; Haque, K.F.; Abdelgawad, A. PredXGBR: A Machine Learning Framework for Short-Term Electrical Load Prediction. *Electronics* **2024**, *13*, 4521. <https://doi.org/10.3390/electronics13224521>

Academic Editor: Pasquale De Falco

Received: 8 October 2024

Revised: 8 November 2024

Accepted: 15 November 2024

Published: 18 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Electricity generation that aligns with fluctuating demand has long been a critical challenge for the power sector. Balancing the needs of industrial and domestic consumers while minimizing excess generation to prevent energy waste is an ongoing struggle. The rapid advancement of technology, coupled with the growing emphasis on sustainable energy, has ushered in numerous innovations within the sector. The integration of photovoltaic systems, wind energy, and other renewable sources has facilitated the development of decentralized, stand-alone grid stations [1]. However, despite these advancements, their full potential remains unrealized if system losses are not adequately mitigated. Consequently, accurate demand prediction has emerged as a key focus area for researchers and industry alike.

While the concept of load prediction is not new, having been applied to grid networks for decades, the evolution of predictive methodologies has been significant. Early approaches relied on qualitative and quantitative methods such as curve fitting, decomposition, regression analysis, and exponential smoothing. These traditional models, while

effective to an extent, gave way to more complex statistical techniques like auto regression (AR), the auto regressive moving average (ARMA), the auto regressive integrated moving average (ARIMA), and the support vector machine (SVM), all of which introduced intricate, multivariable mathematical models [2–5]. As the central grid expanded, these models became increasingly prone to NP-hard problems, exacerbating the complexity of demand forecasting.

To address this growing complexity, recent research has shifted toward data-driven, ML-based approaches, which offer the potential to significantly reduce system intricacies [6–9]. Over the past decade, ML has ascended to the forefront of predictive analytics, particularly in time series forecasting. By mimicking human learning processes, ML algorithms process vast datasets, extract features, and gain insights, offering unprecedented computational speed, accuracy, and adaptability [10]. These characteristics have made ML indispensable in a variety of practical applications, from image and handwriting recognition [11] to home automation and IoT-based smart systems, such as waste management [12]. In the realm of load forecasting, ML methods, particularly those involving supervised learning, have proven transformative. By utilizing labeled datasets for training, these methods not only streamline the prediction process but also enhance the speed and robustness of the resulting models.

Most contemporary research on electric load forecasting focuses on models like LSTM, the RNN, and the convolutional neural network (CNN), and statistical methods based on the ARIMA and the SVM. While effective, these approaches often overlook short-term, definite time-lag features—dependencies between data points over short, fixed intervals, such as the relationship between electricity demand at one hour and demand from earlier hours. These time-lagged features are crucial for capturing the immediate effects of factors like weather or peak usage that drive sudden load fluctuations. Without them, models struggle with randomness and nonlinearity, leading to less precise, generalized predictions that fail to account for short-term variations. Neglecting these temporal dynamics results in models that capture broad trends but miss critical short-term fluctuations, particularly in environments with rapidly changing demand. This lack of specificity can reduce the model's effectiveness in real-time scenarios, leading to either over- or under-generation of electricity. Incorporating definite time-lag features would enable models to better predict short-term variations, improving accuracy and reliability, and addressing a crucial gap in current forecasting methodologies. This integration could lead to more adaptive, precise load management, ensuring better efficiency in power distribution.

To address these challenges, we introduce PredXGBR-1, an XGBoost-based regression model carefully designed to incorporate short-term lag features, which substantially enhance the model's accuracy and robustness in electric load forecasting. Traditional models often overlook the immediate temporal dependencies critical for capturing sudden shifts in electricity demand. In contrast, PredXGBR-1 leverages key short-term lag features, including the mean and standard deviation of load data over recent 6, 12, and 24 h periods, to monitor and adapt to these rapid demand changes. By focusing on these short-term intervals, PredXGBR-1 effectively captures the impact of transient factors—such as abrupt weather fluctuations, spontaneous industrial activity spikes, or peak residential usage—providing a detailed understanding of demand volatility that is crucial for precise, real-time forecasting. This nuanced approach allows PredXGBR-1 not only to deliver high-resolution forecasts but also to maintain resilience against unpredictable demand patterns that might otherwise lead to inefficiencies in power distribution. The model's predictive accuracy and adaptability have been rigorously validated across five diverse datasets, demonstrating that the incorporation of short-term lag features plays an indispensable role in producing precise, responsive forecasts that support dynamic and efficient power management in real-world scenarios.

In addition to the previously outlined attributes of PredXGBR-1 and PredXGBR-2, we emphasize that the primary distinction lies in their feature selection approaches, specifically focusing on different time-lag intervals to capture unique temporal dependencies in the

data. PredXGBR-1 incorporates only short-term lag features, such as the mean and standard deviation of load over recent intervals like the past 6, 12, and 24 h. This design allows it to adapt swiftly to immediate demand fluctuations, which is crucial for short-term load forecasting. Conversely, PredXGBR-2 is tailored for scenarios requiring longer forecasting horizons by integrating long-term lag features that extend to broader time frames, including the mean and standard deviation of load data over the previous 24 h and 48 h, and weekly intervals. These long-term features enable PredXGBR-2 to capture seasonal and weekly patterns that influence load demand, offering stability and improved performance for forecasts that rely on recurring patterns over time. This tailored approach for each model version explains the notable variance in their prediction accuracies across different datasets and time periods, as highlighted in our experimental results.

Summary of Contribution:

- We have developed and implemented PredXGBR-1, a short-term feature-based XG-Boost model with time-lagged features. PredXGBR-1 is designed to capture short-term fluctuations in electricity demand by leveraging data from the previous 24 h, and it has been rigorously evaluated and validated across five different datasets. The integration of time-lagged features significantly improved prediction accuracy, addressing a key gap in existing forecasting methods.
- We performed an extensive analysis to explore how different feature sets influence the performance of the model. This comparative study proved our intuition that short-term lag features are essential for enhancing predictive accuracy, especially in rapidly fluctuating demand environments.
- The proposed model demonstrated high accuracy, achieving an MAPE of 0.98–1.2% across all datasets. This result underscores the model's robustness and reliability for short-term load forecasting in diverse scenarios.
- We pledge to share the whole code repository and the dataset with the community to promote reproducibility and advancements in the field of electric load forecasting.

2. Related Works

In recent years, various approaches of short-term load forecasting have been developed, emerging as some of the most effective methods for electric load prediction. ML and deep learning (DL) models have become prominent due to their ability to handle complex data and provide more accurate forecasts, thus facilitating efficient management, economic dispatch, and scheduling of electrical loads [13]. Load forecasting techniques can generally be categorized into three main groups: statistical models, ML-based models, and hybrid models [14].

One of the most widely used techniques is based on artificial neural networks (ANNs), which have been found to be highly effective for load forecasting. Aly et al. [15] introduced six hybrid models combining ANNs with Wavelet Neural Network (WNNs) and Kalman Filtering (KF), demonstrating improvements in prediction accuracy. Similarly, Singh et al. [16] conducted a regional load forecasting study for the NEPOOL region of ISO New England, utilizing hourly temperature, humidity, and historical load data. However, this study did not consider yearly holiday schedules, which could have an impact on load prediction. A Boosted Neural Network (BooNN), an enhancement of the traditional ANN, was presented by Khwaja et al. [17]. The model reduced forecasting errors by iteratively improving predictions based on the output of previous iterations. Another popular model is LSTM, which has been widely used for accurate load forecasting. Many researchers have proposed both classic and hybrid models involving LSTM. Marino et al. [18] compared conventional LSTM with a Sequence-to-Sequence (S2S) architecture for individual building-level load forecasting, while Ageng et al. [19] designed an hourly load forecasting model for domestic households, combining LSTM with advanced data preparation strategies. This work also considered the segmentation of a day into patterns such as weekends and weekdays, while addressing data quality issues through interpolation and de-noising.

In a comparative study, Ogunjuyigbe et al. [20] evaluated multiple load forecasting models, including Multiple Linear Regression (MLR), the seasonal auto regressive integrated moving average with exogenous variables (SARIMAX), and LSTM. The study highlighted the limitations of univariate approaches, such as the exclusion of holidays, weather conditions, and climate data. Mubashar et al. [21] conducted another comparison, where different classical models were tested with real-time series data. The study confirmed the superiority of LSTM over traditional models such as exponential smoothing and the ARIMA. Bashir et al. [22] proposed a hybrid model combining the Back Propagation Neural Network (BPNN) with Prophet and LSTM. In this model, datasets were trained using the SARIMA and Prophet models, while the residual nonlinear data were trained using LSTM. The outputs were linearly added together and further optimized with the BPNN model. Neeraj and Mathew [23] developed a Singular Spectrum Analysis–Long Short-Term Memory (SSA-LSTM) model, where the dataset was filtered for noise using signal processing techniques. The model's performance was compared to that of traditional ML and DL models, such as Support Vector Regression (SVR), ANNs, and Deep Belief Networks (DBNs), among others. Although no weather or holiday data were considered in the study, this group validated the model using a diverse set of data. In addition, a Discrete Particle Swarm Optimization (DPSO)-LSTM approach was introduced by Yang et al. [24], where DPSO was employed to optimize the selection of features, improving the model's accuracy for weekly load forecasting. However, weekend and working day distinctions were not included in this approach, and no weather forecasting data were incorporated.

In recent years, RNNs have also gained attention in the field of load forecasting. Kong and Dong [25] studied an RNN-based LSTM model to forecast electrical loads, demonstrating that aggregating individual load forecasts provided more reliable results than directly forecasting aggregated loads. The authors also explored the influence of weather forecasting on load prediction accuracy, showing that models incorporating weather data outperformed those that did not. Additionally, a 2D CNN model was examined [26,27], though it struggled to predict full-day load profiles accurately. CNNs have been applied to load forecasting with promising results. Amarasinghe et al. [18] benchmarked a classical CNN against an LSTM-(S2S) model, showing that CNN models could accurately predict peak load demand for a power station. Ibrahim and Rabelo [28] extended this work by applying various CNN structures, such as multivariate CNN, CNN-LSTM, and multi-headed CNNs. Their study found that the multivariate CNN model outperformed LSTM under both noisy and noise-free conditions.

Recent developments in regression models, particularly XGBoost, have contributed to enhanced load forecasting. Wang et al. [27] proposed a model combining linear regression for trend series and XGBoost for fluctuating sub-series, with data decomposed using variational mode decomposition (VMD) and singular value mode decomposition (SVMD). This approach was particularly effective for industrial load forecasting, where environmental factors such as temperature and holiday schedules were considered. Although the model struggled with nonlinearity and uncertainty in industrial loads, the VMD helped stabilize its performance.

In another study, Zheng et al. [29] introduced a hybrid model involving similar day (SD), empirical mode decomposition (EMD), and LSTM. This model used XGBoost to capture similarities between forecast data and historical data, further improving load prediction accuracy. SVMs have also been widely applied in load forecasting. Barman et al. [29] developed a Grasshopper Optimization Algorithm-based SVM (GOA-SVM) to minimize the deviation between forecasted and actual load curves. The GOA-SVM model was validated by comparing it to a genetic algorithm (GA)-SVM model and particle swarm optimization (PSO)-SVM model, demonstrating the effectiveness of the proposed approach using regional climate data. These relevant works are summarized in Table 1.

Table 1. Summary of related literature.

Research	Model	Contribution	Limitation
Aly et al. [15]	ANN, WNN, KF	Six clustering hybrid models combining ANN, WNN, and KF for load forecasting.	Did not account for weekday/weekend patterns.
Singh et al. [16]	Standard ANN	Regional load forecasting using historical temperature, humidity, and load data.	Excludes yearly holiday schedules; limited datasets.
Khwaja et al. [17]	Boosted ANN	Iterative minimization of forecasting error using BooNN, improving prediction accuracy.	No specific limitations mentioned.
Marino et al. [18]	LSTM, S2S	Comparison of LSTM and S2S architectures for building-level forecasting.	Only focused on a single building-level dataset.
Ageng et al. [19]	LSTM, Data Preparation	Addressed data interpolation and de-noising for household load forecasting.	Weather and atmospheric conditions not considered.
Ogunjuyigbe et al. [20]	LSTM, ARIMA	Comparative analysis of LSTM with ARIMA for improved accuracy.	Limited consideration of holidays and weather data.
Mubashar et al. [21]	MLR, ANN, SVR	Use of Gaussian filtering and validation across academic, commercial, and residential datasets.	Did not consider long holidays or special events.
Bashir & Haoyong [22]	Prophet, LSTM	Hybrid Prophet-LSTM model with residual nonlinear data trained by LSTM.	Excluded weekend/weekday patterns; limited dataset validation.
Neeraj & Mathew [23]	SSA-LSTM	Proposed SSA-LSTM model with noise reduction via signal processing.	No weather- or climate-related data; holidays not considered.
Yang et al. [24]	DPSO-LSTM	Combined DPSO algorithm with LSTM for weekly load forecasting.	Did not distinguish weekday patterns or consider weather conditions.
Kong & Dong [25]	RNN, LSTM	Demonstrated improvement in forecast accuracy when using weather data with RNN-based LSTM.	Limited historical data used in evaluation.
Amarasinghe et al. [18]	CNN	Benchmarking classical CNN against LSTM for peak load demand forecasting.	Model was not validated with diverse datasets.
Imani et al. [30]	CNN, SVR	Proposed Nonlinear Relationship Extraction (NRE) using CNN and SVR for load-temperature correlation.	Socio-demographic data and household occupancy not considered.

Table 1. Cont.

Research	Model	Contribution	Limitation
Alhussein et al. [31]	CNN-LSTM	Proposed hybrid CNN-LSTM model for feature extraction and sequence learning.	Did not consider socio-demographic data or household occupancy.
Wang et al. [27]	XGBoost, VMD, SVM	Hybrid XGBoost model combined with trend analysis using VMD for industrial load forecasting.	Model not tested on multiple datasets.
Zheng et al. [29]	LSTM, XG-Boost, EMD	Developed a hybrid model combining EMD, LSTM, and XGBoost for similarity-based forecasting.	No major limitations reported.
Barman et al. [32]	GOA-SVM, GA-SVM, PSO-SVM	Proposed Grasshopper Optimization Algorithm-based SVM for minimizing forecasting errors.	Did not incorporate comprehensive regional climate factors.

3. Background and Preliminaries

Traditionally, electric load forecasting, including daily load demand and long-term load prediction, has relied on statistical and probabilistic models. Notable algorithms in this domain include the ARIMA [33] and SVM [34], which have been widely used in earlier load forecasting approaches, as presented in Section 2.

3.1. ARIMA and Time Series Methods

The ARIMA model is one of the most commonly used methods in time series forecasting [33,35–37]. The ARIMA is designed to handle nonstationary time series data by differencing it to make it stationary. It has been extensively applied in fields such as digital signal processing, economic forecasting, and electric load prediction. Extensions of the ARIMA, including the auto regressive integrated moving average with exogenous variables (ARIMAX) [38,39], which incorporates exogenous variables such as weather data and the ARMA [40,41], which assumes stationary data, are also widely used in load forecasting. The working principle of the ARIMA is presented in Figure 1a. The process begins by checking whether the time series is stationary; if not, differencing or power transformation is applied to achieve stationarity. Once the series is stationary, the model identifies key parameters (P, d, q) through the analysis of the autocorrelation and partial autocorrelation functions. The coefficients are then estimated, followed by a diagnostic check to ensure the model fits the data. If the model passes, it is finalized; otherwise, adjustments are made to improve its accuracy. The ARIMA model uses time and load as its primary input parameters, making it particularly effective for time series analysis where seasonality and trends are key features. Among the ARIMA family, the ARIMAX has gained prominence in electric load forecasting due to its ability to integrate weather variables, which are crucial factors influencing load demand. However, the complexity of interconnected grids and the increasing number of variables in real-world scenarios often lead to large, complex mathematical models, making probabilistic models less practical for some applications.

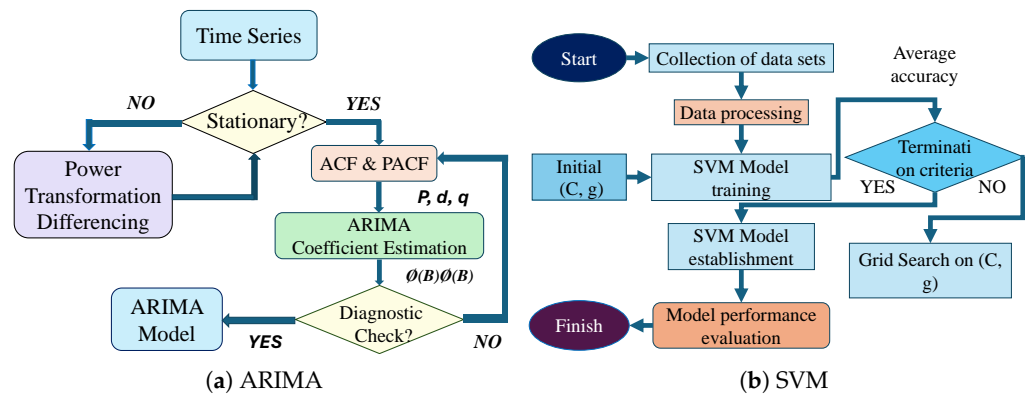


Figure 1. Main steps of ARIMA and SVM.

3.2. SVM

The SVM is another popular approach for load forecasting [42–44]. The SVM is a supervised learning algorithm that excels in both regression and classification tasks. It works by mapping input data to a high-dimensional space, enabling it to classify and predict large volumes of data efficiently. Known for its ability to prevent overfitting, the SVM maximizes the margin between classes, making it particularly effective for large datasets. Kernel methods and large-margin classifiers are central to the SVM’s success in handling nonlinear data. Figure 1b depicts the main steps of SVM operation. After collecting and preprocessing the dataset, the SVM initializes hyperparameters (C , g) for the model. The SVM is then trained using the given data, and its performance is evaluated based on criteria such as accuracy. If the model meets the accuracy criteria, it is finalized. Otherwise, a grid search is conducted to optimize the hyperparameters before retraining to improve performance. The SVM has found applications in a wide range of areas, including time series prediction, feature selection, solar and wind energy prediction, lake water level forecasting, and more [45–47]. However, the SVM’s scalability can become a limitation when handling the large, interconnected systems typical in modern electric load forecasting.

3.3. DL Approaches

DL, a subset of ML, has emerged as a powerful tool for electric load forecasting. DL models, particularly neural networks, consist of multiple layers that automatically learn features from data. These layers, often referred to as “deep” due to the depth of the network, allow the model to capture complex patterns in the data.

3.3.1. RNN

RNNs are connectionist models designed to process sequential data. Unlike traditional feedforward networks, RNNs maintain internal memory, enabling them to process historical data effectively. This makes them well suited for tasks like electric load forecasting, where past load data are crucial for making predictions [48–50]. Despite their strengths, RNNs were initially difficult to train, limiting their widespread use until recent advancements in training techniques. RNNs are designed to handle sequential data by having connections that loop back within the network. As shown in Figure 2a, the RNN takes an input sequence X and passes it through recurrent layers, where each output O at time step i depends on both the current input x_i and the previous output O_{i-1} , with shared weights w . This recurrent mechanism allows RNNs to capture temporal dependencies in sequences, which makes them suitable for tasks like time series forecasting or natural language processing.

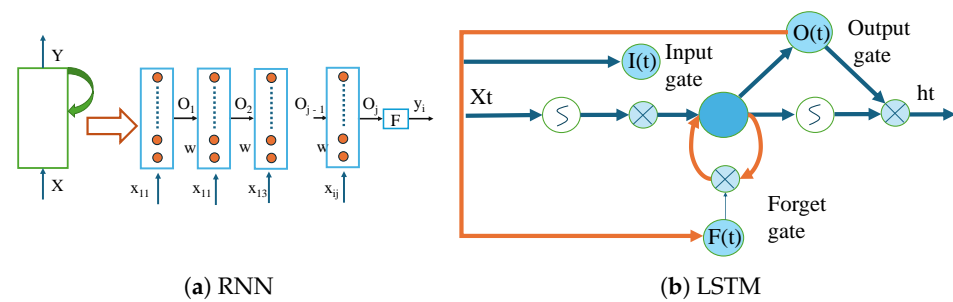


Figure 2. Main steps of RNN and LSTM.

3.3.2. LSTM

LSTM, a variant of RNN, is designed to handle long-term dependencies in data. Its central unit, known as the cell state, acts as memory, retaining relevant information while discarding unnecessary data. This architecture allows LSTM to process historical data more effectively than standard RNNs, making it ideal for applications like handwriting recognition, image processing, and, notably, electric load forecasting [51,52]. Recent developments in LSTM-based algorithms have led to robust models for electric load forecasting [53–55]. LSTM is designed to handle long-term dependencies and mitigate the vanishing gradient problem that can occur in standard RNNs. Figure 2b shows the LSTM's internal architecture, which consists of various gates. The *forget gate* decides what information to discard, the *input gate* regulates the new information added to the cell state, and the *output gate* determines the final output h_t . These gates work together to control the flow of information, making LSTM effective for learning patterns in data over longer time intervals.

3.3.3. Temporal Convolutional Networks (TCN)s

TCNs are convolutional neural network-based architectures specifically designed for sequence modeling tasks, utilizing causal convolutions to capture temporal dependencies in data without the recurrent connections found in RNNs. By stacking layers of 1D convolutions with dilation, TCNs can learn from both short- and long-range dependencies efficiently [56,57]. This makes them suitable for applications like electric load forecasting, where capturing temporal patterns is crucial. Unlike RNNs and LSTM, which rely on sequential data processing, TCNs enable parallel computation, offering lower inference times and improved computational efficiency [58].

3.3.4. Transformer

The Transformer model, introduced by Vaswani et al., revolutionized sequence modeling with its self-attention mechanism, which allows the model to focus on relevant parts of the input sequence without relying on recurrent connections [59]. This attention mechanism enables Transformers to capture dependencies at varying distances in the data, making them highly effective for tasks requiring long-term dependency modeling, such as time series forecasting and natural language processing. Transformers have demonstrated strong predictive performance in electric load forecasting by efficiently handling sequential data with both global and local dependencies [60]. The basic architecture of a Transformer comprises multiple layers of self-attention and feedforward networks, allowing it to process input sequences in parallel, reducing computational complexity and improving scalability.

3.4. XGBoost

XGBoost is a gradient-boosting algorithm known for its high efficiency and predictive accuracy. It uses decision trees as weak predictors and incorporates both linear solvers and tree learning algorithms [61]. The model builds successive trees using residual errors from previous trees, ultimately producing a more optimized prediction. Its application spans various domains, including fingerprint localization, sales forecasting, chronic disease diagnosis, and electric load forecasting [6,62–64]. Due to its ability to handle large

datasets with high computational speed, XGBoost has become a preferred model in many forecasting tasks.

While traditional models like the ARIMA and SVM have provided reasonable accuracy in load forecasting, their limitations, particularly with the increasing complexity of interconnected grids, have led to a shift toward ML-based models. Specially, LSTM and XGBoost have been shown to provide superior performance by adapting to complex data patterns and handling nonlinear relationships more effectively.

The superior performance of LSTM and XGBoost in handling nonlinear relationships arises from their unique model architectures and learning mechanisms. Specifically, LSTM networks are designed with memory cells and gating mechanisms that allow them to capture complex, nonlinear dependencies across time steps by retaining relevant information over extended sequences [65,66]. This design enables LSTM models to adapt to intricate patterns in sequential data, making them well suited for electric load forecasting [53,67]. Similarly, XGBoost leverages a gradient boosting framework, which iteratively refines an ensemble of decision trees. Each tree in this ensemble learns from the residual errors of the previous trees, thereby adapting to complex, nonlinear patterns within the data [68]. This boosting technique, combined with efficient computational methods, enables XGBoost to model nonlinear relationships effectively [63]. These theoretical foundations underscore why both LSTM and XGBoost outperform traditional linear methods, particularly in capturing the nonlinear dynamics essential for accurate load forecasting.

4. Proposed Model: PredXGBR-1

Short-term electrical load prediction poses significant challenges due to the inherent complexity of the data, which is nonlinear, nonstationary, and often imbalanced. Traditional time series forecasting methods such as the ARIMA and exponential smoothing typically fail to capture the nonlinear dependencies and dynamic patterns present in electrical load data. To address these limitations, we propose PredXGBR-1, an XGBoost-based regression model tailored for short-term load forecasting.

The PredXGBR-1 model utilizes the Extreme Gradient Boosting (XGBoost) framework, a high-performance machine learning algorithm widely recognized for its accuracy, efficiency, and scalability. XGBoost operates on the principle of gradient boosting, where an ensemble of decision trees is sequentially added, with each tree aiming to correct the errors of its predecessors. This iterative process allows the model to continuously refine its predictions, making it exceptionally powerful for capturing complex patterns in time series data, such as the short-term fluctuations in electric load.

A key strength of XGBoost lies in its use of a regularized objective function, which balances predictive accuracy with model complexity. This objective function has two main components: the loss function and the regularization term. The loss function, typically squared error for regression tasks, measures the difference between predicted and actual values, driving the model to improve accuracy. The regularization term, on the other hand, penalizes the complexity of the model, thereby preventing overfitting—a common issue in high-dimensional and noisy datasets like those in electric load forecasting.

The regularization term in XGBoost includes two critical parameters: γ , which controls the number of leaves in each tree, and α , which regulates the magnitude of the weights assigned to these leaves. The parameter γ penalizes the model for adding additional leaves, discouraging overly complex trees, while α ensures that the model does not assign excessively large weights to any leaf. Together, these parameters constrain the model's complexity, ensuring that it generalizes well to new data and remains robust across diverse forecasting scenarios. Additionally, XGBoost's use of shrinkage and subsampling during training further enhances model stability by reducing variance and improving resilience to noise.

The PredXGBR-1 model benefits from these regularization strategies by achieving a high level of accuracy while maintaining computational efficiency and avoiding overfitting. Specifically tailored for short-term load forecasting, PredXGBR-1 integrates short-term lag

features—such as the mean and standard deviation of load over recent intervals—which makes it responsive to sudden changes in demand. This targeted use of short-term features, combined with XGBoost’s regularization framework, enables PredXGBR-1 to deliver accurate, adaptive predictions in real time, which is essential for efficient energy management in modern power systems.

PredXGBR-1 leverages short-term lag features to better handle the temporal dependencies and nonlinear relationships within the data while integrating the benefits of gradient-boosting trees to address the challenges of short-term electrical load forecasting.

4.1. Challenges in Short-Term Electrical Load Forecasting

The primary challenges in short-term electrical load forecasting are the following:

- **Nonlinear Relationships:** Electrical load is influenced by various external factors such as weather conditions, time of day, and sudden shifts in demand. These complex and nonlinear relationships are difficult to model using conventional linear methods.
- **Unbalanced Data:** Load datasets are often characterized by periods of stable usage interspersed with sudden spikes or drops in consumption. This imbalance can negatively impact the performance of standard regression models.
- **Temporal Dependencies:** Load at a particular time is dependent on both short-term and long-term historical data, making the selection of features and modeling of temporal dependencies critical.

4.2. How PredXGBR-1 Addresses These Challenges

To address these challenges, PredXGBR-1 incorporates several key innovations based on the strengths of the XGBoost algorithm:

- **Tree-Based Regression:** The model employs the classification and regression tree (CART) as a base learner, enabling it to capture complex, nonlinear relationships within the data. The tree-based structure allows the model to perform well in unbalanced datasets by focusing on regions of the data with the highest residuals.
- **Boosting Mechanism:** XGBoost uses boosting to iteratively refine predictions by correcting the residual errors from previous iterations. This iterative process enables PredXGBR-1 to focus on improving short-term predictions, which are typically more volatile and difficult to forecast.
- **Feature Selection:** The model utilizes short-term lag features—mean and standard deviation of load over the prior 6, 12, and 24 h intervals—which capture the immediate temporal dependencies. This is critical in load forecasting, where short-term variations can greatly impact overall prediction accuracy.

In contrast to conventional XGBoost models, PredXGBR-1 is tailored specifically for short-term load forecasting through its strategic use of short-term lag features, which allow the model to capture immediate fluctuations and micro-trends in demand. By focusing on these recent temporal dependencies, PredXGBR-1 can respond quickly to short-term changes—such as peak demand spikes or abrupt shifts driven by weather conditions—offering a more responsive and accurate forecasting capability. This specialized feature selection makes PredXGBR-1 uniquely suited for short-term predictions, providing it with a nuanced understanding of recent trends that standard XGBoost models, designed without this emphasis on short-term lags, may overlook. As a result, PredXGBR-1 outperforms traditional approaches in short-term load forecasting, as demonstrated by its consistently lower MAPE and higher R^2 values across datasets.

To address different temporal dependencies, our approach incorporates both short-term and long-term lag features. For the models with long-term features, we consider temporal windows that extend to broader intervals, enabling the model to capture extended seasonal patterns and weekly or monthly trends. Specifically, the long-term lag features include aggregate statistics (mean and standard deviation) of load data over the previous 24 h and 48 h, and weekly intervals. These broader windows provide essential context for

models tasked with longer forecasting horizons, allowing them to adapt to recurring cycles in the data.

In contrast, PredXGBR-1 focuses on short-term lag features using intervals from the previous 6, 12, and 24 h. This design choice emphasizes recent data, optimizing the model for accurate short-term forecasting by capturing immediate, short-duration fluctuations.

4.3. Model Structure and Formalization

PredXGBR-1 is a gradient-boosted tree model designed to minimize a convex loss function over time series data. Let N denote the total number of trees in the ensemble, and $f_k(x_i)$ represent the output of the k -th tree for sample x_i . The predicted value \hat{y}_{x_i} for the i -th sample is computed as the sum of the outputs of all trees:

$$\hat{y}_{x_i} = \sum_{k=1}^N f_k(x_i), \quad f_k \in \zeta \quad (1)$$

where f_k belongs to the space ζ of regression trees. Unlike decision trees used for classification, regression trees output continuous values, which are better suited for the prediction of time series data. Each tree is built iteratively, with each subsequent tree aiming to correct the errors (residuals) of the previous trees.

4.4. Illustration of the Model Structure

To illustrate the core working mechanism of PredXGBR-1, Figure 3 provides a diagrammatic view of the iterative process. The figure shows how each regression tree in the ensemble progressively adjusts the model's predictions by learning from the residuals of the previous trees. In this framework, the model begins with an initial prediction \hat{y}_0 , and each subsequent tree $f_k(x; \phi_k)$ adds its output to the prediction based on the residuals from the previous tree.

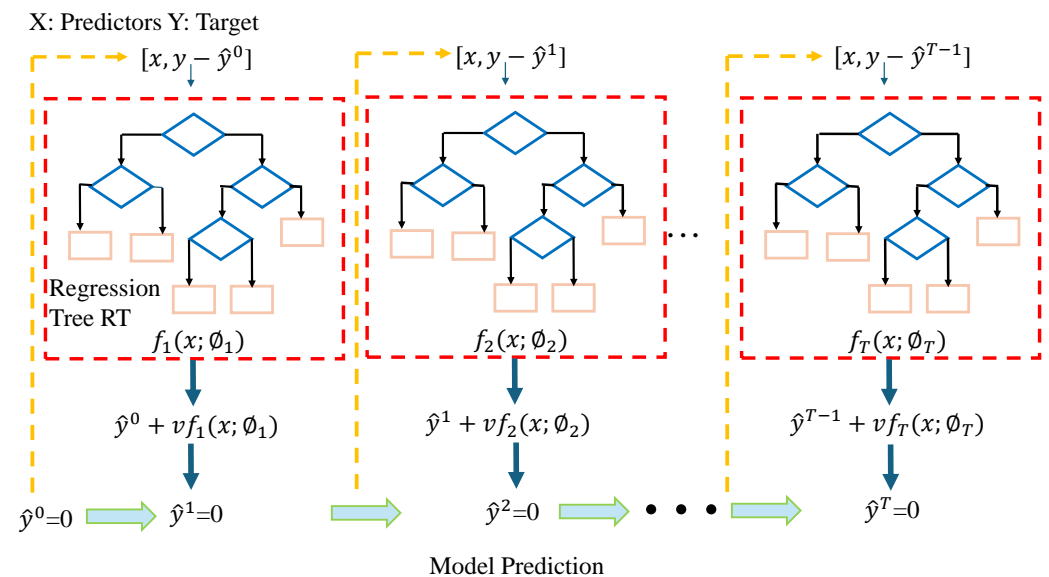


Figure 3. Working principle of the proposed PredXGBR-1 model. The model iteratively refines its prediction by minimizing residuals using successive regression trees. Each new tree improves upon the predictions of its predecessor by learning from the residuals.

In Figure 3, the first tree $f_1(x; \phi_1)$ produces an initial adjustment based on the predictors x . The second tree $f_2(x; \phi_2)$ further refines the prediction \hat{y}_1 by addressing the remaining error. This process continues iteratively until the model converges, with the final prediction \hat{y}_T representing the cumulative output of all trees in the ensemble.

4.5. Objective Function

The learning objective is to minimize a regularized loss function that balances predictive accuracy and model complexity. The objective function is defined as

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2)$$

where $l(\hat{y}_i, y_i)$ is the loss function, typically the squared loss for regression tasks:

$$l(\hat{y}_i, y_i) = \frac{1}{2}(\hat{y}_i - y_i)^2 \quad (3)$$

The term $\Omega(f_k)$ represents the regularization function, which penalizes model complexity to prevent overfitting:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\alpha \|\omega\|^2 \quad (4)$$

Here, T is the number of leaves in the tree, and ω represents the leaf weights. The regularization parameters γ and α help control the depth and weight magnitudes of the trees, respectively, ensuring that the model does not become overly complex and overfit the training data.

4.6. Leaf Weight Optimization

The optimization of the weights assigned to each leaf is critical for improving the predictive power of the model. For each leaf node j , the optimal weight ω_j^* is computed by minimizing the regularized loss function. The weight ω_j^* is given by

$$\omega_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} (h_i + \alpha)} \quad (5)$$

where g_i and h_i are the first and second derivatives of the loss function with respect to the prediction \hat{y}_i , known as the gradient and Hessian, respectively. The index I_j denotes the set of instances assigned to leaf j . This equation is derived from a second-order Taylor expansion of the loss function, allowing the model to efficiently minimize the loss while accounting for both the gradient and curvature of the objective function.

4.7. Tree Quality Evaluation

The quality of each regression tree is assessed by evaluating the reduction in the objective function after adding a new tree. The reduction in the residual sum of squares (RSS) is computed as

$$\tilde{\mathcal{L}}_t(q) = -\frac{1}{2} \sum_{j=1}^T \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} (h_i + \alpha)} + \gamma T \quad (6)$$

This equation represents the improvement in the model's predictions after the addition of each tree, with the regularization term γT acting to penalize overly complex trees.

4.8. Split Candidate Evaluation

The selection of optimal split points is crucial for building effective regression trees. To evaluate the effectiveness of a split, the gain from splitting the data into two subsets, I_L and I_R , is calculated as

$$\mathcal{L}_{\text{split}} = \frac{1}{2} \left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} (h_i + \alpha)} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} (h_i + \alpha)} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} (h_i + \alpha)} \right] - \gamma \quad (7)$$

The gain reflects the improvement in the objective function as a result of the split. A higher gain indicates a better split, leading to more accurate predictions and an overall reduction in prediction error.

4.9. Model Generalization and Overfitting Control

A key strength of PredXGBR-1 is its ability to prevent overfitting through the inclusion of regularization terms in the objective function and the use of early stopping criteria. The parameters γ and α control the complexity of the trees, ensuring that the model does not overfit the training data. Additionally, by monitoring the validation error during training, the model can halt the training process if further iterations do not lead to significant improvements.

PredXGBR-1 combines the flexibility and power of XGBoost with domain-specific adaptations for short-term load forecasting. The integration of short-term lag features and tree-based regression enables the model to accurately predict short-term fluctuations in electrical load. By employing a robust regularization framework, PredXGBR-1 ensures that the model generalizes well to unseen data, making it a reliable tool for short-term load forecasting in real-world applications.

5. Datasets and Feature Extraction

To rigorously evaluate the performance of our model, we selected five diverse datasets from centralized grid stations across different regions in the USA. The selection of these datasets was based on their distinct characteristics, such as geographical location, load patterns, and time span, which together provide a comprehensive validation of our model's robustness. Table 2 summarizes each dataset along with its description and time span.

Table 2. Electrical load forecasting datasets for model evaluation.

Dataset	Description	Time Span
PJM—PJM Interconnection LLC [69]	Regional transmission organization (RTO) in the USA, serving Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, etc.	1998–2002
PJME—PJM East [70]	Hourly data from the PJM East region.	2001–2018
PJMW—PJM West [71]	Hourly data from the PJM West region.	2001–2018
AEP—American Electric Power (AEP) [72]	A major investor-owned electric utility in the USA, delivering electricity across 11 states.	2004–2018
Dayton—Dayton Power and Light Company [73]	Serving over 500,000 customers within a 6000-square-mile (16,000 km ²) area in West Central Ohio, around Dayton.	2004–2018

5.1. Data Preprocessing

To ensure the quality and consistency of the data used in our model, we employed a systematic preprocessing approach to address issues such as missing values, timestamp inconsistencies, and temporal variations in load patterns. These preprocessing steps are detailed below, highlighting the methods and rationale behind each step to illustrate their impact on the model's performance:

- **Handling Missing Values:**

Missing entries in the datasets were primarily due to interruptions in data logging or transmission errors, which can introduce biases and disrupt model training. To address this, we applied two techniques: linear interpolation and forward filling. For extended periods with missing data, linear interpolation was used to generate intermediate values based on surrounding data points, creating smooth transitions and preserving underlying trends. This approach was particularly useful for restoring temporal continuity over multiple time steps. For isolated missing points, forward

filling was employed, where the last available valid value was repeated to fill gaps. This method was beneficial for short, intermittent gaps, ensuring the continuity of time series patterns without distorting the data structure. By addressing missing values, we maintained the temporal integrity of the data, allowing the model to capture continuous patterns in load fluctuations accurately.

- **Organizing Data in Temporal Order:**
Accurate temporal sequencing is essential for load forecasting models, especially those that rely on time-lagged features to capture dependencies over time. To achieve this, we standardized all timestamps across datasets to a 24 h format using Python's `datetime` module, which enabled us to resolve inconsistencies, such as incorrect AM/PM labels. These inconsistencies, if left unaddressed, could lead to misalignment of hourly data, resulting in misleading trends and poor model performance. After standardizing the timestamps, we sorted records in ascending order by hour, ensuring that each observation followed a natural temporal progression. This careful sequencing allowed our model to accurately interpret time-dependent features and reliably capture the structure of load demand patterns.
- **Removing Duplicate Entries:**
Duplicate entries in time series data can skew the model's learning process by overemphasizing certain observations, potentially leading to biased predictions. We performed a systematic search for duplicate records within each dataset, focusing on entries with identical timestamps and load values. Once identified, these duplicates were removed to ensure that each data point represented a unique, distinct observation. This step preserved the dataset's integrity, enabling the model to generalize well by learning from an unbiased representation of historical load patterns.
- **Segmentation of Peak and Off-Peak Hours:**
Load demand often fluctuates significantly between peak and off-peak periods, driven by factors such as residential and industrial activity levels. To capture these fluctuations, we segmented each day's load data into peak and off-peak hours. Specifically, we recorded the maximum load observed during peak hours (typically between 5 p.m. and 9 p.m.) and the minimum load during off-peak hours (usually from midnight to early morning). This segmentation helped the model to distinguish between periods of high and low demand, enhancing its ability to forecast accurately across different times of the day. By providing the model with these segmented values, we enabled it to capture and adapt to the distinct patterns characteristic of peak and off-peak demand, which are critical for short-term load forecasting accuracy.
- **Resolving AM/PM Inconsistencies:**
Time inconsistencies related to AM/PM formatting were common in some datasets and could interfere with temporal ordering. For instance, an entry incorrectly marked as "PM" instead of "AM" could cause significant deviations in the load pattern analysis, leading to inaccurate predictions. Using the `datetime` module, we converted all timestamps to a uniform 24 h format, thus eliminating ambiguity and ensuring that each record corresponded to the correct time of day. This consistency allowed the model to extract reliable time-dependent features and improved its capability to capture daily load cycles accurately.

Through these comprehensive preprocessing steps, we established a high-quality, consistent foundation for our model's training and evaluation phases. By addressing critical issues such as missing values, duplicate entries, and timestamp inconsistencies, we minimized noise and biases in the data. This systematic preprocessing framework enabled our model, *PredXGBR-1*, to accurately capture short-term fluctuations in load demand, thus enhancing its robustness and predictive power in real-world applications.

5.2. Feature Extraction and Analysis

5.2.1. Seasonal Decomposition

The time series analysis of load consumption has been further explored using *Seasonal Decomposition* to break down the data into its core components: *trend*, *seasonal*, and *residual*. These components provide a deeper insight into the patterns of consumer load behavior over the years. Seasonal decomposition typically employs a moving average method to extract the *trend*, which reflects the long-term direction of the load consumption. The *seasonal* component isolates the recurring patterns that happen at specific times, such as daily, monthly, or yearly cycles, and the *residual* represents the remaining fluctuations once both the trend and seasonal effects have been removed. These residuals help capture any irregularities or anomalies in the data. Figure 4 illustrates the original data along with the *trend*, *periodic*, and *residual* patterns of electrical load consumption for the PJM and Dayton datasets, which are shown here as representative examples due to space limitations, although additional datasets were also analyzed.

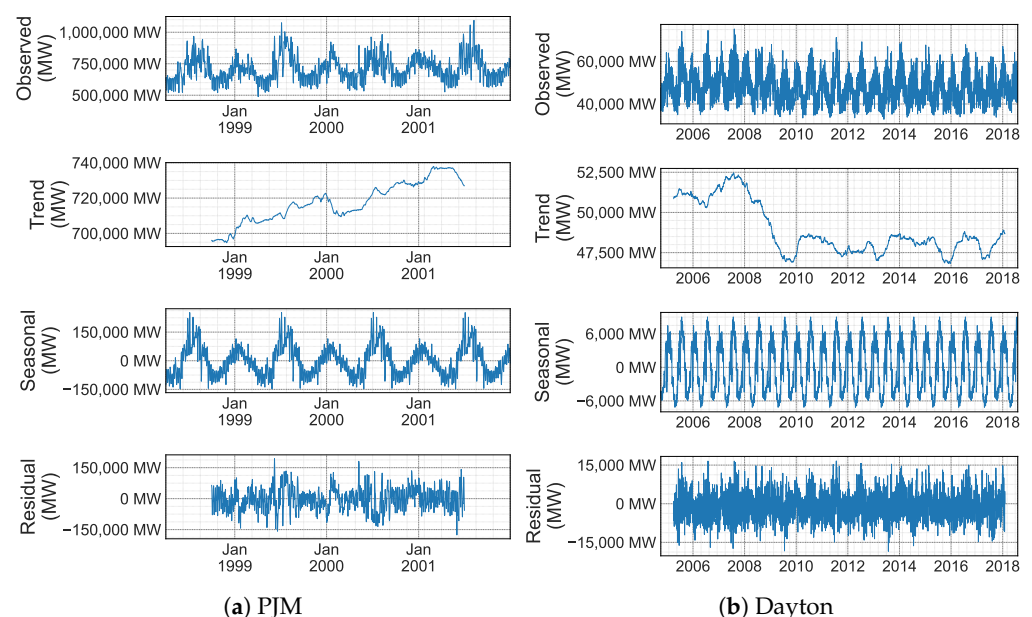


Figure 4. The original data along with the *trend*, *periodic*, and *residual* patterns of electrical load consumption for the PJM and Dayton datasets

We can observe distinct differences in the load consumption behavior between the *PJM* and *Dayton* datasets. The *PJM dataset* exhibits a steady upward trend in load consumption over the years, reflecting a general increase in energy demand. The seasonal component also shows consistent periodic fluctuations, which are likely influenced by recurring yearly or seasonal cycles. The residuals, while present, do not show significant deviations, indicating relatively stable patterns beyond the trend and seasonality. In contrast, the *Dayton dataset* reveals more variable behavior. The trend shows less consistency, with noticeable shifts, particularly between 2008 and 2010, where a decline in load consumption is evident. Additionally, the seasonal component appears more irregular in amplitude compared to PJM, suggesting more pronounced or unpredictable seasonal effects in the Dayton area. The residuals in the Dayton dataset also display greater variability, highlighting more frequent anomalies or short-term fluctuations that cannot be attributed to trend or seasonality.

Moreover, a gradual upward trend, or gentle acclivity, is observed in the *PJM dataset*, indicating a steady increase in load consumption over time. In contrast, the *AEP dataset* reveals an opposing pattern, where the load consumption trend shows a more declining or stagnant behavior compared to PJM. For the *PJMW dataset*, a sharp rise in load consumption is noticeable between 2005 and 2006, reflecting a sudden increase in energy demand during

this period. However, this trend reverses between 2008 and 2010, where a significant decline is evident.

The variation in load consumption trends across these datasets can be attributed to several factors, including changes in human behavior, regional weather conditions, and the broader climate patterns of the respective areas. These factors exert a significant influence on energy usage, highlighting how external conditions and social habits shape the overall trend in energy consumption over time.

5.2.2. Temporal Features of Electric Load Consumption

In this study, three critical features are used to understand the electric load consumption patterns: *hour of the day*, *day of the week*, and *month of the year*. These features offer insights into how electricity demand fluctuates over different temporal dimensions:

- **Hour of the Day:**
This feature captures how electricity consumption varies throughout the 24 h daily cycle. It provides a granular view of consumption patterns on an hourly basis, which is crucial for identifying peak and off-peak hours. Typically, demand is lower during late night and early morning hours (11 p.m. to 7 a.m.), when most residential, commercial, and industrial activities are minimal. Conversely, consumption often peaks during the morning and early evening, when individuals and businesses are most active. This feature allows for a detailed examination of daily demand cycles and helps in load forecasting and grid management.
- **Day of the Week:**
The *day of the week* feature distinguishes between weekdays and weekends, capturing the variation in electricity demand that occurs based on the socio-economic activities typically scheduled during the week. Weekdays (Monday through Friday) usually show higher and more stable demand patterns due to the regular operation of industries, offices, and commercial establishments. Weekends, on the other hand, may exhibit a drop in demand, particularly in commercial and industrial sectors, though residential consumption may fluctuate depending on lifestyle habits.
- **Month of the Year:**
This feature reflects the seasonal variation in electric load consumption over the twelve months, providing insight into how different times of the year impact electricity demand. Seasonal changes drive consumption patterns, with summer months (e.g., July, August) generally showing higher demand due to increased use of cooling systems, while winter months (e.g., December, January) may reflect higher consumption due to heating needs. Transitional seasons, such as fall and spring, tend to exhibit lower and more stable consumption levels compared to the extremes of summer and winter. Analyzing monthly data helps understand the impact of climatic conditions on load demand, allowing for better planning and resource allocation.

This detailed examination of the *hour of the day*, *day of the week*, and *month of the year* features allows for a comprehensive understanding of how electricity consumption fluctuates on multiple temporal scales, which is critical for improving load forecasting models and enhancing grid efficiency. Due to space limitations, Figures 5 and 6 present heatmaps of the PJM and Dayton datasets as representative examples of temporal feature analysis, although other datasets were also considered.

The analysis of load consumption patterns reveals consistent trends across both the PJM and Dayton datasets, particularly in terms of temporal features. **Hourly load variations** are similar between the two regions, with off-peak hours (11:00 p.m. to 7:00 a.m.) showing the lowest demand. This period corresponds to reduced activity in residential, commercial, and industrial sectors. Peak consumption occurs between 11:00 a.m. and 7:00 p.m., driven by daytime activities. These hourly patterns are consistent over multiple years, indicating stable consumption behaviors across both datasets.

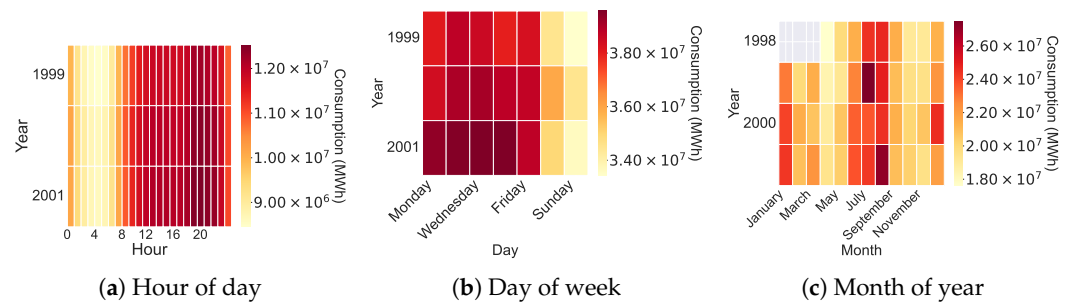


Figure 5. Heatmaps of different temporal features of PJM dataset.

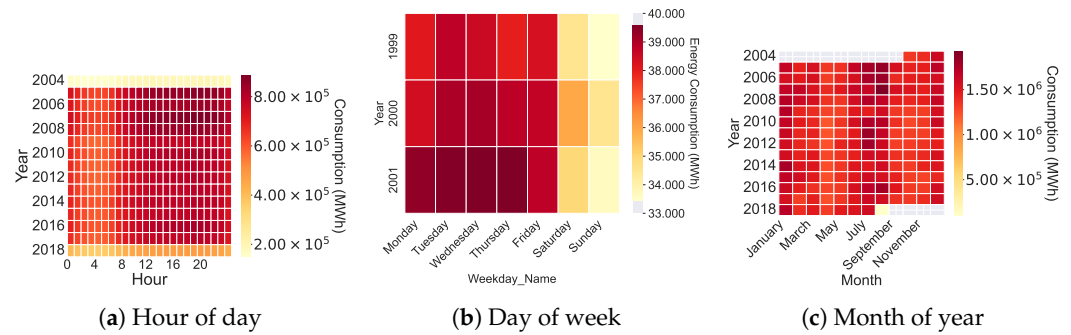


Figure 6. Heatmaps of different temporal features of Dayton dataset.

In terms of **weekly consumption patterns**, although not explicitly visualized in the data, it is likely that both datasets follow typical trends, with higher electricity demand observed during weekdays due to increased industrial and commercial activity. Weekends are expected to show relatively lower demand, predominantly reflecting residential consumption, aligning with common load distribution patterns.

Regarding **seasonal variations**, both datasets exhibit notable changes in consumption across the months of the year. Peak demand is observed during the summer months (July and August), driven primarily by cooling needs. Conversely, the fall months (September and October) show a reduction in load due to milder weather conditions. While the overall trends remain consistent, the PJM dataset exhibits more pronounced seasonal fluctuations compared to Dayton, potentially due to regional differences in population density and economic activity.

6. Performance Evaluation

6.1. Evaluation Metrics

To rigorously evaluate the performance of PredXGBR-1 in short-term load forecasting, we employed the MAPE and coefficient of determination (R^2) as our primary metrics. These metrics were chosen for their interpretability and relevance in assessing predictive accuracy and goodness-of-fit, which are essential for reliable electric load forecasting.

The MAPE provides a measure of prediction accuracy in terms of percentage, making it a useful metric for understanding relative forecasting errors. For each dataset, the MAPE is calculated as follows:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (8)$$

where y_i is the actual load demand for the i -th instance, \hat{y}_i is the predicted load demand for the i -th instance, and n is the total number of instances in the dataset. This metric captures the average percentage deviation between the predicted and actual values, allowing us to compare model performance across datasets with varying load profiles. Lower MAPE values indicate higher accuracy in the model's ability to forecast short-term fluctuations in load demand.

The coefficient of determination (R^2) provides an indication of how well the model's predictions match the actual load demand values. It is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (9)$$

where y_i is the actual load demand, \hat{y}_i is the predicted load demand, and \bar{y} is the mean of the actual load demand values. The R^2 metric captures the proportion of variance in the actual load that is predictable from the model's outputs, with values closer to 1 indicating a better fit.

For each dataset, MAPE and R^2 values were computed to assess the model's performance across different temporal patterns, providing a comprehensive evaluation of the accuracy and reliability of PredXGBR-1. This evaluation approach, applied across five datasets representing diverse load profiles, allowed us to validate the robustness and generalizability of our model.

6.2. Optimal Parameter Selection

To achieve optimal performance for the PredXGBR-1 model in short-term load forecasting, we carried out a systematic hyperparameter tuning process, focusing on adjusting critical parameters that influence model accuracy and efficiency. This process involved a grid search approach combined with cross-validation, allowing us to test a range of values for each parameter and identify the best configuration.

Key parameters tuned for PredXGBR-1 included the learning rate, maximum depth of trees, number of estimators, and regularization terms. The learning rate was varied between 0.01 and 0.3 to balance convergence speed and stability. We tested the maximum depth of trees between 3 and 10 to control model complexity and mitigate the risk of overfitting, particularly for datasets with diverse patterns. For the number of estimators, we explored values from 50 to 300, enhancing model robustness while considering computational efficiency. Regularization parameters, such as gamma (γ) and alpha (α), were adjusted to further prevent overfitting by controlling tree complexity and the magnitude of leaf weights.

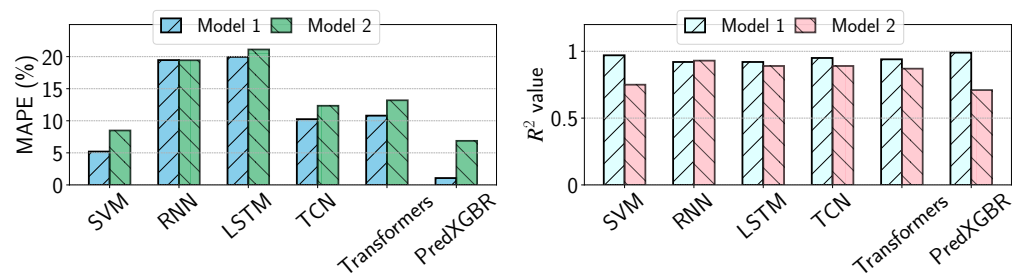
Each parameter configuration was evaluated using five-fold cross-validation, providing a rigorous assessment of model performance across different data subsets. This cross-validation approach ensured that the final parameter set selected offered the best balance between accuracy and generalizability, thereby reducing the risk of overfitting.

6.3. PredXGBR Performance Across Different Datasets

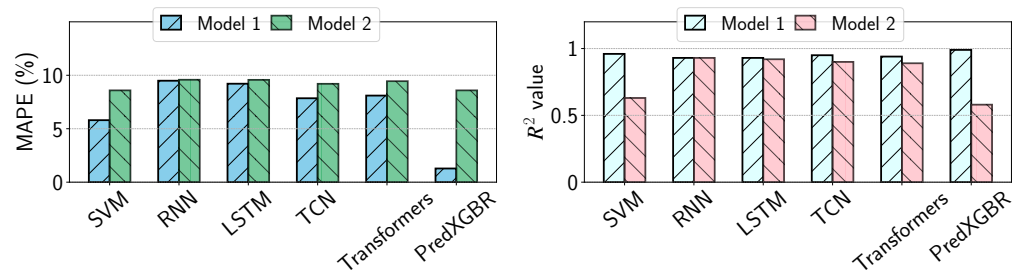
We conducted a comprehensive performance evaluation of the proposed short-term lag-based XGBoost model, PredXGBR-1, alongside baselines including SVM, TCN, RNN, LSTM, and Transformer models. This evaluation, conducted across five datasets (PJM [69], PJME [70], PJMW [71], AEP [72], and Dayton [73]), utilized MAPE and R^2 metrics to assess the accuracy of short-term (Model1) and long-term (Model2) lag-based predictions. The visualization in Figure 7 further supports these insights.

The results demonstrate that PredXGBR-1 consistently achieves the lowest MAPE and highest R^2 values across all datasets, indicating its superior performance in short-term forecasting. For instance, in the PJM dataset, PredXGBR-1 attains an MAPE of 1.07%, significantly outperforming both the SVM and TCN, which have MAPE values of 5.13% and 19.46%, respectively. This represents an approximate 79% improvement over the SVM, highlighting PredXGBR-1's ability to capture short-term micro-trends effectively.

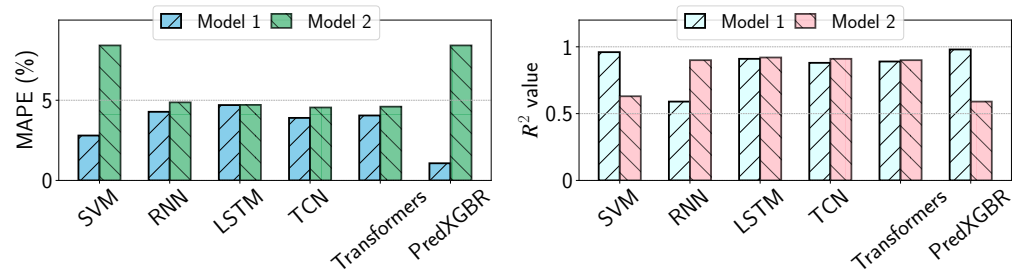
Additionally, PredXGBR-1 excels in R^2 performance, achieving values close to 0.99 in datasets like PJM and AEP. In contrast, TCN and Transformer models, while competitive, are less responsive to short-term demand shifts due to their inherent design, which leans toward long-term data patterns. This contrast underscores PredXGBR-1's strength in reacting to immediate fluctuations, an essential characteristic for short-term electric load forecasting.



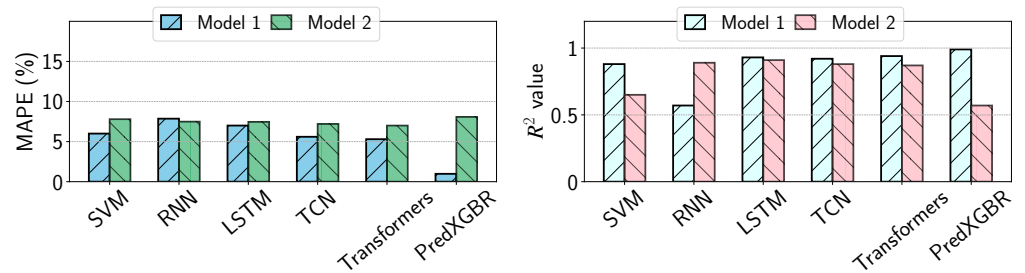
(a) PJM dataset (1998–2002)



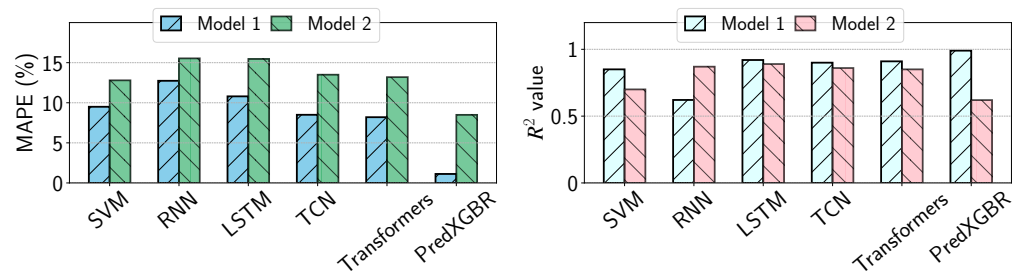
(b) PJME dataset (2001–2018)



(c) PJMW dataset (2001–2018)



(d) AEP dataset (2004–2018)



(e) Dayton dataset (2004–2018)

Figure 7. Comparative analysis of the MAPE and R^2 value of the proposed approach: PredXGBR-1.

The TCN and Transformer models, though effective, do not surpass PredXGBR-1 in capturing rapid, short-term changes in demand, as seen in more volatile datasets like PJMW. While TCN's sequence-processing architecture offers advantages over RNNs and LSTM, it remains less accurate than PredXGBR-1 in settings requiring high sensitivity to micro-trends. Similarly, Transformer models excel in complex, long-sequence patterns but are comparatively less effective in addressing immediate changes in short-term load demand.

The observed differences in model performance across datasets, especially for the RNN model, can be attributed to the unique temporal characteristics and demand patterns within each dataset. RNNs, which rely on sequential dependencies to capture temporal patterns, may encounter difficulties in generalizing effectively across datasets with varying seasonality, load volatility, and trend shifts. For example, datasets like PJM and Dayton exhibit more irregular and volatile patterns, posing challenges for RNN models due to their sensitivity to sequential dependencies and susceptibility to issues like vanishing gradients. These factors can result in inconsistent performance across different datasets. In contrast, PredXGBR-1, with its XGBoost-based structure, demonstrates greater resilience to these variations. By leveraging short-term lag features and a gradient-boosting framework, PredXGBR-1 effectively captures nonlinear relationships and adapts to different demand profiles, achieving more stable and accurate predictions across diverse datasets. This adaptability underscores PredXGBR-1's robustness in handling a range of temporal characteristics, setting it apart from traditional sequential models like the RNN.

These findings, summarized in Table 3, confirm that PredXGBR-1's design, centered on short-term lag features, not only yields precise and stable forecasts but also enhances performances across various datasets. This makes PredXGBR-1 a robust and responsive solution for real-world applications where adaptability and accuracy are paramount.

Table 3. Detailed results for different models and datasets with MAPE and R^2 values for Model1 (short-term lag) and Model2 (long-term lag).

Model	Dataset	Model1 MAPE	Model2 MAPE	Model1 R^2	Model2 R^2
SVM	PJM	5.13	6.87	0.96	0.71
	PJME	5.80	8.59	0.96	0.63
	PJMW	2.80	8.42	0.96	0.63
	AEP	6.23	8.08	0.94	0.57
	Dayton	7.36	8.49	0.93	0.62
RNN	PJM	19.46	19.44	0.92	0.93
	PJME	9.49	9.58	0.93	0.93
	PJMW	4.28	4.87	0.59	0.90
	AEP	7.86	7.49	0.57	0.89
	Dayton	12.74	15.54	0.62	0.87
LSTM	PJM	19.96	21.12	0.92	0.89
	PJME	9.21	9.57	0.93	0.92
	PJMW	4.70	4.71	0.91	0.92
	AEP	7.00	7.46	0.93	0.91
	Dayton	10.80	15.46	0.92	0.89
TCN	PJM	19.46	19.44	0.92	0.93
	PJME	7.85	9.20	0.95	0.90
	PJMW	3.90	4.55	0.88	0.91
	AEP	7.86	7.49	0.57	0.89
	Dayton	12.74	15.54	0.62	0.87

Table 3. Cont.

Model	Dataset	Model1 MAPE	Model2 MAPE	Model1 R^2	Model2 R^2
Transformer	PJM	19.96	21.12	0.92	0.89
	PJME	8.10	9.45	0.94	0.89
	PJMW	4.05	4.60	0.89	0.90
	AEP	7.00	7.46	0.93	0.91
	Dayton	10.80	15.46	0.92	0.89
PredXGBR	PJM	1.07	6.87	0.99	0.71
	PJME	1.28	8.59	0.99	0.58
	PJMW	1.07	8.42	0.98	0.59
	AEP	0.98	8.08	0.99	0.57
	Dayton	1.12	8.49	0.99	0.62

6.4. PredXGBR Generalization Performance

The generalization performance of the proposed PredXGBR-1 model, as shown in Figure 8, underscores its effectiveness in maintaining accuracy when trained on one dataset and tested on others. Across all scenarios, PredXGBR-1 achieves lower MAPE and higher R^2 values compared to the baseline models SVM-1 and TCN-1, indicating a clear advantage in adapting to unseen data.

In detail, when trained on the PJM dataset, PredXGBR-1 demonstrates substantial improvements in MAPE across all test datasets, consistently achieving values below 5%, whereas SVM-1 and TCN-1 reach MAPE values between 7% and 10% for most cases. This pattern persists when trained on other datasets such as PJME, PJMW, and AEP, where PredXGBR-1 consistently records a lower MAPE, often by a margin of 3–5% compared to SVM-1 and TCN-1. For instance, when trained on the AEP dataset, PredXGBR-1 achieves an MAPE of around 1% on other datasets, while SVM-1 and TCN-1 exhibit higher MAPE values, underscoring PredXGBR-1's robustness in preserving forecast accuracy across different test sets.

In terms of R^2 values, PredXGBR-1 maintains values close to or above 0.95 across most test scenarios, highlighting its strong correlation with actual data. In contrast, SVM-1 and TCN-1 show more variability, with R^2 values occasionally dropping below 0.9, particularly in cross-dataset scenarios. This discrepancy reinforces that PredXGBR-1 captures and generalizes short-term load variations better than the baseline models.

Overall, Figure 8 demonstrates that PredXGBR-1 not only achieves a lower MAPE but also maintains high R^2 values across diverse datasets, emphasizing its superior generalization capability. This performance makes PredXGBR-1 a more reliable choice for applications requiring stable short-term forecasting across regions with varying load characteristics.

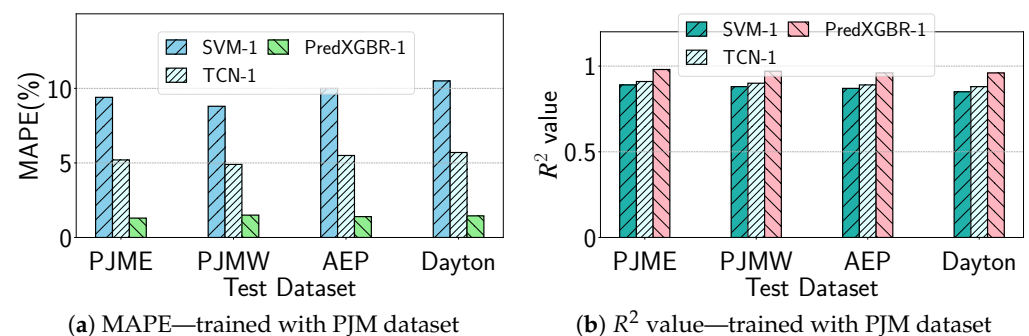


Figure 8. Cont.

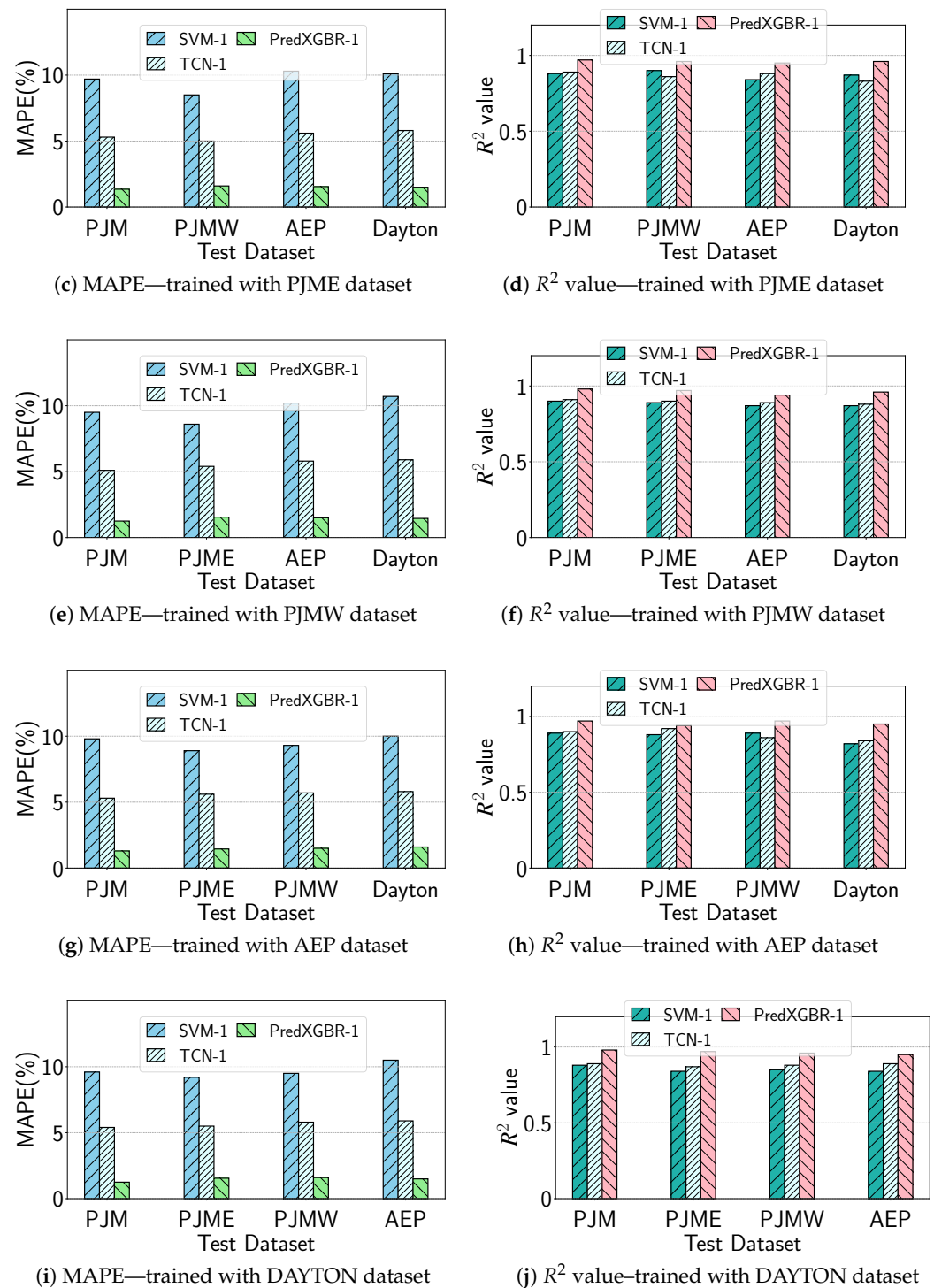


Figure 8. Analysis of the generalization performance of PredXGBR-1 when compared with two of the best-performing models—SVM and TCN. Models are trained with one dataset and tested with others.

6.5. Computational Complexity and Inference Time

In this section, we analyze the computational complexity and inference time of the evaluated models, as shown in Figure 9. The computational complexity, represented in giga floating point operations per second (GFLOPS), estimates the processing power required by each model, while inference time (in milliseconds) reflects the time taken for each model to generate predictions. These inference times were measured on a Linux-based workstation

equipped with a 12th Gen Intel(R) Core(TM) i7-12700K CPU, an NVIDIA RTX A4000 GPU, and 64 GB of RAM.

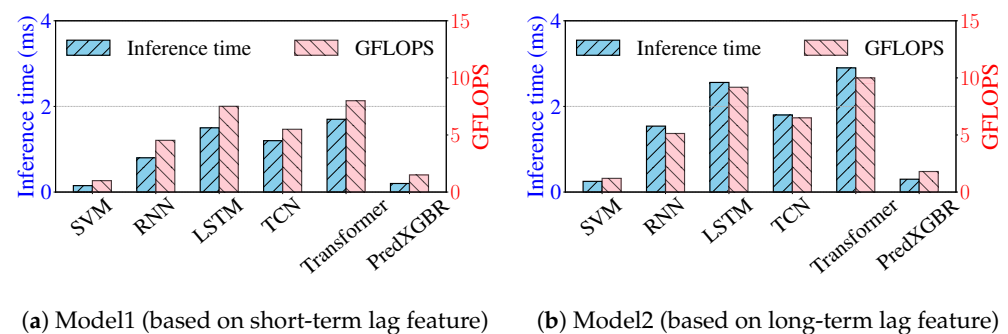


Figure 9. Comparative analysis of the computational complexity (FLOPS) and inference time of PredXGBR-1 (Model1).

The results indicate that models with long-term lag features, such as Transformer-2 and LSTM-2, demand higher computational resources and exhibit longer inference times compared to their short-term counterparts. For instance, Transformer-2 has the highest computational burden at 10 GFLOPS and an inference time of 2.9 ms, followed closely by LSTM-2 at 9.18 GFLOPS and 2.56 ms. This elevated demand stems from the need to capture extended temporal patterns (e.g., weekly and monthly trends), which increases input dimensionality and processing requirements.

Conversely, short-term lag models, particularly PredXGBR-1, are significantly more efficient. PredXGBR-1, with a computational burden of only 1.5 GFLOPS and an inference time of 0.2 ms, is approximately 93% faster in inference time and 85% less computationally demanding than Transformer-2. This efficiency makes PredXGBR-1 well suited for real-time applications, where low latency is crucial. Among neural network models, TCN-1 and SVM-1 also show competitive performance, with TCN-1 requiring 5.5 GFLOPS and an inference time of 1.2 ms, and SVM-1 being the most efficient at only 1.0 GFLOPS and an inference time of 0.15 ms, underscoring their potential for low-resource environments.

Overall, while long-term lag models provide a more comprehensive temporal context, they incur higher computational costs. Short-term lag models, particularly PredXGBR-1, achieve a balance between computational efficiency and predictive accuracy, making them ideal for scenarios requiring rapid responses and low-latency predictions.

7. Conclusions

In this paper, we introduced a short-term feature-based XGBoost model, PredXGBR-1, designed to address the limitations of traditional and probabilistic methods in electric load forecasting. By leveraging time-lagged features from the previous 24 h, our model captures short-term fluctuations in demand with remarkable accuracy. We rigorously evaluated PredXGBR-1 across five different datasets representing twenty years of electrical load data from various regions in the USA. Our comparative analysis between long-term and short-term feature-based models highlighted the significance of focusing on short-term lag features. While traditional models like the RNN and LSTM demonstrated moderate accuracy, PredXGBR-1 consistently outperformed them with an average MAPE of 0.98–1.2% and an R^2 score close to 1, signifying near-perfect predictive performance. This level of accuracy makes PredXGBR-1 a highly reliable model for real-time, short-term load forecasting, especially in environments with rapidly fluctuating demand. The findings presented in this paper underscore the robustness and adaptability of PredXGBR-1, paving the way for more efficient and precise forecasting in the electric power sector.

Author Contributions: Conceptualization, methodology: R.Z. and K.F.H.; Algorithm formation, formal analysis, resources, and data processing, R.Z.; Validation, original draft, and review preparation, R.Z., K.F.H. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets and algorithms supporting the reported results can be found at <https://github.com/rifatzabin/PredXGBR> accessed on 8 November 2024.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saqib, N.; Haque, K.F.; Zabin, R.; Preonto, S.N. Analysis of grid integrated PV system as home RES with net metering scheme. In Proceedings of the 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 10–12 January 2019; pp. 395–399.
2. Rodrigues, F.; Cardeira, C.; Calado, J.M.; Melicio, R. Short-term load forecasting of electricity demand for the residential sector based on modelling techniques: A systematic review. *Energies* **2023**, *16*, 4098. [\[CrossRef\]](#)
3. Habbak, H.; Mahmoud, M.; Metwally, K.; Fouda, M.M.; Ibrahim, M.I. Load forecasting techniques and their applications in smart grids. *Energies* **2023**, *16*, 1480. [\[CrossRef\]](#)
4. Akhtar, S.; Shahzad, S.; Zaheer, A.; Ullah, H.S.; Kilic, H.; Gono, R.; Jasiński, M.; Leonowicz, Z. Short-term load forecasting models: A review of challenges, progress, and the road ahead. *Energies* **2023**, *16*, 4060. [\[CrossRef\]](#)
5. Eren, Y.; Küçükdemiral, İ. A comprehensive review on deep learning approaches for short-term load forecasting. *Renew. Sustain. Energy Rev.* **2024**, *189*, 114031. [\[CrossRef\]](#)
6. Cordeiro-Costas, M.; Villanueva, D.; Eguía-Oller, P.; Martínez-Comesaña, M.; Ramos, S. Load forecasting with machine learning and deep learning methods. *Appl. Sci.* **2023**, *13*, 7933. [\[CrossRef\]](#)
7. Avendano, I.A.C.; Javan, F.D.; Najafi, B.; Moazami, A.; Rinaldi, F. Assessing the impact of employing machine learning-based baseline load prediction pipelines with sliding-window training scheme on offered flexibility estimation for different building categories. *Energy Build.* **2023**, *294*, 113217. [\[CrossRef\]](#)
8. Zhang, D.; Wang, S.; Liang, Y.; Du, Z. A novel combined model for probabilistic load forecasting based on deep learning and improved optimizer. *Energy* **2023**, *264*, 126172. [\[CrossRef\]](#)
9. Aduama, P.; Zhang, Z.; Al-Sumaiti, A.S. Multi-feature data fusion-based load forecasting of electric vehicle charging stations using a deep learning model. *Energies* **2023**, *16*, 1309. [\[CrossRef\]](#)
10. Zhang, L.; Wen, J.; Li, Y.; Chen, J.; Ye, Y.; Fu, Y.; Livingood, W. A Review of Machine Learning in Building Load Prediction. *Appl. Energy* **2021**, *285*, 116452. [\[CrossRef\]](#)
11. Saqib, N.; Haque, K.F.; Yanambaka, V.P.; Abdelgawad, A. Convolutional-neural-network-based handwritten character recognition: an approach with massive multisource data. *Algorithms* **2022**, *15*, 129. [\[CrossRef\]](#)
12. Haque, K.F.; Zabin, R.; Yelamarthi, K.; Yanambaka, P.; Abdelgawad, A. An IoT based efficient waste collection system with smart bins. In Proceedings of the 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2–16 June 2020; pp. 1–5.
13. Chen, K.; Chen, K.; Wang, Q.; He, Z.; Hu, J.; He, J. Short-Term Load Forecasting with Deep Residual Networks. *IEEE Trans. Smart Grid* **2018**, *10*, 3943–3952. [\[CrossRef\]](#)
14. Hammad, M.A.; Jereb, B.; Rosi, B.; Dragan, D. Methods and Models for Electric Load Forecasting: A Comprehensive Review. *Logist. Sustain. Transp.* **2020**, *11*, 51–76. [\[CrossRef\]](#)
15. Aly, H.H. A Proposed Intelligent Short-Term Load Forecasting Hybrid Models of ANN, WNN and KF based on Clustering Techniques for Smart Grid. *Electr. Power Syst. Res.* **2020**, *182*, 106191. [\[CrossRef\]](#)
16. Singh, S.; Hussain, S.; Bazaz, M.A. Short Term Load Forecasting using Artificial Neural Network. In Proceedings of the 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 21–23 December 2017; pp. 1–6.
17. Khwaja, A.; Zhang, X.; Anpalagan, A.; Venkatesh, B. Boosted Neural Networks for Improved Short-Term Electric Load Forecasting. *Electr. Power Syst. Res.* **2017**, *143*, 431–437. [\[CrossRef\]](#)
18. Amarasinghe, K.; Marino, D.L.; Manic, M. Deep Neural Networks for Energy Load Forecasting. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1483–1488.
19. Ageng, D.; Huang, C.Y.; Cheng, R.G. A Short-Term Household Load Forecasting Framework using LSTM and Data Preparation. *IEEE Access* **2021**, *9*, 167911–167919. [\[CrossRef\]](#)
20. Ogunjuyigbe, A.S.; Ayodele, T.R.; Lasarus, C.P.; Yusuff, A.A.; Mosetlhe, T.C. Comparative Analysis of Short-Term Load Forecasting Methods. In Proceedings of the 2021 IEEE AFRICON, Arusha, Tanzania, 13–15 September 2021; pp. 1–6.
21. Mubashar, R.; Awan, M.J.; Ahsan, M.; Yasin, A.; Singh, V.P. Efficient residential load forecasting using deep learning approach. *Int. J. Comput. Appl. Technol.* **2022**, *68*, 205–214. [\[CrossRef\]](#)
22. Bashir, T.; Haoyong, C.; Tahir, M.F.; Liqiang, Z. Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN. *Energy Rep.* **2022**, *8*, 1678–1686. [\[CrossRef\]](#)
23. Neeraj, N.; Mathew, J.; Agarwal, M.; Behera, R.K. Long short-term memory-singular spectrum analysis-based model for electric load forecasting. *Electr. Eng.* **2021**, *103*, 1067–1082. [\[CrossRef\]](#)
24. Yang, J.; Zhang, X.; Bao, Y. Short-term load forecasting of central China based on DPSO-LSTM. In Proceedings of the 2021 IEEE 4th International Electrical and Energy Conference (CIEEC), Wuhan, China, 28–30 May 2021; pp. 1–5.

25. Nespoli, A.; Ogliari, E.; Pretto, S.; Gavazzeni, M.; Vigani, S.; Paccanelli, F. Electrical load forecast by means of lstm: The impact of data quality. *Forecasting* **2021**, *3*, 91–101. [\[CrossRef\]](#)
26. Orr, M. Short-Term Electrical Load Forecasting for Irish Supermarkets with Weather Forecast Data. Ph.D. Thesis, National College of Ireland, Dublin, Ireland, 2021.
27. Wang, Y.; Sun, S.; Chen, X.; Zeng, X.; Kong, Y.; Chen, J.; Guo, Y.; Wang, T. Short-term load forecasting of industrial customers based on SVM and XGBoost. *Int. J. Electr. Power Energy Syst.* **2021**, *129*, 106830. [\[CrossRef\]](#)
28. Ibrahim, B.; Rabelo, L. A deep learning approach for peak load forecasting: A case study on panama. *Energies* **2021**, *14*, 3039. [\[CrossRef\]](#)
29. Zheng, H.; Yuan, J.; Chen, L. Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. *Energies* **2017**, *10*, 1168. [\[CrossRef\]](#)
30. Imani, M. Electrical load-temperature CNN for residential load forecasting. *Energy* **2021**, *227*, 120480. [\[CrossRef\]](#)
31. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access* **2020**, *8*, 180544–180557. [\[CrossRef\]](#)
32. Barman, M.; Choudhury, N.D.; Sutradhar, S. A regional hybrid GOA-SVM model based on similar day approach for short-term load forecasting in Assam, India. *Energy* **2018**, *145*, 710–720. [\[CrossRef\]](#)
33. Chodakowska, E.; Nazarko, J.; Nazarko, L. Arima models in electrical load forecasting and their robustness to noise. *Energies* **2021**, *14*, 7952. [\[CrossRef\]](#)
34. Mohandes, M. Support vector machines for short-term electrical load forecasting. *Int. J. Energy Res.* **2002**, *26*, 335–345. [\[CrossRef\]](#)
35. Kontopoulou, V.I.; Panagopoulos, A.D.; Kakkos, I.; Matsopoulos, G.K. A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet* **2023**, *15*, 255. [\[CrossRef\]](#)
36. Sharma, R.R.; Kumar, M.; Maheshwari, S.; Ray, K.P. EVDHM-ARIMA-based time series forecasting model and its application for COVID-19 cases. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 6502210. [\[CrossRef\]](#)
37. Sirisha, U.M.; Belavagi, M.C.; Attigeri, G. Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison. *IEEE Access* **2022**, *10*, 124715–124727. [\[CrossRef\]](#)
38. Atabay, F.V.; Pagkalinawan, R.M.; Pajarillo, S.D.; Villanueva, A.R.; Taylar, J.V. Multivariate time series forecasting using arimax, sarimax, and rnn-based deep learning models on electricity consumption. In Proceedings of the 2022 3rd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey, 15–16 December 2022; pp. 1–6.
39. Ahn, E.; Hur, J. A short-term forecasting of wind power outputs using the enhanced wavelet transform and arimax techniques. *Renew. Energy* **2023**, *212*, 394–402. [\[CrossRef\]](#)
40. He, K.; Yang, Q.; Ji, L.; Pan, J.; Zou, Y. Financial time series forecasting with the deep learning ensemble model. *Mathematics* **2023**, *11*, 1054. [\[CrossRef\]](#)
41. Song, Z.; Yang, L. Statistical inference for ARMA time series with moving average trend. *J. Nonparametr. Stat.* **2022**, *34*, 357–376. [\[CrossRef\]](#)
42. Ahmad, W.; Ayub, N.; Ali, T.; Irfan, M.; Awais, M.; Shiraz, M.; Glowacz, A. Towards short term electricity load forecasting using improved support vector machine and extreme learning machine. *Energies* **2020**, *13*, 2907. [\[CrossRef\]](#)
43. Aisyah, S.; Simaremare, A.A.; Adytia, D.; Aditya, I.A.; Alamsyah, A. Exploratory weather data analysis for electricity load forecasting using SVM and GRNN, case study in Bali, Indonesia. *Energies* **2022**, *15*, 3566. [\[CrossRef\]](#)
44. Emhamed, A.A.; Shrivastava, J. Electrical load distribution forecasting utilizing support vector model (SVM). *Mater. Today Proc.* **2021**, *47*, 41–46. [\[CrossRef\]](#)
45. Pant, M.; Kumar, S. Fuzzy time series forecasting based on hesitant fuzzy sets, particle swarm optimization and support vector machine-based hybrid method. *Granul. Comput.* **2022**, *7*, 861–879. [\[CrossRef\]](#)
46. Gao, W.; Li, Z.; Chen, Q.; Jiang, W.; Feng, Y. Modelling and prediction of GNSS time series using GBDT, LSTM and SVM machine learning approaches. *J. Geod.* **2022**, *96*, 71. [\[CrossRef\]](#)
47. Ramadevi, B.; Bingi, K. Chaotic time series forecasting approaches using machine learning techniques: A review. *Symmetry* **2022**, *14*, 955. [\[CrossRef\]](#)
48. Aseeri, A.O. Effective RNN-based forecasting methodology design for improving short-term power load forecasts: Application to large-scale power-grid time series. *J. Comput. Sci.* **2023**, *68*, 101984. [\[CrossRef\]](#)
49. Zhao, C.; Ye, J.; Zhu, Z.; Huang, Y. FLRNN-FGA: Fractional-Order Lipschitz Recurrent Neural Network with Frequency-Domain Gated Attention Mechanism for Time Series Forecasting. *Fractal Fract.* **2024**, *8*, 433. [\[CrossRef\]](#)
50. Zhang, X.; Zhong, C.; Zhang, J.; Wang, T.; Ng, W.W. Robust recurrent neural networks for time series forecasting. *Neurocomputing* **2023**, *526*, 143–157. [\[CrossRef\]](#)
51. Misgar, M.M.; Mushtaq, F.; Khurana, S.S.; Kumar, M. Recognition of offline handwritten Urdu characters using RNN and LSTM models. *Multimed. Tools Appl.* **2023**, *82*, 2053–2076. [\[CrossRef\]](#)
52. Masood, F.; Khan, W.U.; Ullah, K.; Khan, A.; Alghamedy, F.H.; Aljuaid, H. A hybrid CNN-LSTM random forest model for dysgraphia classification from hand-written characters with uniform/normal distribution. *Appl. Sci.* **2023**, *13*, 4275. [\[CrossRef\]](#)
53. Abumohsen, M.; Owda, A.Y.; Owda, M. Electrical load forecasting using LSTM, GRU, and RNN algorithms. *Energies* **2023**, *16*, 2283. [\[CrossRef\]](#)
54. Zaboli, A.; Tuyet-Doan, V.N.; Kim, Y.H.; Hong, J.; Su, W. An lstm-sae-based behind-the-meter load forecasting method. *IEEE Access* **2023**, *11*, 49378–49392. [\[CrossRef\]](#)

55. Pavlatos, C.; Makris, E.; Fotis, G.; Vita, V.; Mladenov, V. Enhancing Electrical Load Prediction Using a Bidirectional LSTM Neural Network. *Electronics* **2023**, *12*, 4652. [CrossRef]
56. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
57. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
58. Oreshkin, B.N.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv* **2019**, arXiv:1905.10437.
59. Vaswani, A. Attention Is All You Need. *Advances in Neural Information Processing Systems*. 2017. Available online: <https://user.phil.hhu.de/~cwurm/wp-content/uploads/2020/01/7181-attention-is-all-you-need.pdf> (accessed on 20 June 2022).
60. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 11106–11115.
61. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
62. Oukhouya, H.; El Himdi, K. Comparing machine learning methods—svr, xgboost, lstm, and mlp—for forecasting the moroccan stock market. In Proceedings of the Computer Sciences & Mathematics Forum, Online, 1–5 May 2023; MDPI: Basel, Switzerland, 2023; Volume 7, p. 39.
63. Zhang, T.; Zhang, X.; Liu, Y.; Chow, Y.H.; Iu, H.H.; Fernando, T. Long-term energy and peak power demand forecasting based on sequential-XGBoost. *IEEE Trans. Power Syst.* **2023**, *39*, 3088–3104. [CrossRef]
64. Zhang, L.; Jánošík, D. Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches. *Expert Syst. Appl.* **2024**, *241*, 122686. [CrossRef]
65. Kumar, B.; Yadav, N. A novel hybrid model combining β SARMA and LSTM for time series forecasting. *Appl. Soft Comput.* **2023**, *134*, 110019. [CrossRef]
66. Kumar, I.; Tripathi, B.K.; Singh, A. Attention-based LSTM network-assisted time series forecasting models for petroleum production. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106440. [CrossRef]
67. Mounir, N.; Ouadi, H.; Jrhilifa, I. Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system. *Energy Build.* **2023**, *288*, 113022. [CrossRef]
68. Mishra, P.; Al Khatib, A.M.G.; Yadav, S.; Ray, S.; Lama, A.; Kumari, B.; Sharma, D.; Yadav, R. Modeling and forecasting rainfall patterns in India: A time series analysis with XGBoost algorithm. *Environ. Earth Sci.* **2024**, *83*, 163. [CrossRef]
69. PJM Interconnection LLC. *Regional Transmission Organization (RTO) in the USA, Serving Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, etc.* (1998–2002); PJM Interconnection LLC: Norristown, PA, USA, 2024.
70. PJM Interconnection. *PJM East Hourly Data from the PJM East Region (2001–2018)*; PJM Interconnection: Norristown, PA, USA, 2018.
71. PJM Interconnection. *PJM West Hourly Data from the PJM West Region (2001–2018)*; PJM Interconnection: Norristown, PA, USA, 2018.
72. American Electric Power (AEP). *A Major Investor-Owned Electric Utility in the USA, Delivering Electricity Across 11 States (2004–2018)*; American Electric Power (AEP): Columbus, OH, USA, 2018.
73. Dayton Power and Light Company. *Serving Over 500,000 Customers Within a 6,000-Square-Mile Area in West Central Ohio, Around Dayton (2004–2018)*; Dayton Power and Light Company: Dayton, OH, USA, 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.