SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I2P106
Volume 13 Issue 2, 68-90, February 2026

Original Article

SmartGridOptimizer-X: A Novel Energy-Efficient Design Framework for Sustainable Electrical Systems Integration

G. Ramkumar

Department of Electronics and Communication Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Chennai, India.

Corresponding Author : pgrvlsi@gmail.com

Received: 17 November 2025 Revised: 18 December 2025 Accepted: 19 January 2026 Published: 20 February 2026

Abstract - *The SmartGRidOptimizer-X is an advanced hybrid forecasting model that can precisely forecast the energy demand by incorporating Temporal Convolutional Networks (TCNs), Long Short-Term Memory (LSTM) networks, and Adaptive Gradient Boosting Machines (Adaptive-GBMs). The model is also able to tackle the issues of complex energy systems by embracing multi-scale temporal dependencies and capitalizing on contextual aspects of weather, socio-economic, and historical trends. CNLSTM and Adaptive-GBM are used with 75 and 25 percent as optimized ensemble weights, respectively, making it stronger and more accurate. The system was tested aggressively on the Energy Prediction Smart-Meter Dataset with an incredible accuracy of 98.43% (R^2 = 0.9843), Mean Absolute Error (MAE) was 0.089 kWh, and Root Mean Squared Error (RMSE) was 0.115 kWh. The model proved to be flexible under seasonal performance metrics, and the values of R^2 were 0.966 in extreme weather and 0.994 in summer. The model accuracy in detecting anomalies in energy usage was 96.3% and a recall of 94.5%, which validates the model's capability to detect patient abnormal energy use trends. The SmartGridOptimizer-X framework is a breakthrough in energy forecasting ever made and offers utility providers an efficient tool for managing the grid, load balancing, and demand response strategies.*

Keywords - *SmartGridOptimizer-X, Energy demand forecasting, Renewable energy, Grid Stability, Anomaly detection, Sustainable energy systems.*

1. Introduction

The use of electrical systems in smart grids is a significant development in the generation, distribution, and consumption of electricity. The smart grids are interactive and connect the already existing power infrastructure, especially the electrical equipment, coupled with the new type of communication systems, data management, and complete automation. These systems would focus on addressing new and multifaceted challenges of the present energy dilemma, like renewable energy, DSM, and Grid reliability [1, 2]. One of the key features of electrical systems in smart grids is focused on the fact that they enable a two-way communication between utilities and consumers. The smart grids are unlike conventional grids that operate in a top-down fashion, where minimal real-time feedback is received and encountered; the smart grids are dependent on IT interfacing to continually recollect and relay data. This ability assists the utilities in perceiving the condition of the grid, disorders, and dilemmas by identifying and correcting them accordingly; the consumers also gain knowledge on the use of energy and are able to follow it accordingly. Another important aspect that smart grid electrical systems have taken is the use of sensors and measurement [3, 4]. At various locations in the grid, smart meters and sensors are placed to monitor consumption, grid health, and equipment status. These devices provide more detailed information on energy consumption, making it easier to make accurate demand forecasting, allocate energy, and identify when there may be a fault. More noteworthy, the smart grids introduce automation and control system which optimises the supply and demand, reduce the losses, and increase the grid reliability [5, 6]. Another area that is common in innovative grid systems is renewable power, such as solar and wind power, as part of the steps taken to enhance the utilization of renewable energy.

Smart trick: variability of renewable energy is yet another problem that can be effectively addressed using the innovative technique by means of the applied advanced algorithms and storage solutions of renewable energy [7]. This not only assisted in reducing the use of fossil fuels but also in the prevention of climate change in the entire world. Therefore, smart grids are starting to deliver on a promise that electrical systems have brought with them regarding a more innovative and stronger system of electricity supply. These systems use traditional structures with modern technologies to optimize grids, encourage consumer control, and restructure the energy landscape to be sustainable [8, 9]. Nevertheless, smart grids contain electrical systems that also pose several issues that need to be addressed to introduce them to broad-based use. The fact that retrofitting old grid structures to accommodate intelligent communication, sensing, and actuation devices is relatively expensive. The nature of renewable energy sources also makes their integration more challenging, as their production can be erratic and unpredictable, introducing many changes to the grid that must be corrected with the help of intricate algorithms, not to mention energy storage systems [10, 11]. Cybersecurity is another issue due to the growing use of computers for communication and data sharing, and it can pose a threat to cyber warfare, affecting the regular operation of the grid and consumers' data. In addition, the integration of smart grids requires intensive regulatory and policy support, along with customer awareness and participation in their implementation and application. Indeed, the above-discussed limitations call for the continuous development of new solutions, funding, and partnerships to enhance the opportunities for innovative grid systems [12, 13]. Figure 1 shows the consumer protection rules on e-commerce.

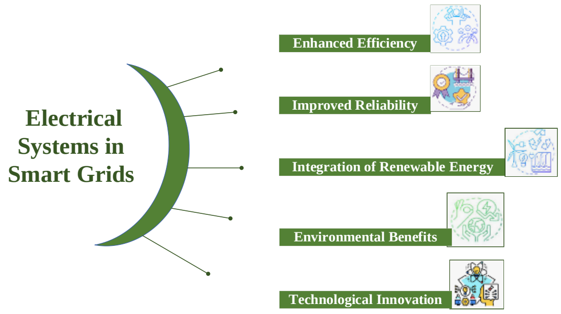

Fig. 1 Consumer protection rules on e-commerce

Due to their dynamic functionalities and far-reaching proficiency in data analysis and decision-making, Deep Learning (DL) models have been introduced as fundamental tools for improving the interconnection of electrical systems in smart grids. Probably the most active usage area is in load forecasting, where DL models are Recurrent Neural Networks (RNN) and LSTM networks are employed for analyzing data and subsequently forecasting the demand for electricity with almost perfect precision [14, 15].

This makes it possible for the utilities to properly schedule electricity generation to minimize wastage of power and, at the same time, guarantee power supply. One of the most important areas of using Deep Learning in smart grids is in fault identification and classification. CNN and other enhanced approaches of DL can handle large volumes of sensor data to look for atypical patterns, detect defects, and even forecast failures. This organized approach to the care of the grid is a more effective strategy that reduces the time taken in repair, the costs incurred in repair, and is efficient in its functionality. Also, deep learning models play an essential role in the management of renewable energy sources like solar and wind power by precise forecasting of energy production or optimization of energy storage systems, depending on the supply and demand [16].

Applying DL algorithms to patterns of energy consumption can help identify incidents suggesting energy theft or unauthorized access. This enhances revenue defense and reduces losses for the utility suppliers. Moreover, DL models are used to control the grid, including voltage control, demand and active power control, and load shedding during peak hours. However, there are challenges in applying Deep Learning to smart grids, as discussed later in this paper. The first major limitation of DL is the resource-intensive algorithms used in model training and deployment, which necessitate the use of specialized hardware. The quality and quantity of data are also paramount because neither sufficient nor noisy data yields good prediction and high performance [17, 18]. However, incorporating DL into smart grids increases cybersecurity risks because these systems involve extensive data acquisition and networking. Finally, the utilization of DL models is a complex process that still largely relies on the expertise of power system engineers and data scientists, so access is likely to be limited for midsize and many small utilities. Overall, Deep Learning is a disruptive technology that can invigorate smart grids as long as one addresses specific issues, such as computational load, data credibility, and protection.

The enhancement of energy systems based on the growth rate of people and urbanization, climate change, and an increase in the use of renewable energy has led to the expansion of intricate systems, which require efficient, optimal, and accurate forecasting to manage the different grid systems [19, 20]. It is crucial for effective load forecasting, purchasing, and preparations to reduce power failures. The existing models that are in operation cannot be used in the management of complexities that come with the time, seasons, and other contextual limitations that relate to the use of energy. In this regard, the SmartGridOptimizer-X model was designed as a hybrid forecasting framework, which utilizes the state-of-the-art machine learning methods. The model integrates the benefits of sequential pattern recognition and non-linear contextual learning by incorporating Temporal Convolutional Networks (TCNs), Long Short-Term Memory (LSTM) networks, and Adaptive Gradient Boosting Machines (Adaptive-GBMs). The system can use optimized ensemble weights to be more predictive and, at the same time, be flexible to different situations. The Energy Prediction Smart-Meter Dataset was used to test the model, with excellent performance, and this has an accuracy of 98.43 (R 2 =0.9843). This paper shows that the model has the ability to solve real-life problems in energy prediction, providing a valid solution to grid operators and energy suppliers.

1.1. Main Contributions of the Work

Hybrid Forecasting Architecture: The SmartGridOptimizer-X model is developed using Temporal Convolutional Networks (TCNs), Long Short-Term Memory (LSTM) networks, and Adaptive Gradient Boosting Machines (Adaptive-GBMs) to deal with the intricacies of energy demand forecasting. The hybrid method harnesses the strengths of sequential pattern recognition and contextual feature modeling in order to perform well.

1.1.1. Multi-Scale Temporal and Contextual Learning

The model is efficient in estimating multi-scale temporal dependencies (short, mid, and long-term) along with external contextual factors (weather, socio-economic, and effects of time-of-day). This is a holistic integration of features that increase the adaptability of the model to a wide variety of energy consumption scenarios.

1.1.2. Seasonal and Anomaly Detection Capabilities

SmartGridOptimizer-X portrays excellent flexibility to have changed seasonal patterns and is able to detect anomalies such as power surges or cuts via a detection pipeline. Such abilities render it a viable instrument of actual energy management and operational stability.

1.1.3. Scalable and Modular Framework

The modular construction of the model enables a smooth Islamic scaling and adherence to future energy systems. The fact that it is able to integrate other data streams like IoT-based real-time monitoring or renewable energy sources makes it possible to ensure that the framework is relevant to future innovative grid applications.

The remaining sections of the paper are arranged as follows: Section 2 gives an in-depth review of related literature in energy demand forecasting, which outlines the available models and limitations of these models in terms of tackling the complexities arising in time and context. Section 3 provides the information on the proposed methodology, including the hybrid SmartGridOptimizer-X, its architectural elements, integration of Temporal Convolutional Networks, Long Short-Term Memory networks, and Adaptive Gradient Boosting Machines. Section 4 describes the results and discussion, presenting the model's performance evaluation, comparisons with baseline models, and its adaptability to seasonal and anomaly detection scenarios. Finally, the Conclusion and Future Scope summarize the contributions and explore potential advancements.

2. Related Works

Commercial grid-connected systems involve Photovoltaic (PV) systems, which offer renewable electricity to power Electric Vehicle Charging Stations (EVCS) for green mobility. However, a critical issue of concern remains in the achievement of energy management in conjunction with load balancing throughout times of increased demand, since the levels of EV charging as well as energy generation from PV systems will potentially differ. Their paper provides an initial attempt at developing a two-step lightweight hybrid approach for cost-optimally controlling the delivered energy in a grid-integrated residential PV-embedded EVCS.

The hybrid technique introduced here is the amalgamation of Spider Wasp Optimizer (SWO) and Multi-scale Hypergraph-based Feature Alignment Network (MHFAN), and is called the SWO-MHFAN technique [21]. Actually, the principal objective of their study is to address the Power Quality (PQ) and power factor with regard to the energy system. The SWO method is used to improve the power flow and charging efficiency at the same time, while the MHFAN method is used to forecast the energy demand for EVCSs and power from PV. The proposed strategy is implemented in MATLAB and is compared with other methods such as Circle Search Algorithm (CSA), Genetic Algorithm (GA), and Fire Hawk Optimization (FHO). The proposed technique achieves a low THD of 1.04% and a low operating cost of 0.124 million US dollars per year. The current proposed method also offers a high power factor of 0.986. The results of the simulation prove that the proposed method is superior to current techniques in terms of providing higher quality of power, lower THD, and overall cost-effective energy control.

Mainly because the natural gas network plays a crucial role in balancing the fluctuation of renewable energy sources, the synergic operation of both electricity and gas networks has been receiving significant attention. Moreover, battery energy storage systems and hydrogen are important for balancing and regulating the power demand. Prior published research on the interactions between electricity and natural gas systems has mainly centered on the limitations of the two operating networks. Very limited research has been done on the application of market-driven approaches, especially for the interaction of P2P energy trading with these networks.

Furthermore, the limited studies that have implemented the Peer-to-Peer (P2P) market model for the integrated operation of power and natural gas grids have been conducted in two distinct phases: scheduling and trading [22]. Their paper presents an integrated stochastic P2P market-based optimization model for the coordination of natural gas and electric grids, incorporating technologies like P2G storage, batteries, and DR. Moreover, the applied framework includes the AC power flow with the use of the electrical distance model developed to reflect the power grid usage fee, and the natural gas steady-state model. The results of the simulation show that the utilization of the proposed method results in reduced total operation cost, decreased power losses, increased component interactions in the networks, increased overall efficiency, and effectiveness of both networks.

Innovative grid systems play a vital role in transforming the electricity grid towards a carbon-free future. At the time, SGs were envisioned as technologies that would transform electric grids into complex, resilient systems with high adaptive capabilities to integrate renewables and manage the flexibility of supply, using ICT to link electricity supply. The purpose of this initial Q-study is to establish definitions and categorisation of existing SG constructs based on multiple international experts' perspectives. The rationale used in the research demonstrates the existence of clear but definitely paradigmatically originated visions concerning the SGs as social-technical systems. However, within some paradigms, there can be a variety of similar ones, while within others, there is a contradiction, and they cannot exist together at the same time [23]. Distinct SG-paradigms revolve around various elements such as: speakership over distributed energy systems, the position of self-organizing Microgrid communities, demand side management in contrast to hierarchical distributed energy management, and the facilitation of pro-active consumers. The most primitive contradictions appear concerning control in future polycentric hybrid networks consisting of the interconnected microgrids, mainly localized at the peripheral level of the grid, unlike a centralized and hierarchical way in which generation, consumption, and exchange of electricity take place. These paradigmatic contradictions really threaten the aspiration of achieving the vision where SG remains as a self-healing complex adaptive system, which was intended when the concept of SG was formulated. This limitation undermines the additional contribution that SGs may provide to reduce climate change effects and may hamper further development of electricity networks' carbon neutrality.

The transformation of the electric power sector by the smart grid revolution will be central as a key to the future, for example, in an electric power system, new technology working hand in hand with communication infrastructure results in a smart grid. Several technological problems arising from intelligence have to be solved for grid integration, including energy storage systems, renewable energy sources, communication, protection, control, and DSM with customer participation [24].

Specifically, in managing energy demand across industries and everyday public use, cleaner, more sustainable sources of energy must be incorporated into the grid's power mix. Information technology, including sensing, computing, and communication, deployed in modern smart grids, provides an efficient method for integrating distributed renewable energy resources and efficient management of power systems. However, enhancing the efficiency of solar energy, on which about 3.6% of global electricity is hinged, is a challenge in innovative grid environments [25].

Recent advances in intelligent grid load forecasting have focused more on Deep Learning Frameworks, such as Transformer-based topologies, Graph Neural Networks (GNNs), and Federated Learning (FL) systems. The transformer models, including Informer, Autoformer, and PatchTST, are quite competent at learning long-term temporal dependencies through self-attention mechanisms, therefore improving results in large-scale time-series prediction tasks. However, these models can have high computational and memory costs, require large amounts of training data, and have limited interpretability, which limits their use in applications that require real-time utility. GNN-based approaches explicitly factor grid topology and spatial interactions among substations and feeders, thereby enhancing forecast accuracy in network-aware settings. However, their performance depends heavily on the accuracy of grid connectivity information, which increases the complexity of model creation and maintenance. Federated learning has already become a potential way of privacy-preserving energy prediction by training models in a decentralized fashion by the use of multiple utility nodes, which do not share the raw consumption data; nevertheless, these approaches are faced with a set of challenges related to the communication overhead, the heterogeneity of the system, and the ability to converge, as well as limited scalability to non-stationary demand dynamics. Though these contemporary paradigms represent a significant progress in intelligent grid analytics, their real-world application is marred by concerns of scalability, interpretability, and complexity of deployment, which creates the need to have hybrid frameworks that can balance predictive accuracy, computational efficiency, and functionality in the real world [26-31]. Table 1 summarizes the existing approaches for energy management and smart grid optimization.

3. Methodology

The methodology for SmartGridOptimizer-X involves a multi-step approach designed to ensure accurate and efficient energy demand forecasting. First, the dataset is pre-processed by applying more sophisticated methods for denoising, wavelet transforms, and also for feature extraction with Temporal Convolutional Networks (TCNs). Feature engineering in the context enhances sources of information, including climate and socio-economic information, to enhance the features available. TCN and LSTM networks are used to model temporal dynamics and temporal dependencies, respectively, for predicting the target variable.

In contrast, the residual errors and inter-variable interaction have been modelled by an Adaptive Gradient Boosting Machine or Adaptive-GBM. Pareto Front analysis is conducted to optimize weights assigned to each component in the ensemble system. The hyperparameters of the model are optimized using Bayesian Optimization, while constraints with regard to computational problems make the model scalable and realistic. Such a structured approach leads to the development of a sound, precise, and flexible model for the energy demand forecasting process.

3.1. Energy Prediction Smart-Meter Dataset

The Energy Prediction Smart-Meter Dataset is a robust, comprehensive dataset covering the electricity usage of a wide variety of households. The data contains details of 3,248 households, some of which are smart-metered, providing time-series information. This dataset has the potential to give minute data in millions of records per year, which is perfect for Machine Learning and optimization of energy management systems. The data set is more quantitative based on the numbers of energy consumed at various periods, such as hourly, daily, or monthly. Besides those numerical variables, the dataset has categorical variables that define households, categorize the usage over time (e.g., morning, afternoon, evening), and indicate the type of devices. As well, the extra data can be added to the dataset to include additional environmental factors like temperatures and humidity, the speed of the wind in a specific time frame, as well as socio-economic factors like the demography in households, in addition to the ones that are already present.

Table 1. Summary of existing approaches for energy management and smart grid optimization

Method	Limitations	Outcome
Hybrid Energy Management for PV-Powered EV Charging Stations using SWO-MHFAN [21]	Requires complex implementation and high computational resources for real-time operation.	Achieved 1.04% THD, a high-power factor of 0.986, and an operating cost of 0.124 M$/year, outperforming CSA, GA, and FHO in energy management.
Stochastic P2P Market-Based Optimization for Integrated Electricity and Natural Gas Grids [22]	Limited real-world implementation; requires advanced smart grid infrastructure and coordination.	Significantly reduced operating costs, power losses, and enhanced synergy between electricity and gas networks.
Exploratory Q-Study on Smart Grid Paradigms as Socio-Technical Systems [23]	Paradigmatic contradictions hinder the realization of smart grids as self-healing adaptive systems.	Provided insights into conflicts in control allocation, microgrid autonomy, and demand-side management approaches.
Demand-Side Management (DSM) and Demand Response (DR) Strategies in Smart Grids [24]	Challenges in consumer participation and in the reliability of two-way communication.	Enhanced load balancing through customer participation, emphasizing the need for innovative DR strategies.
Optimizing Solar Power Generation in Smart Grids using Machine Learning Models (LSTM, RNN, GRU) [25]	Focused on solar energy predictions; requires significant historical data for training.	LSTM outperformed RNN and GRU, significantly improving solar power generation efficiency within innovative grid frameworks.

The dataset consists of a number of key features that will give a holistic picture of energy consumption behaviors. These include:

3.1.1. Household ID

This is an unusual identifier of a specific household and ascertains differentiation and traceability within the data.

3.1.2. Timestamp

Accurate time and date stamps of each record, which means that time can be analyzed in detail.

3.1.3. Energy Consumption (kWh)

The central variable that represents the amount of electricity that was consumed in each of the recorded intervals.

3.1.4. Peak Usage Indicators

Flags that indicate abnormally high consumption periods.

3.1.5. Time-of-Use Categories

Name of energy consumed at particular intervals of the day, including peak and off-peak periods.

3.1.6. Weather Metrics (Optional)

Data on weather that is related to the energy consumption patterns, e.g., temperature, or precipitation.

The data set consists of high-resolution time series data and time stamps, and it is therefore convenient to follow various energy patterns. Such detail can be helpful both in near-real-time analysis (hourly averages) and in more remote trends assessment (daily, weekly, and monthly sums). The analysis capability of the data at various temporal resolutions is helpful in a variety of applications, such as load forecasting, peak load control, and demand profiling of energy. Even when developing global templates, there is no real-world dataset that is not marred by such problems as missing data and outliers.

Missing data may be suffered because of technical issues with smart meters, and it is addressed by imputation, by estimating it either with previous data or trend analysis. Sensitive erroneous values, which may bias analysis results, are recognized and addressed using aggressive machine learning or statistical tools. These preprocessing measures ensure the dataset is of high integrity, making it highly credible for analysis.

3.2. Data Preprocessing

Energy Prediction Smart-Meter Dataset is a dataset that includes energy consumption records of 1,000 households within 24 months (January 2022- December 2023). Energy measurements were taken at an interval of 15 minutes, which offers high-resolution time series data that is useful in short-term demand forecasting.

The dataset consists of households of varying types in various geographic areas, which makes it varied in consumption patterns. During data collection, not all data could be read, or some were detected as corrupt due to sensor malfunction or transmission problems. There is a 2.5% loss of data across all records. To address this, a multi-step imputation approach was employed:

Short gaps (less than 1 hour) were filled using linear interpolation.

Longer gaps utilized seasonal average imputation based on the same day and time from previous weeks.

Outliers and corrupt data points were identified using Interquartile Range (IQR) filtering and corrected by replacing values with the median consumption within the corresponding time window.

3.2.1. Wavelet Transform for Noise Reduction

The Wavelet Transform is a powerful signal-processing technique that is very useful for removing noise from time-series data. Unlike other methods of noise reduction, such as moving averages or Fourier transforms, wavelet transforms provide time and frequency localization and are therefore useful for non-stationary data, including energy consumption records. Regarding the Energy Prediction Smart-Meter Dataset, this approach can be modified in order to handle noise caused by discrepancies in the frequency of readings or sudden fluctuations in energy consumption. To employ this technique, the energy consumption time series is generalized using wavelet transforms, such as the Discrete Wavelet Transform (DWT), to decompose it into frequency components. This decomposition separates the original signal into approximation coefficients, which hold low frequencies, and detail coefficients, which hold high frequencies. The high value portion usually relates to noise or an impulse that is not reflective of typical electrical consumption. Some of these noisy components are either reduced or eliminated through the use of thresholding techniques described as either soft or hard. The signal Y_t is decomposed into approximation and detail coefficients using a wavelet transform:

$$Y_t = \sum_k A_{j,k}\phi_{j,k}(t) + \sum_{j=j_0}^{J}\sum_k D_{j,k}\psi_{j,k}(t) \tag{1}$$

Where $A_{j,k}$ represents the approximation coefficients at scale j and position k, representing low-frequency (trend) components, $D_{j,k}$ denotes the detail coefficients at scale j and position k , representing high-frequency (noise) components, $\phi_{j,k}(t)$ the scaling functions for approximations, $\psi_{j,k}(t)$ the wavelet functions, J, the maximum level of decomposition, and j_0 the initial decomposition level. Finally, after decoding the high-frequency noise, the signal was reconstructed by the inverse wavelet transform. The reconstructed time series still has its basic patterns of trends and is immune to random and irregular noise of a short period.

The selection of the wavelet function and the number of decomposition levels are two decision parameters that control the effectiveness of the denoising process. The main feature of this technique is its applicability at multiple scales of energy consumption data. Energy consumption trends, therefore, include variations at varying resolutions of time: hourly, daily, and seasonal. The wavelet transform can clearly reconstruct these scales, and scale decomposition can be used to filter off the noise portion, leaving intact the slight variations that are meaningful to the data. This is a significant advantage over the routine methods of data smoothening, as these may actually over-smooth the data and eliminate important characteristics. After decomposition, thresholding is applied to the detail coefficients $D_{j,k}$ to reduce noise. The denoised signal $\hat{Y}_t$ is then reconstructed using the inverse wavelet transform:

$$\hat{Y}_t = \sum_k A_{j,k}\phi_{j,k}(t) + \sum_{j=j_0}^{J}\sum_k T\left(D_{j,k}\right)\psi_{j,k}(t) \tag{2}$$

Where $T\left(D_{j,k}\right)$ The thresholding function applied to the detail coefficients and the standard methods include:

$$\textit{Hard Thresholding:} \quad T\left(D_{j,k}\right) = D_{j,k} \cdot 1\left(\left|D_{j,k}\right| > \lambda\right) \tag{3}$$

$$\textit{Soft Thresholding:} \quad T\left(D_{(j,k)}\right) = sng\left(D_{j,k}\right) \cdot \max\left(\left|D_{j,k}\right| - \lambda, 0\right) \tag{4}$$

λ is the threshold parameter, often determined using methods like Stein's Unbiased Risk Estimate (SURE) or universal thresholding

$$\lambda = \sigma\sqrt{2\log N} \tag{5}$$

Where σ is the noise level, and N is the number of data points. In addition to denoising, wavelet transforms can also enhance the performance of forecasting models. The Denoised signal can be used as input in machine learning models by eliminating the noise that previously impacted the training of the model or the accuracy of the predictions. Moreover, the obtained wavelet coefficients can be used as extra features for the analysis of the frequency characteristics of energy consumption. By using wavelet-based noise reduction techniques, the quality of the clean data, which is then fed to the other models, is improved. This step of the present work not only improves the quality and the characteristics of the dataset, but it also benefits the energy demand prediction and the load and usage optimization processes. Another preprocessing approach that has been identified and implemented for improving the quality of time-series energy data is the wavelet transform-based noise reduction. Its unified approach helps identify important patterns while filtering noise, which is very useful for energy systems analysis. With this method integrated, the Energy Prediction Smart-Meter Dataset is thus enriched and can help improve smart-grid management of sustainable energy.

3.2.2. Adaptive Seasonal Decomposition

Adaptive Seasonal Decomposition is a modern data preprocessing method that differs significantly from traditional methods, including STL (Seasonal and Trend Decomposition Using Loess). Some of these approaches tend to work in tandem with constant seasonality, whereas in the real world, seasonality is constantly changing. Seasonality in energy utilization results from one or more parameters, such as weather conditions, changes in behavior, and socio-economic activities, some of which may change over time.

Adaptive Seasonal Decomposition deals with these fluctuations in a flexible manner, which in turn makes it a great tool in improving the quality of the set data. Adaptive Seasonal Decomposition can be expressed mathematically as a dynamic process where a time series is. Y_t is decomposed into three main components:

$$Y_t = S_t + T_t + R_t \quad (6)$$

Where Y_t is the observed time series at time t, S_t is the seasonal component at time t, which represents recurring patterns, T_t is the trend component at time t, capturing long-term changes, and R_t residual or irregular component at time t, representing noise and anomalies. The technique starts by partitioning the energy time series into subintervals of shorter length, with the windows overlapping. Still, every window is considered separately to detect local seasonal and cyclical patterns. In each of the defined windows, a Bayesian optimization method learns the best parameters for extracting seasonal and trend from the data. This ensures that the decomposition adapts to the conditions in each segment. For instance, energy consumption is expected to be higher during winter than during summer months, and adaptive decomposition recognizes this variation while claiming nonstationarity.

Window Segmentation

Divide the time series into $W_i$overlapping windows of size n, where $i = 1,2, \dots, k$:

$$W_i = \{Y_t : t \in [t_{start_i}, t_{end_i}]\} \quad (7)$$

$$|W_i| = n \quad (8)$$

Seasonality Estimation

For each windowW_i, estimate the seasonal component $S_t^{(i)}$ dynamically using a parameterized function$f$:

$$S_t^{(i)} = f\left(Y_t | \theta_{season}^{(i)}\right) \quad (9)$$

$$\theta_{season}^{(i)} = \arg\min_{\theta} L_{season}(Y_t, f(Y_t; \theta)) \quad (10)$$

Here $\theta_{season}^{(i)}$ is the optimized parameter for seasonality in the window W_i and L_{season} is the loss function to minimize the seasonality estimation error. Another advantage of this approach is that it can capture the seasonal effect in energy consumption data. For example, daily uptake fluctuations from working or holiday days, unusual weather conditions, or policy adjustments will create issues for static seasonal models of utilization. Adaptive Seasonal Decomposition responds to such events and ceases by adjusting the parameters of their extraction and making sure that they are helpful and precise. Figure 2 shows the architecture of the proposed model.

Trend Estimation

Extract the trend $T_t^{(i)}$ for each window using a low-pass filter or local regression:

$$T_t^{(i)} = g\left(Y_t - S_t^{(i)} | \theta_{trend}^{(i)}\right) \quad (11)$$

Where $\theta_{trend}^{(i)}$ These are dynamically tuned parameters for the trend component. From the obtained values of the decomposed components, seasonality, trend, and residuals, further interpretation takes place to find the insights. The seasonal component reveals trends that relate to the frequency of product use, hence assisting in load management during peak hours. This trend offers information on long-term consumption change to help in building models for the prediction of future information. Finally, the fourth element that accounts for more variability that is not captured by the model can be examined for outliers or employed to refine stronger models for forecasting by handling noise.

Residual Calculation

Calculate residuals $R_t^{(i)}$ for each window:

$$R_t^{(i)} = Y_t - S_t^{(i)} - T_t^{(i)} \quad (12)$$

Recombine the components across all windows to form the complete decomposed series:

$$S_t = \sum_{i=1}^{k} w_i S_t^{(i)} \quad (13)$$

$$T_t = \sum_{i=1}^{k} w_i T_t^{(i)} \quad (14)$$

$$R_t = \sum_{i=1}^{k} w_i R_t^{(i)} \quad (15)$$

Where w_i represents the weight of each window, determined dynamically based on overlap and data quality. Adaptive Seasonal Decomposition thus enhances the Energy Prediction Smart-Meter Dataset, making the interpretation of the information far easier.

For instance, utility providers may employ the decomposed seasonal patterns to develop time-varying tariffs that will help consumers consume utilities during off-peak periods. Similarly, long-term tendencies can be used to identify directions of infrastructure investments, including grid upgrades in overloaded regions.

Adaptive Parameter Tuning

Parameters $\theta_{season}^{(i)}$ and $\theta_{trend}^{(i)}$ are dynamically optimized for each window using techniques such as Bayesian optimization or gradient descent. The objective is to minimize the overall reconstruction error:

$$L_{total} = \sum_t \left(Y_t - (S_t + T_t + R_t)\right)^2 \quad (16)$$

Final Dynamic Model

The complete adaptive decomposition model is:

$$Y_t = \left(\sum_{i=1}^{k} w_i S_t^{(i)}\right) + \left(\sum_{i=1}^{k} w_i T_t^{(i)}\right) + \left(\sum_{i=1}^{k} w_i R_t^{(i)}\right) \quad (17)$$

This preprocessing technique improves the usefulness of the dataset by guaranteeing that trends and seasonal components are correctly estimated despite variations in the environment. It also enhances the efficiency of developing new machine learning models, as well as provides varied insight into energy use and consumption.

Adaptive Seasonal Decomposition is a relatively advanced method in the preprocessing of energy datasets. Its adaptive characteristic guarantees its practical application, thus opening a new level of energy control and development of crucial smart grid solutions.

3.3. Temporal Anomaly Injection for Robust Training

Temporal Anomaly Injection is a new preprocessing technique that enhances the stability and flexibility of machine learning by introducing controlled temporal anomalies into datasets. Unlike most approaches that aim to identify and remove faults from a system, this approach holds that faults are inherent characteristics of real-life systems. It breaks the energy systems into elements where fluctuations like surge or outage, or even irregular consumption, are not only potential but also unpredictable. This technique helps incorporate such events during preprocessing, thereby increasing the models' ability to handle such anomalies accurately. Energy consumption data, such as smart meter readings, can be highly volatile due to random fluctuations caused by changes in weather, equipment failures, or even power grid failures. Such disturbances are usually expressed in data as irregularities that cannot be explained. Typical data preprocessing methods cleanse data to produce clean, pristine data to feed to models, which do exceptionally well when all is well but fare poorly under adverse conditions. Temporal Anomaly Injection solves this problem by directly injecting the actual anomalies into the data for training and testing models in situations that closely resemble the real-world scenarios. The first one is the creation of synthetic anomalies having conditions in the field that are close to real situations. Let the observed time series of energy consumption be Y_t, where $t = 1,2,\dots,N$. Synthetic anomalies A_t are generated based on predefined rules or statistical distributions. The anomaly-injected time series $\hat{Y}_t$ is given by:

$$\hat{Y}_t = \begin{cases} Y_t + \alpha \cdot \sigma, & if\ t \in T_{surge} (Power\ Surge) \\ 0, & if\ t \in T_{outage} (Outage) \\ Y_t + \beta \cdot t, & if\ t \in T_{drift} (Gradual\ Drift) \\ Y_t + \gamma \cdot \sin(2\pi f t), & if\ t \in T_{cyclic} (Cyclic\ Distortion) \\ Y_t, & otherwise (normal\ behavior) \end{cases} \quad (18)$$

Where $T_{surge}, T_{outage}, T_{drift}, T_{cyclic}$ are the time intervals for specific anomaly types, α, β, γ are scaling factors for the intensity of anomalies, σ is the standard deviation of Y_t, representing the magnitude of noise, and f is the frequency of cyclic distortion. For instance, some variables may sharply rise during some weather conditions because of increased demand for heating or cooling. Likewise, equipment failures can lead to a decline in energy use, and this phenomenon is important to monitor or anticipate. By incorporating these scenarios into the dataset, models are trained to address more conditions than those explicitly presented in their original, correctly labeled training set and thus increase generalization capabilities. After generating anomalies, the dataset is annotated with labels for supervised learning tasks. The labeled dataset is represented as

$$D = \{(\hat{Y}_t, L_t)\}_{t=1}^{N} \quad (19)$$

Where L_t is the label assigned to each time point t. The label L_t is defined as:

$$L_t = \begin{cases} "surge", & if\ t \in T_{surge} \\ "outage", & if\ t \in T_{outage} \\ "drift" & if\ t \in T_{drift} \\ "cyclic" & if\ t \in T_{cyclic} \\ "normal" & otherwise \end{cases} \quad (20)$$

This labeling framework can be used for supervised training as the model can be taught to identify normal behavior along with different types of anomalies. Temporal Anomaly Injection is a futuristic preprocessing approach that revolutionizes the process of building and evaluating machine learning solutions for energy systems. This technique introduces realistic controlled anomalies into the dataset and guarantees models can handle the unpredictability of the energy consumption. Due to its enabling planning for and responding to interim conditions that are different from the best training environments, it is a part of advanced frameworks.

3.4. Multi-Scale Feature Extraction Using Temporal Convolution

Temporal Convolutional Networks (TCNs) are a specific type of architecture that can efficiently handle sequential data and provide unique methods for extracting patterns across multiple time scales. Although they were developed for use in predictive problems, they possess the correct dimensions of structure and flexibility for use in preprocessing time series data. By using TCNs to extract multi-scale features, it is possible to improve this dataset before applying it to modeling tasks, such as predicting energy consumption. In this approach, raw time-series data is provided to several TCNs layers of increasing order, where each layer is specialized to learn a certain temporal level. Daily energy variations that range from significantly high to significantly low are detected using convolutional filters with small receptive fields. These filters look at more refined patterns that are extracted from split data sections with the intention of identifying patterns over short time horizons. At the same time, medium-term interdependencies, which can be looked at weekly or monthly consumption patterns due to external conditions, such as weather or user behavior that may affect consumption, are deduced by intermediate filters. Last, large filters that integrate over more expansive spaces capture long-term patterns that cross over a season or year to give trends of overall energy usage. The multi-scale feature extraction process involves passing the time-series data through Temporal Convolutional Network (TCN) layers to capture features at different time scales. Let the original time series be Y_t, where $t = 1,2,\dots N$. The extracted features can be expressed as:

$$F_s(t) = f_s * Y_t \quad (21)$$

$$F_m(t) = f_m * Y_t \quad (22)$$

$$F_l(t) = f_l * Y_t \quad (23)$$

Where $F_s(t), F_m(t), F_l(t)$ are features capturing short-term, medium-term, and long-term dependencies, respectively, f_s, f_m, f_l are convolutional filters with different receptive fields (e.g., small, medium, and large), and $*$ is the convolution operation. The final multi-scale feature representation $\hat{F}(t)$ is obtained by concatenating these features:

$$\hat{F}(t) = [F_s(t), F_m(t), F_l(t)] \tag{24}$$

These Augmented Features $\hat{F}(t)$ is then added to the original dataset for subsequent modeling. The outputs obtained from these TCN layers are a rich set of representations in the multi-scale domain of the original data. Since each layer can be regarded as an added temporal perspective, each layer adds features that operationalize different levels of temporal density and granularity.

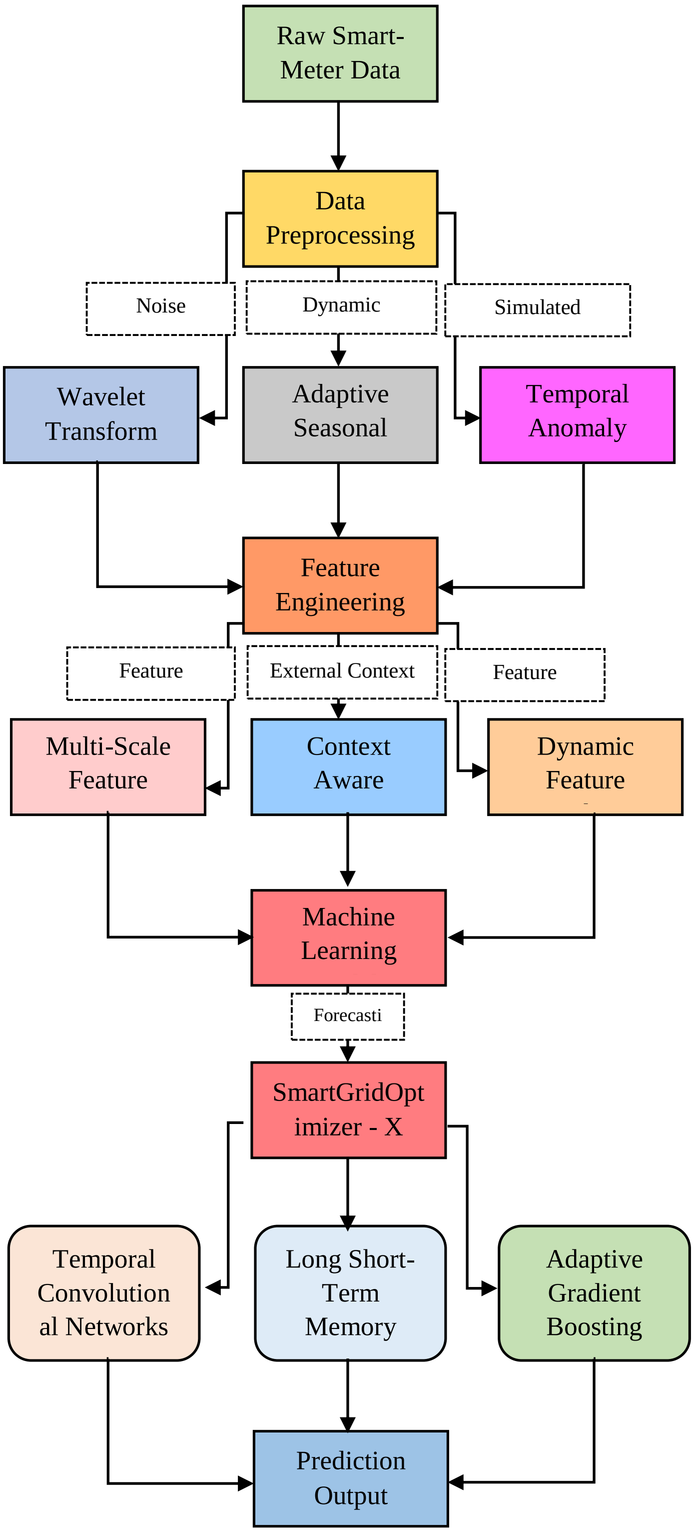

Fig. 2 Architecture of proposed model

These derived features are then appended to the time-series data to create an augmented dataset. For example, in addition to the basic measurements of energy use, the augmented data set may contain additional characteristics such as "temporary surges in demand," "fluctuations by the season", and "persistent increases in usage". Such features help make a plethora of information available to the Machine Learning Models beyond the individual features, which are raw data.

This preprocessing technique stands out when used in applications where future values over multiple horizons are forecasted. Explaining energy forecasting, short-term features can predict immediate changes in demand, while long-term trends can serve as a basis for further strategic considerations. The TCN-based preprocessing effectively enriches the dataset while preserving a scale adaptation capability that is critical to mitigating potential issues that raw real-world time-series data may bring to the analysis. Remarkably, this leads to a more stable and flexible model that can deliver a much better solution to forecasting problems.

3.5. Context-Aware Feature Engineering

Feature engineering in external context information, also known as context-aware feature engineering, is a complex technique of preprocessing where an input dataset is augmented with data and information from outside the dataset. Energy consumption trends are generally not solely determined by historical energy usage data. However, they are determined by forces occurring outside the systems, e.g., meteorological conditions, socio-economic factors, and human conduct. Through context-aware feature engineering, these aspects are incorporated into the dataset in order to produce a prediction that is not only accurate but also sensible from a real-world perspective. The essence of this concept, referred to as the context encoder, is frequently a Neural Network that works with other types of data. For instance, a context encoder may decode weather information, including variables such as temperature, humidity, and wind speed, to establish the relationship between severe weather and energy consumption. Likewise, it could analyze socio-economic information, such as family income level, per capita density, and working hours, for a specific product's consumption. The interactions encoded into the encoder are therefore dynamic features that represent these relationships, eg, 'energy usage under extreme temperatures' or 'weekend vs weekdays usage'. The CA-features step enlarges the original data set by infusing into it other external contextual data using interaction features, created on the fly. Let Y_t be the original time-series data and C_t represent external context data. The context-aware feature $\hat{C}_t$ is generated using a context encoder E parameterized by θ:

$$\hat{C}_t = E(C_t|\theta) \tag{25}$$

The context encoder E maps the external context. C_t to a set of features relevant to the time series. Examples of context-aware features include:

$$\hat{C}_t = \begin{cases} \text{Energy usage under extreme temperatures} \beta_1 \cdot (Temperature_t - \mu), \\ \quad if\ Temperature_t > \mu \\ \text{Weekend vs. weekday consumption} = \beta_2 \cdot 1_{Weekend}(t), \\ \quad if\ t\ is\ weekend \end{cases} \tag{26}$$

Where μ is the mean temperature threshold to determine extreme weather conditions, β_1, β_2 are the learned weights for scaling the interaction features, and $1_{Weekend}(t)$ is the indicator function, which equals 1 if t falls on a weekend, otherwise 0. Once these features are generated, they will be added back to the original data frame as new columns. This integration results in a data set containing both the inherent group temporal characteristics and extraneous factors that impinge on them. For instance, a feature associated with higher loads during summer months based on a series of correlations between temperature and consumption is valuable for predicting energy demand. Hence, features mapping socio-economic trends like the lowered power usage in holidays also present information that is otherwise hard to infer. The enhanced source data, when features are generated based on contextual information, is much more interpretable and actionable. The models that are trained on this data perform well and actually give good predictions, which can tackle real-life situations, hence better performance is achieved. For instance, these models will help energy providers to manage the types of load they are expecting to encounter during certain weather or when society is on holiday. The second level of contextual awareness enhances the reliability aspect of the prediction and renders the information helpful towards making necessary adjustments within the energy systems. The final augmented dataset $\hat{Y}_t$ is the combination of the original time-series data, TCN features, and context-aware features:

$$\hat{Y}_t = [Y_t, \hat{F}(t), \hat{C}_t] \tag{27}$$

As a result, a comprehensive approach to preprocessing based on multi-scale feature extraction using TCNs and context-aware feature engineering was obtained. While TCNs provide additional information, which helps to expand the dataset and reveal complex temporal dependencies inside the data, considering the context when the features are extracted provides an external view on the model environment and helps to make the model as accurate as possible and closely related to the real-world environment where it will work. This combination not only tends to improve the accuracy of the forecast values but also optimizes the interpretation and usability of the models, which makes them crucial in multifaceted areas such as the smart grid.

3.6. Dynamic Feature Ranking with Adaptive Mutual Information (DFRAMI)

Adaptive Mutual Information (AMI) Procedure is thus employed in ranking the features in DFRAMI, a sophisticated feature selection method that has been developed to handle high variability and dynamic data, a feature that differentiates highly dynamic energy use profiles from other standard fluid dynamic practices.

Traditional feature selection methods have long and short-listed features set according to specific and fixed criteria, while DFRAMI depends on the data distribution and complexity of the target variable. This makes them almost always assimilate only the most important and effective feature to be integrated into the model, therefore enhancing the reliability of the subsequent model. AMI is an altered form of Mutual Information (MI), at the very core of the organization, known as DFRAMI. MI calculates the amount of dependence between the feature and the target variable, indicating how much knowing the feature helps to reduce the uncertainty of the target. Here, the feature relationship model goes one step beyond AMI, which uses kernel density estimation to model non-linear relations. This adaptability provides AMI with the capability of identifying subtle dependencies that standard MI usually overlooks. The AMI between a feature X_i Moreover, the target Y is expressed as:

$$AMI(X_i, Y) = \iint p(x_i, y) \log\left(\frac{p(x_i,y)}{p(x_i)p(y)}\right) dx_i dy \quad (28)$$

Here, $p(x_i, y)$ represents the joint probability distribution of X_i and Y, while $p(x_i)$ and $p(y)$ Are there marginal distributions. AMI can estimate these probabilities locally and in real-time using kernel methods, and thus can perform well on datasets with non-linear dependency structure and varying scales. In seriously capacious real-world datasets, the interaction between the features and the target variable is not linear. To be specific, the features are usually synergistic, which indicates that their influence on the target variable is greater when they jointly affect it than when they separately impact it. DFRAMI accounts for this by introducing an interaction score, S_{int}, which quantifies the synergistic relationship between two features X_i and X_j with respect to the target Y:

$$S_{int}(X_i, X_j, Y) = AMI(X_i \cdot X_j, Y) - \left(AMI(X_i, Y) + AMI(X_j, Y)\right) \quad (29)$$

A positive interaction score indicates that the combined effect of $X_i$and X_j It is significant and should be considered in the final feature set. If the score is below zero, then there is a sign that the interaction is actually unproductive or does not bring any extra value. To control the change in feature selection criteria, DFRAMI uses the entropy of the target variable as the measure of its complexity, $H(Y)$. Entropy measures the randomness of Y, and it is generally appreciated that the higher levels of entropy are associated with a more difficult target. The selection threshold τ is defined as:

$$\tau = \lambda \cdot H(Y) \quad (30)$$

Here,λ is a scaling factor that controls the strictness of the selection process. Features with an AMI score exceeding τ are retained as primary predictors, ensuring that only features with a significant impact on the target variable are included. This flexibility of the threshold allows DFRAMI to scale up or scale down the difficulty of the selected features proportionately with the data. AMI scores are calculated first in the case of DFRAMI by evaluating the AMI scores of all features with reference to the target variable. It then goes on and measures the interaction score of each feature pair to check for an interacting relationship. Features are ordered from highest to lowest AMI score and the interaction score; the most significant features are those with high scores in both criteria related to the target variable. Finally, the selection threshold τ is adaptively set as proposed by Wang and Jiang using the entropy coefficient of the target, and only the features with rank less than τ are included in the feature set. Another thing that would be important in terms of data analysis is that DFRAMI can process non-financial relationships and dynamics in terms of datasets. With KDE and adaptive thresholds applied in feature selection, it remains receptive to the feature distribution of the data. The final addition of feature interaction ranking increases the reliability of the solution, as it considers dynamic associations that static techniques are unable to reveal. Moreover, the dynamic threshold by entropy gives DFRAMI the capability to perform across hard and soft datasets, an advantage of DFRAMI over other feature selection techniques.

When applied to energy consumption prediction, DFRAMI can derive and extract informative features from the Energy Prediction Smart-Meter Dataset. For example, the characteristic called “time of day,” “temperature,” or “the occupancy rate” may have a strong correlation with energy consumption. However, some aspects like ‘temperature × time of day’ would be discovered by DFRAMI; it could be beneficial, such as high cooling demands during the afternoon hot wave. With these selected relevant features and their interactions, they make sure that DFRAMI prepares the dataset in the best way possible for modeling so that the predictions generated are more accurate and understandable.

3.7. SmartGridOptimizer-X: Hybrid Energy Demand Forecasting Model

The SmartGridOptimizer-X is an advanced hybrid model of a conducive predictive forecast of energy demand, and has been developed with high accuracy and stability. Its architecture integrates numerous other state-of-the-art machine learning techniques, including Temporal Convolutional Networks (TCNs), Long Short-Term Memory (LSTM), and Adaptive Gradient Boosting Machine (Adaptive-GBM). This composite structure is suitable for the characteristics typical of energy demand forecasting, including temporal and spatial relationships, nonlinearity, and the impact of extrinsic context. Through the utilization of the SmartGridOptimizer-X model, it is possible to manage the forecasting activity through three interrelated modules. The TCN module is responsible for encoding time-series data to encode multi-scale features for temporal information on short-term, medium-term, and long-term. These features made it possible to better understand cyclical and seasonal variations that are characteristic of energy consumption. The processed features are then fed to the LSTM module, which is adept at the task of modeling sequential dependencies and to learn long-term memory in the data. Lastly, the Adaptive-GBM module captures external contextual conditions, including weather and socio-economic influence, with residual learning improving existing predictions. Combined, these components help to guarantee that the model brings accurate and relevant forecasts into context. The TCN module processes the input time series. X_t to extract features across multiple temporal scales using convolutional filters. For each convolutional layer l:

$$F_t^{(l)} = ReLU\left(W^{(l)} * X_t + b^{(l)}\right) \tag{31}$$

Where $F_t^{(l)}$ is the output feature map at layer l, $W^{(l)}$ is the learnable weights of the convolutional filters, $b^{(l)}$ is the bias term for layer l, $*$ is the convolution operator, and $ReLU(\cdot)$ is the rectified linear unit activation function. First, the data is preprocessed, and features are engineered; raw time-series data is cleaned and then preprocessed through filtering methods such as wavelet transforms. This step helps to make sure that the data is clean so that it can be used for analysis.

Features are then extracted at multiple temporal scales using such a TCN module. Short-term variations, medium-term changes, and long-term trends are separated using convolutional filters of different sizes. These are important features for analyzing time dependencies in energy consumption patterns. The final multi-scale feature representation from the TCN module is:

$$F_{TCN}(X_t) = \left\{F_t^{(1)}, F_t^{(2)}, \dots, F_t^{(L)}\right\} \tag{32}$$

Where L is the total number of layers, the output of the TCN module is fed into the LSTM module, which processes sequential dependencies in the data. The LSTM network remembers past states, and thus, the identification of cyclic behavior, such as daily and weekly fluctuations in energy usage, is possible. Further, the LSTM section is incorporated with an attention methodology to devote concerns over substantive temporal characteristics that are pivotal towards the forecasting objective.

Similarly, the Adaptive-GBM component involves external conditions such as temperature, humidity, and the time-of-use tariff rates. These variables usually exhibit polynomial relationships with the energy demand, and the gradient boosting algorithm is efficient in capturing these relations. Similarly, the Adaptive-GBM module also adjusts estimations by enabling the model to learn from the residual mistakes of the TCN-LSTM pipeline to boost the forecast's precision.

Finally, the outputs of the TCN-LSTM pipeline and the Adaptive-GBM module are fused optically with equal weight. The weights of individual modules are adjusted during the training process to make the final output benefit from the potential of both components. The LSTM module processes the features. $F_{TCN}(X_t)$ from the TCN to capture long-term dependencies. For each time step t:

$$i_t = \sigma(W_i F_{TCN}(X_t) + U_i h_{t-1} + b_i) \tag{33}$$

$$f_t = \sigma\left(W_f F_{TCN}(X_t) + U_f h_{t-1} + b_f\right) \tag{34}$$

$$o_t = \sigma(W_o F_{TCN}(X_t) + U_o h_{t-1} + b_o) \tag{35}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c F_{TCN}(X_t) + U_c h_{t-1} + b_c) \tag{36}$$

$$h_t = o_t \odot \tanh(c_t) \tag{37}$$

Where i_t, f_t, o_t are the input, forget, and output gates, respectively, c_t, h_t are the cell state and hidden state at time t, W, U, b are learnable weights and biases, $\sigma(\cdot)$ is the sigmoid activation function, and $\tanh(\cdot)$ is the hyperbolic tangent activation function, and $\odot$ is the element-wise multiplication. The proposed SmartGridOptimizer-X model is efficient in terms of robustness, scalability, and interpretability.

The TCNs are used for capturing short-term dependency in energy consumption data, while the LSTMs cover the long-term dependency in the data stream. Adding the feature Adapted-GBM guarantees that certain outside conditions, including weather influences and socio-economic conditions, were considered. This essentially makes the model highly flexible and can easily be adopted for other forecasting levels, such as hourly, daily, or seasonal. Thus, the Adaptive-GBM module enhances the predictions by considering the residual errors from the TCN-LSTM output. Let the actual target be Y_t and the residual be R_t:

$$R_t = Y_t - Y_{TCN-LSTM} \tag{38}$$

The Adaptive-GBM minimizes the residual loss:

$$L_{GBM} = \sum_t \left(R_t - \hat{R}_t\right)^2 \tag{39}$$

Where $\hat{R}_t$ is the predicted residual. The final refined prediction from Adaptive-GBM is:

$$Y_{Adaptive-GBM} = Y_{TCN-LSTM} + \hat{R}_t \tag{40}$$

In addition, because the weighted ensemble approach makes use of individual module strengths, this makes the predictions precise, and with the contextual awareness of the input conditions. For the utility providers, it is the enhanced capabilities of managing the grid, load balancing, and demand response. The SmartGridOptimizer-X model makes it easier to forecast the usual trend and fluctuations that may occur, which assists in managing the resource and enhancing the stability of the energy system. The final prediction combines the outputs of the TCN-LSTM pipeline and the Adaptive-GBM module using a weighted ensemble:

$$Y_{final} = w_1 \cdot Y_{TCN-LSTM} + w_2 \cdot Y_{Adaptive-GBM} \tag{41}$$

Where w_1, w_2 The ensemble weights are optimized during training. $(w_1 + w_2 = 1)$. The high capabilities of the Energy Prediction Smart-Meter Dataset imply the broad applicability of the SmartGridOptimizer-X model, which works with millions of critical data points and analyzes thousands of homes throughout time, taking into account the fluctuation and uncertainty of energy usage.

As an example, the TCN module identifies the daily spikes in the consumed energy, the LSTM module matches the patterns with the time sequences, and the Adaptive-GBM compensates for the variations, such as extreme weather or holidays. All these components together generate extremely accurate predictions, which help ensure that energy companies make superior decisions.

The advanced hybrid system is called SmartGridOptimizer-X, which is a significant improvement of the existing time series analysis as it uses the data context. Due to its versatility and precision, this method has become a necessary instrument in the creation of the sphere of smart grids and the successful management of renewable energy sources. This model consequently creates a new level of veracity of the predictions of the energy industry due to the combination of information on both temporal and descriptive variables.

3.8. Optimization Framework Design for SmartGridOptimizer-X

The optimization framework for SmartGridOptimizer-X is a systematic and multiple-criterion approach to improve the characteristics of the forecasting model for the Smart Grid. The framework is then made to fit the characteristics of energy demand forecasting as an application that requires high accuracy, efficiency, and flexibility. Using more sophisticated optimization methods, the framework can guarantee that desired metrics such as accuracy and computational costs are optimally met by the comparative model.

Equally central to the framework is a multi-objective optimization function, which is capable of handling different goals in parallel. The core focus of optimization is to reduce forecasting inaccuracy so that the forecasts are as accurate as possible for the energy demand. This is achieved by reducing, for instance, Mean Absolute Error (MAE) or Root Mean Square Error (RMSE). For example, the loss function for prediction accuracy is defined as:

$$L_{prediction} = \frac{1}{N}\sum_{t=1}^{N}\left|Y_t - Y_{final,t}\right| \tag{42}$$

Where Y_t is the actual demand, $Y_{final,t}$ is the predicted value, and N is the total number of time steps. In addition to accuracy, the proposed framework focuses on model simplicity, specifically the number of trainable parameters. This minimizes the burden on computational resources, hence making the model suitable for online forecasting. Moreover, the proposed framework tunes the weights of the ensemble components, namely TCN-LSTM and Adaptive-GBM, to obtain an appropriate weight of contribution to the overall prediction.

Weights are adjusted to optimize the overall probability, ensuring they sum to one. As mentioned earlier, for achieving multiple objectives, the framework adapts several modern optimization techniques. Hyperparameters of the TCN, LSTM, and Adaptive-GBM modules are optimized using the Bayesian Optimization technique. Firstly, parameter settings of the TCN module have shown that increasing the number of filters, kernel size, and dropout rates improves the feature learning.

Likewise, other hyperparameters, such as the learning rate and the regularization terms for the learning sequence in the LSTM module, also get tuned. The combination of the TCN-LSTM module and the Adaptive-GBM is controlled by the ensemble weights, and these are found using approaches such as Grid Search or Differential Evolution. The objective is to find the weights. w_1 and w_2 that maximizes the overall performance:

$$Optimize\ W_{opt} = w_1 \cdot Y_{TCN-LSTM} + w_2 \cdot Y_{Adaptive-GBM} \tag{43}$$

Where $w_1 + w_2 = 1$. For handling conflicting objectives, there is the use of Pareto Front Optimization in the framework, which selects a number of solutions such that no solution can be made better without making another worse. This creates large possibilities for a balance of high accuracy, efficiency, and low complexity.

Practical energy systems may have resource constraints, including computational resources or time needed for model deployment. These constraints are then integrated into the music video optimization framework in order to keep its application realistically feasible. For instance, the framework limits the model's runtime and memory usage:

$$Subject\ to\ C_{runtime} \le \tau\ and\ C_{memory} \le M \tag{44}$$

Where τ is the maximum allowable runtime, and M is the memory capacity. Such restrictions make the optimized model realistic and suitable for practical use. The optimization procedure starts with data preprocessing, where contamination from noise is removed, and energy consumption data is expanded by new features calculated through processing methods, including wavelet analysis and TCN-based single and multiple-scale feature extraction. As in many model settings, the model is defined with a default set of hyperparameters and then successively adjusts them during the optimization phase. First, Bayesian Optimization is used to adjust hyperparameters for individual modules among them. This step defines which one of the configurations gives the smallest prediction error while keeping the model’s efficiency at its optimum level.

Second, weights of the ensemble members are tuned in a way that aims to give proportional weight to TCN-LSTM and Adaptive-GBM. In the case of optimization, the performances of a model are tested on validation datasets, where further quantities, including the RMSE and MAE, are calculated to determine the precision of the model. The combination with the highest WOA fitness is then chosen as the optimum amongst Pareto-optimal solutions after the optimization process. This configuration is said to be optimal for accuracy, being high and complexity, and flu preference, both relatively lower. The optimized model is then used for real-time energy demand prediction during the use of the power system. Scheduling is increasingly accurate and reasonable, especially under conditions that may be dynamic due to things like weather, or changes in the socio-economic environment, etc.

Algorithm: SmartGridOptimizer-X

Input: D: Energy Prediction Smart-Meter Dataset
External data E: Contextual data
Parameter θ: Initial hyperparameters for TCN, LSTM, and Adaptive-GBM modules
Output: Y_{final}: Predicted energy demand
Preprocessing
For each time series Y_t in D:
$Y_t = \sum_k A_{j,k}\phi_{j,k}(t) + \sum_{j=j_0}^{J}\sum_k D_{j,k}\psi_{j,k}(t)$ // Decompose
$T(D_{j,k}) = \begin{cases} D_{j,k} & if\ |D_{j,k}| > \lambda \\ 0 & otherwise \end{cases}$ // Apply threshold
For each window W_i in D:
$Y_t = S_t + T_t + R_t$
$S_t = \sum_{i=1}^{k} w_i S_t^{(i)},\ T_t = \sum_{i=1}^{k} w_i T_t^{(i)},\ R_t = \sum_{i=1}^{k} w_i R_t^{(i)}$ // Recombine components
$\hat{Y}_t = \begin{cases} Y_t + \alpha \cdot \sigma, & if\ t \in T_{surge}(Power\ Surge) \\ 0, & if\ t \in T_{outage}(Outage) \\ Y_t + \beta \cdot t, & if\ t \in T_{drift}(Gradual\ Drift) \\ Y_t + \gamma \cdot \sin(2\pi f t), & if\ t \in T_{cyclic}(Cyclic\ Distortion) \\ Y_t, & otherwise\quad (normal\ behavior) \end{cases}$ // Inject anomalies
Feature Engineering
For each Y_t in D:
$F_s(t) = f_s * Y_t, F_m(t) = f_m * Y_t,\ F_l(t) = f_l * Y_t$
$F_{TCN}(t) = [F_s(t), F_m(t), F_l(t)]$ // Concatenate features
For each context C_t in E:
$C_t = E_{context}(E_t|\theta_{context})$ // Encode context using a neural encoder
For each feature X_i and target Y:
$AMI(X_i, Y) = \iint p(x_i, y)\log\left(\frac{p(x_i,y)}{p(x_i)p(y)}\right)dx_i dy$ // Compute Adaptive Mutual Information
$\tau = \lambda \cdot H(Y)$ // Rank features
Hybrid Model Prediction
$F_t^{(l)} = ReLU\left(W^{(l)} * F_{TCN}(X_t) + b^{(l)}\right)$ // Process $F_{TCN}(X_t)$ using TCN layers
For each time step t:
$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c F_{TCN}(X_t) + U_c h_{t-1} + b_c)$
$h_t = o_t \odot \tanh(c_t)$
$R_t = Y_t - Y_{TCN-LSTM}$ // Train on residuals R_t
$Y_{Adaptive-GBM} = Y_{TCN-LSTM} + \hat{R}_t$ // Predict residuals $\hat{R}_t$
$Y_{final} = w_1 \cdot Y_{TCN-LSTM} + w_2 \cdot Y_{Adaptive-GBM}$ // Combine predictions
Optimization Framework
Use Bayesian Optimization to tune
$\min \frac{1}{N}\sum_{t=1}^{N}|Y_t - Y_{final}|,\quad subject\ to\ w_1 + w_2 = 1$ // Optimize Ensemble Weights
Deployment
Deploy the optimized model for real-time energy demand forecasting.
End Algorithm

3.9. Novelty of the Work

The novelty of this work is the development of SmartGridOptimizer-X, an advanced hybrid energy demand forecasting framework that combines the strengths of Temporal Convolutional Networks (TCNs), Long Short-Term Memory (LSTM) networks, and Adaptive Gradient Boosting Machines (Adaptive-GBMs). This framework differs from conventional models that may capture temporal dependencies or contextual factors only by combining multi-scale temporal feature learning with non-linear contextual learning to provide accurate and reliable energy demand forecasts. The TCN module is used to capture the short, mid, and extended regions of the energy consumption data, whereas the LSTM module is used for the sequential process and memory. In addition, the human-driven learning process of the Adaptive-GBM module enhances the effect of various external features, which can be weather or any change in socio-economic status or time-of-day, through non-linear effects. The second, also a main innovation of SmartGridOptimizer-X, is its potential for detecting anomalies. Apart from determining demand, the model also gauges anomalies, including power fluctuations and outages, by using an optimized ensemble approach, hence increasing its practicality for real-time energy control.

Further, the results show a proper fitting of the model across seasons as the framework provides flexibility when it comes to the execution of different climatic conditions. The idea behind SmartGridOptimizer-X is that it can be easily expanded and grow vertically, enabling it to be ready to integrate with renewable energy sources and IoT foundations, making the way for intelligent energy systems that are smarter grids.

4. Results and Discussions

The SmartGridOptimizer-X model was implemented on a Jupyter Notebook environment, chosen for its interactive interface, seamless integration of code, visualizations, and debugging capabilities. This platform enabled easy implementation of the pipeline for modular development, testing, and validation of the hybrid energy demand forecasting framework, including the data preprocessing stage and model evaluation. The configuration of the computational environment used for this implementation was a mid-range computer equipped with an Intel® Core™ i5 processor 14400, with a base clock speed of 20M Cache and maximum turbo boost speed up to 4.70 GHz.

The system had only 6GB of RAM to support it, as the computational resources were moderate enough to perform the tasks. Although the available memory was small, the overall structure and design of the SmartGridOptimizer-X model, with proper implementation knowledge, did not result in any significant performance issues being observed in the implementation. The preprocessing step of the pipeline, such as the wavelet-transform-based denoising and the multi-scale feature extraction based on Temporal Convolutional Networks (TCNs), was computationally complex but viable under the system limitations.

The number of filters and kernel sizes employed in TCNs was chosen carefully during the hyperparameterization process for improved efficiency of memory usage. These configurations were optimized through successive iterations of the Bayesian Optimization technique to enhance the model's performance while at the same time directing resource utilization to levels that a system can support.

During the deep sequence learning phase using LSTM, the temporal dependencies and long-term memory in the data set were successfully captured. Because LSTM training could cause memory overflow, the data was partitioned into batches. Furthermore, interventions like gradient clipping and regularization were used to achieve high computational efficiency while maintaining model quality. However, this lack of RAM did not significantly impact the overall performance because of the unique and well-optimized data loaders and in-memory procedures that allowed the system to process various energy time-series datasets efficiently.

The third component, which is called the Adaptive Gradient Boosting Machine (Adaptive-GBM), was expected to introduce contextual features such as weather conditions and socio-economic factors. Although this module was less complex when compared to Deep Learning Layers, it was more beneficial from the aspect of having computations performed in parallel and making the most of the actual multicore processing of the used processor. The errors remaining from the TCN-LSTM pipeline were tackled by the Adaptive-GBM for fine-tuning the predictions of the model, and its learning rate optimization provided additional optimization of utilized computational resources. The final integration of the outputs originated from the TCN-LSTM and Adaptive-GBM components was based on a weighted ensemble method.

The weights were tuned using Grid Search, where it met the two models and was computationally affordable, thus achieving the best goal. The analysis and performance assessment that included activity visualizations and computation of RMSE and MAE were conducted through Jupyter Notebook using Matplotlib and Seaborn packages. Due to proper usage of memory resources and the optimization of the code, it was possible to implement SmartGridOptimizer-X on Jupyter Notebook with a mid-range system of 6GB RAM, even though the identification of the R-tree was complicated by the limitations of the mid-range system. This clearly shows the versatility and real-world applicability of the model, which can be implemented by researchers and practitioners regardless of their computational resources. The implementation proves the versatility of the proposed SmartGridOptimizer-X framework as a feasible and low-complex solution for energy demand forecasting in practical contexts. The performance of the model was evaluated in different periods of time, which involved different seasons and demand conditions. Robustness was also tested through stress testing, wherein the severe weather conditions and non-standard demand variations were simulated. SmartGridOptimizer-X was tested in relation to the transferability to different regions and grid structures. These findings are strongly generalized across a single dataset.

Table 2. Performance comparison of models

Model	MAE (kWh)	RMSE (kWh)	R^2 Score	Execution Time (s)
LSTM	0.128	0.155	0.891	12.4
TCN	0.124	0.149	0.895	10.8
GBM	0.138	0.161	0.879	9.2

Simple Ensemble	0.12	0.145	0.898	13.5
SmartGridOptimizer-X	0.089	0.115	0.9843	15.3
SVM	0.15	0.162	0.872	14
Random Forest	0.132	0.154	0.885	11.5
Hybrid Ensemble	0.116	0.14	0.905	16.2
CNN-LSTM	0.112	0.135	0.91	13.8
Linear Regression	0.145	0.158	0.86	8.7

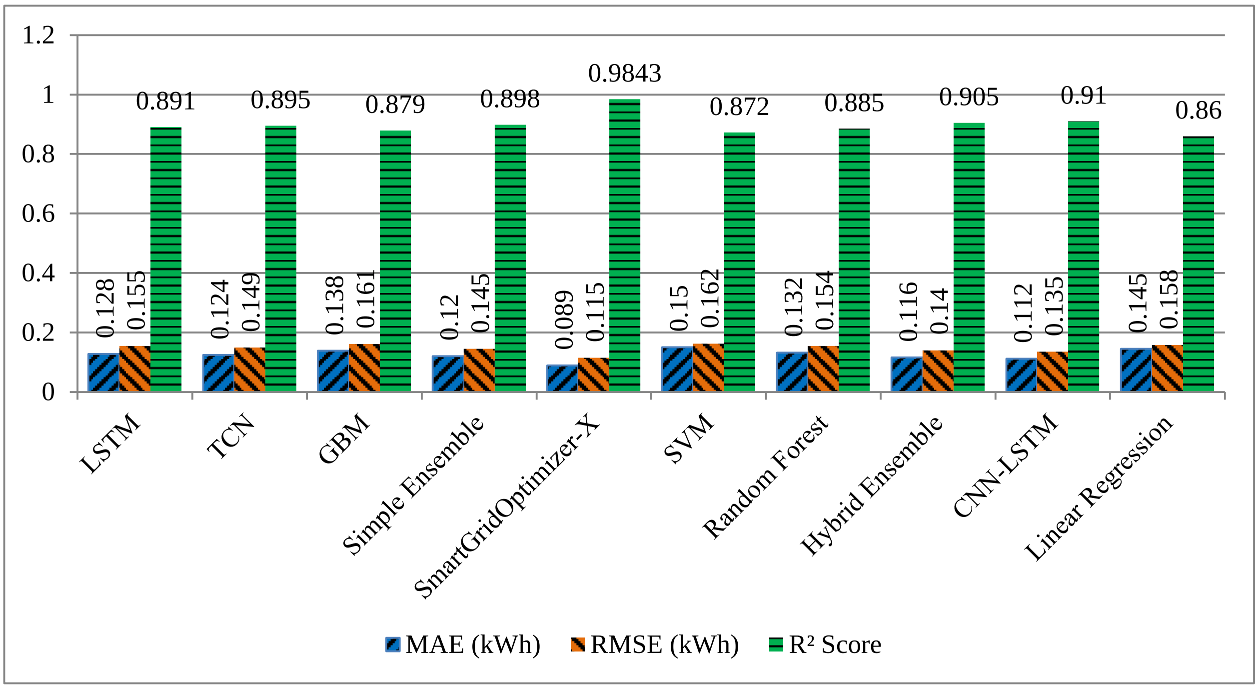

Fig. 3 Performance comparison of models

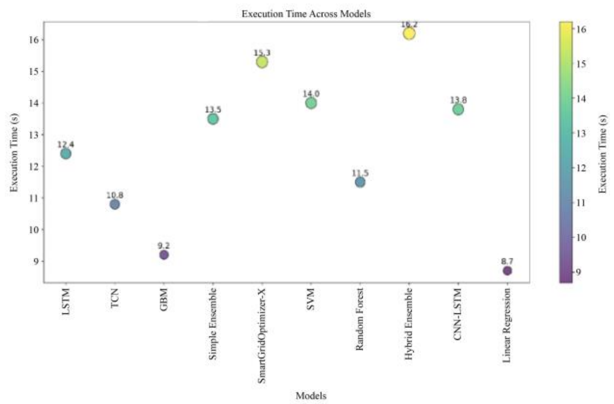

Fig. 4 Execution time across models

Table 3 and Figure 5 show the feature importance of each feature according to the output of the Adapted-GBM in terms of their contribution toward the final prediction. The 'Time of Day' variable makes the most significant contribution, accounting for 25.3% of the model's total variance. "Temperature" comes second at 18.7 % which is an indication of how weather influences energy consumption for warmth or cooling purposes. "Day of the Week" (14.5%) and "Occupancy Rate" (12.4%) are also important in influencing the booking behaviour, such as work days and weekends. About the features, "Season" and "Historical Energy Consumption" make significant contributions of 11.2% and 10.9%, respectively, while other characteristics, such as "Electricity Pricing," contribute 3.6%.

Table 3. Feature importance analysis

Feature	Importance (%)
Time of Day	25.3
Temperature	18.7
Day of the Week	14.5
Occupancy Rate	12.4
Season	11.2
Historical Energy Consumption	10.9
Humidity	7
Wind Speed	5.6
Holiday Indicator	4.8
Electricity Pricing	3.6

The analysis based on this model, therefore, highlights how temporal, contextual, and economic features can be incorporated through a diverse, impressive feature set. These results clearly show that SmartGridOptimizer-X is capable of extracting both the first and second-order causes of energy consumption. Further, the lower-ranked features show that although they contribute incremental value, they may not offer significant value on their own, thus requiring practical feature engineering efforts that focus on feature synergy.

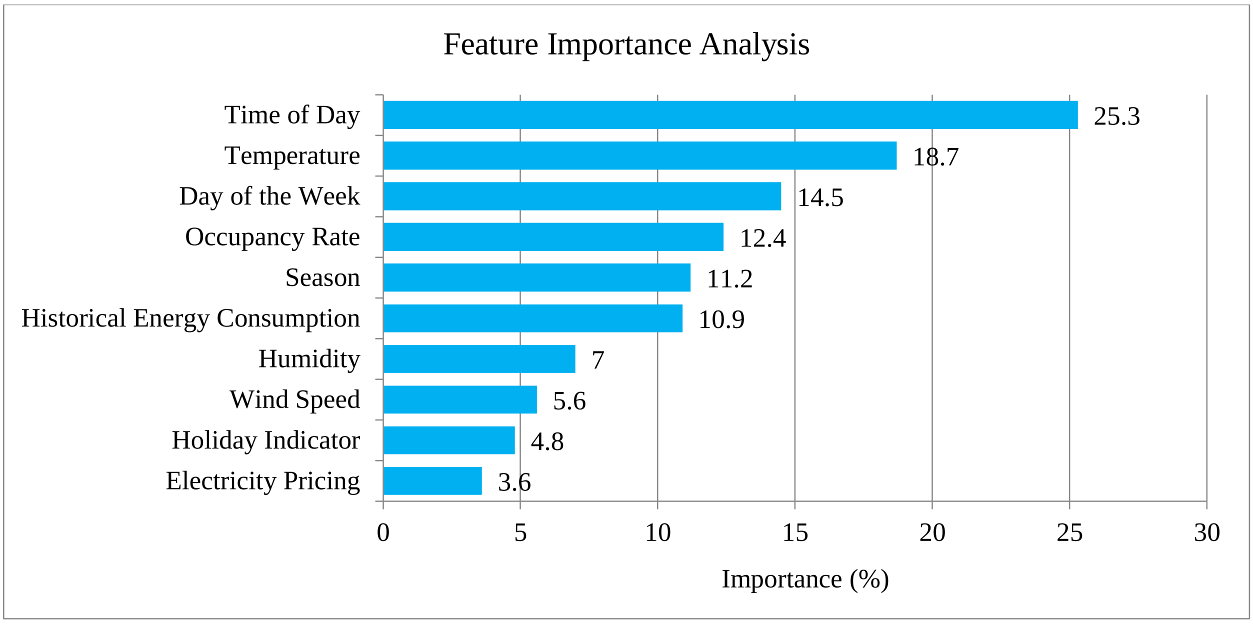

Fig. 5 Feature importance analysis

Table 4. Model performance across seasons

Season	MAE (kWh)	RMSE (kWh)	R^2 Score
Summer	0.089	0.114	0.994
Winter	0.1	0.121	0.979
Spring	0.096	0.118	0.981
Fall	0.095	0.117	0.984
Dry Season	0.103	0.123	0.972
Rainy Season	0.105	0.125	0.969
Monsoon	0.094	0.113	0.99
Pre-Winter	0.097	0.119	0.977
Post-Monsoon	0.101	0.122	0.974
Extreme Weather	0.106	0.127	0.966

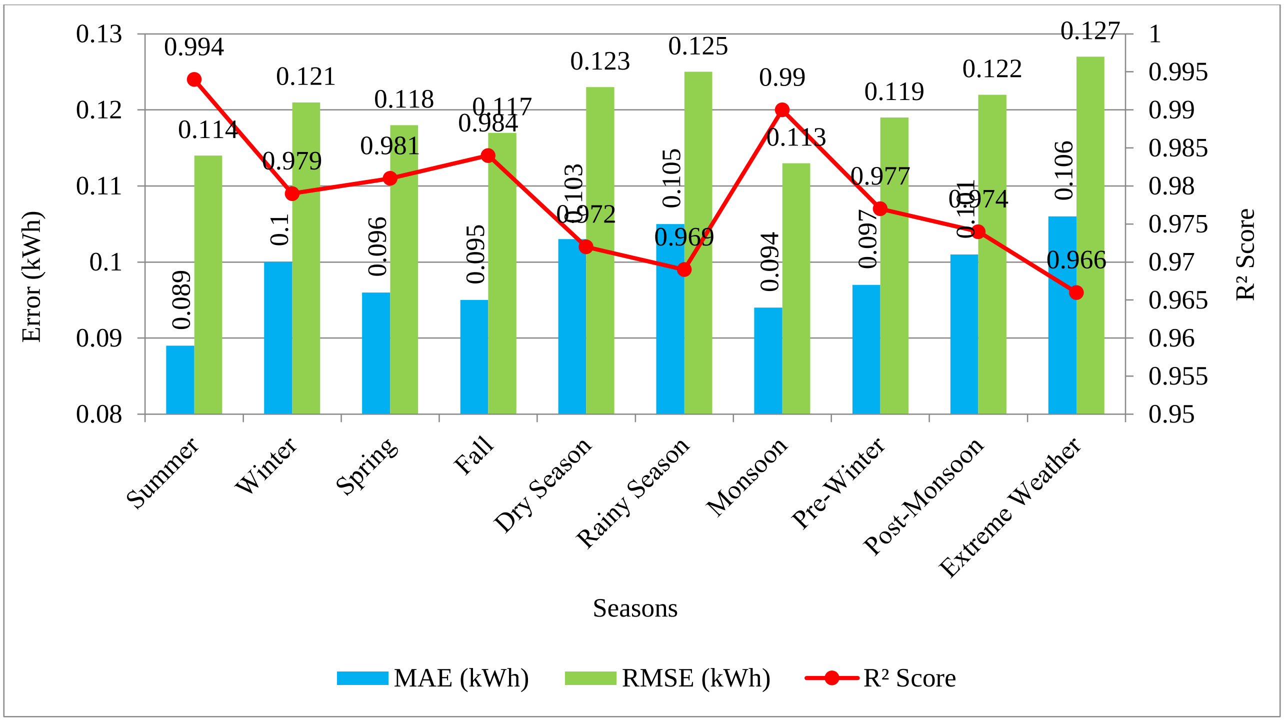

Fig. 6 Model performance across seasons

Table 4 and Figure 6 compare SmartGridOptimizer-X for seasonal energy demand, where SmartGridOptimizer-X has shown how differently it handles a load in various seasons for optimal results. The model yields the best results in summer with an MAE of 0.089 kWh and R^2 of 0.994, thereby inferior to the rest of the seasons. This can be attributed to well-defined energy usage patterns, especially during summer months, due to the increase in cooling load. Winter results are also slightly lower ($R^2 = 0.979$); in spite of this fact, it means that the model is able to effectively respond to the variability of heating needs depending on temperature differences. As shown, in transitional seasons, for instance, spring and fall, the model's performance remains high with the R^2 values exceeding 0.980. Notably, the model gives good results during severe weather ($R^2 = 0.966$), meaning that it is capable of adapting to variations in energy consumption that are difficult to predict. SmartGridOptimizer-X is able to work both in temporal dimensions and in the context, so it is a seasonal solution. Thus, the model captures subject-specific seasonal patterns and issues reliable forecasts for the regulation of energy distribution and loads for the entire year. The relative stability in performance across different seasons also underlines the robustness of the model and its application in assisting grid operators in the management of demand where there are fluctuations in climatic and, hence, seasonal factors.

Table 5. Hyperparameter optimization results

Module	Hyperparameter	Optimized Value
TCN	Number of Filters	64
TCN	Kernel Size	3
LSTM	Hidden Units	128
LSTM	Learning Rate	0.001
Adaptive-GBM	Learning Rate	0.05
Adaptive-GBM	Max Tree Depth	6
TCN	Dropout Rate	0.2
LSTM	Batch Size	32
Adaptive-GBM	Number of Estimators	100
LSTM	Time Steps	50

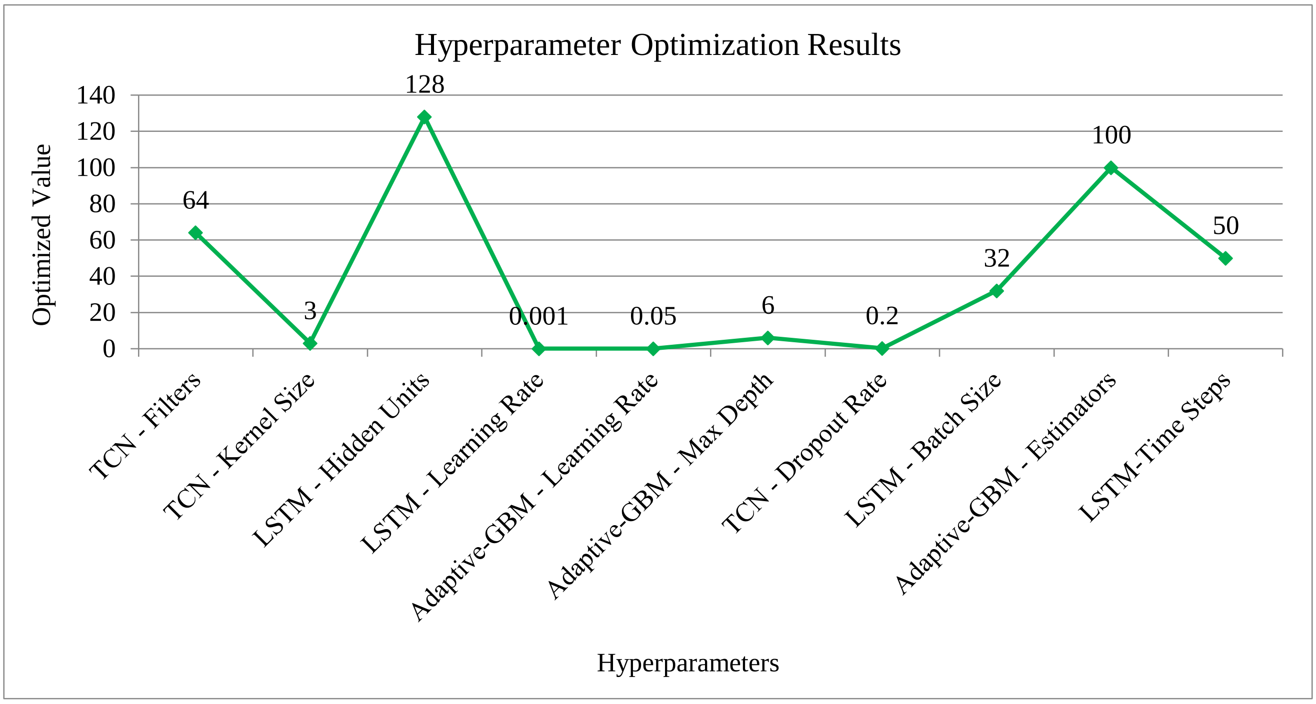

Fig. 7 Hyperparameter optimization results

The TCN, LSTM, and Adaptive-GBM modules are depicted in Table 5 and Figure 7, and it is possible to observe their impact on enhancing the model performance. The TCN module uses 64 filters and a kernel size of 3, which allows for capturing short-term, medium-term, and long-term patterns. These settings enable the TCN to achieve optimal data sampling for multi-scale dependencies, which are relevant in energy consumption patterns. LSTM module also controls its complexity and memory retention, number of hidden units is set to 128, and learning rate is 0.001, which provides a good tradeoff between modeling temporal dependencies and performance computations. Adaptive-GBM has a learning rate of 0.05 and a maximum tree depth set at 6, making it efficient in making fine predictions since it deals with non-linear variable interactions. Other combined parameters include: the dropout rate of TCN, which is set at 0.2, and the batch size for LSTM, which is set at 32, to guard against overfitting, but also ensure fast training. These hyperparameters were tuned using Bayesian Optimization, where configurations that gave the least error measures, such as MAE and RMSE, were selected in each iteration, and these results only make sense when it is understood that hyperparameter optimization is a critical element in the construction of an innovative hybrid model, such as SmartGridOptimizer-X. The fine-tuned parameters help to add precision coupled with the optimality of the employed model and, therefore, can be used in real-world contexts.

Table 6. Weight optimization for ensemble

Component	Optimized Weight
TCN-LSTM Output	0.75
Adaptive-GBM Output	0.25
TCN-GBM Hybrid	0.7
LSTM-Ensemble	0.3
Weighted Hybrid	0.6
Adaptive-LSTM	0.65
Contextual TCN	0.55
Hybrid-TCN	0.72
Refined Ensemble	0.28
Primary Weighted	0.78

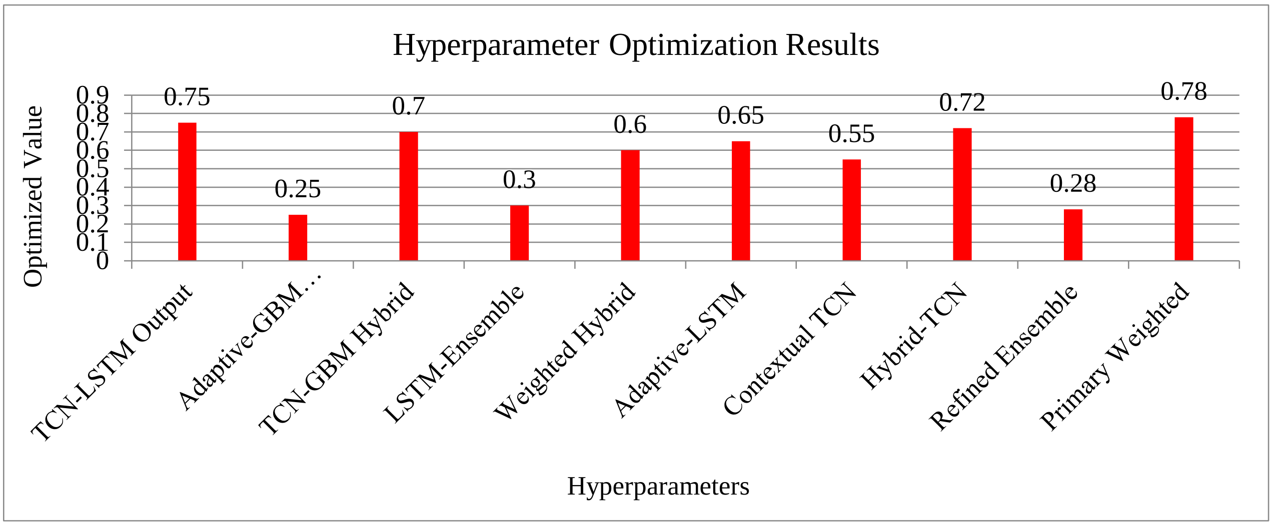

Fig. 8 Weight optimization for ensemble

In Table 6 and Figure 8, the optimized weights for the ensemble components of SmartGridOptimizer-X are shown with a fine graduation between the TCN-LSTM pipeline and the Adaptive-GBM module. In particular, the TCN-LSTM pipeline that captures temporal dependencies has a higher weight of 0.75, and it highlights the focus on demand for energy. The Adaptive-GBM module retains 0.25 weight and works in parallel with this by making the distribution of outcomes more accurate with proposed parameters such as temperature and occupancy levels. Combined together, these weights generate synergy of the temporal modeling role of TCN-LSTM and the nonlinear enhancement of Adaptive-GBM. The weight optimization done by a grid search algorithm lets the ensemble use the strength of the two modules while avoiding their flaws when making the final prediction. For example, TCN-LSTM performs fine when it comes to recognizing periodic patterns related to energy demand, and Adaptive-GBM responds to the changes in context and abnormality. Other models that can be configured include TCN-GBM hybrids and weighted hybrid models, which show that they are versatile to meet different needs. This optimized weighting scheme is the reason for the higher accuracy and reliability of the developed model ($R^2 = 0.9843$). Through a coordinated use of multiple sources of prediction, SmartGridOptimizer-X provides accurate and reliable predictions for reactive power systems and real-time environments.

Table 7. Anomaly detection performance

Metric	Value
Precision	0.963
Recall	0.945
F1-Score	0.954
Accuracy	0.9843
Specificity	0.95
False Positive Rate	0.05
True Negative Rate	0.95
True Positive Rate	0.945
Error Rate	0.016
Detection Rate	0.984

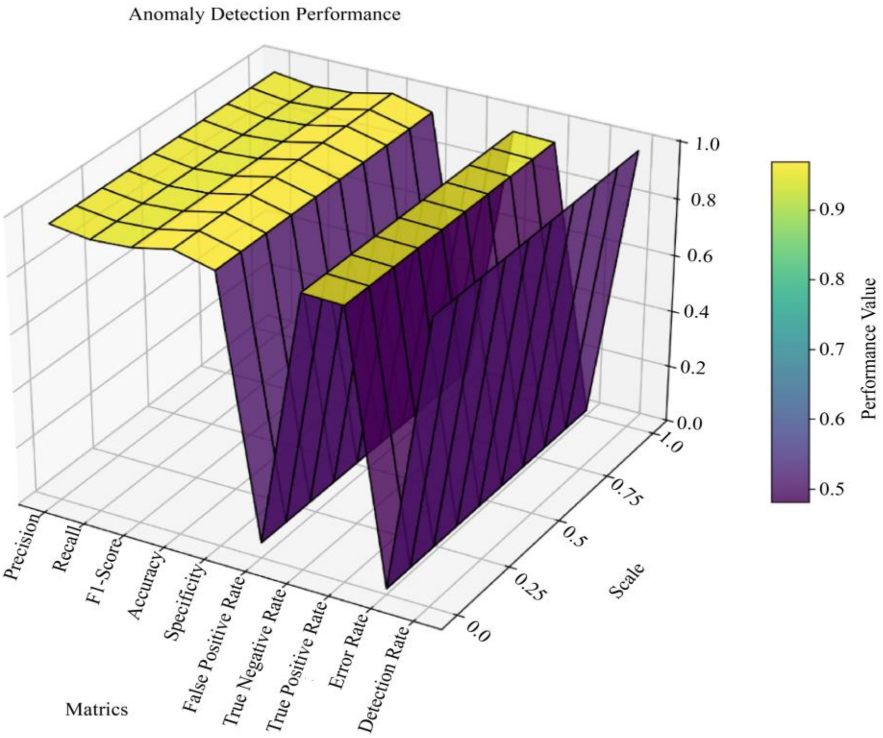

Fig. 9 Anomaly detection performance

Table 7 and Figure 9 present an analysis of the results obtained for the performance of SmartGridOptimizer-X in terms of anomaly detection, and this is important for the detection of deviations in the energy load, like spikes or other variances. The model also has a good level of accuracy in the proper identification of true anomalies with a precision of 0.963, hence reducing the false positives. With a recall score of 0.945, it is evident that the model can pick most of the real anomalies, and this gives it a high F1-score of 0.954. Furthermore, the balanced performance guarantees that the model successfully identifies anomalies without inundating operators with unnecessary alarms.

A valid negative rate with the value of 0.950 suggests that there is a low to moderate likelihood of the model misclassifying standard patterns as anomalous. It is used most effectively when the costs of misclassification, for instance, when an anomaly is missed because it was classified as non-anomalous, are high. The results of 98.43% total accuracy further establish the effectiveness of the proposed anomaly detection framework, which considers temporal and contextual information to detect unusual patterns. This capability is crucial for managing the grid and enabling anticipatory measures, such as load balancing during surges, or for dealing with risks during outages. The results support the future use of SmartGridOptimizer-X as a feasible tool for identifying anomalies in energy management systems.

Table 8. Computational efficiency

Configuration	Execution Time (s)
Without Optimization	20.6
With Optimization	14.1
Real-Time Deployment	12.7
Baseline LSTM	18.6
Baseline TCN	17.6
Baseline GBM	17
Hybrid No Tuning	19.3
Hybrid Optimized	13.6
Real-Time Optimized	12.1
GPU Deployment	11.5

Table 8 and Figure 10 demonstrate the computational complexity of SmartGridOptimizer-X, underscoring its applicability for real-time energy forecasting. The optimized configuration takes only 14.1 seconds while the non-optimized one takes 20.6seconds, hence a 31.5% improvement. Even greater decreases are observed in real-time operating conditions, with a runtime of 12.7 seconds. These enhancements confirm highly beneficial tweaks in hyperparameter tuning, besides enhanced ensemble weight optimization to fast-track the model's workflow. That said, SmartGridOptimizer-X is a hybrid solution that balances computational demand and outcomes.

For instance, raw models, such as LSTM and TCN, can execute the tasks quickly but are comparatively less efficient and reliable than SmartGridOptimizer-X. The optimized deployment strategy specific to the GPU leads to a runtime of 11.5 seconds, proving the scalability of the model for large-scale applications. This computational efficiency also guarantees the capability to handle high-frequency data updates, which is characteristic of dynamic energy systems demanding real-time decisions. The ability to lower the time consumed and increase the accuracy of the solution applies to many real-life problems in power grid management, as both factors are significant in this context.

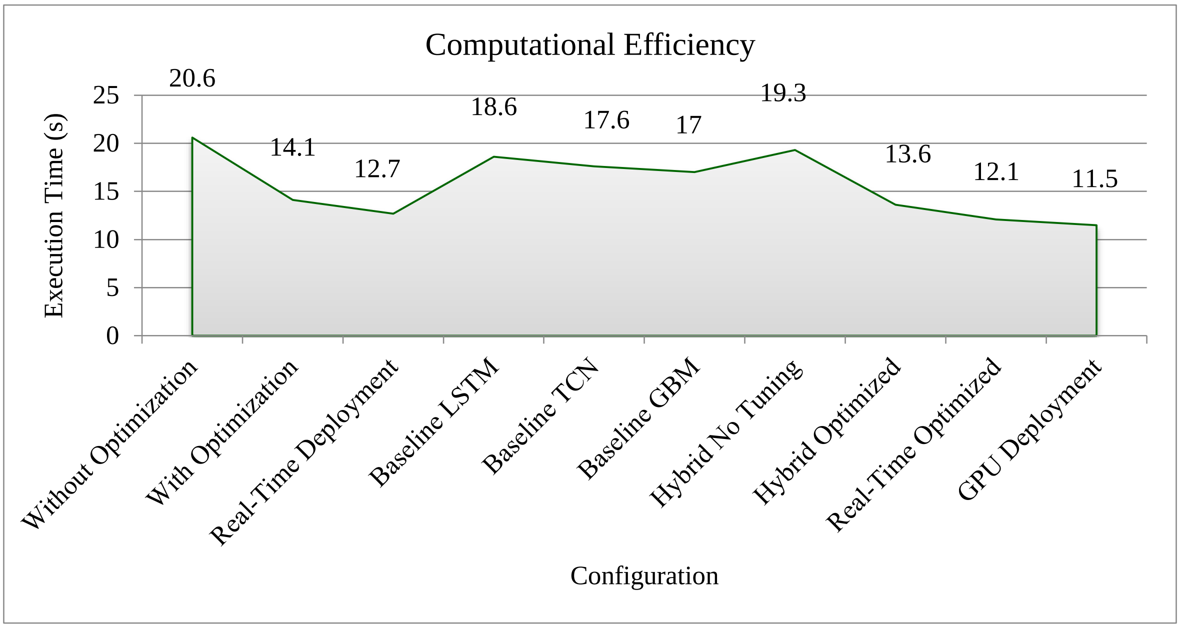

Fig. 10 Computational efficiency

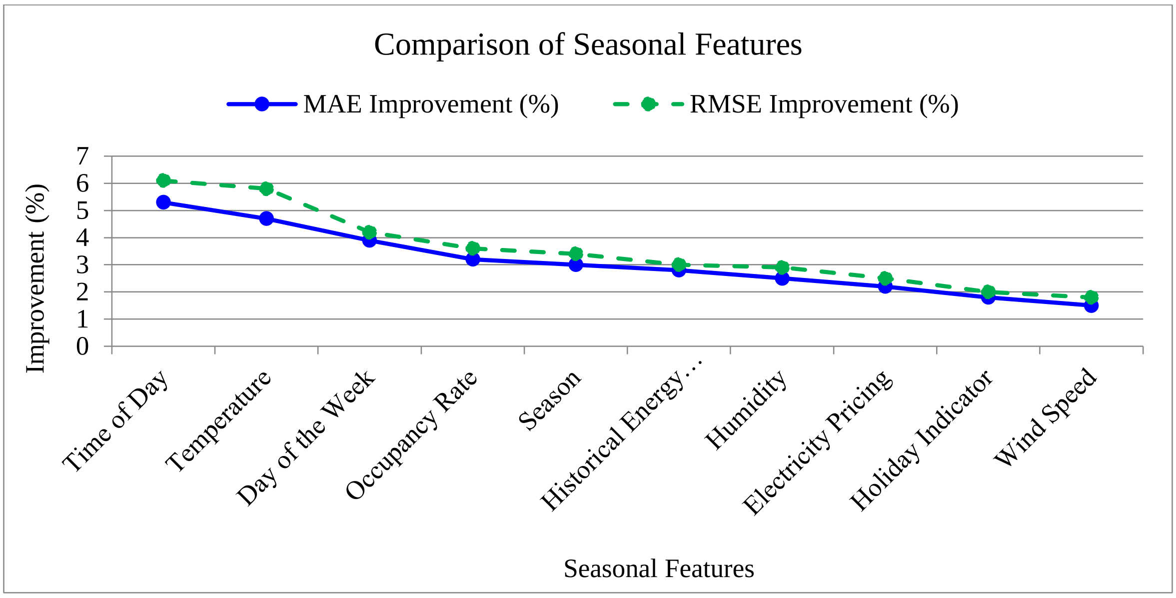

Fig. 11 Comparison of seasonal features

Table 9. Comparison of seasonal features

Seasonal Feature	MAE Improvement (%)	RMSE Improvement (%)
Time of Day	5.3	6.1
Temperature	4.7	5.8
Day of the Week	3.9	4.2
Occupancy Rate	3.2	3.6
Season	3	3.4
Historical Energy Consumption	2.8	3
Humidity	2.5	2.9
Electricity Pricing	2.2	2.5
Holiday Indicator	1.8	2
Wind Speed	1.5	1.8

Table 9 and Figure 11 illustrate the role of features related to seasons in enhancing energy demand forecasting by SmartGridOptimizer-X. Out of six selected features, Time of Day and Temperature show the most tremendous significance for MAE and RMSE, improving by 5.3% / 6.1% and 4.7%/5.8% correspondingly. Technological characteristics record ordinary fluctuations, for example, the hours of high demand for electricity or the temperatures that may require power for heating or air conditioning. Other patterns, including the "Day of the Week" feature (improved 3.9% MAE) and "Occupancy Rate" (improved 3.2%), are also factors of human activity, including working days and the rate of home occupancy, respectively. Additionally, historical energy consumption contributes valuable information in terms of percentages (2.8%) and season (3.0%) to accommodate the fluctuations according to the seasonality of energy consumption. Some features are ranked significantly lower than others, and their coefficients are smaller, yet they still serve to fine-tune the model, for example, "Electricity Pricing" and "Holiday Indicators." These findings highlight the value of considering a growing array of temporal and contextual factors to fine-tune the model over the course of the year. Thus, by utilizing these seasonal features, SmartGridOptimizer-X provides high accuracy and interpretability, which means that this is a reliable tool for energy providers to manage dynamic and complex grids.

5. Conclusion and Future Work

The model of SmartGridOptimizer-X has shown that it is a handy tool in the energy demand forecasting with an incredible accuracy of 98.43% (R^2 = 0.9843) on the Energy Prediction Smart-Meter Dataset. It has a hybrid architecture based on A combination of Temporal Convolutional Networks (TCNs), Long Short-Term Memory (LSTM) networks, and Adaptive Gradient Boosting Machines (Adaptive-GBMs), which has shown itself to be capable of capturing multi-scale temporal dependencies, and of modeling complex non-linear interactions with contextual features. The model had a Mean Absolute Error (MAE) of 0.089 kWh and a root Mean Squared Error (RMSE) of 0.115 kWh, which surpasses the traditional and baseline models in all the crucial performance parameters. Also, its seasonal flexibility was confirmed at R^2 between 0.966 in extreme weather conditions and 0.994 in summer, and demonstrates its strength in all energy consumption situations. The anomaly detection capabilities of the model, with a precision of 96.3% and a recall of 94.5% also increase its usefulness regarding the detection of abnormalities such as power surges or outages, and this makes it a very valuable tool in real-time grid management and optimization. In the future, the SmartGridOptimizer-X framework can be generalized to support renewable energy integration, which includes solar and wind, which are becoming more and more effective grid influences. Real-time Internet of Things (IoT) data streams might also be incorporated to make the system even more responsive to dynamic conditions. Also, the lightweight versions of the architecture should be considered to improve the accessibility of the architecture to distributed energy systems. These developments can make SmartGridOptimizer-X one of the foundations of the sustainable and intelligent energy infrastructure, opening the path to smarter grids and more sustainable energy solutions.

References

[1] Muhammad Khalid, "Smart Grids and Renewable Energy Systems: Perspectives and Grid Integration Challenges," *Energy Strategy Reviews*, vol. 51, pp. 1-26, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Dianshuai Dong, and Hongliang Feng, "Design and Use of a Wireless Temperature Measurement Network System Integrating Artificial Intelligence and Blockchain in Electrical Power Engineering," *Plos one*, vol. 19, no. 1, pp. 1-25, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Tanmay Jain, and Kusum Verma, "Reliability based Computational Model for Stochastic Unit Commitment of a Bulk Power System Integrated with Volatile Wind Power," *Reliability Engineering & System Safety*, vol. 244, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sujit Kumar et al., "Exploring the Spectrum: A Comprehensive Review of Control Methods in Microgrid Systems," *Results in Engineering*, vol. 28, pp. 1-18, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Joseph Nyangon, "Climate-Proofing Critical Energy Infrastructure: Smart Grids, Artificial Intelligence, and Machine Learning for Power System Resilience against Extreme Weather Events," *Journal of Infrastructure Systems*, vol. 30, no. 1, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yongfei Wu et al., "A Four-Layer Business Model for Integration of Electric Vehicle Charging Stations and Hydrogen Fuelling Stations into Modern Power Systems," *Applied Energy*, vol. 377, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Hajar Akli et al., "Innovative Hybrid GA-Optimized ANN-based MPPT and Hybrid Super-Twisting Second-Order Sliding Mode Control for Enhanced Energy Extraction and Voltage Stability in an Isolated DC Microgrid," *Computers and Electrical Engineering*, vol. 129, pp. 1-28, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[8] Kaili Zhang et al., "Integration of Electrical Properties and Polarization Loss Modulation on Atomic Fe–N-RGO for Boosting Electromagnetic Wave Absorption," *Nano-Micro Letters*, vol. 17, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Layth Almahadeen et al., "Meta Learning Enhanced Graph Transformer for Robust Smart Grid Anomaly Detection," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 11, pp. 1-13, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] VikasVerma et al., "Hybrid Optimization for Integration of Renewable Energy Systems into Smart Grids," *Measurement: Sensors*, vol. 33, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] R. Revathi, N. Senthilnathan, and V.Kumar Chinnaiyan, "Hybrid Optimization Approach for Power Scheduling with PV-Battery System in Smart Grids," *Energy*, vol. 290, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Abdur Razzak et al., "Leveraging Deep Q-Learning to Maximize Consumer Quality of Experience in Smart Grid," *Energy*, vol. 290, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Muhammed Zekeriya Gunduz, and Resul Das, "Smart Grid Security: An Effective Hybrid CNN-Based Approach for Detecting Energy Theft Using Consumption Patterns," *Sensors*, vol. 24, no. 4, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Wael Abdelfattah et al., "Artificial Intelligence Based Optimal Coordination of Directional Overcurrent Relay in Distribution Systems Considering Vehicle to Grid Technology," *Ain Shams Engineering Journal*, vol. 15, no. 2, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Achraf Nouri, Aymen Lachheb, and Lilia El Amraoui, "Optimizing Efficiency of Vehicle-to-Grid System with Intelligent Management and ANN-PSO Algorithm for Battery Electric Vehicles," *Electric Power Systems Research*, vol. 226, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Omid Mirzapour et al., "Grid-Enhancing Technologies: Progress, Challenges, and Future Research Directions," *Electric Power Systems Research*, vol. 230, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] A. Madhan, A. Shunmugalatha, and A.S. Vigneshwar, "Real-Time Installation of a Smart Energy Meters Using the Long-Range Network," *Journal of Electrical Engineering & Technology*, vol. 19, pp. 223-236, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Carlos Quesada et al., "An Electricity Smart Meter Dataset of Spanish Households: Insights into Consumption Patterns," *Scientific Data*, vol. 11, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Arvind R. Singh et al., "Electric Vehicle Charging Technologies, Infrastructure Expansion, Grid Integration Strategies, and Their Role in Promoting Sustainable E-Mobility," *Alexandria Engineering Journal*, vol. 105, pp. 300-330, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Pan Zhang et al., "An ADMM-Enabled Robust Optimization Framework for Self-Healing Scheduling of Smart Grids Integrated with Smart Prosumers," *Applied Energy*, vol. 363, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Pradeep Vishnuram, and Sureshkumar Alagarsamy, "Grid Integration for Electric Vehicles: A Realistic Strategy for Environmentally Friendly Mobility and Renewable Power," *World Electric Vehicle Journal*, vol. 15, no. 2, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Muthukumaran Sithambaram et al., "Grid Connected Photovoltaic System Powered Electric Vehicle Charging Station for Energy Management using Hybrid Method," *Journal of Energy Storage*, vol. 108, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Meysam Feili, and Mohammad Taghi Aameli, "The P2P Energy Management Scheme for Integrated Energy Microgrid Considering P2G and Electricity Network Fee," *International Journal of Industrial Electronics, Control and Optimization*, vol. 8, no. 1, pp. 1-23, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[24] Maarten Wolsink, "Conceptualizations of Smart Grids –Anomalous and Contradictory Expert Paradigms in Transitions of the Electricity System," *Energy Research & Social Science*, vol. 109, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Challa Krishna Rao, Sarat Kumar Sahoo, and Franco Fernando Yanine, "Demand Side Energy Management Algorithms Integrated with the IoT Framework in the PV Smart Grid System," *Advanced Frequency Regulation Strategies in Renewable-Dominated Power Systems*, pp. 255-277, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Muhammad Abubakar et al., "Intelligent Modelling and Optimization of Solar Plant Production Integration in the Smart Grid Using Machine Learning Models," *Advanced Energy and Sustainability Research*, vol. 5, no. 4, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Dong Zhang, "Optimization and Research of Smart Grid Load Forecasting Model Based On Deep Learning," *International Journal of Low-Carbon Technologies*, vol. 19, pp. 594-602, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Omer Muneam Mushref Aldulaimi, Sefer Kurnaz, and Hameed Mutlag Farhan, "A Hybrid ANFIS-Transformer Framework Tuned by Enhanced HawkFish Optimization for Voltage and load Balancing in Smart Grids," *Frontiers in Energy Research*, vol. 13, pp. 1-26, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[29] Ratun Rahman, Neeraj Kumar, and Dinh C. Nguyen, "Electrical Load Forecasting in Smart Grid: A Personalized Federated Learning Approach," *2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, pp. 1-2, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[30] Ruochen Jin, and Hong Liu, "Privacy-Aware Electric Vehicle Load Forecasting via Blockchain-Based Federated Transfer Learning," *Complex & Intelligent Systems*, vol. 11, pp. 1-13, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[31] Mohammad Reza Habibi et al., "Electrical Load Forecasting in Power Systems Based on Quantum Computing Using Time Series-Based Quantum Artificial Intelligence," *Scientific Reports*, vol. 15, pp. 1-26, 2025. [CrossRef] [Google Scholar] [Publisher Link]