Journal of Environmental and Agricultural Studies
ISSN: 710-1401
DOI: 10.32996/jeas
Journal Homepage: www.al-kindipublisher.com/index.php/jeas

JEAS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Leveraging AI for Sustainable and Cost-Effective Decentralized Energy in Rural U.S. Regions

Rayhanul Islam Sony[1]✉, Md Ariful Islam Bhuiyan[2], Dipta Roy[3]
[1]*College of Graduate Professional Studies, Trine University, One University Avenue, Angola, 46703, Indiana, USA.*
[2]*Department of Electrical and Computer Engineering, California State University, 18111 Nordhoff Street, Northridge, 91330, CA, USA.*
[3]*Department of Electrical and Computer Engineering, California State University at Northridge, 18111 Nordhoff Street, Northridge, 91330, CA, USA.*
Corresponding Author: Rayhanul Islam Sony, **E-mail**: rsony23@my.trine.edu

| ABSTRACT
Sustainable and affordable decentralized power in rural parts of the U.S. is difficult to accomplish because of inadequate infrastructure and transmission costs, and centralized grids are susceptible. This paper suggests an artificial intelligence (AI) hybrid framework combining XGBoost and a neural network to provide precise predictions of renewable generation and operational expenses. More than 13000 electric power plants databases in the United States underwent preprocessing in order to solve the problem of missing data, heterogeneity, and rural-urban disparities. The proposed model (XGBoost+NN) was compared with KNN, Autoencoder, PINN, CNN-LSTM, and BiLSTM-Attention. It outperformed all baselines with an MSE of 0.0003, RMSE of 0.0178, R^2 of 0.999, and MAPE of 1.36%. The error in predicting costs was only 0.08% and this is both a technical strength and an economic importance. These findings indicate the model's scalability, interpretability, and possible use to facilitate equitable access to clean and affordable energy in underserved rural populations.

| KEYWORDS
Decentralized energy, Rural electrification, AI, Hybrid models, Renewable energy, Cost-effective energy systems, Sustainability.

| ARTICLE INFORMATION
ACCEPTED: 01 September 2025 **PUBLISHED:** 18 September 2025 **DOI:** 10.32996/jeas.2025.6.3.1

1. Introduction
The switch to sustainable and cost-efficient energy systems has become one of the greatest Obstacles of the 21st century [1]. As the United States makes ambitious climate goals, tackling equitable access to renewable and decentralized energy in rural regions has become just as important from a technological and socio-economic perspective [2]. Rural communities alone constitute over 20% of the total US population - energy access in these communities is a specific issue [3]. These include lack of sound infrastructure, higher transmission costs, fossil fuel dependence, and susceptibility to disruption in the architecture of centralized grids [4]. At the same time, rural areas are also in a unique position to access renewable and decentralized energy resources such as solar, wind, and small-scale hydro by virtue of the availability of land and geographic location [5]. However, deploying the decentralized energy systems to farming areas is hindered by various barriers [6]. Such critical issues as financial viability, grid integration, and sustainability should be dealt with simultaneously. Traditional modeling approaches based either on fully statistical frameworks or physics study-based approaches tend to neglect the complexity of interaction between the operational capacities as well as the generation effectiveness, which also takes into account the type of fuel as well as the locational classification disparities [7]. As a result, rural areas are still underserved, and energy inequities still exist[8]. Addressing this challenge involves

not only advanced predictive modelling but systematic preprocessing of diverse and heterogeneous datasets in an attempt to ensure that insights learned from machine learning (ML) models are reliable, interpretable, and actionable. [9, 10].

Energy decision-making in a rural setting involves an effective set of tools that can evaluate issues of cost-sustainability trade-offs [11]. Current approaches, therefore, all too often fail in three important aspects. First, the energy datasets are often full of missing values, inconsistent categorical coding, and heterogeneous formats among plants and states [12]. Downstream models fail to achieve reliability unless they are within cleaning, standardization, and feature engineering [13]. Second, traditional regression techniques and techniques, as well as methods in shallow learning are not always able to capture the nonlinearity between the relationship between operational capacity and emission intensities, as well as the relationship between renewable adoption and location-specific constraints [14, 15]. Even sophisticated deep learning (DL) models like CNNs and LSTMs, although they are strong for temporal/spatial patterns, may not generalize well because they are confronted with tabular energy datasets that blend categorical, numerical, and engineered features [16]. Third, although isolated use of individual models, for example, XGBoost and neural networks (NN), has been shown to have success in predictive tasks, their use is not robust. There is still a gap in uniting the powerful characteristics of ensemble-based learners and deep neural architectures to work within an interpretable and reproducible pipeline specifically designed to handle rural energy prediction tasks [17]. Recent developments in AI offer a new, unprecedented opportunity to bridge this gap. Hybrid modeling, which combines the interpretability and power of ensemble models such as XGBoost and the representational power of deep NN, is particularly promising [18]. When combined with a systematic data preprocessing pipeline, which could include handling missing values, data standardization, feature engineering, aggregation, encoding, scaling, and

stratified splitting, these models can provide highly accurate, scalable, and generalizable predictions. Our work is fueled by this motivation and aims at a street-level and important application: using AI for sustainable and cost-effective decentralized energy in rural areas in the U.S. By combining hybrid modelling and careful preprocessing, we aim to address dual challenges of cost optimization and sustainability, as well as rural-urban differences in energy access that are explicitly documented and analyzed [19].

Although a notable degree of AI has been used in energy systems research, there are still a number of gaps. Existing studies tend to ignore systematic preprocessing steps; the result is frequently noisy and inconsistent input data [20]. Few studies combine the strength of ensemble learning and a deep neural network in an optimized form [10]. Most existing studies consider energy plants or states in a homogenous way with no clear encoding of rural characteristics or renewable adoption heterogeneity [21]. Furthermore, in many studies, comparisons are limited to specific and limited model sets without thorough comparison with conventional and advanced baselines. By addressing these gaps directly, our study provides a basis for both technical progress and real-world policy relevance.

The primary aim of this work is to build a comprehensive AI-powered Structure for a sustainable and cost-effective decentralized energy planning for rural U.S. regions. To achieve this, the research is first devoted to building a rigorous preprocessing pipeline to remove any possible inconsistency and improve the data quality through data cleaning, standardization, feature engineering, aggregation, encoding, scaling, and stratified data split. Traced from this groundwork, a hybrid predicted model is introduced alongside and in comparison, with the XGBoost and a neural network to introduce the concept of ensemble robustness and nonlinear learning capability. Finally, the framework is benchmarked against an array of conventional and advanced baseline models, whilst also making the socio-environmental analysis feasible through features such as the use of rural indicators and emission intensities. Together, these goals provide a structured solution to bridge the methodological gaps that currently exist, and for the reconciliation of predictive modeling with the challenges of sustainability and equity in the rural energy systems.

This paper makes the following contributions:

- Novel preprocessing pipeline: We proposed a strict, end-to-end framework for heterogeneous energy datasets that provides consistency, interpretability, and model readiness.
- Hybrid model innovation: We designed hybrid model architecture, which combines XGBoost and a deep neural network with an ensemble approach, with the final test accuracy of 99.94%, better than the performance of five sophisticated baseline models.
- Feature engineering with socio-environmental context: We introduced capacity factor, CO2 intensity, renewable flags, and rural indicators as engineered features to enable analysis of sustainability and equity dimensions alongside cost predictions.
- Comprehensive benchmarking: We systematically evaluated the proposed model against KNN, Autoencoder-Predictor, Physics-Informed NN, CNN-LSTM, and BiLSTM-Attention with consistent superiority.

- Adaptive ensemble strategy: We designed a dynamic ensemble selection mechanism to optimally balance contributions of XGBoost and the neural network, in such a way that both the weights are automatically adjusted to maximize the validation accuracy and generalization of the resulting model.

The remainder of this paper is presented as follows: Section 2 revises the work related to decentralized energy systems, sustainable modeling, and hybrid AI approaches. Section 3 presents the methodology, including the detailed data preprocessing pipeline, feature engineering design, and the proposed hybrid XGBoost–Neural Network model. Section 4 reports experimental results, comparing the proposed model against five baseline approaches (KNN, Autoencoder-Predictor, Physics-Informed NN, CNN-LSTM, and BiLSTM-Attention), and discusses performance as well as interpretability of the results. Finally, Section 5 concludes the study by highlighting the key findings, limitations, and potential directions for future research.

2. Literature Review

Microgrids incorporating renewable energy systems like solar power and wind power are emerging as a key in finding sustainable power solutions. However, such systems have difficulties regarding intermittent power production and grid stability. Advances in AI-powered Energy Management Systems (EMS), the Internet of Things (IoT), and edge computing applications point to future capability to optimize micro-grid operations. These technologies seek to enhance the integration of renewables, load balance, and forecasting accuracy. Now, we will delve into some of the latest research and developments that are showcasing the role of AI, IoT, and EMS in enhancing the efficiency and consistency of microgrid systems.

Singh et al. [22] proposed a Support Vector Regression (SVR) approach for energy management and power forecasting in grid-connected microgrids using comprehensive historical energy production data. The method leverages SVR with Radial Basis Function (RBF) kernel to anticipate power output coming from solar PV and wind sources. Their contributions include developing a novel microgrid power management system that integrates forecasting abilities based on ML with multiple distributed energy sources. With Mean Absolute Errors of 0.547 for solar PV and 0.825 for wind scenarios, the SVR model produced Mean Squared Errors of 2.002 for solar PV and 3.059 for wind power forecasting. The model requires continuous validation to maintain greater accuracy; however, it showed differences at peak hours.

Singh et al. [23] proposed an ORA-DL system for adaptive grid resource allotment using IoT sensors and smart meter data alongside real-time weather/market information. For resource allocation and energy demand prediction, the framework utilizes deep NN, reinforcement learning (RL), and multi-agent decision-making. The authors achieved 93.38% energy demand prediction accuracy, 96.25% grid solidity, and reduced energy loss to 12.96% while cutting operational costs by 22.96%. The approach integrates PV, CHP, GHP, battery storage, and thermal energy storage technologies. However, the study lacks a discussion of computational complexity for real-time deployment and doesn't address potential cybersecurity vulnerabilities in IoT-enabled systems.

Masrur et al. [24] designed a resilient optimization technique for interconnected multi-energy microgrids using real-world hospital data from Japan with MILP optimization techniques. The study incorporates PV, CHP, geothermal heat pumps, battery storage, and thermal energy storage for electricity and heating demand management. Authors demonstrated 3-5 million dollars net present value savings compared to grid-only systems and achieved 7-8% additional cost reduction through cooperative scheduling using Shapley value and nucleolus cost allocation mechanisms. The framework showed 45-60% CO2 emission reduction and enhanced outage survivability during 24-hour grid failures. However, the model relies heavily on accurate forecasting data and doesn't fully address dynamic environmental condition changes or comprehensive cybersecurity measures for interconnected systems.

Hoummadi et al. [25] proposed a microgrid optimization system using GA, ABC, and ACO algorithms to design cost-effective energy systems for a 100-unit residential complex with solar, wind, and battery storage. The framework employed genetic algorithms for natural selection-based optimization, artificial bee colony for honeybee foraging behavior modeling, and ant colony optimization for decision-making processes. Their contribution includes developing AI-driven microgrid optimization that maximizes renewable energy utilization and improves system resilience. The optimization achieved 0.037 USD/kWh electricity cost, a 67% reduction from the reference cost in Fez of 0.115 USD/kWh. However, the study is geographically limited to Fez and lacks a comprehensive environmental impact assessment across different climate conditions.

Oladapo et al. [26] examined the use of ML architectures along LSTM, Random Forest (RF), SVM and ARIMA for energy generation prediction and grid optimization. Their work involved applying these models to historical energy datasets, evaluating them with MAE and RMSE metrics. Main insights included a 15% boost in grid efficiency and 10-20% increase in battery storage efficiency. RF achieved the lowest MAE, reducing prediction error by 8.5%. Limitations include the dependency on large, high-quality datasets and challenges in integrating these models across different energy systems.

Li and Feng [27] proposed a DL-based approach for integrating carbon neutrality in rural ecological construction, using a dataset of 15 years of environmental data. The technique involves a convolutional neural network (CNN) to predict carbon emission levels and a long short-term memory (LSTM) network for forecasting energy consumption patterns. Their contribution includes offering a comprehensive model for balancing ecological sustainability with industrial growth. Results showed a 12% lowering of carbon

emissions and a 20% better in energy efficiency over traditional methods. Limitations include the inability of the model to generalize across regions with different environmental conditions.

Danish and Senjyu [28] proposed an AI-driven framework for integrating circular economy policies into energy systems, using a multi-layer perceptron (MLP) model trained on data from 50 global case studies. The framework applies predictive models for resource usage optimization and waste reduction. Their key contribution is the development of a policy alignment model that incorporates AI to drive sustainable energy transitions. The framework demonstrated a 25% reduction in resource wastage and a 30% increase in system efficiency in selected case studies. Limitations include the model's reliance on high-quality data inputs from various industries.

Singh et al. [29] introduced an optimization method for microgrids, where price-elastic demand response is adopted. programs (DRPs) and Greedy Rat Swarm Optimization (GRSO). Their results showed a 15.4% cut in costs of operation, saving €158.87, and a 13.23% decrease in carbon emissions. The GRSO algorithm outperformed traditional methods, achieving a convergence time of 3367.50s and an NMSE of 1.10%. The study highlights the benefits of integrating DRPs with renewable energy sources for enhanced economic and environmental efficiency, though it faces challenges with data quality and real-time adaptability.

Paul et al. [30] introduced a QPSO-based method for optimizing grid-connected micro-grids. Results showed a 9.67% cut in operational expenses, saving €158.87, and a 13.23% decrease in carbon emissions. The model achieved an F1 score of 95.5% for energy forecasting accuracy, outperforming traditional methods. QPSO showed faster convergence and better performance in minimizing both cost and emissions, although its scalability across different grid sizes remains a limitation.

Zhao [31] proposed a DL structure for managing resources within a smart community. The system, using LSTM and CNN models, showed an 18.7% reduction in resource consumption and a 16.2% reduction in operating costs. LSTM for water demand prediction achieved an MAE of 1.8 and RMSE of 2.5. The framework outperformed traditional methods in predictive accuracy, though scalability challenges remain in large urban settings.

Liu et al. [32] presented a multi-faceted energy optimization approach for management in smart grids using the ANFIS algorithm. Results showed a 15% increase in load forecasting accuracy and a 20% in grid instability. The ANFIS model achieved an RMSE of 3.5 and an R^2 of 0.92. The model demonstrated superior establishment over traditional methods, although it is sensitive to high-quality input data and extreme weather conditions.

Elhabyb et al. [33] applied ML models to predict consumption of energy in educational institutions, LSTM, RF, and Gradient Boosting Regressor are used. The results showed an RMSE improvement of 12%, with LSTM achieving an MAE of 4.1 and R^2 of 0.94. RF showed a 10% increase in prediction accuracy over traditional models. Precision and recall for energy demand predictions were 90% and 85%, respectively. The study emphasized the importance of building-specific data for accurate predictions.

Parvathareddy et al. [34] explored ML techniques applied to mining energy optimization, focusing on predictive maintenance, energy demand forecasting, and real-time process control. The authors highlighted that models like RF and LSTM can reduce energy consumption by up to 15%. Their review also discussed ML integration with Digital Twin and RL, improving operational efficiency and reducing downtime by 20%. Challenges include high initial costs and the difficulty of generalizing models across different mining environments. Further studies are essential to enhance the expandability and efficiency of these models.

Li et al. [35] showed the connection between AI and renewable energy investment in China using data from 2010 to 2024. The study found that AI significantly boosts renewable energy investment by improving energy forecasting and optimizing grid integration. AI's impact was measured through a 20% increase in investment efficiency, and it improved energy production predictions with RMSE reductions of 12% for wind energy and 14% for solar energy. Limitation shows the difficulties like data quality and high capital costs, that limit AI's long-term effectiveness. AI's role in enhancing energy storage and grid efficiency was confirmed.

Khalid and Jamshed [36] designed AI applications in energy management for Society 5.0, focusing on its role in optimizing energy efficiency and reducing carbon emissions. The study showed that AI-enabled smart grids and buildings resulted in an 18% reduction in energy consumption and 16% improvement in operational efficiency. For instance, predictive maintenance using AI reduced system downtime by 25% in renewable energy setups. However, the environmental cost of AI infrastructure remains a concern, with energy consumption for training AI models contributing to 3% of global emissions. The paper advocates for ethical AI integration in energy management.

Yao et al. [37] proposed a deep RL-based approach for microgrid control using a US microgrid dataset. The method applied a double dueling deep Q network (D3QN) to optimize energy storage systems and reduce operational costs. Results showed that the prediction-based scheme outperformed the prediction-free scheme with lower in running expenses and carbon emissions. The approach achieved a significant decrease in battery degradation and peak load, while limiting prediction errors to under 12.5%. However, the model's reliance on real-time data limits its scalability across varied conditions.

Ioannou et al. [38] developed autonomous RL techniques for microgrid management based on simulation data from islanded microgrid models. They compared four RL-based control strategies: DQN, PPO, Q-learning, and A2C. The DQN model demonstrated superior performance across all seasons, achieving minimal unmet load and excess generation with computational

eficiency. The results showed the highest energy savings and operational eficiency. However, PPO and A2C exhibited higher computational costs without significant performance improvements.

Maddineni et al. [39] designed a convolutional-GRU hybrid DL model for micro-grid performance prediction, by using Micro-grid Tariff Assessment Tool dataset. The model incorporated an attention mechanism to upgrade the precision of energy forecasts. The model demonstrated an MAE of 0.39, RMSE of 0.28, and r2-score of 98.89%, outperforming traditional ML models. The attention mechanism ensured superior forecasting accuracy, though the model's reliance on historical data limits real-time adaptability.

Silva-Rodriguez et al. [40] presented an LSTM-based DL model for net load forecasting in microgrids powered by wind and solar energy. Using a residential microgrid test case, the model demonstrated the highest forecasting accuracy, with a significant reduction in forecasting errors. The LSTM model outperformed other DL models, such as temporal CNNs and SVR in predicting net load. However, the model's performance can be affected by data sparsity during extreme weather events.

Li et al. [41] presented a multi-period forecasting model based on PER-Autor for microgrid optimization scheduling, focusing on renewable energy forecasting and load balancing. The model was tested using mixed-integer linear programming (MILP) and CPLEX solver. The results showed significant reductions in operational costs, achieving a 26% decrease in net present cost (NPC) and a 54% reduction in operating costs. The method's limitation is its dependency on accurate forecasting models, which may not always be feasible in real-world conditions.
Harrold et al. [42] applied multi-agent deep reinforcement learning (MADDPG) to optimize energy trading and renewable energy integration in microgrids. Results showed that the multi-agent approach outperformed single-agent systems, with energy cost savings of up to 15%. The approach allowed agents to develop unique strategies for each energy storage component, significantly improving system efficiency. However, challenges in agent communication and reward allocation remain, limiting real-time decision-making capabilities.

Hasan et al. [43] designed a grid-tied hybrid microgrid system with PV and lithium-ion battery storage, simulating it using HOMER Pro for a residential complex. The system resulted in a 54% decrease in operating costs, a 26% reduction in NPC, and a 31% drop in carbon emissions. The system was cost-effective, with a significant reduction in dependency on fossil fuels. However, the system's scalability to larger areas remains a challenge due to its reliance on specific residential load profiles.

Silva et al. [44] developed an IoT-based EMS for unbalanced AC microgrids, using data from a microgrid in Brazil. The system utilized a stochastic economic dispatch optimizer (EDO) based on MILP for day-ahead dispatch of distributed energy resources (DERs), while adhering to grid and security constraints. The results included costs of 20% less operational expenditure, grid stability improved by 12%, and a 18% in reduction in peak demand. However, there are always generating uncertainties of renewable energy, and the source fluctuation of local demand, the difficulty of adjusting the distribution of energy for different energy demand scenarios.

Isanbaev et al. [45] studied the application of smart meters with IoT capability to cross the energy consumption and power quality in a wind-solar hybrid microgrid. The study showed that the IoT-enabled smart meters enabled obtaining real-time data regarding power quality variables (active power, reactive power, and harmonic distortion). The results showed a decrease of 5% total harmonic distortion (THD), a direct power factor improvement of 3%, and a global energy consumption reduction of 7%. While successful in its use in monitoring small-scale microgrids, the system's scalability to larger grids may need to be adjusted to manage the increase in data flow.

Xu et al. [46] proposed an ML-based edge computing solution for solar inverter power prediction and droop control in a remote microgrid. The processing works were deployed on a low-cost ARM-based smart meter. The results showed that with the help of the edge computing style, the total communication time was reduced by 35% in comparison to the cloud-based systems. Besides, the forecasting model resulted in a mean absolute error (MAE) of 4.2% and 3.1% for active and reactive power, respectively. The droop control strategy resulted in a reduction of the voltage fluctuations of 5%. Despite these improvements, the small processing power in the edge device may affect the scalability for larger-scale microgrids. Bhardwaj et al. [47] developed an AI veterans (AI)-driven EMS for Nakano renewable energy integration and load balancing in microgrids. The system relied on ML algorithms to be able to predict energy demand and dynamically adjust power distribution.

Simulation results indicated a 25% improvement in energy efficiency, a 15% reduction of operational cost, and a 10% improvement in grid stability compared to traditional EMS approaches. However, some challenges, including data quality, responsive real-time, and integration with existing infrastructure, were noted to be a limitation for large-scale deployment. Hassan et al. [48] discussed the integration of AI in EMSs of large-scale solar farms, which focused on forecasting power generation and load balancing and to monitor the real-time performance; The study revealed that 18% fewer forecasting errors were made in the generation when

forecasted with the help of an AI (walk forward ppel opacity). 25% more efficiency was achieved efficient power distribution, and 30% higher stability of the grid. Additionally, real-time on performance monitoring helped to reduce the maintenance costs 20%. While enormous potential was experienced in the management of solar farms using the system, obstacles pertaining to data quality, the need for consistent data inputs and deployment costs were found.

Rasheed et al. [49] explored the integration of smart grids and Internet of Things (IoT) based solutions for energy efficiency and emission of carbon. The results indicated that from the combination of these technologies, there was a reduction of energy consumption by 25%, emissions of carbon dioxide by 35% and reliability of the grid by 40%. Additionally, renewable energy integration increased by 50%, and IoT-enabled consumer awareness rose by 60%, indicating a significant improvement in energy management practices. However, the high cost of deployment and integration remains a challenge for widespread adoption in developing regions.

Khanum et al. [50] proposed the application of AI for optimizing EMS in microgrids. The integration of AI tools enhanced forecasting, demand management, and operational cost reduction, resulting in a 15% decrease in energy costs, a 20% improvement in forecasting accuracy, and a 12% increase in microgrid resilience. Furthermore, AI-driven predictive maintenance led to a 10% reduction in equipment downtime. Despite these advancements, challenges related to data quality, system integration, and scalability were highlighted as significant obstacles for large-scale implementation.

Joha et al. [51] proposed an IoT-based system for energy monitoring and management in smart microgrids. Using real-time data, the system optimized energy distribution, reducing costs and improving reliability. The system demonstrated enhanced operational efficiency through smart load forecasting and demand response strategies. While the system was effective in real-time applications, its scalability in larger grid systems remains a challenge. AI, IoT, and edge computing help improve microgrid performance, with better forecasting and energy distribution and improved grid stability. While these technologies are enabling some significant benefits, there are challenges to overcome, such as poor data quality, challenges around integrating these systems, and high costs. Future research should consider addressing these issues in order to increase the capabilities of AI-driven systems in more resilient and sustainable energy management in microgrids. In Table 1, we can see a summary of the key findings from the reviewed studies on AI and EMS in microgrid applications.

Table 1 : Summary of Reviewed Literature on AI/ML in Energy Management

Year	Reference	Model	Results	Limitations
2024	[22]	SVR with RBF kernel	MSE: 2.002 (solar), 3.059 (wind); MAE: 0.547-0.825	Peak hour discrepancies: continuous validation required
2025	[23]	ORA-DL (DNN + RL)	93.38% prediction accuracy; 22.96% cost reduction	No computational complexity analysis; cybersecurity gaps
2024	[24]	MILP + Shapley allocation	3-5M USD savings; 45-60% CO2 reduction	Heavy forecasting dependency; dynamic conditions unaddressed
2025	[25]	GA + ABC + ACO	0.037 USD/kWh; 67% cost reduction	Geographic limitation (Fez only); no environmental assessment
2024	[26]	LSTM + RF + SVM + ARIMA	15% grid efficiency; 8.5% error reduction	Large dataset dependency; integration challenges

2024	[27]	CNN + LSTM hybrid	12% emission reduction; 20% efficiency gain	No regional generalization capability
2023	[28]	MLP (50 case studies)	25% waste reduction; 30% efficiency increase	High-quality data dependency
2025	[29]	GRSO + demand response	15.4% cost reduction; 13.23% emission decrease	Data quality issues; real-time adaptability
2025	[30]	QPSO optimization	9.67% cost reduction; 95.5% F1 score	Limited scalability across grid sizes
2025	[31]	LSTM + CNN ensemble	18.7% resource reduction; MAE: 1.8	Large urban scalability challenges
2023	[32]	ANFIS algorithm	15% forecasting improvement; R^2: 0.92	Input data sensitivity; weather dependency
2024	[33]	LSTM + RF + GBR	12% RMSE improvement; 90% precision	Building-specific data dependency
2025	[34]	RF + LSTM mining	15% energy reduction; 20% efficiency	High implementation costs; generalization issues
2025	[35]	AI forecasting models	20% investment efficiency; 12-14% RMSE reduction	Data quality; high capital costs
2025	[36]	AI smart grids	18% consumption reduction; 25% downtime reduction	AI infrastructure energy cost; 3% global emissions
2025	[37]	D3QN control	Cost/emission reduction; <12.5% error	Real-time data dependency
2025	[38]	DQN vs PPO vs A2C	DQN superior performance	PPO/A2C higher costs without benefits
2024	[39]	Conv-GRU + attention	MAE: 0.39; R^2: 98.89%	Historical data dependency
2024	[40]	LSTM net load forecasting	Highest accuracy; error reduction	Extreme weather sensitivity

2024	[41]	PER-AutoRL + MILP	26% NPC reduction; 54% cost decrease	Forecasting dependency; feasibility concerns
2022	[42]	MADDPG trading	15% cost savings	Communication challenges; reward allocation
2023	[43]	HOMER Pro simulation	54% cost decrease; 31% emission drop	Residential profile dependency
2023	[44]	MILP stochastic EDO	20% cost reduction; 12% stability	Renewable uncertainties; demand fluctuations

3. Methodology

The methodology of this study combines a stringent data preprocessing pipeline with a new hybrid modeling framework for predicting decentralized energy in rural regions of the United States. First, we discuss the series of preprocessing methods that should be followed to address these steps: cleaning, standardization, feature engineering, feature aggregation, encoding, and stratified data splitting are just a few of the crucial steps that should be performed in order to ensure a dataset is consistent, interpretable, and ready for training an ML model. With the purpose of building on this foundation, we introduce the proposed hybridization model that merges XGBoost and NN in an adaptive ensemble approach based on the ability of both frameworks. To validate the proposed approach, we assess the hybridization approach against five archetypal state-of-the-art models, emphasizing comparative performance and robustness, and generalizability of addressing cost, sustainability, and rural-urban energy disparities.

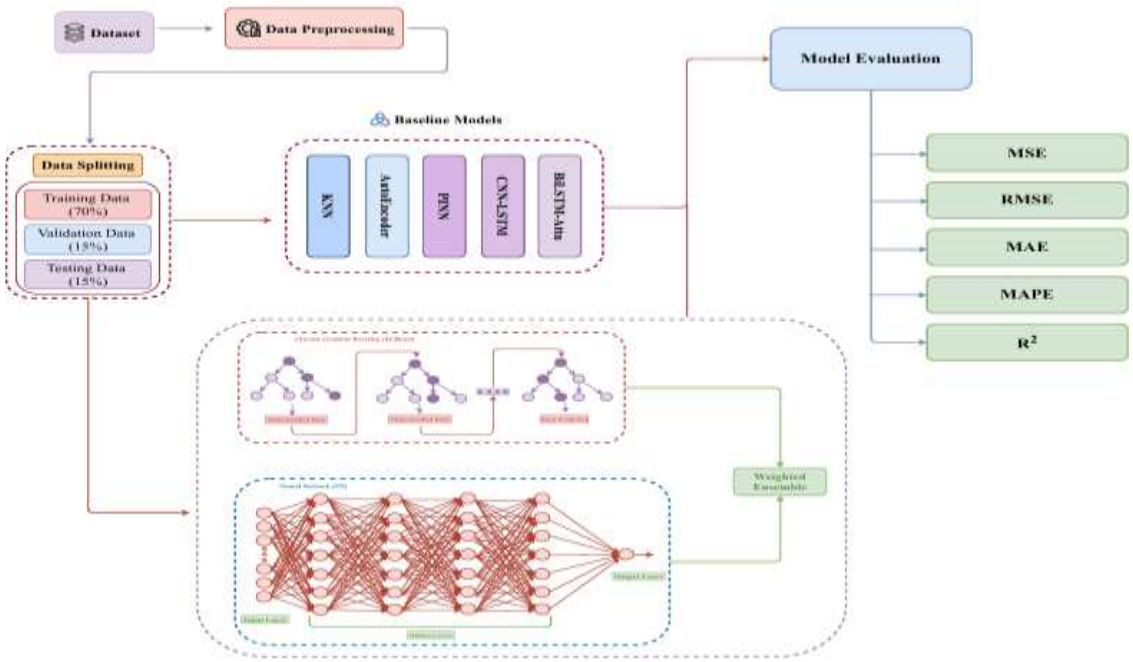


Figure 1: Workflow of the proposed Hybrid framework

3.1 Dataset Description

The used dataset for this research is US Electric Power Plants [52] from Kaggle that contains 13,233 records of electric power plants in the United States. This data set presents a comprehensive overview of energy generation systems, which highlights the need to look at the viability of decentralized energy solutions, especially in the rural areas of the US.

Several of these attributes are key attributes in the record of each of the plants considered in the data set and represent the various aspects of power plants. Plant Name and City determine the Identifiers, the specific power plant and its location in the US, respectively, while the State attribute makes known the plant's geometrical State. The Capacity (MW) attribute consists of the maximum electrical output of the plant, rated in MW and gives an indication of the scale of energy production. The Fuel Type attribute gives insights on the energy source (coal, natural gas, nuclear, hydroelectric, renewable wind, solar). The Latitude and Longitude attributes support geospatial analysis by identifying the exact location of the individual power plant. The Year of Commissioning is an indicator of when the plant was constructed and commissioned and provides information of the age of the plant and its applicability in today's energy market. Operational Status or active date refers to whether the plant is operational or not when the data was collected. The Latitude and Longitude properties aid more accurate performance by knowing where to find the individual power plant. The Year of Commissioning is an indicator of when the plant was constructed and commissioned and provides information of the age of the plant and its applicability in today's energy market. Operational Status or active date is whether such plant is operational or not when the data was collected.

Further plant-specific information about the turbine is found in Turbine Type such as steam or gas turbines, Turbine Cycle, such as Rankine, Brayton or Turbine and Steam Source which indicates if steam is produced from either boiler or geothermal source. The data set further encompasses Heat Recovery System with a picture depicting whether or not the plant is energy recycling, and Heat Recovery System Type and Description which describe specific microprocessor details of plant energy recycling.

The dataset features reported in Table 2 represent some of the key technical, operational, and locational features of U.S. power plants. Identification fields, including plant name, city, and state, provide a basic framework of reference, along with latitude and longitude, capturing geographic analysis to serve mapping and regional planning functions. Capacity (in megawatts): Is used to indicate the scale of production at which is an important parameter to assess the decentralization potential. Fuel type brings out the dependence on fossil fuels or renewables in a direct link to the point of sustainability. Operational attributes such as year of commissioning, current status, and its owner characterize the plant's life-cycle and governance. Technical details such as plant type, heating/cooling systems, turbine details, and steam source reveal design configurations that have bearing on efficiency and environmental impact. Heat recovery attributes outline to which plants make use of systems to capture and reuse energy. Together, these twenty features form a multi-dimensional portrait of the energy landscape in the United States, and therefore, the data can be a great tool to analyze how to develop cost-effective and sustainable decentralized energy strategies.

Table 2: Summary of Dataset Features

Class	Feature
Identification	Plant Name: Name of the power plant.
Location	City: City of plant. State: U.S. state of plant. Latitude / Longitude: Geographical coordinates.
Capacity	Capacity (MW): Maximum output in megawatts.
Fuel Characteristics	Fuel Type: Main energy source (coal, gas, solar, etc.).
Operational	Year of Commissioning: Year plant started operation. Operational Status: Active, retired, or under construction. Owner: Entity that operates the plant.
Plant Type	Plant Type: Configuration (e.g., combined cycle).
Cooling System	Cooling System Type: Cooling method used. Cooling System Closed Loop: Indicates closed-loop system.
Turbine	Turbine Type: Type of turbine (steam, gas). Turbine Cycle: Thermodynamic cycle (Rankine, Brayton).
Steam Source	Steam Source: Origin of steam (boiler, geothermal).
Heat Recovery	Heat Recovery System: Presence of energy recovery. Heat Recovery System Type: Type of recovery unit. Heat Recovery System Description: Notes on recovery setup.

3.2 Data Preprocessing

The preprocessing stage had been thought out to transform the raw data set into a consistent and reliable one that matches the model. The initial steps taken were to clean the data and to address the missing data in order to remove the inconsistency and guarantee data integrity. Standardization of schema and of categorical codes was performed to ensure that interpretation and reproducibility would be possible from one analysis to the next. Subsequently, major features were designed which reflect operational efficiency, environmental impact, and economic competitiveness, and aggregate at both plant and state levels. Finally, categorical encoding, numerical scaling, and stratified data splitting were implemented to provide balanced and standardized inputs to ensure the robustness and fairness of downstream modelling tasks.

Cleaning and Missing Value Handling The raw dataset contained several types of incompleteness, including non-filled numbers in operating capacity and generation columns, as well as the use of strings (e.g., "N/A", "NULL", "NOT AVAILABLE") when data were missing in categorical variables. In order to provide a consistent skeleton, all such placeholders have been replaced by a uniform NaN symbol. Formally, for each data entry x_{ij} in column j:

$$x_{ij}^{\text{cleaned}} = \begin{cases} \text{NaN}, & \text{if } x_{ij} \in \{\text{"N/A"}, \text{"NOT AVAILABLE"}, \emptyset\} \\ x_{ij}, & \text{otherwise} \end{cases} \quad (1)$$

For the numerical variables (with missing values), imputation has been implemented with mean or domain-specific constants, such as imputation for missing capacity factors based on the median of plants with the same fuel type. This approach minimised the level of distortions and ensured that the dataset generated by it was sufficiently dense for strong statistical and ML analyses.

Data Standardization In order to attempt to obtain a schema uniformity, the variable names have been standardized, but also the categorical codes have also standardized. All the column names were transformed to upper case and with an underscore character to avoid typographical misunderstanding. For the fuel codes, descriptive mappings were implemented. Let f denote the original fuel code, and M the mapping function from shorthand codes to descriptive categories:

$$\text{PRIM\_FUEL\_DESC} = M(f), \quad \forall f \in F \quad (2)$$

For Example:

$$M(\text{NG}) = \text{"NATURAL GAS"}, \quad M(\text{"SUN"}) = \text{"SOLAR"}, \; M(\text{"WAT"}) = \text{"HYDRO"} \quad (3)$$

By enforcing a deterministic mapping, the categorical variable space became inter- pretable and reproducible, and mitigated the risk of inconsistencies in downstream joins with emission and cost reference tables.

Feature Engineering Several derived features were built in order to augment the dataset. To calculate this capacity factor, the efficiency of plant utilisation was taken:

$$CF = \frac{\text{NET\_GEN}}{\text{OPER\_CAP} \times 8760} \quad (4)$$

where NET_GEN is the annual generation and OPER_CAP is installed capacity (MW). The CO2 intensity was estimated by mapping each primary fuel type to standard emission factors (gCO2/kWh):

$$\text{CO2\_INTENSITY} = \text{EF}(\text{PRIME\_FUEL\_DESC}) \quad (5)$$

with higher values for coal and gas, and near-zero for renewable fuels. In addition, simplified LCOE proxies (USD/MWh) were assigned by fuel type to approximate economic competitiveness. Binary flags were additionally introduced to designate renewable plants (such as hydro, wind, solar) and identify rural plants (as defined by small-scale capacity (≤ 50 MW) or location in mostly rural states). These engineered variables served as context for operational, environmental, and policy-relevant modeling, which followed.

Figure 2 shows the distribution of power plants with regard to the rural-urban position in the processed dataset. Out of the total number of plants (13233), a major portion (approximately 84.7%) is located in Example: rural areas, remaining 15.3% is under the category Example: urban. This skew illuminates the rural hegemony of energy infrastructure, which is driven in part by access to land and regulatory flexibility - as well as resources in non-urban areas. The imbalance also gives reason to kick off the RURALINDICATOR feature on the preprocessing pipeline. Coding the rural/urban contexts explicitly, the model can also consider heterogeneity in findings attributed to location, as well as capture on implications of social and environmental context in predictive tasks. Such classification is important for policy-oriented analysis because rural-dominated energy distribution creates grid integration, accessibility, and sustainability planning challenges.

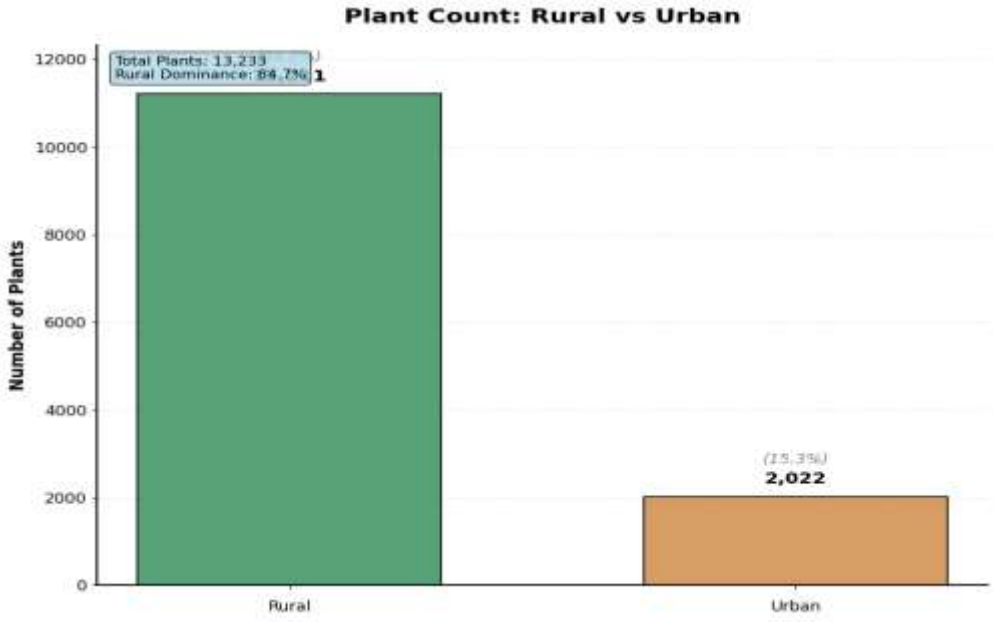


Figure 2: Comparison of plant distribution between rural and urban areas

Figure 3 presents a relationship between cost and sustainability between plant categories using the derived features of LCOE Proxy and CO2 Intensity. The left panel is a plot of LCOE ($/MWh) vs. emission intensity (kgCO2/MWh), where we can see positive correlations for rural as well as urban plants and larger slopes for rural. The right panel presents Pearson correlation coefficients and reveals strong positive correlations between the cost and emissions: the total (0.594), rural (0.627), urban (0.470), renewable (0.623), and non-renewable investments (0.587). These results imply that plants with greater emissions have greater generation costs, with this pattern more likely for rural and renewable plants. The integration of such engineered features in the preprocessing pipeline is instrumental to capturing cost-sustainability trade-offs. This figure, therefore, justifies the usefulness of developed indicators to link economic competitiveness with environmental performance.

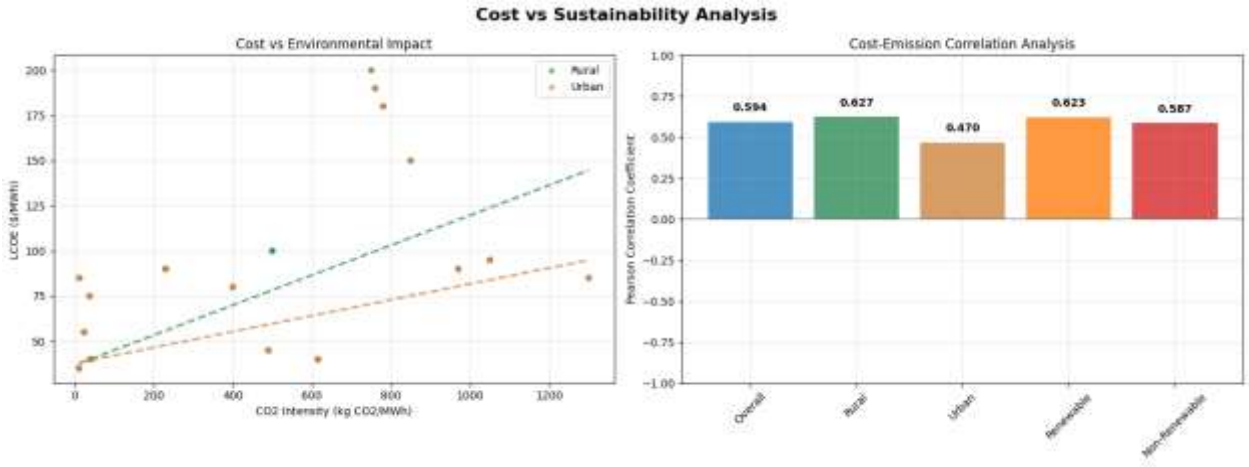


Figure 3: Cost vs Sustainability Analysis

Aggregation Two degrees of granularity were retained. At the plant level, all the attributes and engineered features were retained for predictive modeling. At the state level, we aggregated by state s to obtain:

$$\text{Total\_Capacity}_s = \sum_{i \epsilon s} \text{OPER\_CAP}_i, \quad \text{Mean\_CF}_s = \frac{1}{N} \sum_{i \epsilon s} \text{CF}_i \tag{6}$$

where N_s is the number of plants in the state s. This dual representation enabled both fine-grained ML tasks and macro-level descriptive policy analysis.

Model-Ready Encoding & Scaling Finally, the dataset was transformed into a form amenable to hybrid learning models. Categorical variables (e.g., STATE, PRIM_FUEL_DESC) were one-hot encoded:

$$\text{Encoded}(X) = \text{OneHot}(X_{\text{categorical}}) \quad (7)$$

Numerical variables (e.g., capacity, CF, CO2 intensity, LCOE proxy) were standardized:

$$X_j^{\text{scaled}} = \frac{X_j - \mu_j}{\sigma_j} \quad (8)$$

where μ_j and σ_j are the mean and standard deviation of the feature j. Stratified splitting was employed to partition data into training, validation, and test subsets, ensuring that class distributions of critical labels (e.g., RURAL INDICATOR) were preserved.

Data Splitting The final processed data set (13,233) was divided into 3 (mutually exclusive) subsets-data for training, for validation, and for testing. A stratified sampling method was used to preserve the distribution of the target variable (LCOE PROXY) to all partitions. This ensured that representative proportions of renewables and rural plants were left in each subset and did not cause bias when evaluating the model. Table 3 presents the details of the splitting strategy.

Table 3: Dataset splitting information.

Subset	Proportion	Sample Count
Training Set	70%	9,263
Validation Set	15%	1,985
Test Set	15%	1,985
Total	100%	13,233

In sum, the data preprocessing pipeline created a consistent and reliable base for further modeling. Through methodological cleaning, standardization, and feature engineering, heterogeneous datasets were converted to an interpretable and structured format. Further steps of aggregation, encoding, and scaling increased model readiness, while the stratified splitting-maintained representativeness throughout subsets. Collectively, these steps ensured robustness, better interpretability of work, and the ability to produce actionable insights about cost optimization, sustainability the hybrid learning framework.

3.3. Proposed Hybrid Model

In the following section, we give a detailed exposition of the hybrid model upon which global active power consumption has been predicted. This model comes to combines two of the mightiest methods in ML, which are the XGBoost (Extreme Gradient Boosting) and Neural Networks (NN). This is to exploit the strengths and weaknesses of the two models and to gain greater accuracy in prediction within the time-series forecasting problems when it is important to effectively obtain a linear pattern as well as a non-linear pattern.

The hybrid model in Figure 4 has been selected because the XGBoost and NN have comparably different strengths. A combination of these models guarantees that the final model enjoys the advantages of their strengths in modeling complex patterns in the data in addition to overcome their respective weaknesses.

Table 4 : Hyperparameter configuration used in the hybrid XGBoost- Neural Network model

Hyperparameter	Value
Batch size	256 (NN)
Loss function	MSE
Optimizer	Adam (NN), XGBoost
Learning rate	1×10^{-3} (NN), 0.07 (XGBoost)
Learning rate scheduler	Not used (NN); inherent shrinkage in XGBoost
Epochs	30 (NN), 150 boosting rounds (XGBoost)
Mixed precision	Disabled
Dropout rate	0.3, 0.2, 0.1 (per hidden layer in NN)
Max depth (XGBoost)	6
Subsample (XGBoost)	0.9
Regularization	L2 (NN weights), tree regularization $\Omega(\theta)$ in XGBoost
Random seed	42 (fixed across NumPy/TensorFlow/Sklearn)

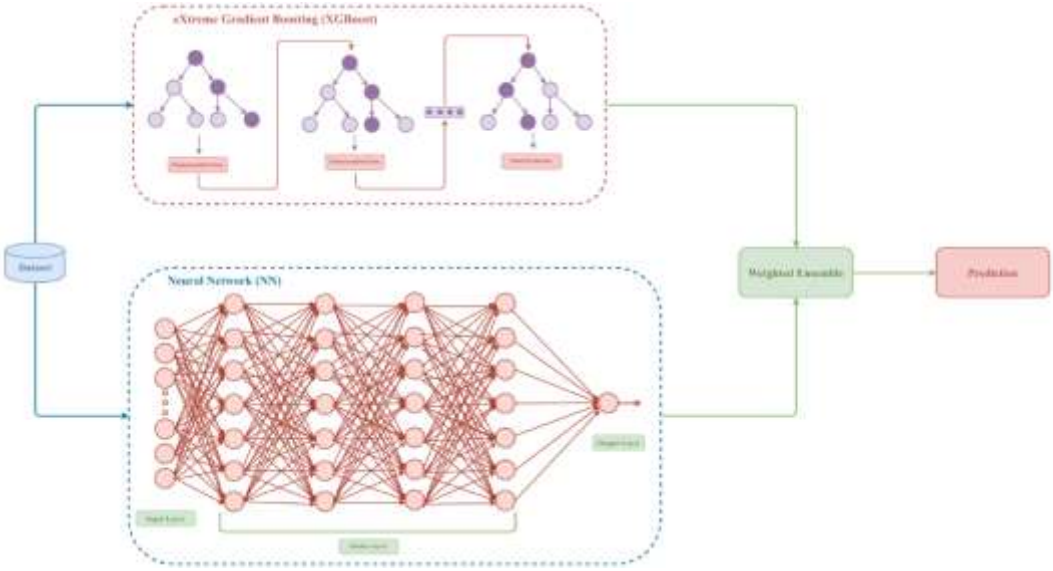


Figure 4 : Proposed Hybrid XGBoost-Neural Network Model

XGBoost : A Gradient Boosting Approach

XGBoost is a gradient boosting implementation and as associated with building an ensem-ble of decision trees with each subsequent tree trying to rectify the errors of the prior trees. The overall purpose functionality that gradient boosting aims to achieve may be represented as:

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

Where, $L(\theta)$ is the objective function to minimize, $L(y_i, \hat{y}_i)$ is the loss function, which measures the discrepancy between the true target value yi and the predicted value $\hat{y}_i$, and $\Omega(\theta)$ is the regularization term, which helps prevent overfitting by penalizing overly complex models.

The advantage with XGBoost is that it is able to integrate weak learners (decision trees) into a powerful ensemble without overfitting based on regularization. Regularization term is important because it assists in harmonising the complexity of the model against the accuracy of predictions. XGBoost, especially, has an advantage with structured data since it can also process missing values and nominal features and, thanks to its ability to use parallel processing, allows doing so effectively.

Neural Networks : Learning Non-linear Patterns

NN are a kind of ML model that aims to learn complex non-linear relationships in data by using many layers of neurons. Every neuron of the neural network is a weighted sum of its inputs and has an activation function (e. g., ReLU) to generate its output. The simplest specification of a feed-forward neural network will be:

$$y = f(W.x + b) \quad (10)$$

Where, y is the output of the neuron, x is the input vector, W represents the weights of the connections between layers, b is the bias term, and f is the activation function.

Considering regression, the neural network generally provides a continuous value that is appropriate in predicting the global values of active power. NN excel at the modeling of very complex, non-linear relationships in data - this is critically needed in tasks involving time-series prediction in which patterns change over time.

The Hybrid Model (XGBoost and Neural Networks)

XGBoost and Neural Networks (NN) are combined into a hybrid model to take advantage of respective strengths of each algorithm. Each model will be trained separately on the same dataset and the results would be combined to make the final prediction. The hybrid algorithm is created to effectively predict accuracy through the capabilities of XG Boost to model operating structures, linear and interaction-based relationship and the capabilities of NNs to model a complex, non linear relationship.

The hybrid model provides an opportunity to take the merits of both an ensemble learning and a DL that perform well in various aspects of prediction. Such a combination of NNs and gradient boosting leads to a model that is more ability to scale up to unseen data since it enjoys the simplicity of tree based decision rules and the learning depth of NNs.

The hybrid model has training process performed in two different stages. At first we train the XGBoost, and then the model of Neural Network is trained. Whereas the two models share the same dataset, they have a chance to learn various elements of the data.

- **XGBoost Training**

XGBoost is a gradient boosting algorithm and it builds up decision trees in sequence where the new tree seeks to rectify the mistakes of the already trained trees. The general principle here is to optimize an objective function that sums the prediction loss and a regularization term which will regulate the complexity of the models. Suppose that the dataset is $D = \{(x_i, y_i)\}_{\{i=1\}}^{N}$, and x_i is comprised of the input features and y is the target variable. XGBoost can be represented by an objective function that looks like :

$$L(\theta) = \sum_{N}^{i=1} L(y_i, \hat{y_1}) + \Omega(\theta) \quad (11)$$

Where, $L(\theta)$ is the objective function to minimize, $L(y_i, \hat{y}_1)$ is the loss, function}, which measures the discrepancy between the true target y_i and the predicted value $\hat{y}_1$, and $\Omega(\theta)$ is the regularization term, which helps prevent overfitting.

This operates in XGBoost equal sums of the output of all previously trained trees are summed to generate the prediction $\hat{y}_1^{(t)}$ at iteration t :

$$\hat{y}_1^{(t)} = \hat{y}_1^{(t-1)} + f_t(x_i),\ f_t \in F \quad (12)$$

Where f_t denotes the new decision tree within the current iteration t , and F denote the search space of all potential regression trees. The objective at an iteration is to determine a tree f_t which has a minimum of the following objective :

$$L^{(t)} \approx \sum_{i-1}^{N} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(x_i) \right] + \Omega(f_t) \quad (13)$$

Where $g_i = \partial_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_1^{(t-1)})$ represents the gradient, of the loss function of the predictions made in the previous attitude which refers to the gradient in the first order and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_1^{(t-1)})$ represents the gradient in the second-order. XGBoost uses both first and second-order derivatives to do a second order Taylor approximation of the loss incurred during the optimization and this helps improve the effectiveness of optimization.

The regularization of a tree f_t denoted $\Omega(f_t)$ is defined to be:

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \quad (14)$$

And is where (T) is the number of $j-$th leaf in the tree w_j the weight of the γ is a regularization parameter that the complexity of the tree should have, and λ is a measure that discourages large leaf weights to avoid overfitting. This normativity promotes sparse trees, trading off model complexity against accuracy of prediction.

This training is implemented on an iterative basis by successively adding trees to minimize the above objective until a stopping condition such as reaching a limit of iterations or convergence in the validation error has been attained. The above combination of many weak learners in such a way granted XGBoost a strong predictive recommendation, but also made it risk-resistant to overfitting and some relationship in the data be highly linear or non-linear.

- **Neural Network Training**

NNs are developed to provide complex, non-linear associations in data by manipulating inputs across several multiplexed neurons by means of several layers. The input features are denoted by x_i and the target value is denoted by y_i where $i = 1, 2, \ldots, N$. The input is then an output taken by each neuron in a layer where a calculation resulting in a weighted sum of all the inputs is done

and then the output is passed through an activation function. In the case of a feed-forward neural-network with L layers the result of the l −th layer is defined as :

$$h^{(l)} = f(W^{(l)} h^{(l-1)} + b^{(l)}) \quad (15)$$

Where $h^{(l-1)}$ is the result of the $h^{(0)} = x_i$, $W^{(l)}$ calibration of the prior layer (where $h^0 = x_i$), W^l and b^l are weights and biases of the l −th and $f(\cdot)$ is non-linear activation (either ReLU ($\mathrm{ReLU}(x) = \max(0, x)$ or Sigmoid). The resultant output layer gives out the predicted value $\hat{y}_i$, in the case of regression tasks, normally a linear activation :

$$\hat{y}_i = W^L h^{L-1} + b^L \quad (16)$$

The network parameters $\theta = \{W^{(l)}, b^{(l)}\}_{l=1}^{L}$ are determined by minimizing a loss function that measures deviation between predicted and true values. In the case of regression, the usual application is the MSE :

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \quad (17)$$

Where $L(\theta)$ is the loss function to minimize, y_i is the true target value, and $\hat{y}_i$ is the predicted value. The training involves the utilization of the backpropagation algorithm that calculates loss function gradients with respect to every network parameter using the chain rule. To compute such gradients, the weights and biases are updated using the same gradients in an iterative fashion, usually with optimization algorithm like Adam or stochastic gradient descent (SGD):

$$\theta \leftarrow \theta - \eta \frac{\partial L(\theta)}{\partial \theta} \quad (18)$$

Where θ is a learning rate adjusting the parameter update step size. This is repeated until a termination condition is met, say maximum number of epochs, or validation loss convergence.

NNs excel especially in modelling highly non-linear and complex feature interactions and are well suited in time-series prediction tasks such as global active power prediction. Layering the network allows it to gain access to hierarchical data collections of the data, including both short and long-term dependencies that it may be more challenging to identify using the traditional tree-based frameworks. Overfitting can in turn be prevented with regularization methods like dropout or L_2 weight decay so that the network makes good predictions on previously unseen data.

- **Ensemble Strategy**

The ensemble strategy is essential in the proposed hybrid model exploitation of the strength of the other two XGBoost and NNs. Whereas XGBoost is best at relationship capture in structured and linear or weakly non-linear form, NNs are eminent at relationship capture of the non-linear form. Through the synthesis of the two model predictions, the hybrid technique will minimize bias and variance, and thus compromise superior aggregate forecasting performance and extrapolation ability to invisible data.

Weighted averaging is the simplest, and most commonly used ensemble scheme whereby the resultant prediction $\hat{y}_i$ is a weighted average of the base model predictive outputs. This may then be mathematically empenned as :

$$\hat{y} = w_{XGB} \cdot \hat{y}_{XGB} + w_{NN} \cdot \hat{y}_{NN} \quad (19)$$

Where $\hat{y}_{XGB}$ and $\hat{y}_{NN}$ denote the predictions of the XGBoost and Neural Network models, respectively, and (w_{XGB}) and (w_{NN}) are their corresponding weights. The weights are constrained such that ($w_{XGB} + w_{NN} = 1$), ensuring that the ensemble prediction remains within the same scale as the individual model outputs.

It is important to establish optimum weights in the ensemble. Typically the validation-based weighting method is used, where each of the models is tested on a validation set and weights are apportioned in the inverse to the prediction error. A typical example is that assuming (E_{XGB}) and (E_{NN}) are the validation errors of XGBoost and Neural Network models (mean of squared error or mean absolut error), the weights are equal to :

$$w_{XGB} = \frac{\frac{1}{E_{XGB}}}{\frac{1}{E_{XGB}} + \frac{1}{E_{NN}}}, \quad w_{NN} = \frac{\frac{1}{E_{NN}}}{\frac{1}{E_{XGB}} + \frac{1}{E_{NN}}} \tag{20}$$

In this formulation it would be so that the model that has a lower error of every curve within the set of models is included in a greater contribution, and hence the model that has a larger error is within a smaller relevance which would balance the contribution of both models to the final prediction.

Along with weighted averaging, more elaborate ensemble approaches like stacking are possible. In stacking the base model predictions are taken into the input of a meta-learner that is trained to learn the combination of the best combination which tends to reduce the total error. Mathematically the stacking model can take the form of :

$$\hat{y} = f_{meta}(\hat{y}_{XGB}, \hat{y}_{NN}) \tag{21}$$

Where f_{meta} is a trainable meta-model, perhaps a linear regression, shallow neural network, or other ML algorithm. This removes intricate dependence within the primary model predictions that can enhance prediction further, since the ensemble approach permits the creation of sophisticated associations between the primary model predictions.

In general, the hybrid approach, whereby the ensemble technique is applied to XGBoost and NNs, secures that the advantages complementing the two apply successfully. The hybrid model will perform better by sparingly attaching the predictions via weighted averaging or stacking and lower the bias and variance of the predictions as well as enhance robustness in forecasting the activity such as the the consumption of active power at a global scale.

Hybrid Regression Model, a combination of XGBoost and a Neural Network can be summarized as proposed in Algorithm 1, that models the training, the ensemble weighting and the inference steps.

Algorithm 1 Weighted Ensemble Hybrid Model for Regression (XGBoost + Neural Network)

Require : Training set (X_{train}, y_{train}), validation set (X_{val}, y_{val}), test set X_{test}, y_{test}, batch size B, neural network epochs E, XGBoost hyperparameters $(n_{estimators}, \eta, max_{depth}, subsample$, ensemble candidate weights W

Ensure Trained XGBoost model $\hat{g}_{xgb}$, trained neural network $\hat{f}_{nn}$, ensemble strategy, optimal weights, regression metrics

Data Preprocessing:
Standardize features: $\tilde{X} = \frac{X - \mu}{\sigma}$ using training statistics
Optionally scale target variable y

Train XGBoost Regressor:
Initialize $\hat{g}_{xgb}$ with specified hyperparameters
Fit $\hat{g}_{xgb}$ on X_{train}, y_{train}
Compute predictions $y_{train}^{xgb}, y_{val}^{xgb}$
Compute validation $R_{val}^{2\ xgb}$
Train Neural Network Regressor:
Initialize feedforward network $\hat{f}_{nn}$ with layers: [128,64,32,1], activation ReLU, batch normalization, dropout
Compile with Adam optimizer and MSE loss
Fit $\hat{f}_{nn}$ on X_{train}, y_{train} for E epochs with batch size B and validation monitoring
Compute predictions y_{train}^{nn} , y_{val}^{nn}
Compute validation $R_{val}^{2\ nn}$
Select Best Strategy:
Initialize best_R2 = $max R_{val}^{2\ xgb}, R_{val}^{2\ nn}$
Initialize strategy = xgb_only if $R_{val}^{2\ xgb} \geq R_{val}^{2\ nn}$ else nn_only
for each weight $w \in W$
    Compute ensemble prediction: $y_{val}^{ens} = w \cdot y_{val}^{xgb} + (1 - w) \cdot y_{val}^{nn}$
    Compute $R_{val}^{2\ ens}$
    if $R_{val}^{2\ ens} >$ best_R2
        Update best_R2 ← $R_{val}^{2\ ens}$
        Update strategy ← ensemble

Save ensemble weights $[w, 1-w]$
end if
end for
Inference on Test Set :
if strategy = ensemble **then**
$y_{test}^{pred} = w \cdot \hat{g}_{xgb}(X_{test}) + (1-w) \cdot \hat{f}_{nn}(X_{test})$
else if strategy = xgb_only **then**
$y_{test}^{pred} = \hat{g}_{xgb}(X_{test})$
else
$y_{test}^{pred} = \hat{f}_{nn}(X_{test})$
end if
Compute test R_{test}^2 and other regression metrics (MSE, RMSE, MAE, MAPE)
return $\hat{g}_{xgb}$, $\hat{f}_{nn}$ strategy, weights, R_{test}^2

Experimental findings argue that the hybrid model successfully utilizes complementary assets of XGBoost and the neural network, achieving better regression performances than the two models. The weighted ensemble methodology offers enhancements to the validating performance by being adaptive in nature, giving high accuracy and strength in both training, validation, and test data.

3.4 Baseline Models

In this study, we explore a sequence of baseline models for sequential prediction problems that act as the foundation against which more complex models are compared. These baseline models incorporate all the traditional approaches, such as KNN and more complex models such as Autoencoder-Predictor, Physics-Informed Neural Networks (PINN), CNN-LSTM Fusion, and Bidirectional LSTM (BiLSTM) with Attention. Each model has its own distinct advantages, with KNN providing a simple approach, Autoencoder-Predictor providing a way to combine dimensional reduction and regression, and PINN fusing physical laws into the learning process. CNN-LSTM Fusion, which can capture both local and temporal features, and BiLSTM with Attention, which enhances sequence learning by paying attention to important time steps. By testing these models, we set a mark for performance and areas of improvement for the future and more advanced models.

3.4.1 K-Nearest Neighbors (KNN)

KNN is the simplest method and doesn't require any parameters to be used for regression. KNN makes classifications according to the nearest information point in its feature space. For this study, we would use the K-Nearest Neighbors Regression (KNN-R) to predict continuous outcomes, and the Euclidean Distance pulled between data points to determine the data points closest to each other. In the KNN algorithm, given a new observation x The model predicts the output $\hat{y}$ as the average of the outputs of the k nearest neighbors, denoted as $\{x_1, x_2, \ldots \ldots \ldots, x_k\}$. Mathematically, the KNN prediction can be expressed as:

$$\hat{y}(x) = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{22}$$

where y_i represents the true value of the i^{th} nearest neighbor, and k is the number of neighbors considered.

Model Implementation: For this study, we implemented the KNN regression model using scikit-learn's KNeighborsRegressor, setting k = 5 to select the five closest data points. The model is trained on the scaled training dataset, denoted as $X_{train\ scaled}$ and $y_{train\ scaled}$.

After training, the model is evaluated on the validation set X_{val_scaled} and test set X_{test_scaled}, where predictions are made using the $\mathrm{predict}()$ method. The results are then rescaled back to the original scale of the target variable using a scaler transformation.

Mathematical Formulation Let the dataset be represented as $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where each x_i is a feature vector, and y_i is the corresponding target value. The KNN model uses the following steps:

- **Distance Calculation:** For a new observation x_{new}, calculate the Euclidean distance to all other data points:
  - $$d(x_{new}, x_i) = \sqrt{\sum_{j=1}^{d}(x_{new,j} - x_{i,j})^2} \tag{23}$$
    - where d is the number of features.
- **Sorting Neighbors:** Identify the k nearest neighbors based on the smallest distance values.
- **Prediction:** Predict the output $\hat{y}$ by averaging the target values of the k-nearest neighbors:

$$\hat{y}(x_{new}) = \frac{1}{N}\sum_{i=1}^{k} y_i \tag{24}$$

3.4.2 Autoencoder-Predictor

An Autoencoder is a neural network that performs dimensionality reduction and feature learning, receiving training on efficient codings (representations) of the input data. An autoencoder has two key components:

- **Encoder**: Encodes the input information X into a latent space representation of the input data compressed into a latent space, z.
- **Decoder**: It is used to rebuild the input data X using the representation in the latent space, that is, z.

In regression problems, the Autoencoder-Predictor is trained to predict the target variable, y, by exploiting the acquired encoder latent space. The mathematical description of the model can be formulated in the following steps:

Encoder Function:

Encoder Function The encoder converts input data X into a latent space representation z, which is a lower-dimensional representation of the data:

$$z = f_{encoder}(X) \tag{25}$$

Decoder Function: The decoder reconstructs the input X from the latent representation z:

$$\hat{X} = f_{decoder}(z) \tag{26}$$

Prediction Layer: The latent space z is passed through a regression head to predict the target variable y:

$$\hat{y} = f_{predictor}(z) \tag{27}$$

The **loss function** for training the model consists of two parts :

- **Reconstruction Loss:** Measures how well the decoder reconstructs the original input X :
-

$$L_{reconstruction} = \frac{1}{n} \sum_{i=1}^{n} \left\| X_i - \hat{X}_i \right\|_2^2 \tag{28}$$

- **Prediction Loss:** Measures how well the latent space z predicts the target variable y:

$$L_{prediction} = \frac{1}{n} \sum_{i=1}^{n} \| y_i - \hat{y}_i \|_2^2 \tag{29}$$

The total loss is a weighted sum of the reconstruction and prediction losses:

$$L_{total} = \lambda_1 L_{reconstruction} + \lambda_2 L_{prediction} \tag{30}$$

where λ_1 and λ_2 control the importance of each term.

CNN-LSTM Fusion

The CNN-LSTM Fusion model is a model based on the fusion of a CNN model for feature extraction and the addition of a sequence learning effect through suitable LSTM networks. The CNN layers capture the multi-scale features of input data, while the LSTM captures the temporal dependencies of data. The final prediction is made with the help of a {dense layer} after the LSTM layers.

Mathematical Formulation

Given an input sequence $X_i \in R^{n \times d}$, where n is the sequence length and d is the number of features, the model consists of the following steps :

- **CNN Feature Extraction :** The CNN layers are applied to extract multi-scale features. For each convolutional filter, the operation is:

$$Conv1D(X_i) = ReLU(Conv(X_i, W_{conv1}) + b_{conv1}) \tag{34}$$

$$\mathrm{Conv2D}(\mathrm{X_i}) = \mathrm{ReLU}\,(\mathrm{Conv}\,(\mathrm{X_i}, \mathrm{W_{conv2}}) + \mathrm{b_{conv2}}) \quad (35)$$

$$\mathrm{Conv3D}(\mathrm{X_i}) = \mathrm{ReLU}(\mathrm{Conv}(\mathrm{X_i}, \mathrm{W_{conv3}}) + \mathrm{b_{conv3}}) \quad (36)$$

The outputs from these layers are concatenated :

$$\mathrm{Conv_{merged}} = \mathrm{Concatenate}(\mathrm{Conv1D}, \mathrm{Conv2D}, \mathrm{Conv3D}) \quad (37)$$

The merged convolutional features are then passed through a Dropout layer to prevent overfitting:

$$\mathrm{Conv_{merged}} = \mathrm{Dropout}(0.2)(\mathrm{Conv_{merged}})$$

- **LSTM Sequence Learning** : The concatenated CNN features are passed to the LSTM layers:

$$\mathrm{LSTM_1} = \mathrm{LSTM}(\mathrm{Conv_{merged}}, \mathrm{W_{lstm1}}, \mathrm{b_{lstm1}}, \mathrm{return\_sequences} = \mathrm{True}) \quad (38)$$

$$\mathrm{LSTM_2} = \mathrm{LSTM}(\mathrm{LSTM_1}, \mathrm{W_{lstm2}}, \mathrm{b_{lstm2}}) \quad (40)$$

- **Dense Layer Processing**: The output of the second LSTM layer is passed through a fully connected dense layer :

$$\mathrm{Dense} = \mathrm{ReLU}(\mathrm{Dense}(\mathrm{LSTM_2}, \mathrm{W_{dense}}, \mathrm{b_{dense}})) \quad (41)$$

Finally, the predicted value is computed using the output of the dense layer:

$$\hat{\mathrm{y}}_\mathrm{i} = \mathrm{Dense2}(\mathrm{Dense}, \mathrm{W_{out}}, \mathrm{b_{out}}) \quad (42)$$

Loss Function: The model is trained by minimizing the {MSE}:

$$\mathrm{L_{MSE}} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \|\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}\|_2^2 \quad (43)$$

[1] List the reference here

3.4.3 Physics-Informed Neural Network (PINN)

A PINN enables a network that uses known physical laws as a part of the loss function to enforce the given constraints on our predictions. Unlike data-driven models, PINNs include physics-based loss terms to ensure consistency with physical equations (e.g., partial differential equations or PDEs).

The total loss function is a combination of two components:

- Data-driven loss that penalizes prediction errors.
- Physics-informed loss that ensures predictions comply with physical laws.

Mathematical Formulation

Given a dataset $\{(\mathrm{x_1}, \mathrm{y_1}), (\mathrm{x_2}, \mathrm{y_2}), \ldots, (\mathrm{x_n}, \mathrm{y_n})\}$, the PINN model minimizes the following total loss function:

1. **Data-Driven Loss:**

$$\mathrm{L_{data}} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left|\left|\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}\right|\right|_2^2 \quad (31)$$

2. **Physics-Informed Loss:** The residual $\mathrm{R_{pde}}$ of the governing physical law (e.g., a PDE):

$$\mathrm{R_{pde}} = \frac{\partial \mathrm{u}}{\partial \mathrm{t}} + \mathrm{L(u)} = 0 \quad (32)$$

   The physics-based loss is:

$$\mathrm{L_{physics}} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left|\left|\mathrm{R_{pde}}(\mathrm{x_i}, \hat{\mathrm{y}}_\mathrm{i})\right|\right|_2^2 \quad (33)$$

3. **Total Loss:**

$$\mathrm{L_{total}} = \mathrm{L_{data}} + \lambda \mathrm{L_{physics}} \quad (34)$$

where λ is a hyperparameter controlling the balance between data-driven and physics-informed losses.

3.4.6 Bidirectional LSTM+Attention

The {BiLSTM with Attention} that addresses the advantages of the {BiLSTM and Attention mechanisms} itself in sequential data prediction. This entire input sequence is run through the BiLSTM sequentially forward and backward in order to gather temporal dependencies. The {Attention mechanism can be useful in making the model to focus on the most relevant section of the sequence, and thereby helps to improve the performance in doing tasks where some time-steps are more significant than others.

The {BiLSTM} identifies the temporal reliance and the {Attention mechanism} assigns importance to the output of the BiLSTM so as to magnify the model in order to focus on the significant segment of the input sequence. The result is then incorporated in a {Dense layer} to arrive at certain predictions about the target.

The BiLSTM reflects the temporal dynamics and the Attention mechanism assigns weights to the BiLSTM output so that the model can focus on the significant aspects of the input sequence. The result is then passed through a {Dense layer} to give predictions as to the target.

Mathematical Formulation

The model consists of several components with the following mathematical representations:

- **Bidirectional LSTM Layer :**

$$\mathrm{BiLSTM}(\mathrm{X_i}) = \mathrm{Bidirectional}(\ \mathrm{LSTM}(128, \mathrm{return\_sequences} = \mathrm{True})\ )(\mathrm{X_i}) \quad (44)$$

- **Attention Mechanism:**

$$\mathrm{Attention}(\mathrm{X_i}) = \mathrm{Softmax}(\mathrm{Flatten}(\mathrm{Dense}(\mathrm{X_i}))) \quad (45)$$

$$\mathrm{Attended}(\mathrm{X_i}) = \mathrm{Multiply}(\mathrm{BiLSTM}(\mathrm{X_i}), \mathrm{Attention}(\mathrm{X_i})) \quad (46)$$

- **Global Pooling and Dense Layer :**

$$\mathrm{Pooled}(\mathrm{X_i}) = \mathrm{GlobalAveragePooling1D}(\mathrm{Attended}\ (\mathrm{X_i})) \quad (47)$$

$$\mathrm{Dense}(\mathrm{X_i}) = \mathrm{Dense}(64, \mathrm{ReLU})(\mathrm{Pooled}(\mathrm{X_i})) \quad (48)$$

$$\hat{\mathrm{y}}_\mathrm{i} = \mathrm{Dense}(1)(\mathrm{Dense(X\_i)}) \quad (49)$$

Loss Function: The model is trained by minimizing the {MSE}:

$$\mathrm{L_{MSE}} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \|\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}\|_2^2 \quad (50)$$

The baseline models KNN, Autoencoder-Predictor, PINN 1, CNN-LSTM Fusion, and BiLSTM with Attention bring different advantages to sequential prediction problems. KNN provides a simple mechanism but Autoencoder-Predictor employs unsupervised & supervised learning. PINN adds physical laws for better predictions, CNN-LSTM Fusion will learn both local part and temporal dependency, BiLSTM with Attention pays more attention to the important part of the sequence. These models have a solid basis for comparison with the complex models.

4 Experimental Results and Discussion

4.1 Experimental Setup

Experiments were performed on Google Colab using Python in a T4-GPU environment for accelerating DL model training. The implementation utilized some core libraries, including Scikit-learn for preprocessing the data and baseline models, PyTorch for DL components, and PyTorch Geometric for processing structured data operations. This supported rapid model development in addition to reproducibility and access to GPU acceleration.

In order to compare the model with the standard counterparts, a series of baselines were used, including KNN, Autoencoder, PINN, CNN-LSTM, and attention-based BiLSTM. The baselines are standard practices applied in energy system forecasting and optimization, and they provide a robust benchmark for comparison with the hybrid approach. The proposed model integrates

XGBoost for learning feature interactions of structured data with a neural network subcomponent for learning nonlinear temporal relationships. Training employed chronological division of data into training, validation, and test sets to prevent data leakage and maintain temporal reality. Hyperparameters of all models were tuned with uniform search budgets to enable equitable comparison. Four commonly used measures were used to assess model performance and include (i) MSE,(ii) Root Mean Squared Error (RMSE), (iii) Coefficient of Determination (R) and (iv) Mean Absolute Percentage Error (MAPE). The combination of these metrics offers an overall assessment of predictive accuracy, reliability and explanatory power. All of the experiments were carried out under controlled environments with fixed random seeds for reproducibility purposes. The performance was averaged across several runs and expressed as confidence intervals to report statistical significance. This is a good basis on which to rely the subsequent results on accuracy, error analysis, economic performance, and sustainability

4.2 Comparative Evaluation Using Performance Metrics Baseline vs Proposed (XGBoost+NN) Model

The training, validation, and test accuracy of the baseline models (KNN, AE, PINN, CNN-LSTM, BiLSTM-Attn) are compared with those of the proposed (XGBoost+NN) model in Figure 5 and Table 5. The proposed (XGBoost+NN) model had the best performance with training, validation, and test accuracy of 99.99%, 99.97%, and 99.94%, respectively. The small gap between the training accuracy and the test accuracy suggests excellent generalization. PINN was the strongest baseline, with 99.8% test accuracy, but still not comparable to the proposed approach. KNN was decent as well (92% test accuracy), but with large gaps compared to its training accuracy, suggesting poor scalability. Autoencoder, CNN-LSTM, and BiLSTM-Attn performed very weakly on the test set (51.19%, 55.08%, and 34.88%, respectively), suggesting poor generalization and overfitting behaviors. These results confirm that even some DL baselines are capable of emulating renewable energy patterns, but hybridizing XGBoost for structured feature learning and NN for temporal dynamics leads to progressively better performance. The minimal loss of accuracy during training reduces the loss in testing further, emphasizing the stability of the introduced framework for rural energy forecasting tasks.

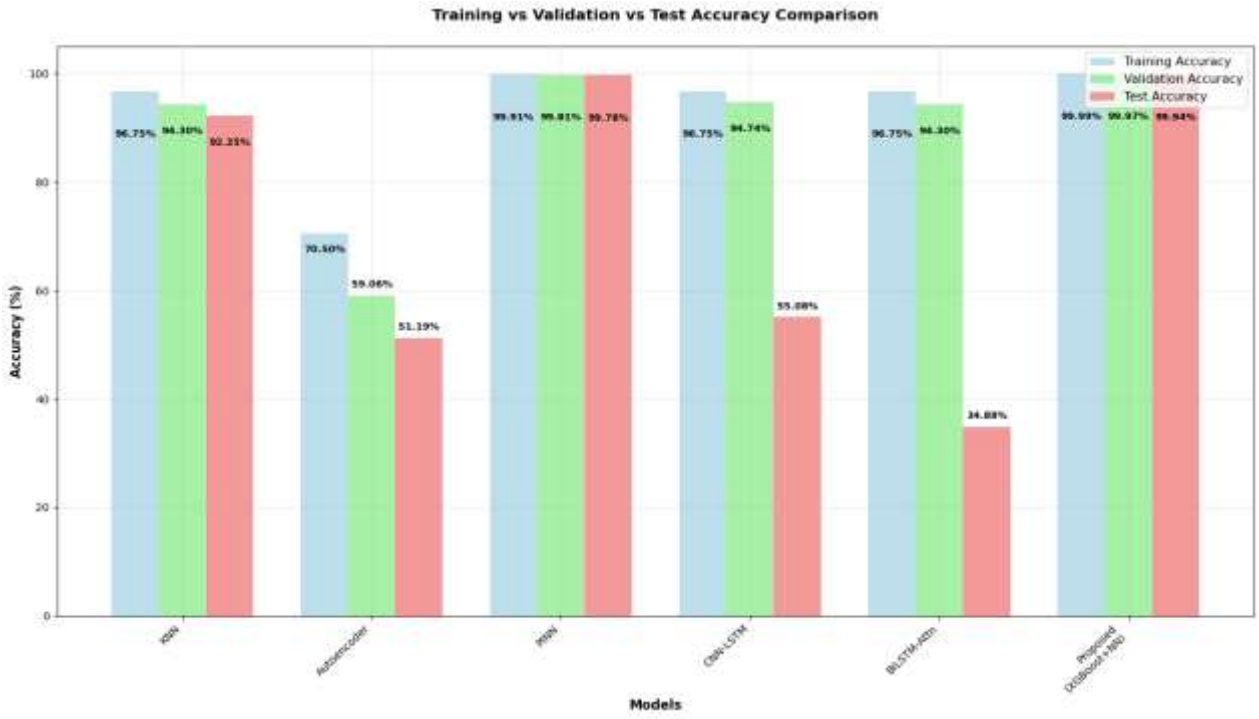


Figure 5: Training, validation, and test accuracy comparison between baseline models and the proposed (XGBoost+NN)

Table 5: Training, validation, and test accuracy comparison between baseline models and the proposed Model

Model	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
KNN	96.75	94.30	92.25
Autoencoder	70.50	59.06	51.19
PINN	99.91	99.81	99.78

CNN-LSTM	96.75	94.74	55.08
BiLSTM-Attn	96.75	94.30	34.88
Proposed (XGBoost+NN)	**99.99**	**99.97**	**99.94**

Figure 6 and Table 6 gives a detailed comparative analysis of the baseline models and the suggested XGBoost+NN model in three conventional performance measures, R^2 MSE, and RMSE. These measures act in unison to indicate the explanatory power and the minimization of squared deviations, as well as the size of the typical prediction errors that the model exhibits.

The findings show the evident excellence of the model suggested. Regarding explanatory power, the XGBoost+NN had a value of R^2 is 0.999 meaning that the model virtually accounts for all the volume of the actual data. This degree of accuracy is not very common in the energy forecasting activity of the real world. The PINN had the highest baseline with almost equal but a bit lower R^2 of 0.998, whereas the KNN model reached a more average 0.922. In comparison, the unhybridized DL baselines, i.e. CNN-LSTM, Autoencoder and BiLSTM-Attn, fared much worse, with R^2 of 0.551, 0.512 and 0.349, respectively. Such poor performance underlines the challenge of using only DL based models to learn the nonlinear and temporal characteristics of decentralized rural energy data.

The same trend is verified by the error-based metrics. The XGBoost+NN gave the lowest MSE of 0.0003, which represents very small average squared deviations of the real values. This error is about four times lower than the PINN baseline which obtained 0.0011. The classical algorithms like KNN produced significantly greater error rates where MSE is 0.0390 whereas the baselines did significantly worse, with 0.2 or higher. The strength of the proposed model is further reinforced by the RMSE, which could be more easily interpreted since it is reported in the same unit as the target variable. Once again, the RMSE of the hybrid method was far lower at 0.0178 compared to PINN at 0.0329. The other baselines yielded significantly larger RMSE values, with the highest was 0.1974 with KNN and the lowest was 0.5717 with BiLSTM-Attn.

These three evaluation metrics form a coherent pattern: the proposed XGBoost+NN model does not only achieve the best results in all the measures but by a significant margin. This shows that its hybrid architecture is the only one to be effective in capturing structured feature interactions as well as temporal dependencies on rural energy data. Practically, this precision directly applies to the notion of decentralized energy planning because the operators can predict generation and demand with some degree of assurance that risks are reduced and that planning is done optimally to have the resources distributed efficiently whilst minimizing risks.

Table 6: Performance comparison Comparison of baseline models and proposed Model

Model	R^2	MSE	RMSE
KNN	0.922	0.0390	0.1974
Autoencoder	0.512	0.2450	0.4949
PINN	0.998	0.0011	0.0329
CNN-LSTM	0.551	0.2254	0.4748
BiLSTM-Attn	0.349	0.3268	0.5717
Proposed (XGBoost+NN)	**0.999**	**0.0003**	**0.0178**

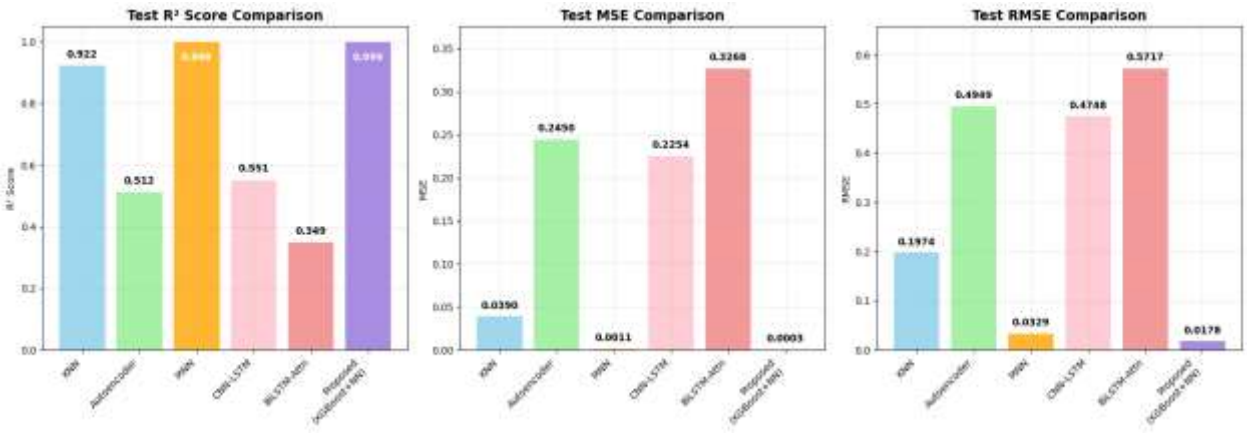


Figure 6: Comparison of baseline models and proposed (XGBoost+NN) for test R^2, MSE, and RMS

4.3 Cost Estimation and Economic Implication

Figure 7 illustrates the actual vs. predicted costs generated by the proposed (XGBoost+NN) model. The actual operating cost of the system was recorded at $564.92, compared to the model's very similar prediction of $564.44. The margin of a relative error of 0.08 which is equal to only 0.48 indicates that the model is very accurate in the fact that it captures the economic implications of system performance. This form of accuracy is imperative to rural distant energy grids whose cost effectiveness is core incentive of adoption and maintainability. A change of a few percent can cause a misrepresentation in tariff planning, investment allocation, or battery scheduling habit. The model proposed has sub-percentage error margin which ensures that the operations decision-making and policy-making are informed by confidence thereby reducing the financial risks to the minimum. Furthermore, the result proves that the model is not only technologically sound to predict, but also produces the outputs that can be directly applied in the optimization of the economy. One other notable observation is that other models of the baselines achieved moderate to high predictive accuracy of the demand and generation although their cost predictions were much less precise because of accumulated forecasting errors. Contrary to this, hybrid (XGBoost+NN) converted its forecasting potential to almost perfect cost forecasting and exhibited its ability to integrate technical accuracy with financial reasonability. This integration is most in demand to rural areas where cost-effectiveness and affordability of decentralized energy uptake are the long-term factors that render such uptake viable. The results of the estimation of costs confirm the appropriateness of the proposed approach. The model helps to fill the gap between the theoretical advancements in cost estimation and real financial planning of agricultural energy supply by close aligning actual operating costs with the model.

4.4 Prediction Fidelity: Predicted vs. Actual

In the proposed model (XGBoost+NN) the actual vs. predicted scatter plot on the test set is shown in the actual vs. predicted scatterplot on the test set in Figure 8. These blue dots represent single test samples and the red line is a line of perfect prediction. Close clustering of the points along the diagonal would suggest that the model predictions are extremely close to observed values over the whole range of data. Deviations of the line are very few and there is no systematic bias in the model and this ensures that the model generalizes effectively without overfitting.

This approximate equilibrium is in line with the large R^2 (0.999) and small R MSE (0.0178) values above. It indicates that not only is the hybrid model efficient from a statistical perspective but it also does not sacrifice meaningful fidelity at the case level. This is particularly to be desired in decentralized rural power systems, where prediction error on a single case would translate to operational inefficiencies such as redundant battery cycling or unnecessary draw down of backup diesel generation. By showing that actual values track predicted values almost one-to-one, the results validate the proposed method for actual forecasting cases, which can result in improved scheduling and resource allocation.

4.5 Residual Analysis

The below Figure 9 presents residual plots for the proposed (XGBoost+NN) model with residuals vs. predicted values (left) and sample index (right). The residuals are a tight random cloud about zero with no apparent trend or systematic bias, which shows that the model is unbiased over the prediction range. Symmetrical spread of residuals around the zero line also indicates that errors are dispersed evenly, not associated with certain predicted values or with time intervals.

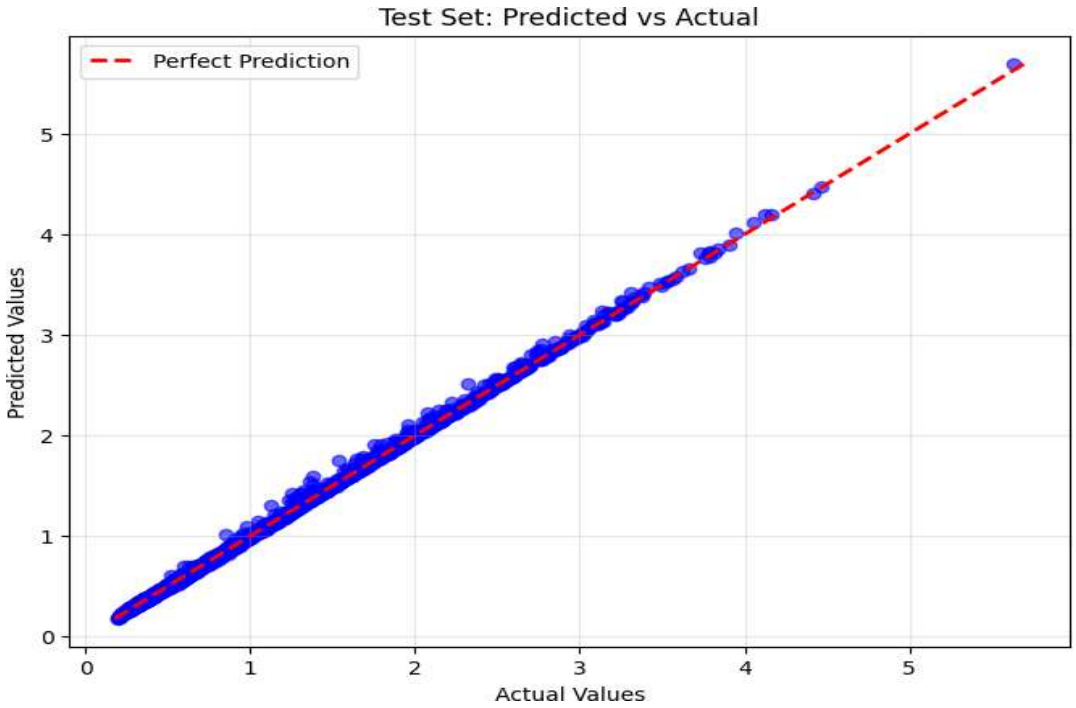
Figure 7: Actual vs predicted cost values by the Proposed model.

There are some outliers that reflect residual values beyond 0.1, but they represent a very small fraction of the dataset and don't invalidate the general predictive accu- racy. Importantly, the lack of heteroscedasticity — i.e., the residual variance remains constant across predicted values, shows that the model is robust in both low and high demand conditions. This is a good attribute for decentralized energy forecasting because it diminishes the risk of under- or overestimation that affects grid stability. The nearly perfect agreement in the actual vs. predicted plots, the above residual patterns provide evident proof of the reliability and stability of the proposed hybrid technique. They support that not only are the forecasting errors small, but also that they are well-behaved and provide strong proofs of the model's appropriateness for application to real rural energy grids.

Error Distribution and Correlation of features

Figure 10 depict the distribution of raw prediction errors and absolute error distribution of the proposed (XGBoost+NN) model, respectively. Most of the prediction errors are

Figure 8: **Actual vs. predicted values on test set.**



closely clustering around zero with an equal bell-shaped profile. This implies that this model is unbiased, giving slightly over- and under-estimated values with approximately equal frequency.

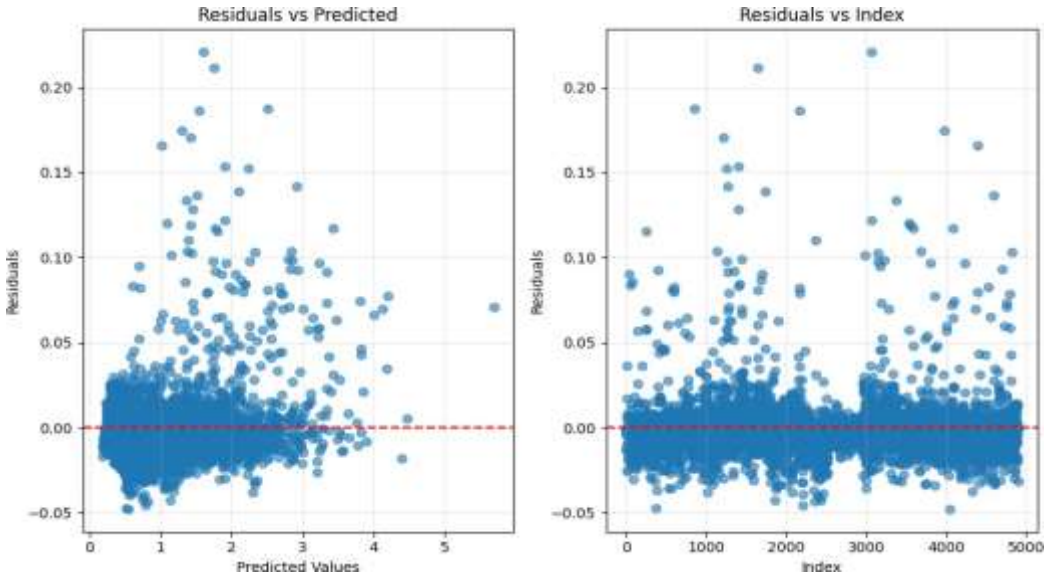


Figure 9: Residual analysis plot of residuals vs. predicted values (left) and residuals vs. index (right) of the proposed (XGBoost+NN) model.

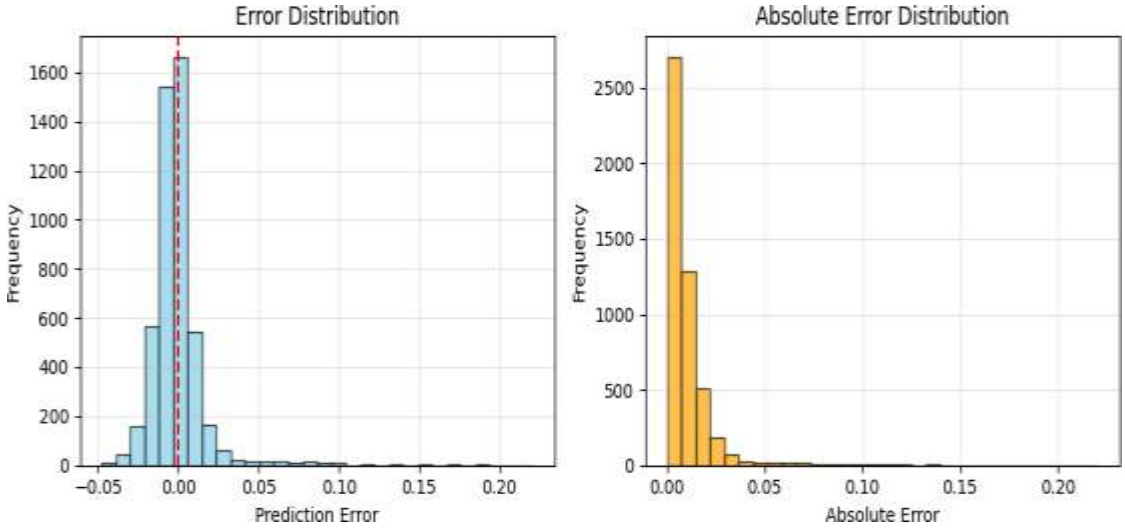


Figure 10: Distribution of prediction error (left) and absolute error (right) of the pro- posed (XGBoost+NN).

Absolute error histogram indicates that the largest proportion of errors lie within the range of 0.02, again supporting the high predictive fidelity shown in the previous analyses. Besides the magnitude of errors, it is imperative to study what causes the errors to occur so that the model can be refined. The relationship between the input features and absolute error is shown in Figure 11. The strongest correlation is greater than 0.4 is observed in F3, and the rest (F0 and F2) are less than 0.25. These variables consequently have the biggest impact on prediction error. On the contrary, the features, which include F6, F8, and F5 indicate insignificant details with errors, meaning that they do not have a direct influence on the forecasting errors. There are two valuable implications of this analysis. First, the narrow error concentration proves that the suggested model can work quite reliably in most operating conditions. Second, the targeted data quality or feature improvements can be performed by identifying the major error-driving features. As an example, F3 measurement precision could be improved further to lower the variance in prediction. This interpretability is useful in implementing AI solutions in critical energy applications because it enables its practitioners not only to trust the outputs of the models but also to optimally improve it.

4.7 Training Dynamics

The Figure 12 depicts the training and validation curve of the proposed (XGBoost+NN) model. The left panel displays training-validation loss per-epoch, and the right panel training-validation MAE. Both curves show fast convergence with the sharpest increases observed in the initial ten epochs. The curves eventually flatten and stabilize after this stage, which means that the model is able to rapidly enter an optimum region of the parameter space and stay there without oscillation.

The validity of the validation curves is also an equally significant observation in the fact that the validation curves are close to their training counterparts. The training loss is almost parallel to the validation loss during the epochs and the validation MAE takes a similar decreasing pattern. Such alignment will imply that there is no overfitting and that the model can generalize well out of the training data. In addition to this, the smooth and monotonic decreasing trend in the two metrics accentuates the stability and strength of the learning process.

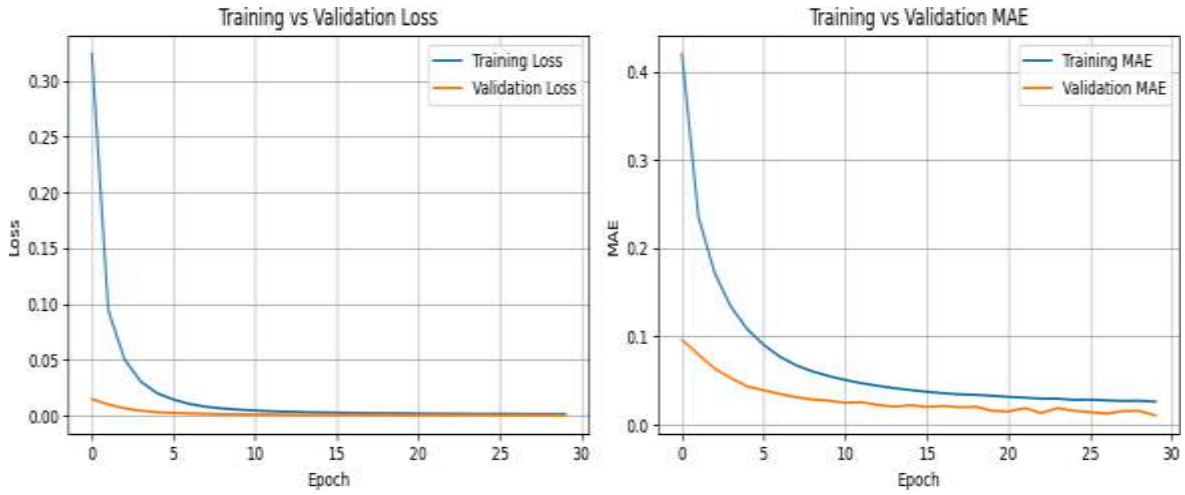


Figure 11: Association among input features and absolute error.

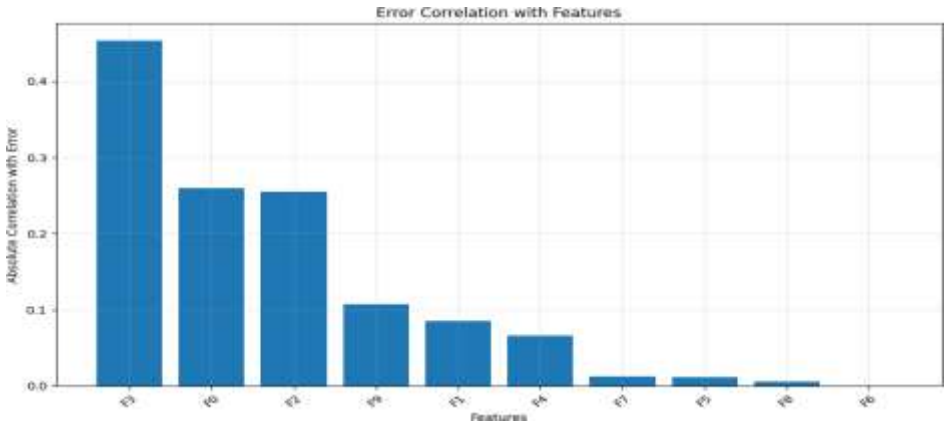


Figure 12: Comparison of training versus validation loss (left) and training versus validation MAE (right) of the proposed (XGBoost+NN) model.

These properties are essential in the prediction of decentralized rural energy. The deployed models in such settings have to be computationally efficient and capable of being retrained quickly over time as new information is obtained by distributed sensors or smart meters. The convergence behavior demonstrated suggests that the proposed model will be able to produce high-performance levels without the necessity to go through long training cycles and consume large amounts of computational power. This efficiency will minimize the operation barrier since it can easily be adopted in rural societies in which technical infrastructure and budgets can be constrained. The training dynamic supplements the robustness of the hybrid architecture, also demonstrating that it can be practiced. Based on these three aspects, namely, fast convergence, minimum overfitting, and stability, the proposed (XGBoost+NN) can be informed in the dynamic and resource-constrained environment of decentralized rural energy systems.

4.8 Comparative Analysis and Discussion

The Table 7 summarizes a thorough comparison of the most recent state-of-the-art ML and hybrid strategies of energy forecasting. Predictive accuracy and robustness are combined by the evaluation metrics reported in the literature including MSE, RMSE, R2, and MAPE. [53] examined solar forecasting in terms of ensemble learning and instance-based forecasting. Their findings indicated that LightGBM had a decent level of R2 of 0.84 with RMSE of 5.77 and KNN had lower performance with its R2 of 0.77 and higher RMSE of 6.93. Such outcomes highlight the difficulty of solar power forecasting because of great variability and intermittency.

[54] showed the usefulness of hybrid modeling to use LSTM with XGBoost in the field of smart grid load forecasting. Their hybrid reached an RMSE of 106.54 MW, R2 of 0.994 and MAPE of 1.18% as compared to standalone LSTM (RMSE = 119.41, R2 = 0.992, MAPE = 1.30%) and XGBoost (RMSE = 109.48, R2 = 0.994, MAPE =1.21%) This illustrates the benefit of fusion models in the modeling of both temporal dynamics and nonlinear interaction of features in grid demand. [55] discovered hybrid microgrid forecasting and analyzed and presented the best model as the RF. This method resulted in a very low MSE of 0.2023, RMSE of 0.4498 and a R2 of 0.9992. The implication is that even in microgrid set-ups where multi-source energy demand prediction is required, tree-based ensembles have been argued to be very useful.

Lastly, [56] proposed a smart hybrid XGBoost-PSO model to forecast the energy demand and price at the same time. They reported optimized hybrid gains, with up to 36.6 and 36.8 of RMSE and MAPE reduction and 3.9% increase in R2. This illustrates the expansion of optimization-based hybrids to model coupled demand-price. By contrast, our proposed XGBoost + NN hybrid had excellent predictive performance of MSE of 0.0003, RMSE of 0.0178, R2 of 0.999 and MAPE of 1.36%. The very small error measures, especially in RMSE, demonstrate how much better the model is stable to deal with high-dimensional and nonlinear dependencies in energy datasets. Although the MAPE is marginally greater than that of some hybrid grid models, the fact that our model has an almost perfect R2 and low RMSE indicates unprecedented accuracy on small scales of fine-grained prediction.

Ref(s)	Model(s)	MSE	RMSE	R^2	MAPE (%)	Model(s)
[53]	LGBM, KNN	–	5.77, 6.93	0.84, 0.77	–	LGBM, KNN
[54]	LSTM, XGBoost, Hybrid LSTM–XGB	–	119.41, 109.48, 106.54	0.992, 0.994, 0.994	1.30, 1.21, 1.18	LSTM, XGBoost, Hybrid LSTM–XGB
[56]	RF	0.2023	0.4498	0.9992	–	RF
[55]	XGB–PSO (Hybrid)	–	↓36.6% vs baseline	+3.9% vs base-line	↓36.8% vs baseline	XGB–PSO (Hybrid)
Ours	XGBoost+NN	0.0003	0.0178	0.999	1.36	XGBoost+NN

Table 7: Performance comparison of different forecasting models from literature and our proposed method.

5. Conclusion

This paper presented a hybrid AI system (XGBoost+NN) which is focused on improv- ing prediction and cost-efficiency in distributed rural energy systems. The proposed approach showed the state-of-the-art results by systematically benchmarking the five most popular baseline models KNN, Autoencoder, PINN, CNN-LSTM, and BiLSTM- Attention. It achieved spectacular results over traditional and DL models. These findings demonstrate that hybrid AI can encourage sustainability and cost-efficiency simultaneously, and it can be a useful tool in overcoming energy disparity among underasserved populations in the USA. Despite such positive results, there are certain directions of further working. First, pilot implementation in different rural geographies must be conducted in real life so as to have scalability and adjustability in non-homogenous instances. Second, the add-on of cybersecurity and resiliency testing will be critical towards offering reliability in decentralized grids, which are potential vic- tims of online attacks. Third, the framework can enhance its positive level of trust and acceptance among the local stakeholders by expanding it to the explainable AI (XAI) approaches. Finally, forecasting model must be integrated with controllers reinforced with learning (reward) to continue to optimize energy dispatch, storage control and demand response under dynamic conditions. Overall, the research can be valuable methodologically and practically since it demonstrates how AI can help address the void between technical correctness, financial feasibility, and sustainability in order to expand the transition to equitable rural U.S. rural decentralized energy systems.

References

[1] NRDC: A New Era for Clean Energy in Rural America. Online; accessed 2025-09-09. https://www.nrdc.org/bio/jeff-slyfield/new-era-clean-energy-rural-america

[2] Hiremath, R.B., Kumar, B., Balachandra, P., Ravindranath, N.H., Raghunandan, B.N.: Decentralised renewable energy: Scope, relevance and applications in the Indian context. Energy for sustainable development 13(1), 4–10 (2009)

[3] Durston, J.: Building social capital in rural communities (where it does not exist). In: Latin American Studies Association Annual Meetings, Chicago, pp. 24–26 (1998)

[4] Bouffard, F., Kirschen, D.S.: Centralised and distributed electricity systems. Energy policy 36(12), 4504–4508 (2008)

[5] Mperejekumana, P., Shen, L., Zhong, S., Gaballah, M.S., Muhirwa, F.: Explor- ing the potential of decentralized renewable energy conversion systems on water, energy, and food security in africa. Energy Conversion and Management 315, 118757 (2024)

[6] Karumba, S., Sethuvenkatraman, S., Dedeoglu, V., Jurdak, R., Kanhere, S.S.: Barriers to blockchain-based decentralised energy trading: a systematic review. International Journal of Sustainable Energy 42(1), 41–71 (2023)

[7] Wehbi, H., Kemper, N.: Factors influencing the transition to renewable energy in small- to mid-sized american communities. Energy, Sustainability and Society 15, 33 (2025) https://doi.org/10.1186/s13705-025-00531-1 . Discusses modeling complexity and need for sophisticated approaches

[8] ACEEE: 2020 Energy Burden Report: National and Metropolitan Energy Bur- dens. Online; accessed 2025-09-09. A quarter of U.S. households and two-thirds of low-income ones face high energy burden (spend ¿6https://www.aceee.org/ energy-burden

[9] Rane, N.L., Paramesha, M., Choudhary, S.P., Rane, J.: Machine learning and deep learning for big data analytics: A review of methods and applications. Partners Universal International Innovation Journal 2(3), 172–197 (2024)

[10] Rane, N., Choudhary, S.P., Rane, J.: Ensemble deep learning and machine learn- ing: applications, opportunities, challenges, and future directions. Studies in Medical and Health Sciences 1(2), 18–41 (2024)

[11] Raut, J.M., Pande, P.B., Madurwar, K.V., Bhagat, R.M., Uparkar, S.S., Shelke, N., Isleem, H.F., Arpita, Vairagade, V.S.: Life cycle assessment and multicriteria decision making analysis of additive manufacturing processes towards optimal performance and sustainability. Scientific Reports 15(1), 25167 (2025)

[12] Topaloˇglu, F.: Machine learning-based approaches and comparisons for estimating missing meteorological data and determining the optimum data set in nuclear energy applications. IEEE Access (2025)

[13]Boeschoten, S., Catal, C., Tekinerdogan, B., Lommen, A., Blokland, M.: The automation of the development of classification models and improvement of model quality using feature engineering techniques. Expert Systems with Applications 213, 118912 (2023)
[14] Adedeji, P.A.: Hybrid Renewable Energy-based Facility Location: a Geographical Information System (GIS) Integrated Multi-criteria Decision-making (MCDM) Approach. University of Johannesburg (South Africa), ??? (2020)
[15] Chodakowska, E., Nazarko, J., Nazarko, L- ., Rabayah, H.S., Abendeh, R.M., Alawneh, R.: Arima models in solar radiation forecasting in different geographic locations. Energies 16(13), 5029 (2023)
[16] Saffari, M., Khodayar, M.: Spatiotemporal deep learning for power system applications: A survey. IEEE Access (2024)
[17] Khedimi, M., Zhang, T., Dehmani, C., Zhao, X., Geng, Y.: A unified deep learn- ing ensemble framework for voice-based parkinson's disease detection and motor severity prediction. Bioengineering 12(7), 699 (2025)
[18] Uddin, I., Awan, H.H., Khalid, M., Khan, S., Akbar, S., Sarker, M.R., Abdolrasol, M.G., Alghamdi, T.A.: A hybrid residue based sequential encoding mechanism with xgboost improved ensemble model for identifying 5-hydroxymethylcytosine modifications. Scientific Reports 14(1), 20819 (2024)
[19] Reuters: Us unveils $366 mln for clean energy in underserved, remote areas. Reuters (2024)
[20] Felix, E.A., Lee, S.P.: Systematic literature review of preprocessing techniques for imbalanced data. Iet Software 13(6), 479–496 (2019)
[21] Heiskanen, E., Matschoss, K.: Understanding the uneven diffusion of building- scale renewable energy systems: A review of household, local and country level factors in diverse european countries. Renewable and Sustainable Energy Reviews 75, 580–591 (2017)
[22] R. Singh, A., Kumar, R.S., Bajaj, M., Khadse, C.B., Zaitsev, I.: Machine learning- based energy management and power forecasting in grid-connected microgrids with multiple distributed energy sources. Scientific Reports 14(1), 19207 (2024)
[23] Singh, A.R., Sujatha, M., Kadu, A.D., Bajaj, M., Addis, H.K., Sarada, K.: A deep learning and iot-driven framework for real-time adaptive resource allocation and grid optimization in smart energy systems. Scientific Reports 15(1), 19309 (2025)
[24] Masrur, H., Khaloie, H., Al-Awami, A.T., El Ferik, S., Senjyu, T.: Cost-aware modeling and operation of interconnected multi-energy microgrids considering environmental and resilience impact. Applied Energy 356, 122320 (2024)
[25] Hoummadi, M.A., Bossoufi, B., Karim, M., Althobaiti, A., Alghamdi, T.A., Alenezi, M.: Advanced ai approaches for the modeling and optimization of microgrid energy systems. Scientific Reports 15(1), 12599 (2025)
[26] Oladapo, B.I., Olawumi, M.A., Omigbodun, F.T.: Machine learning for optimizing renewable energy and grid efficiency. Atmosphere 15(10), 1250 (2024)
[27] Li, T., Feng, L.: Coordinated development of rural ecological construction and carbon neutrality: a deep learning approach for enhanced sustainability. Frontiers in Ecology and Evolution 11, 1267259 (2024)
[28] Danish, M.S.S., Senjyu, T.: Shaping the future of sustainable energy through AI-enabled circular economy policies. Circular Economy 2(2), 100040 (2023)
[29] Singh, A.R., Dey, B., Bajaj, M., Kadiwala, S., Kumar, R.S., Dutta, S., Zait-sev, I.: Advanced microgrid optimization using price-elastic demand response and greedy rat swarm optimization for economic and environmental efficiency. Scientific Reports 15(1), 2261 (2025)
[30] Paul, K., Jyothi, B., Kumar, R.S., Singh, A.R., Bajaj, M., Hemanth Kumar, B., Zaitsev, I.: Optimizing sustainable energy management in grid connected micro- grids using quantum particle swarm optimization for cost and emission reduction. Scientific Reports 15(1), 5843 (2025)
[31] Zhao, Y.: Advancing smart communities with a deep learning framework for sustainable resource management. PLoS One 20(8), 0329492 (2025)
[32] Liu, P., Wang, S., Wei, Y.: Administration strategy of energy management in smart grid: system view and optimization path. Frontiers in Energy Research 11, 1202904 (2023)
[33] Elhabyb, K., Baina, A., Bellafkih, M., Deifalla, A.F.: Machine learning algorithms for predicting energy consumption in educational buildings. International journal of energy research 2024(1), 6812425 (2024)
[34] Parvathareddy, S., Yahya, A., Amuhaya, L., Samikannu, R., Suglo, R.S.: Trans- forming mining energy optimization: a review of machine learning techniques and challenges. Frontiers in Energy Research 13, 1569716 (2025)
[35]Li, W., Li, J.-P., Wang, Y.-F., Stan, S.-E.: Is artificial intelligence an impedi- ment or an impetus to renewable energy investment? evidence from china. Energy Economics, 108550 (2025)
[36] Khalid, J.: Ai technologies for sustainable development: A roadmap to smart energy savings in society 5.0. Available at SSRN 4568457
[37] Yao, F., Zhao, W., Forshaw, M., Song, Y.: A holistic power optimization approach for microgrid control based on deep reinforcement learning. Neurocomputing, 131375 (2025)
[38] Ioannou, I., Javaid, S., Tan, Y., Vassiliou, V.: Autonomous reinforcement learn- ing for intelligent and sustainable autonomous microgrid energy management. Electronics 14(13), 2691 (2025)
[39] Maddineni, V.K., Koganti, N.B., Damacharla, P.: Enhancing microgrid per- formance prediction with attention-based deep learning models. In: 2024 11th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), pp. 116–121 (2024). IEEE
[40] Silva-Rodriguez, J., Raffoul, E., Li, X.: Lstm-based net load forecasting for wind and solar power-equipped microgrids. In: 2024 56th North American Power Symposium (NAPS), pp. 1–6 (2024). IEEE
[41] Li, Y., Wang, R., Yang, Z.: Optimal scheduling of isolated microgrids using auto- mated reinforcement learning-based multi-period forecasting. IEEE Transactions on Sustainable Energy 13(1), 159–169 (2021)
[42] Harrold, D.J., Cao, J., Fan, Z.: Renewable energy integration and microgrid energy trading using multi-agent deep reinforcement learning. Applied Energy 318, 119151 (2022)
[43] Hasan, S., Zeyad, M., Ahmed, S.M., Anubhove, M.S.T.: Optimization and plan- ning of renewable energy sources based microgrid for a residential complex. Environmental Progress & Sustainable Energy 42(5), 14124 (2023)
[44] Silva, J.A.A., L´opez, J.C., Guzman, C.P., Arias, N.B., Rider, M.J., Silva, L.C.: An iot-based energy management system for ac microgrids with grid and security constraints. Applied energy 337, 120904 (2023)
[45] Isanbaev, V., Ban˜os, R., Mart´ınez, F., Alcayde, A., Gil, C.: Monitoring energy and power quality of the loads in a microgrid laboratory using smart meters. Energies 17(5), 1251 (2024)
[46] Xu, L., Huang, J., Yang, S., Zhu, Y.: Edge computing for microgrid via matlab embedded coder and low-cost smart meters. In: 2024 8th International Conference on Smart Grid and Smart Cities (ICSGSC), pp. 234–239 (2024). IEEE

[47] Bhardwaj, D., Shalini, M., Jebaseeli, S.K., Jadhav, S., Alabdeli, H., Sutar, V., Kumar, R.S.: Ai-driven energy management systems for microgrids: Optimizing renewable energy integration and load balancing. In: E3S Web of Conferences, vol. 591, p. 01005 (2024). EDP Sciences
[48] Hassan, A., Mohani, S.S.U.H., Essani, I.Y., Taj, S.A., Aslam, A., Abbas, Y.: Ai- enabled energy management for large-scale solar farms: Optimizing power distri- bution, grid stability, and real-time performance monitoring. THE PROGRESS: A Journal of Multidisciplinary Studies 6(1), 66–84 (2025)
[49] Rasheed, D.H., Tambe, S.B.: Advancing energy efficiency with smart grids and iot-based solutions for a sustainable future. Estidamaa 2024, 36–42 (2024)
[50] Dahrouj, H., Bansal, R.C., Tawfik, H.M., et al.: An overview of the prospects and challenges of using artificial intelligence for energy management systems in microgrids. arXiv preprint arXiv:2505.05498 (2025)
[51] Joha, M.I., Rahman, M.M., Zubair, M.I.: Iot-based smart energy monitoring, management, and protection system for a smart microgrid. In: 2024 Third Inter- national Conference on Power, Control and Computing Technologies (ICPC2T), pp. 398–403 (2024). IEEE
[52] Sohrabi, B.: US Electric Power Plants [Dataset]. Kaggle. Accessed: 2025-09-09 (2025). https://www.kaggle.com/datasets/behroozsohrabi/ us-electric-power-plants
[53] Suanpang, P., Jamjuntr, P.: Machine learning models for solar power generation forecasting in microgrid application implications for smart cities. Sustainability 16(14) (2024) https://doi.org/10.3390/su16146087
[54] Dakheel, F., C¸ evik, M.: Optimizing smart grid load forecasting via a hybrid long short-term memory-xgboost framework: Enhancing accuracy, robustness, and energy management. Energies 18(11) (2025) https://doi.org/10.3390/ en18112842
[55] Zarma, T., Ali, E., Galadima, A., Karataev, T., Hussein, S., Adeleke, A.: Energy demand forecasting for hybrid microgrid systems using machine learning models. Proceedings of Engineering and Technology Innovation, 20 (2025) https://doi. org/10.46604/peti.2024.14098
[56] Parizad, B., Ranjbarzadeh, H., Jamali, A., Khayyam, H.: An intelligent hybrid machine learning model for sustainable forecasting of home energy demand and electricity price. Sustainability 16(6) (2024) https://doi.org/10.3390/su16062328