Research Article

Towards Sustainable Living: IoT Enabled Predictive Approach for Domestic Solution

Moazam Ali[1], Israr Akhter[1], Saba Mahmood[1*], Fatima Khalique[2], Abrar Ahmed[1] and Farah Akif[1]

[1]Department of Computer Science, Bahria University, E8, Islamabad, Pakistan; [2]Center of Excellence in Artificial Intelligence (CoE-AI), Department of Computer Science, Bahria University, Islamabad, Pakistan.

Abstract: With the advent of 5G technologies, IoT infrastructures are the key enablers in providing smart solutions like smart homes, smart industrial infrastructure, and smart cities. With the increasing availability of Internet of Things (IoT) devices as an interconnected network, smart homes have emerged as promising solutions for enhancing energy efficiency by intelligently regulating energy consumption. Traditional home energy management systems rely on predefined schedules or manual user interventions to regulate energy usage. This limitation restrains the realization of substantial energy savings from a domestic perspective. To overcome these challenges, our study proposes an algorithm IPA-DS. IPA- DS estimates the current energy usage statistics in real-time home environment. The estimated statistics are then exploited as features to train the machine learning algorithms. The trained algorithms are then tested and validated using error functions and processing overheads.

Received: December 25, 2025; **Accepted:** March 31, 2026; **Published:** May 01, 2026
***Correspondence:** Saba Mahmood, Department of Computer Science, Bahria University, E8, Islamabad, Pakistan; **Email**: smahmood.buic@bahria.edu.pk
Citation: Ali, M., Akhter, I., Mahmood, S., Khalique, F., Ahmed, A. and Akif, F., 2026. Towards Sustainable Living: IoT Enabled Predictive Approach for Domestic Solution. *Journal of Engineering and Applied Sciences,* 45: 01-11.
DOI: https://dx.doi.org/10.17582/journal.jeas.2026/45.01.11
Keywords: Adaboost, Home automation, Home energy management, IoT solution; LSTM, Machine learning, Random forest, Sustainable living

Introduction

Energy conservation has garnered much interest from the scholarly community because of the rising demand for energy over the past few decades. Residents of modern cities are currently troubled by their expensive residential energy costs (Ehsan *et al.*, 2021; Qadeer and Ehsan, 2021; Nawaz *et al.*, 2023; Ehsan, 202). A variety of readily available and, in some cases, essential household appliances are necessary for a reasonable way of life. Some appliances can fail undetected. Users can save significant money by closely monitoring and controlling the current drawn by each appliance in the house. Recently, the combination of machine learning and smart home technology has grabbed a lot of attention and promises a revolution in home energy management (IoT Based Digital Wattmeter, 2021). Smart Home Energy Management Systems (SHEMS) use a variety of sensors, smart meters, and Internet of Things (IoT) devices to monitor and control energy consumption as modern living environments become more networked. Machine learning frameworks are essential in this environment because they offer flexible and innovative solutions. These frameworks enable SHEMS to forecast energy consumption trends and collect and analyze various datasets, allowing for real-time efficiency improvements (Girisha *et al.*, 2023). This research article proposed an innovative and sustainable living solution to reduce energy costs via monitoring power consumption. We make an IoT device that integrates various intelligent components to achieve this. After getting the data through an IoT device, we apply Machine-learning algorithms by training the model for prediction. It will be like a decision node, which tells how to respond to a specific input and what output should be given to house owners in response to that particular input. Once the IoT device and application are integrated, managing and testing power consumption will be done through the machine learning model that was trained earlier. The main objectives of this paper are:

- We configure a complete IoT device by integrating various components such as microcontroller, sensor, Wifi and cloud settings.
- With the help of an IoT device-generated dataset, we applied advanced methods for feature extraction.
- We predict energy consumption Using Random Forest, Adaboost and LSTM.
-

The order of the entire document is as follows: Segment II defines the technique framework. Section III describes and compares the experimental procedure to competitive state-of-the-art methodologies. The final section of the paper is section IV. The novelty of the proposed IPA-DS framework lies in its combined architecture that use real-time IoT data acquisition, structured preprocessing, correlation and K-best feature selection, and comparative evaluation of ensemble and deep learning models.

Related Work

The authors of (Rastogi *et al.*, 2016) address the architecture and execution of an Internet of Things-based smart metering system, emphasizing the system's significance in power and communication costs. A smart meter, a digital gadget, keeps track of how much electricity is used and transmits the data to utility companies for billing and monitoring. They employed a Metre Data Management (MDM) system to process and aggregate the data and an Advanced Metering Infrastructure (AMI) to gather data from every meter. They asserted that distributed management systems, customer management, outage management, and other subsystems might all benefit from the data collected from smart meters. This system's communication component is crucial, including cellular, Wi-Fi, ZigBee, Power Line Carrier (PLC), Broadband over Power Lines (BPL), and more. The IRM serves as an interface between PLCs and the Server, receiving the meter readings from the household meters and sending them to the PLC. Along with outlining a cryptographic framework for secure communication between the meter and the server, the article also addresses security issues. The paper explains several attack types, including chosen plaintext attacks, known plaintext attacks, ciphertext-only attacks, and chosen ciphertext attacks. The study also describes the Wide Area Network (WAN), Neighborhood Area Network (NAN), and Home Area Network (HAN) network model for collecting meter data. Every network has a distinct function, such as linking household appliances or gathering and transmitting meter data from several residences to the Power Management System (PMS).

The system (Mufassirin and Hanees, 2018) facilitates efficient administration and monitoring of electricity consumption. The front-end metering (smart meter) and back-end cloud interfacing (cloud server) subsystems of the Internet of Things intelligent metering system are described in this article by the authors, along with the Composite Design Methodology (CDM) that was utilized to develop it. These subsystems are communicated effectively with one another. This makes it possible to monitor and control the amount of electricity used effectively. To achieve an effective system, it is intended to combine two smart systems. An Allegro ACS712 current sensor for monitoring current flow, an IoT Arduino UNO chipset for the core controller board, and a SIM800L GSM module for communicating with the cloud server are the components used in the system. The hardware design involves interfacing the various components and sensors with the Arduino board. The ACS712 sensor has been used to check the amount of current, and the Arduino communicates with the GSM module to establish data link with the cloud server. A relay is also used to switch the meter on or off based on the customer's usage and budget.

Authors in (Condon *et al.*, 2023) designing and implementing a home energy management system that uses cloud computing and the Internet of Things. The solution enables smart home appliance control, data storage, and gathering to facilitate effective energy consumption management. The authors created two scenarios: an AWS IoT Core-based cloud-based HEMS and a local HEMS utilizing edge devices. The perception layer of the suggested system consists of sensors, smart plugs, and smart meters collectively," things." These gadgets make it possible to gather and process data. Using technologies like ZigBee, Lora, and mobile networks, the communication layer in the system makes it easier for the perception layer and the middleware layer to connect.

Authors in (Devi *et al.*, 2023) described The Smart Home Energy Management System (SHEMS) that uses machine learning and Internet of Things (IoT) devices to optimize energy use in a smart home setting. The system gathers real-time data on energy consumption using Internet of Things (IoT) devices, such as sensors and smart plugs. The authors used Deep Neural Networks (DNNs) to forecast future consumption patterns based on previous data and environmental variables. The architecture of the system that the authors have presented is intended to track and regulate the energy consumption of domestic appliances, giving consumers the ability to modify their energy use in response to real-time feedback. Using features taken from IoT sensor data, the DNN is constructed using either Keras or Tensor Flow.

Authors in (Kumar *et al.*, 2023) explored machine learning techniques to optimize energy consumption in smart homes. Sensors and other devices that gather information on temperature, energy use, and other factors are a feature of smart homes. Before delving into the machine learning model section, the authors covered several approaches, including feature selection and data preparation, which are crucial for a model to produce reliable predictions. Machine learning algorithms then use this data to analyze and forecast how best to use energy, which increases security, comfort, and energy efficiency. The authors explore several machine learning algorithms, such as stochastic gradient descent (SGD), decision trees, and support vector machines, and they suggest applying SGD to energy optimization.

The outdated home energy management systems are ineffective in helping households reduce their energy consumption and costs. Many homes still use inefficient and time-consuming manual energy management techniques, such as manually turning off lights and appliances. In addition, because homeowners aren't always aware of how much energy their appliances and devices use in real-time, appropriately optimizing household energy use can be difficult. Another problem is that energy costs are rising, and many households are looking for ways to reduce their energy bills without sacrificing comfort and convenience. If users have some forecasting system that can accurately predict energy consumption in the future, they can make timely decisions to avoid energy wastage. To address these energy wastage problems, we will create a system through which users can test their power consumption in homes, automating energy-saving actions and ultimately reducing power consumption and costs while promoting Sustainability.

Material and Methods

An exhaustive literature review shows that certain energy management systems lacked prediction for real-time energy and power consumption. They mainly operated as cloud-based platforms, facilitating energy monitoring without empowering users to assess past consumption for informed future projections. Furthermore, prevailing systems often relied on a singular machine learning model, compromising accuracy. To address these limitations, our research is to enhance the system's efficacy. By collecting and preprocessing data from the cloud, coupled with rigorous feature selection, our approach aims to view future predictions through a machine learning model. Incorporating machine learning algorithms, namely Random Forest, AdaBoost, and Long Short-Term Memory (LSTM), our study intends to assess their performances for optimal model selection. This interdisciplinary approach seeks to tie existing gaps, contributing to the evolution of accurate and user-empowering energy management systems. Algorithm 1 (Table 1) shows the complete picture of our proposed method.

Figure 1 depicts the practical layout of the proposed system.

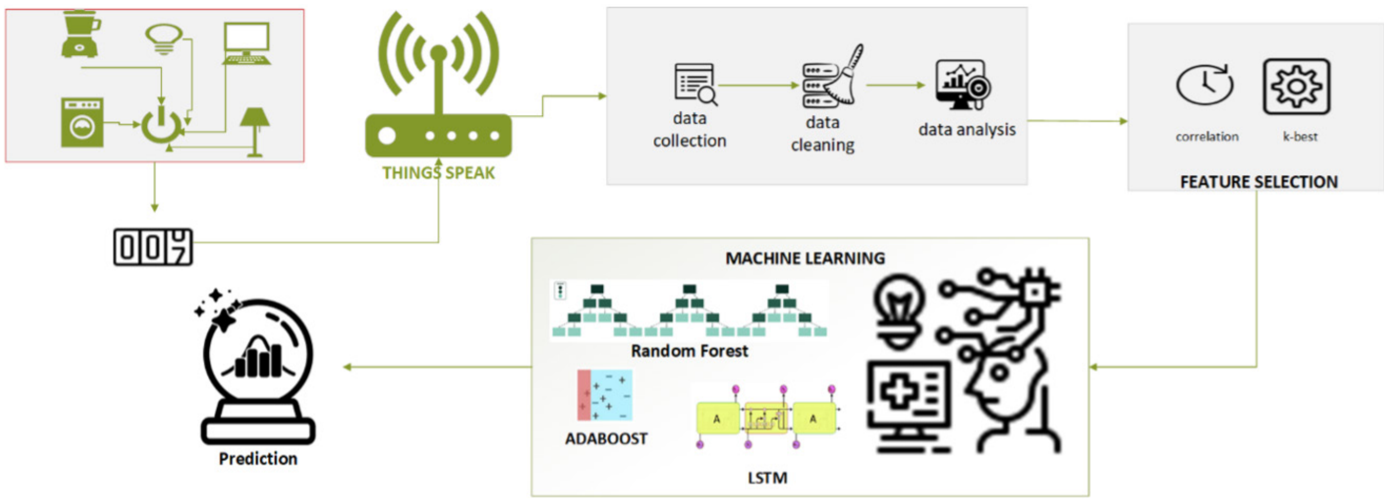

Figure 1: *The architectural diagram of the designed framework.*

Table 1*:* **Algorithm***: IPA- DS Algorithm.*
Input*: IoT Data (Data Gathered via IoT Device)*
Output*: Predicted Results*

Data Collection	
a. Acquire data from IoT device.	
b. Transmit data to the IoT Cloud (Things Speak).	
Data Preparation	
a. Data Cleaning	
i. Identify and handle missing values.	
ii. Detect and remove outliers using the IQR approach.	
b. Data Analysis	
i. Examine dataset characteristics.	
ii. Verify data integrity.	
Features Extraction	
a. Correlation	
i. Compute correlation matrix.	
ii. Select features with significant correlation.	
b. K-Best	
i. Employ K-Best feature selection method.	
ii. Choose top-performing features based on evaluation scores.	
Machine Learning	
a. Random Forest	
i. Implement Random Forest algorithm.	
ii. Train the model using selected features.	
b. AdaBoost	
i. Apply AdaBoost algorithm.	
ii. Enhance model performance through boosting.	
c. LSTM	
i. Implement Long Short-Term Memory (LSTM) neural network.	
ii. Utilize LSTM for sequence modeling in time series data.	
Output	Predicted results for the given IoT data.

Data collection

The data collection process from the IoT device is enabled and supported by the Things Speak cloud in this research. The Arduino IDE code incorporates the channel's API key and the WIFI credentials required for connectivity. After configuring the time zone during data collection, we export the information as a CSV file. By employing two separate channels, one administers the initial dataset, and the other collects often unseen datasets in preparation for model evaluation. The training dataset, denoted by the API Key RS9JIEVR9PBNO7EH, consists of 14,512 entries. A week's worth of data is gathered for the unseen or testing dataset, which the API Key AS69A06RVF8ZB55H denotes and comprises 2,055 entries. Figure 2 illustrates the visual representation of the initial data structure. Figure 3 shows the representation of the unseen dataset.

Data Preprocessing

As our research explores the complexities of real-world datasets, we commence with the fundamental procedure of data cleansing, understanding its critical significance in ensuring reliable insights. We apply a basic preprocessing method to avoid computational complexity and time. The dataset was chronologically sorted before modeling. For tree-based models, lag-based temporal consistency was maintained through ordered splitting. For LSTM, sequential input windows were constructed to preserve temporal data dependencies. The integrity of the dataset was verified in our research effort by stringently examining it for missing values using the pandas and NumPy libraries. By implementing the Interquartile Range (IQR) method, prospective disruptions were effectively mitigated by effectively managing outliers. This action was taken to correct any errors and strengthen the dataset's structure. This research-driven undertaking highlights the significance of data cleansing, which guarantees the reliability and accuracy of our analyses and contributes to the overarching objective of extracting valuable insights. Algorithm 02 describes the overview of data preprocessing.

Table 2: Algorithm 02: *Dataset Cleaning*
Input: *IoT Dataset*
Output: *Cleaned dataset suitable for analysis*

1. Identify Missing Values
a. Check for missing values in the dataset.
b. If missing values exist: Remove or replace with an appropriate test statistic.
2. Outlier detection
a. Use the Interquartile Range (IQR) approach to identify outliers.
b. Decide whether to remove or relocate outliers based on the use case.
3. Datetime formatting
a. Recognize the importance of datetime formatting, especially for time series forecasting.
b. Convert the date column into datetime format.
4. Utilize Python Libraries
Implement Python libraries such as NumPy and pandas for efficient data cleaning.
5. Ensure Consistency
Maintain consistency and reliability throughout the cleaning process.
6. Iterative Approach
a. If issues or errors are encountered during subsequent analyses: Consider an iterative approach to revisit and refine data cleaning steps.

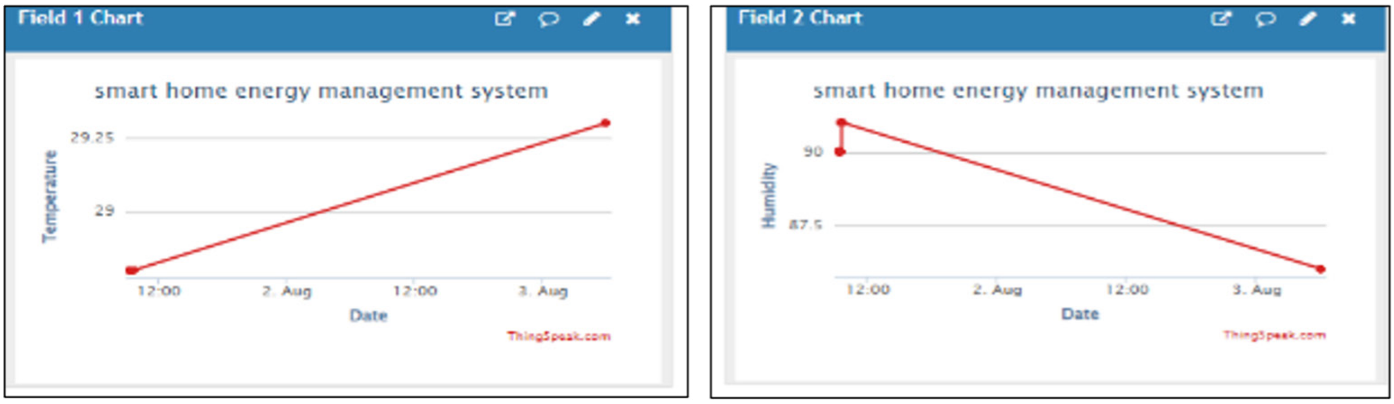

Figure 2: *Data Collection For original or training dataset.*

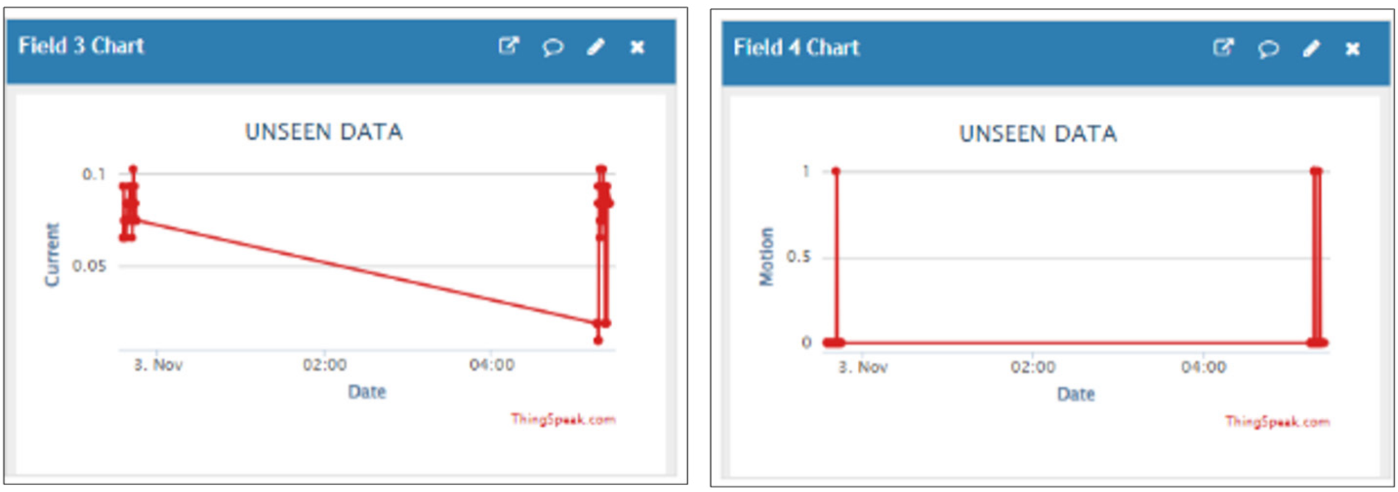

Figure 3 *Data Collection For unseen or testing datasets.*

Calculating Power Consumption

Power consumption is calculated as P = V × I using 220V AC and current from the calibrated ACS712 sensor, with energy estimated by integrating power over discrete sampling intervals. In this research, we use a current sensor as a small LED for input, allowing us to get small current values. After getting the data, the next step is machine learning. Initially, the model could not predict because there were not enough variations in the dataset, which caused the model not to learn anything and continuously overfit (Qadeer and Ehsan, 2021).

Features extraction

After working with data preprocessing, we have moved to feature selection. We have used two methods for feature selection. One is Co-relation and the other is k-best. After using both methods, we have two columns in return: temperature and current. The model identified two critical columns, temperature and current, for their selection based on fewer variations in the dataset than the other two. Fig. 4 represents the feature selection approach using Co-relation.

	Temperature	Humidity	Current	Motion
Temperature	1.000000	-0.800993	-0.037832	-0.036697
Humidity	-0.800993	1.000000	0.016572	0.047281
Current	-0.037832	0.016572	1.000000	0.054903
Motion	-0.036697	0.047281	0.054903	1.000000
Power Consumption	0.016004	0.000474	-0.029281	0.003047

Figure. 4. *The feature selection approach using Co-relation.*

Prediction via Machine Learning

Predicting a result based on past or present circumstances is the practice of forecasting. If we look at the artificial intelligence field, we forecast the future based on prior knowledge and current information. These issues are dealt with by the subset of the artificial intelligence field known as Machine Learning. Machine learning is the most effective method now available for assisting organizations in increasing efficiency. In this, we train the machine, create a model, and then forecast the desired outcome based on the model. Regression is the process of predicting an outcome based on previous inputs. We have four independent variables or inputs in our problem.

- Temperature
- Humidity
- Current
- Motion and the result or dependent variable
- Power Consumption

To mitigate overfitting, hyperparameter tuning was conducted using validation data. Early stopping was use for LSTM training. Tree depth and number of estimators were restricted for ensemble models. Additionally, feature selection reduced redundant inputs. The dataset requires the application of a time-series regression model and a forecast of the outcome obtained during the data analysis. We used our data to apply three models, evaluate the results, and choose the most accurate one based on MAE and MSE (Root Mean Squared Error) and by analyzing the results of these models on unseen datasets. Hyperparameters were optimized using grid search combined with time-series split validation. For Random Forest and AdaBoost, parameters such as number of estimators, tree depth, and learning rate were tuned. For LSTM, hidden units, batch size, learning rate, and number of layers were experimentally adjusted. The configuration yielding the lowest validation RMSE was selected. Hyperparameters were tuned using grid search with time-series split validation. Random Forest achieved optimal performance at 60 estimators, max depth 14, and min samples split 4, while AdaBoost performed best with 80 estimators, learning rate 0.1, and tree depth 11. The LSTM used three layers with 200 units each, batch size 32, learning rate 0.001 (Adam), and early stopping to prevent overfitting. All experiments were conducted using Python 3.10 on a system with Intel i7 processor 12th generation and 16GB RAM. The dataset was chronologically divided into 75% training data and 25% unseen testing data to preserve temporal structure. Hyperparameters were optimized using grid search combined with time-series split validation.

Random Forest

A regression-based supervised machine learning method is called Random Forest (Nawaz *et al.*, 2023). The following parameters are used to create the Random Forest object. N estimators specify how many decision trees will be used in the random forest. The intended number of decision trees is kept in the variable n in the code snippet you offered. In our case, we get 60 trees as optimal as per the random forest model. Therefore, we chose 60 trees. Random state: a seed that is selected from at random and used to duplicate the same results over several runs. Setting this parameter to a fixed value ensures that the algorithm's random initialization is the same in every run, resulting in reproducible results. OOB Error determines an optimal number of decision trees and fully tunes the model. Out-of-bag Error is defined as the proportion of inaccurate classifications to out-of-bag samples. The optimal number of decision trees suggested using OOB error is 60. Figure 5 depicts the OOB error as a function of the tree plot.

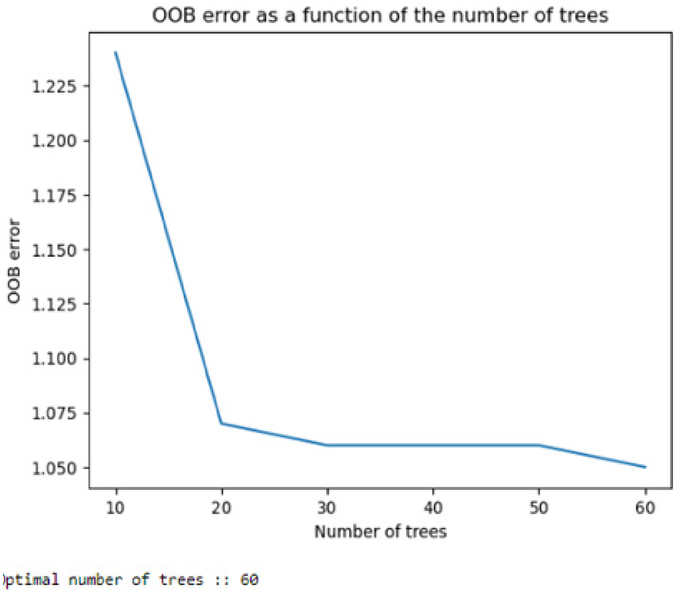

Figure. 5. *OOB error as a function of trees Plot.*

Figure 6 shows the random forest predictions on the test dataset.

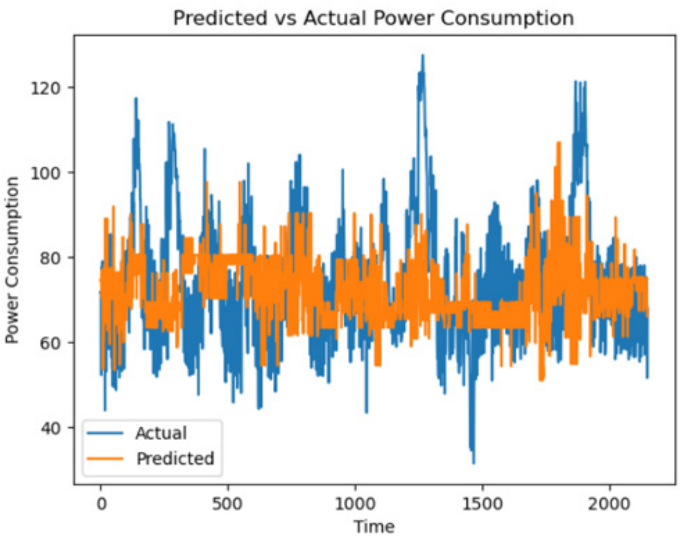

Figure. 6. *Random forest predictions on the test dataset.*

Adaboost

AdaBoost (Girisha *et al*., 2023) Regression is an ensemble learning technique. To produce a powerful, precise regression model, it integrates several weak learners' predictions, typically decision trees or other basic models. AdaBoost uses an iterative procedure to function. Using an iterative strategy, the ensemble concentrates on more challenging samples, lowering the final model's bias and variance. A weighted sum of the predictions from each weak learner is the final prediction in an AdaBoost Regression. AdaBoost Regression is frequently used in fields where continuous numerical value prediction is essential. It can be applied to population growth modeling, home price estimation, and stock price prediction projects. In our project, we selected decision trees as the base estimator for adaboost. These base estimators are called weak learners. The decision trees give good prediction accuracy on our dataset. The max depth we have selected is 11, in which the model is not overfitting. We have also tested with other depths, but their model is overfitting. Every model has its way of learning patterns and behavior in a dataset. Therefore, results on the same dataset can be different for every model. Figure 7 depicts the training vs test error plot.

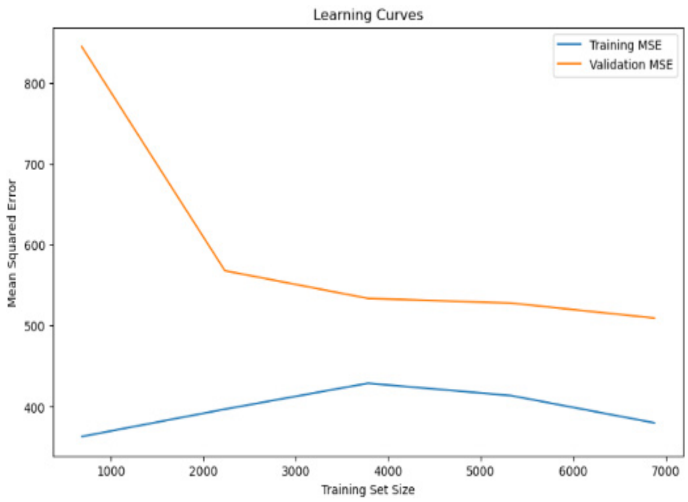

Figure. 7. *The training vs test error plot*

Figure 8 shows the adaboost predictions on the test dataset.

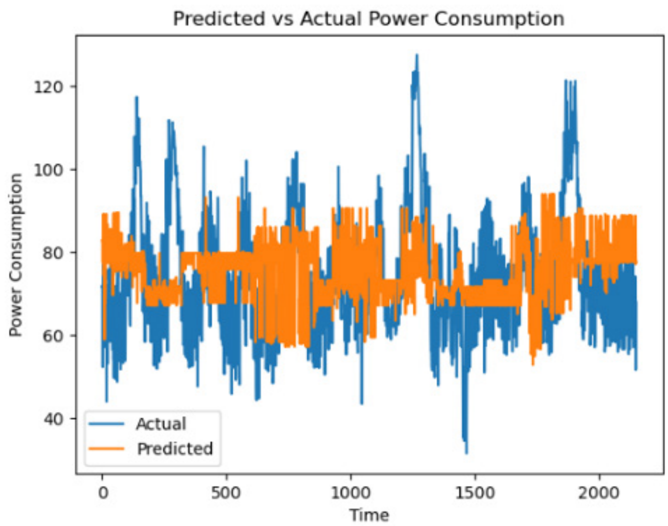

Figure. 8. *Adaboost forest predictions on the test dataset.*

Links Researchers

Long short-term memory (LSTM)

Long short-term memory (LSTM) (IoT Based Digital Wattmeter, 2021) neural networks are a type of Recurrent neural network that can capture long-term dependencies in data and make more accurate predictions based on that dependency. LSTM is used to predict weather forecasts and stock prices. The first LSTM layer in our model, LSTM 18, has shape weights (2, 200). The first dimension represents the number of input features or units from the previous layer. In contrast, the number of units in the current LSTM layer is represented by the second dimension. The second LSTM layer in our model, LSTM 19, has shape weights (50, 200). The first dimension represents the number of input features or units (50 in our case) from the previous layer. In contrast, the second dimension represents the number of units (200) in the current LSTM layer. The third LSTM layer in our model, LSTM 20, has shape weights (50, 200). The first dimension represents the number of input features or units (50) from the previous layer. In contrast, the second dimension represents the number of units (200) in the current LSTM layer. Figure 9 shows the loss graph of the training vs test set. The higher LSTM RMSE is due to limited dataset size and low temporal variability, making tree-based models such as Random Forest and AdaBoost more effective; future work will use larger datasets to capture long-term dependencies.

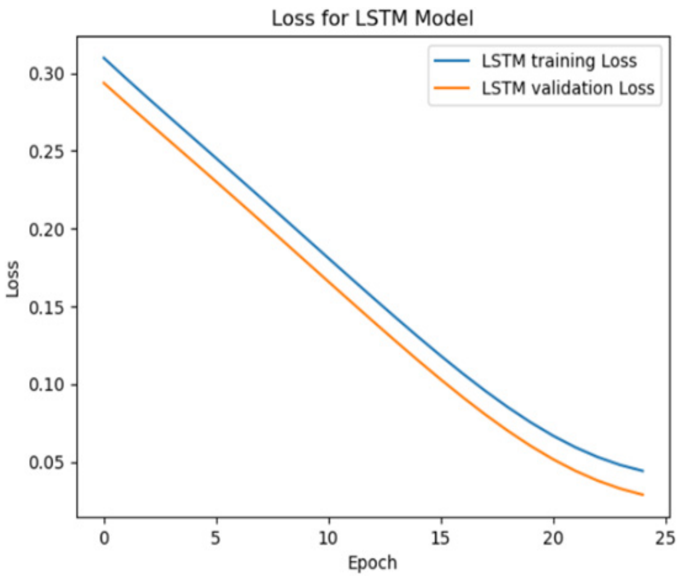

Figure. 9. *Loss Graph of Training vs Test Set*

Figure 10 shows the LSTM predictions on the test dataset.

Models Evaluation Metrics Comparison

Time-series cross-validation with a five-fold rolling window was applied to preserve chronological order, using earlier windows for training and subsequent ones for validation. The average RMSE was 16.12 for Random Forest, 15.89 for AdaBoost, and 20.31 for LSTM, confirming AdaBoost's superior stability. Random Forest may struggle with long-term temporal dependencies. AdaBoost is sensitive to noisy data and outliers. LSTM requires large datasets and higher computational resources, which may limit deployment on low-power edge devices.

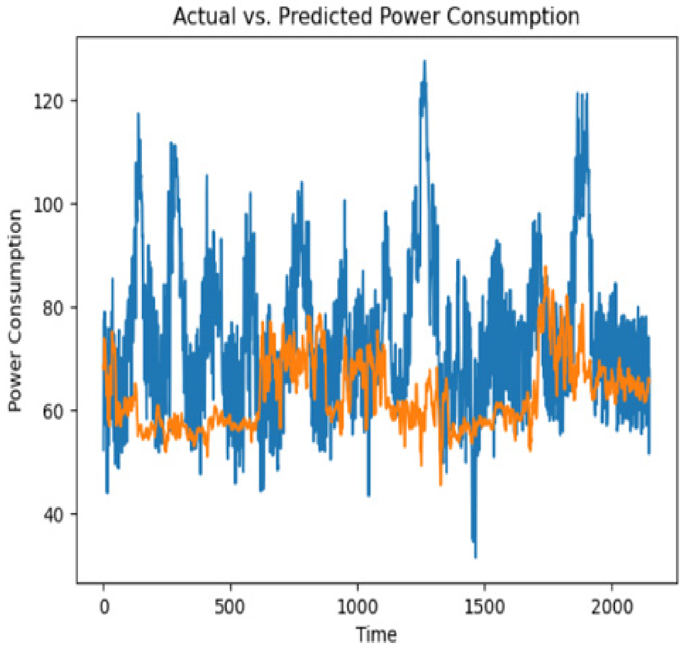

Figure. 10. *LSTM forest predictions on the test dataset.*

System Testing and Evaluation

Hardware Test

We tested the hardware device to ensure it worked perfectly. Hardware validation was performed under a regulated 220V AC supply using resistive loads from 20W to 200W. The ACS712 sensor was calibrated against a Fluke 17B+ multimeter with a mean absolute deviation of 0.018 A, while the DHT11 showed ±1.8°C temperature and ±4% humidity deviation compared to a reference thermo-hygrometer. Data were sampled every 15 seconds and transmitted via NodeMCU ESP8266 to ThingSpeak with 1.8–2.6 s latency, and the PIR sensor range was verified up to 5 m at 120°. The software architecture follows a three-layer model: perception layer such as sensors, communication layer which are WiFi and cloud API, and application layer Python-based machine learning engine. Data preprocessing and model training were implemented using Python with NumPy, pandas, scikit-learn, and TensorFlow libraries.

Links Researchers

Table 3. *Comparison of models*

Models	MAE	RMSE	Time Complexity (sec)		Space Complexity (KB)	
			Train	Prediction	Train	Prediction
Random Forest	12.06	15.86	1.5226	0.0414	0.0001	0.0140
LSTM	14.84	19.52	188.2562	12.1843	672.0312	146.7226
Adaboost	12.50	15.64	1.2621	0.0482	0.0001	0.0140

Table 4: *Test case of hardware testing and data transferred testing.*

Steps	Task	Expected Outcome	Actual Outcome
1	Sensors are calibrated properly	Pass/Fail	Pass
2	Connections of the device are configured properly	Pass/Fail	Pass
3	Verify that the PIR motion sensor is giving one on motion detection and 0 otherwise.	Pass/Fail	Pass
4	Verify that the PIR motion sensor is within its range	Pass/Fail	Pass
5	The current sensor is giving an output value of the same as the multimeter	Pass/Fail	Pass
6	Verify Wireless connectivity module in NodeMCU works accurately	Pass/Fail	Pass
7	Data from the device is transferred to ThingSpeak until connected to the internet.	Pass/Fail	Pass

Table 5: *Case of software performance*

Steps	Task	Expected Outcome	Actual Outcome
1	Verify the dataset uploading speed should be within 3-5 seconds	Pass/Fail	Pass
2	Verify predictions will be displayed within 5 seconds after uploading the dataset	Pass/Fail	Pass
4	After pressing login, details will be fetched from the database within 2 seconds.	Pass/Fail	Pass
5	The main page will load within 1 second after pressing login	Pass/Fail	Pass
6	Verify that the model will be loaded within 2-3 minutes	Pass/Fail	Pass

Software Performance Testing

Software performance testing is the process of evaluating software's functionality. This procedure assesses the system's dependability, efficiency, and accuracy.

Discussion

The proposed IPA-DS system demonstrates a comprehensive approach to smart home energy management with integrating IoT data acquisition, machine learning-based prediction, and real-time monitoring. The framework's novelty lies in its combined architecture, which incorporates structured preprocessing, correlation and K-best feature selection, and comparative evaluation of collective and deep learning models. This multi-model approach ensures that the most accurate prediction model is selected for residential energy optimization, distinguishing IPA-DS from existing methods that typically rely on single-model predictions and only provide monitoring functionality. Experimental results indicate that the system can effectively support residential energy monitoring, demand-response optimization, and anomaly detection. Hyperparameter tuning was conducted using grid search with time-series split validation, and the dataset was chronologically divided into 75% training and 25% unseen testing data to maintain temporal consistency. These measures improved model generalization and minimized overfitting, ensuring reliable predictive performance across different scenarios. Despite these strengths, deployment challenges remain. Large-scale implementation requires robust cloud infrastructure to handle streaming IoT data, and network latency may affect real-time decision making. Data privacy and cybersecurity risks must be addressed through secure communication protocols, while periodic model retraining is necessary to adapt to seasonal changes and evolving user behavior. Additionally, the system's predictive accuracy is influenced with the size and variability of the collected dataset, highlighting the need for extended data collection and potential inclusion of voltage fluctuations for more precise power estimation. Overall, the IPA-DS framework provides a practical and flexible solution for smart energy management in residential settings, combining real-time monitoring, predictive analytics, and adaptive decision-making, while outlining clear directions for addressing real-world deployment constraints.

Conclusions and Recommendations

This research concludes with the design and implementation of an IoT-based smart home energy management system integrated with machine learning for optimized energy consumption. The IoT device is equipped with DHT11, ACS712, and PIR motion sensors, controlled via NodeMCU. The software implementation includes a machine-learning model trained on a self-generated dataset to predict future power consumption, with the gathered data contributing to a comprehensive dataset. The primary objective is to enable users to make informed energy usage decisions. The system is enhance energy efficiency in smart homes and small buildings by detecting anomalies, scheduling appliances, and supporting predictive maintenance, with potential integration into demand-response programs. Deployment challenges involve cloud scalability, network latency, cybersecurity, and periodic model retraining to adapt to seasonal and behavioral changes. Our ongoing work focuses on further enhancements to expand applicability and improve system robustness. Although the proposed system demonstrates effective prediction performance, limitations include dataset size, constant voltage assumption, and limited seasonal variability. Future work will extend data collection duration and incorporate dynamic voltage monitoring for more accurate power estimation.

Acknowledgments

We acknowledge Bahria University for providing research infrastructure and computational resources. The authors declare that no external funding was received for this research.

Novelty Statement

This study proposes a novel IPA-DS algorithm for real time energy usage estimation in smart homes, enabling adaptive and data driven energy management. It improves efficiency over traditional methods by integrating real time analytics with machine learning while sustaining low computational overhead.

Author's Contribution

Moazam Ali: Conceptualization, methodology, statistical analysis, technical design and experiments implementation, writing
Israr Akhter: Research design, data acquisition, experimentation, quality assessment
Saba Mahmood: Review and editing, critical revision
Fatima Khalique: Research design, conceptualization supervision, manuscript preparation
Abrar Ahmed and Farah Akif: Literature review, review and editing, formatting

Generative AI and AI-assisted technology statement
No generative AI and AI-assisted tools were used in the preparation of this manuscript.

Conflict of interest
The authors have declared no conflict of interes.

Refrences

Ehsan, M.K. *et al*. 2021. Characterization of sparse WLAN data traffic in opportunistic indoor environments as a prior for coexistence scenarios of modern wireless technologies. Alexandria Engineering Journal, 60. https://dx.doi.org/10.1016/j.aej.2020.08.029.

Qadeer, I. and M. K. Ehsan. 2021. Improved channel reciprocity for secure communication in next generation wireless systems. Computers, Materials and Continua, 67, no. 2. https://dx.doi.org/10.32604/cmc.2021.015641.

Nawaz, M.U.S., M.K. Ehsan, A. Mahmood, S. Mumtaz, A.H. Sodhro, and W.U. Khan. 2023. Efficient resource prediction framework for software-defined heterogeneous radio environmental infrastructures. Advanced Engineering Informatics, vol. 56, https://dx.doi.org/10.1016/j.aei.2023.101976.

Ehsan, M.K. 2020. Performance analysis of the probabilistic models of ISM data traffic in cognitive radio enabled radio environments. IEEE Access, vol. 8, https://dx.doi.org/10.1109/ACCESS.2019.2962143.

IoT Based Digital Wattmeter, 2021. International Journal of Advanced Trends in Computer

Science and Engineering, vol. 10, no. 2, https://dx.doi.org/10.30534/ijatcse/2021/711022021.

Girisha, K.M., M.V. Shamitha, G.D. Shimsha, G. Thanushree. 2023. Energy monitoring device using Arduino. Int J Res Appl Sci Eng Technol, vol. 11, no. 7, https://dx.doi.org/10.22214/ijraset.2023.54524.

Rastogi, S., M. Sharma, and P. Varshney. 2016. Internet of things based smart electricity meters. Int J Comput Appl, vol. 133, no. 8, https://dx.doi.org/10.5120/ijca2016907903.

Mufassirin, M.M.M. and A.L. Hanees. 2018. Development of an IOT-based smart energy meter reading and monitoring system. in Proceedings of 8th International Symposium.

Condon, F., J.M. Martínez, A.M. Eltamaly, Y.C. Kim. 2023. Ahmed. Design and implementation of a cloud-IoT-based home energy management system. Sensors, vol. 23, no. 1, https://dx.doi.org/10.3390/s23010176.

Devi, M., S. Muralidharan, R. Elakiya, and M. Monica. 2023. Design and implementation of a smart home energy management system using IoT and machine learning. in E3S Web of Conferences, https://dx.doi.org/10.1051/e3sconf/202338704005.

Kumar, N., K. Sundaram, R. Reena, and S. Madhumathi. 2023. Optimizing energy consumption in smart homes using machine learning techniques. in E3S Web of Conferences, https://dx.doi.org/10.1051/e3sconf/202338702002.