Al-Khwarizmi Engineering Journal
ISSN (printed): 1818 – 1171, ISSN (online): 2312 – 0789
Vol. 22, No. 1, March, (2026), pp. 74–90

Al-Khwarizmi Engineering Journal

Smart Energy Management System Based on Embedded Wireless Communication

Rabee M. Hagem[1]*, Mustafa Qassab[2], and Amar Daood[3]
[1,2,3] *Department of Computer Engineering, College of Engineering, University of Mosul, Mosul, Iraq*
*Corresponding Author's Email: rabeehagem@uomosul.edu.iq

(Received 7 September 2025; Revised 1 November 2025; Accepted 29 December 2025; Published 1 March 2026)
https://doi.org/10.22153/kej.2026.12.015

Abstract

A Smart Energy Management System (SEMS), which uses the Internet of Things (IoT), has been created in response to the increasing demand for energy-efficient solutions. A cost-effective and efficient IoT-based SEMS that optimizes energy usage, reduces costs, and improves sustainability is presented in this work. The suggested solution uses smart meters, cloud-based analytics, and inexpensive sensors (total cost is less than US$20) to track and control energy use in real time. Through the use of Machine Learning (ML) algorithms and data-driven decision-making, the system can forecast future consumption trends and provide consumers with relevant data. Energy conservation is achieved through the system's affordability, which makes it appropriate for residential, commercial, and industrial applications without requiring significant infrastructure investments. The system's effectiveness in reducing energy waste while maintaining user convenience is demonstrated by experimental results. The considerable potential of IoT-based technologies in creating an economical and sustainable framework for energy management is demonstrated by this study. Furthermore, energy consumption prediction systems based on ML are presented. In addition, a scalable approach for more intelligent and environment-friendly energy management systems in future smart cities is developed using OPNET simulation for about 1000 nodes (smart meters). The findings demonstrated that boosting family classifiers achieved the best accuracy with a 98% prediction rate. When 1000 nodes are simulated, the latency for the proposed system may reach as low as 0.21 ms with approximately 20 bytes/s of network traffic, according to the network simulation results that used OPNET.

Keywords: *Embedded wireless communication; Smart meters; Smart energy management systems; IoT; Machine learning for energy optimization*

1. Introduction

Smart monitoring systems for different applications are essential [1-5]. The growing need for energy worldwide, along with the urgent requirement for sustainability and environmental protection, has led to the creation of smart systems that consume a considerable amount of energy. Many traditional energy management systems are not highly efficient. They do not allow users to interact with them frequently and in real time. Meanwhile, a Smart Energy Management System (SEMS) enables the dynamic, automated, and effective management of energy use through the application of the latest embedded technologies and wireless communication protocols.

Low-power embedded devices and wireless networks are combined in a SEMS that uses embedded wireless communication to track, evaluate, and manage energy usage in real time. This system can collect information from a variety of sensors, including temperature, voltage, and current sensors. Then, it can wirelessly send these data to cloud platforms or central controllers to make intelligent choices that can improve operational efficiency and reduce waste. Applications are used in residences, enterprises, and

This is an open access article under the CC BY license:

manufacturing facilities, but they are particularly useful for smart homes, buildings, and energy grids.

Wi-Fi, LoRa, ZigBee, and Bluetooth Low Energy represent embedded wireless communication technologies that allow for the flexible, scalable, and cost-effective implementation of SEMS. This system can work with Internet of Things (IoT) platforms and cloud computing to support predictive analytics, remote control, user feedback, and incorporation of renewable energy sources.

Previous studies have suggested IoT systems for smart energy management. However, these proposed systems lack Artificial Intelligence (AI) integration. Accordingly, this study presents the implementation of an SEMS that utilizes embedded wireless technologies. It also determines how well this system works, how much energy it can save, and security problems that it might face. Machine Learning (ML) algorithms for energy consumption prediction are implemented, along with OPNET simulation for about 1000 nodes (smart meters). The objective is to make energy use smarter and greener in future smart cities. Meters can be classified into two major types: traditional meters, which record data once a month, need to be read and billed manually, and only have local access without alerts; and smart meters, which record data every 15–30 min, send data automatically via IoT, bill in real time, send alerts automatically, and can be accessed remotely.

2. Related Work

In the last 10 years, many researchers and developers have investigated how to design and build an SEMS that uses built-in wireless communication. To maximize energy use, reduce costs, and assist in the integration of renewable resources, this system requires real-time monitoring and control. The primary contributions to this field are examined in this section, with a focus on different technologies, designs, and performance-enhancing techniques. Although SEMS has several benefits, certain problems prevent its widespread use. The vulnerability of IoT devices to hackers and data theft is one such problem. Interoperability device compatibility is affected by difficulties and a lack of standardization. An increase in the number of devices connected to a network causes scalability problems, which can lead to less efficient operation. Despite long-term cost-effectiveness, early deployment expenses are still extremely high for some users.

To give consumers a complete picture of their power usage, the authors of [6] suggested a system that incorporates tracking and billing energy meters. The initiative used IoT to provide energy companies with a detailed picture of how much energy is used in homes and businesses. Installing a mobile application on their Android smartphones allows users to keep track of their energy usage and examine their invoices, making saving energy possible at work and at home. The system can be used in homes, businesses, power plants, and offices. It provides advantages, including quicker disaster recovery, more accurate readings, and better customer support.

The primary objective of [7] was to leverage IoT technology enabling the automatic creation of invoices for residential appliances based on how much electricity is used. The decentralized nature of an energy system makes incorporating new energy sources possible.

The authors of [8] suggested an IoT-based kWh meter that will assist power companies in tracking each customer's power usage. Real-time remote monitoring and control are made feasible by the connection of the proposed IoT kWh meter to the Internet and a cloud server. To reduce operating expenses for electricity suppliers, this previous research provided a prototype that combined ESP32 and PZEM-004T. This prototype exhibits remarkable accuracy in electrical power monitoring.

In [9], a method for automating meter readings and replacing labor-intensive tasks was presented. This method allows customers to calculate how much energy their appliances consume each day and place measures to lower the amount, and thus, save energy. The suggested solution uses voice control and real-time monitoring to allow Android-based software to regulate electrical switches and devices remotely. Through a relay module, the Blynk application leverages IoT to manage and keep track of household appliances. The house has a sensor that monitors humidity and temperature. The Wi-Fi module is used to send all sensor data to Arduino.

The primary objective of [10] was to use IoT technology to measure how much energy is consumed by household appliances while enabling automated bill payments. The energy consumption of loads equipped with digital energy meters is measured using NodeMCU. The data gathered by NodeMCU will be sent to the server via the Wi-Fi module. All output load is managed by the relay module. The entire system is automated, and thus, loads can be turned on and off using a website or a smartphone application.

The primary objective of [11] was to encourage energy conservation and increase awareness on how much energy is consumed in daily life. An alarm will sound if energy usage surpasses a predetermined peak threshold that has been set in the system. In addition, the project will provide notifications about power theft, power outage, and overloading by utilizing a PIC16F877A microcontroller and a GSM module.

The authors of [12] focused on allowing the monitoring of consumers' real-time energy consumption by gathering data from their designed Smart Energy Meter (SEM). Their approach included a PZEM-016 AC energy meter, NodeMCU ESP8266, an RS-485 UART serial converter, Arduino Uno R3, an LCD Arduino keypad module shield board, and Blynk IoT application. The SEM utilizes PZEM-016 to measure voltage, current, power, frequency, power factor, and overall energy consumption. NodeMCU transmits the parameters to the Blynk IoT application. Subsequently, Blynk provides real-time measurement of the selected parameters.

The authors of [13] presented a GSM-based automatic remote meter-reading system. The readings were monitored and captured using a microcontroller. In case of customer default, sending someone from the power company to cut off or restore the client's service connection is unnecessary because this process can be completed via short messaging service. In this technology, energy meter readings are transmitted via GSM.

In [14], an SEMS was utilized in residential electric power distribution systems for smart power monitoring using NodeMCU, a PZEM sensor, and relay. Data are sent using an embedded controller with an ESP8266 module after processing to reduce energy consumption. This method benefits domestic consumers.

An affordable IoT energy monitoring system was designed and prototyped in [15]. This system can be utilized in many applications, including home automation, smart grid energy management, and power billing. Captured data are gathered from sensor nodes and transmitted to a Blynk server via the Internet by utilizing a low-cost ESP32 microcontroller coupled with noninvasive Current Transformer (CT) sensors and voltage sensors.

An Android application-based smart energy monitoring system was introduced in [16]. Real-time data collected from energy meters via the NodeMCU module were efficiently and easily displayed by the system. The application provides users access to real-time energy usage data, allowing them to monitor their household energy consumption. This previous study described how an integrated NodeMCU was used to convert analog data into digital data, which were then supplied to the Arduino controller. An Arduino Integrated Development Environment (IDE) module, NodeMCU, and a power supply were used by the Android smartphone to send and receive data.

To save human labor when measuring power consumption and increase user awareness of excessive electricity usage, the authors of [17] presented a system that combined an Android application with an IoT-based electric meter monitoring system. An electrical meter's pulse is gathered using an optical sensor and Arduino Uno. An affordable wireless sensor network is used, accompanied by a mobile application that autonomously analyses meter readings, minimizing human error and associated energy consumption costs.

The design and implementation of an IoT-based SEMS for observing and recording energy usage in residences and enterprises were detailed in [18]. This system was built to be affordable, easy to set up, and user-friendly. The approach is scalable and its implementation is suitable for small and large-scale applications.

The IoT-based SEMS presented in [19] comprised an ESP8266 Wi-Fi module to connect to the Internet, achieving ease of use and control all devices in accordance with the power they consume. This system has control over finances.

The energy consumption of residential loads are tracked automatically by the IoT-based SEMS presented in [20]. The meter is capable of sending consumption data to the consumer and the electricity supplier. The HLW8012 sensor automatically acquires readings. A predefined set of programs computes the total energy costs spent within a specified interval by using the ESP32 microcontroller. The bill is updated on a smartphone via IoT, through which the user can monitor the consumption of units at any moment.

In [21], the dataset was collected manually by recording the readings of traditional kWh meters. Measurements were taken four times per day at intervals of 6 h, with each entry including the corresponding date and time. The data cover three different categories of residential houses in Mosul City. Notably, this dataset does not involve the use of any hardware-based monitoring system, ML prediction models, smart meters, or network simulations. Furthermore, in contrast with the system proposed in the current work, which envisions large-scale data acquisition for approximately 1000 households, the aforementioned dataset is limited to manual recordings from conventional meters.

The authors of [22] suggested a three-tier edge–fog–cloud SEMS architecture in which Long Short-Term Memory (LSTM)-based federated learning is operated by smart meters. Data and model updates are protected by an advanced encryption standard-backed blockchain to guarantee secure aggregation, confidentiality, and integrity.

In [23], a design of a blockchain-enabled cyber-physical smart meter for virtual power plants was suggested. This design uses smart contracts and tamper-evident ledgers to allow for safe peer-to-peer energy trading and real-time monitoring.

Table 1 compares the aforementioned studies.

The major contributions of the current study include the following:

- Designing a cost-effective SEM that measures and monitors energy consumption in real time
- Integrating different types of ML algorithms into the proposed system for predicting energy consumption
- Designing a complete interactive framework by starting with gathering, storing, and processing data to be visualized using a friendly front-end app for customers with real-time notifications
- Implementing a comprehensive scalability evaluation by using OPNET simulation, demonstrating network criteria, such throughput and latency with different topology.

Table 1,
Comparison of related studies.

Reference Number	Hardware Design	Real-Time GUI Platform	Energy Consumption Calculation	Security	ML and Prediction	Scalability	Simulation (1000 SM)
[6]	✓	✓	✓				
[7]	✓	✓	✓				
[8]	✓	✓	✓				
[9]	✓	✓	✓				
[10]	✓	✓	✓				
[11]	✓	✓	✓				
[12]	✓	✓	✓				
[13]	✓	✓	✓				
[14]	✓	✓	✓				
[15]	✓	✓	✓				
[16]	✓	✓	✓				
[17]	✓	✓	✓				
[18]	✓	✓	✓				
[19]	✓	✓	✓				
[20]	✓	✓	✓				
Proposed work	✓	✓	✓	✓	✓	✓	✓

3. System Architecture (Hardware and Software)

The SEMS architecture is composed of five major layers: (1) the sensing layer, which gathers immediate electricity data, such as voltage and other parameters; (2) the decision layer, which processes these data by using rule-based logic and ML models; (3) the actuation layer, which regulates appliances in accordance with user preferences to reduce electricity costs; (4) the communication layer, which synchronizes data to Firebase for remote access and control; and (5) the user interaction layer, which provides real-time monitoring and manual overrides through a Flutter-based mobile application. Figure 1 illustrates the architecture of the major layers of the proposed system.

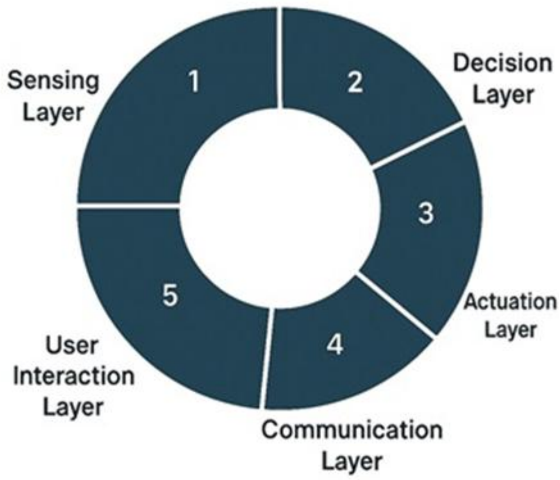


Fig. 1. Major layers of the proposed system architecture.

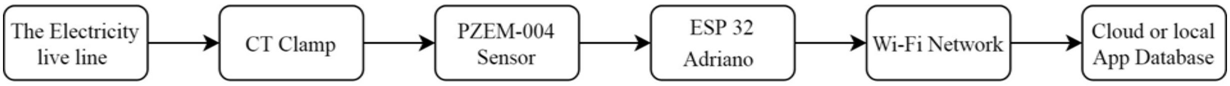


Fig. 2. Hardware design for the proposed SEMS.

The block diagram of the hardware design and components are depicted in Figure 2. A smart energy monitoring system that tracks electricity use in real time is depicted in this diagram. Without cutting the circuit, a CT clamp senses electricity flowing through the live line. After receiving the input, a PZEM-004 sensor makes monitoring a number of electrical properties, such as voltage, current, and energy, possible. These variables are obtained using an ESP32 microcontroller, which then sends them via Wi-Fi to a cloud or local application database for storage, visualization, and analysis via a Web-based or mobile interface. This setup enables users to monitor energy usage remotely and make informed decisions to improve energy efficiency. Figure 3 illustrates the top and bottom layers of SEMS Printed Circuit Board (PCB), which are designed using EasyEDA software. The box is printed using a 3D printer and designed with Tinkercad software.

The total cost of the prototype of the proposed system is less than ($20), which is divided into: ESP32 ($3), PZEM-004 sensor ($8), CT ($4), and PCB with resistors and capacitors ($2). Therefore, the prototype is affordable.

(a) (b) (c) (d)

Fig. 3. (a) Top layer of SEMS. (b) Bottom layer of SEMS. (c) Components of SEMS implementation on the PCB. (d) SEMS PCB container.

The proposed system consists of three major components: firmware (ESP32 code), Firebase real-time database, and Flutter mobile application. Sensor data are gathered every 180 s (or at a configurable interval) and sent as a JSON object format via a secure Wi-Fi network by ESP32, which is programmed using Arduino IDE. Then, data are sent to the Firebase real-time database, with automatic reattempts in case of transmission failure or disconnection. A Network Time Protocol (NTP) server ensures precise time stamping for each reading. The Firebase real-time database is selected because it allows features such as scalability, low latency, and real-time synchronization. It stores sensor data as JSON records under a sensor node, with each record containing voltage, current, frequency, and time stamp, enabling efficient querying and time-based visualization. A real-time dashboard to display information, such as energy consumption readings, charts, and event notifications, is provided by the Flutter-based mobile applications. Figure 4 shows the flowchart of the proposed system design [24].

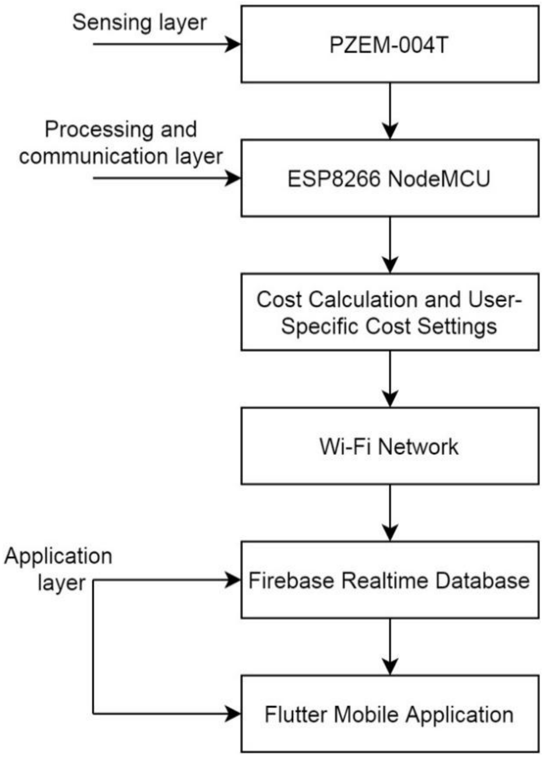


Fig. 4. Proposed design for the system.

3.1. Proposed System Workflow

ESP32 runs a periodic sensing and communication cycle, including the cloud. This cycle starts with connecting the microcontroller ESP32 to a predefined Wi-Fi network, sensor initialization with accurate timestamping, and authentication through the NTP configuration.

Firebase uses the stored credentials. ESP32 reads the sensor data every 3 min during the phase of acquiring periodic data. Then, the data are formatted into JSON object. This object is transmitted to the Firebase real-time database. The system automatically retries during the next cycle if network connectivity fails. The workflow of the proposed system is shown in Figure 5 [24]. Mobile application on the user side is developed using Flutter. It is initiated by authenticating with Firebase via Flutter Firebase SDK.

Real-time listeners are established to monitor the sensor node, and any new data trigger automatic updates to the Graphical User Interface (GUI), as shown in Figure 6. The most recent sensor readings are displayed in text format, while historical data are visualized through interactive charts, such as line graphs, for tracking energy consumption trends over time. It is calculated based on the following equations:

$$P = V\ x\ I, \quad (1)$$

$$E = P\ x\ t, \quad (2)$$

$$Energy\ Cons. = CR - PR, \quad (3)$$

$$Total\ cost = Cons.\ x\ Price\ (per\ KWh), \quad \ldots(4)$$

where P denotes power, V is for voltage, and I represents current. Similarly, E is for energy, and t is for time. In addition, CR denotes current reading, and PR denotes previous reading. Cons. is for consumption. KWh means kilowatt × hour.

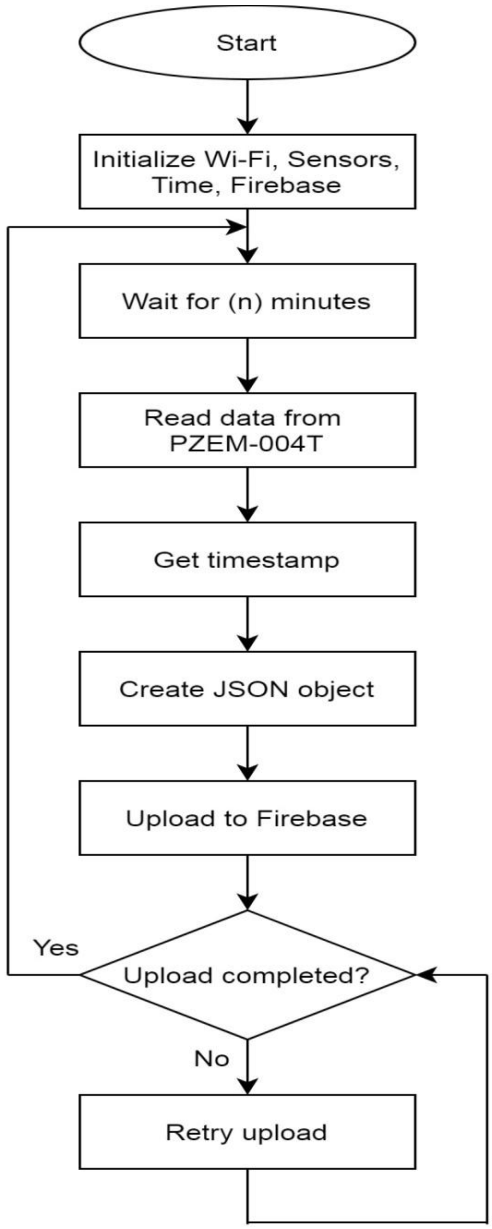
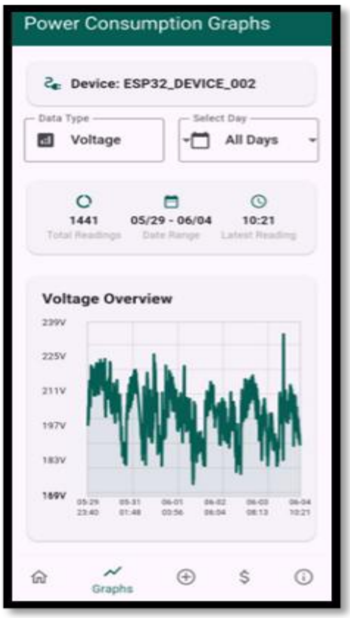
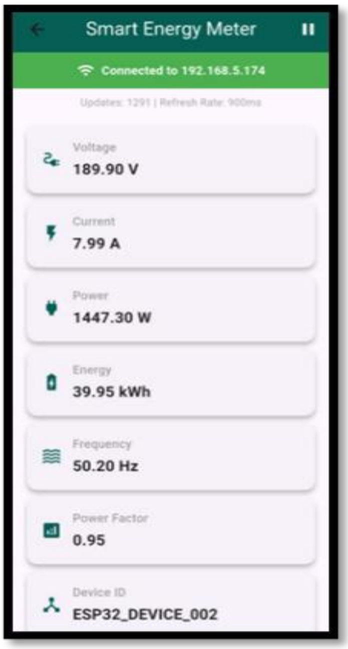


Fig. 5. Overflow of the ESP microcontroller function.



(a)



(b)

(c)

Fig. 6. (a) and (b) Samples of reading SEM data. (c) Logo of the GUI platform for SEM.

3.2. ML Implementation

Recently, AI and ML methods have been used in different types of applications [25-29]. Therefore, AI and ML techniques may provide effective solutions to overcome various challenges in our daily lives. One of the most primary aspects in modern daily life is electricity consumption, which ranges from simple human behavior, such as lighting or heating, to more sophisticated tasks, such as industrial operations, communication, and transportation. The use of electronic devices and automated systems have increased widely in our societies due to human needs. Such increase demands an effective mechanism to save energy consumption. Meanwhile, global warming and climate change have motivated individuals and governments to invent intelligent techniques to manage and optimize electrical consumption. Hence, researchers have widely addressed the management of electricity usage to provide essential methods for reducing costs and minimizing power consumption. Pattern identification and future prediction of electrical usage can be implemented using AI methods to manage electrical usage by examining consumers' data records.

Therefore, our project pipeline started by contacting the electrical department of Mosul City to collect real data from users' consumption. Thereafter, we obtain access to a dataset of more than 1000 consumers, with their average consumption for 1 year. Data cleaning and preprocessing were applied to eliminate noisy and redundant measurements. Then, a clustering process was implemented to label dataset entries automatically into two classes: normal class for reasonable consumption and abnormal class for high consumption. Clustering consumers into two classes in accordance with electrical usage can be an effective solution to managing energy consumption. An in-depth understanding of usage patterns, including identifying peak demand and guiding the creation of strategic intervention strategies to encourage energy saving, may be gained by classifying users into various consumption tiers. For example, a reward reinforcement scheme can encourage regular customers to maintain their existing energy usage levels. Conversely, anomalous users who exhibit excessive energy use can be directed toward methods that can reduce their usage habits, such as personalized recommendations and incentive schemes to lower their costs. In addition, unique consumers can be motivated to change their views and electricity use by implementing sustainable consumption and renewable energy sources.

Our dataset has two types of consumption levels: normal and irregular. Temperature, humidity, wind speed, average historical consumption, and electricity use are all included in the dataset. Accurate classification can be improved by incorporating environmental elements, such as wind, humidity, and temperature. A strong association exists between energy consumption patterns and the inclusion of climatic parameters, especially with regard to heating, ventilation, and air-conditioning systems.

Thereafter, two parts were created from our dataset. The first section of our data was used to train different ML algorithms. Then, we used the second section of the dataset to measure the performance of the proposed system and assess the efficiency of the trained models. We used a cross-validation technique to evaluate the trained model after the learning process. A 10-fold cross-validation analysis was utilized, wherein we repeated the random splitting of data 10 times. At each random split, we used the training part of the data in the training phase to learn the prediction models. In the testing phase, the testing dataset was used to evaluate the efficiency of the classification model. In the end, average performance was calculated by combining the results of the 10 experiments.

A wide range of classification techniques are available. Each of these methods may provide a set of unique characteristics, strengths, and weaknesses. Therefore, selecting the best classification method for a particular problem is extremely challenging. To address this issue, a broad spectrum of classification methods was implemented by applying the training process by using our dataset, which included approximately 18 types of classification models. Model training of these methods is an essential component of the proposed system to recognize customers into two classes: normal and abnormal electricity consumption patterns. The implemented classification models can be categorized into linear and nonlinear algorithms. For example, using the training part of the dataset, we trained support vector machine, linear discriminant analysis, and logistic regression algorithms as linear models. Furthermore, we trained decision trees, quadratic discriminant analysis, and *k*-nearest neighbors as nonlinear models. The block diagram of classification model implementation is provided in Figure 7.

Alternatively, the implemented classification algorithms in our study can be grouped into two types: single classifier and ensemble classifier approaches. All the previously discussed classification algorithms are considered a single classifier learning technique. By contrast, an ensemble of learning techniques combines multiple classifiers and merges their results to achieve better predictive performance. As part of our ensemble technique, we trained the extra trees, random forest, and AdaBoost classifiers as an ensemble of approaches.

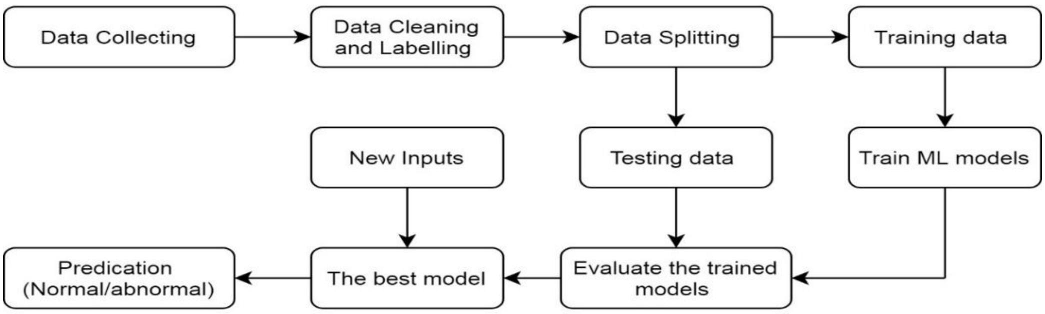


Fig. 7. Block diagram of ML implementation.

3.2.1. Results of ML Algorithms

In this section, we present all the experiments conducted in the proposed system to identify the most appropriate classification algorithm and create the optimal model for developing the core prediction model that is responsible for detecting abnormal consumption patterns. As mentioned previously, the selected ML classification was trained using a 10-fold validation method, dividing the dataset into 70% for training and 30% for testing. Table 2 presents a comprehensive summary of all the experiment results conducted in our study. For a comprehensive examination, we calculated metrics for accuracy, recall, precision, and F1-score. In addition, the execution time of the trained classification model was measured to evaluate the speed metric and real-time response.

As indicated in Table 2, the trained classifiers were sorted from highest to lowest accuracy. Boosting family classifiers achieved the highest accuracy, with a prediction rate of 98% due to the nonlinear decision boundary provided by this classifier, producing a perfect fit for our real-world data. In addition, the confusion matrix of the trained model was computed, as shown in Figure 8. To conduct a comprehensive analysis, we calculated the importance of each feature in our dataset. Feature importance can provide the significant value of the features in our dataset and determine their rank. The results of this analysis are presented in Figure 9.

Table 2,
Results of comparison.

Model	Accuracy	Recall	Precision	F1-score	Execution time
CatBoost Classifier	0.9889	0.9880	0.9879	0.9873	3.6560
Extreme Gradient Boosting	0.9886	0.9874	0.9873	0.9872	0.0940
Gradient Boosting Classifier	0.9883	0.9866	0.9866	0.9863	0.7920
MLP Classifier	0.9880	0.9854	0.9854	0.9842	5.0570
Ada Boost Classifier	0.9880	0.9843	0.9838	0.9831	0.2290
Light Gradient Boosting Machine	0.9874	0.9660	0.9639	0.9588	0.8430
Decision Tree Classifier	0.9866	0.9586	0.9555	0.9436	0.0530
Extra Trees Classifier	0.9854	0.9557	0.9480	0.9381	0.2370
Random Forest Classifier	0.9843	0.9500	0.9025	0.9256	0.8070
K-Nearest Neighbors Classifier	0.9660	0.9500	0.9025	0.9256	0.1190
Naive Bayes	0.9586	0.9500	0.9025	0.9256	0.0560
Quadratic Discriminant Analysis	0.9557	0.9500	0.9025	0.9256	0.0380
Logistic Regression	0.9500	0.9500	0.9025	0.9256	0.0530
SVM - Linear Kernel	0.9500	0.9500	0.9025	0.9256	0.0390
SVM - Radial Kernel	0.9500	0.9880	0.9879	0.9873	0.2830
Gaussian Process Classifier	0.9500	0.9874	0.9873	0.9872	16.2290
Ridge Classifier	0.9500	0.9866	0.9866	0.9863	0.0360
Linear Discriminant Analysis	0.9500	0.9854	0.9854	0.9842	0.0350

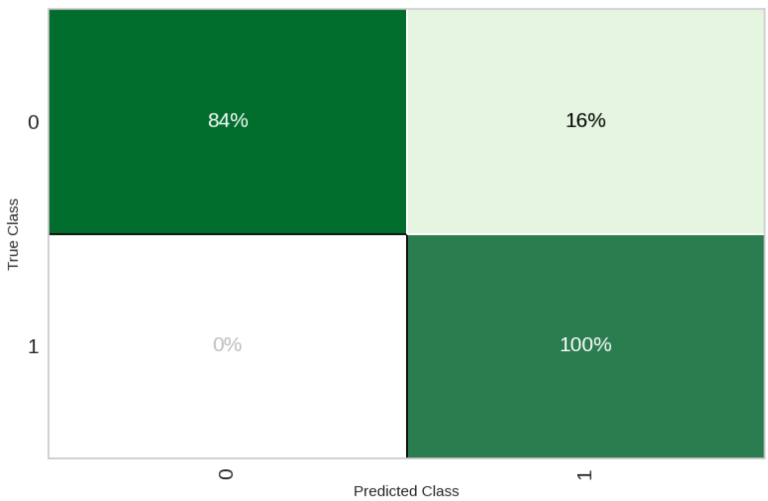


Fig. 8. Confusion matrix of the trained model.

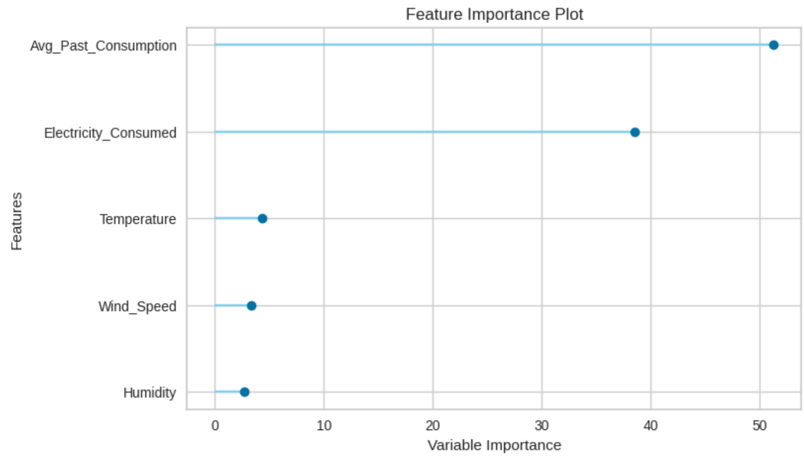


Fig. 9. Significance of the features in the dataset.

4. Simulation Approach by Using OPNET

To obtain approximate figures with regard to system connectivity and networking, the OPNET simulator is used.

4.1. Simulation Setup by Using OPNET

To evaluate and compare the effectiveness of the suggested communication techniques for smart meter data transmission, a simulation model was created using the OPNET simulation platform. A thorough performance evaluation of important characteristics, such as latency, throughput, packet delivery ratio, and scalability, is made easier by using tools such as OPNET to simulate these network designs. From heavily populated urban regions to rural areas, these simulation studies provide crucial information for choosing the best communication strategy for specific deployment circumstances [30].

Only two communication methods, direct WiMAX connectivity for each smart meter and the Wi-Fi network via home infrastructure, were used to assess their viability and performance for the suggested smart meter system due to tool and modeling restrictions.

Although it was not simulated in OPNET, a third model, namely, dedicated Wi-Fi infrastructure, was provided for theoretical comparison. This scenario was not replicated in OPNET. Nevertheless, it implies that smart meters connect directly to strategically located Access Points (APs) owned by the power company. It eliminates dependency on home networks but requires significant infrastructure deployment.

All scenarios assumed mains-powered meters with low data transmission requirements.

4.1.1. OPNET Configuration per Scenario

Scenario 1: Wi-Fi Network via Household Infrastructure

Meters formed a multi-hop hybrid star network that relays packets across clusters. The structure consisted of 1 super cluster (1,000 houses), 10 large clusters (100 houses each), and 10 subclusters (10 houses each). Various Wi-Fi protocols with different data rates were tested. Figure 10 provides additional details.

Scenario 2: Direct WiMAX Connection

Each smart meter was equipped with a WiMAX modem that transmitted data to a base station. The network operated in a star topology with cellular uplinks and centralized cloud access.

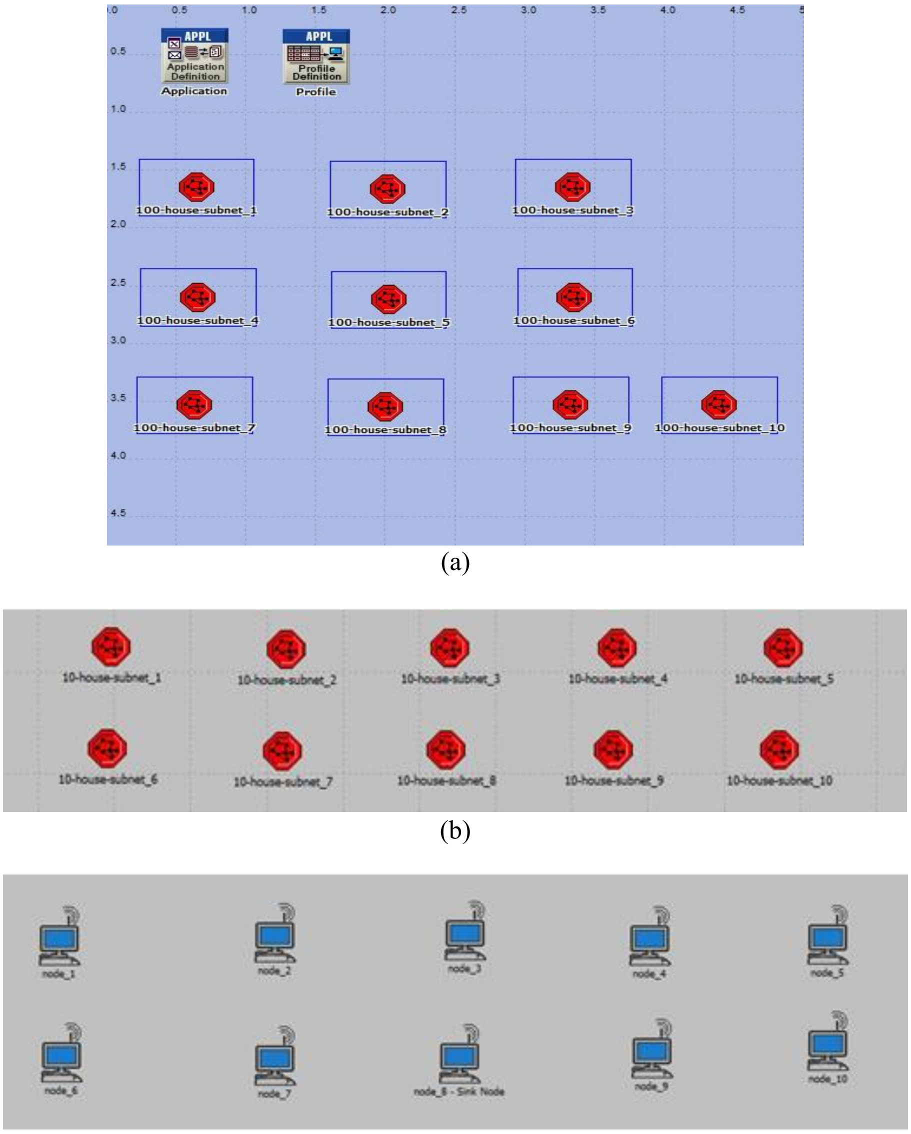


Fig. 10. OPNET simulation representation of clusters: (a) 100 node clusters, (b) 10 node clusters, and (c) 1 cluster of 10 nodes.

4.2. Simulation Settings

OPNET offers numerous options to configure any required network. These settings are adopted for the simulation. Table 3 provides details of the simulation parameters.

Table 3,
Summary of simulation parameters.

	Parameter	Value
	Number of Nodes	1000
	Area	5 km × 5 km
	Message Size	8 bytes
	Message Rate	1 message/h
	Simulation Duration	1 week
	Wireless Protocol	Wi-Fi (Scenario 1) WiMAX (Scenario 2)
Wi-Fi Settings	Protocol	802.11 b (1 Mbps) 802.11 b (5.5 Mbps) 802.11 b (11 Mbps) 802.11 g (54 Mbps) 802.11 a (54 Mbps)
	Topology	Hybrid (multi-star)
WiMAX Setting	Cluster Size	10 nodes
	Scheduling Type	rtPS
	Maximum Traffic Rate	5 Mbps
	Minimum Traffic Rate	1 Mbps
	Service Class	Gold
	Modulation and Coding	64 QAM 3/4

4.3. Simulation Results from OPNET

The OPNET simulation produced consistent metrics for all tested Wi-Fi and WiMAX variants. Given the simulation tool's constraints and lack of support for newer protocols, the results for scenarios that used newer connection protocols, e.g., 3G Universal Mobile Telecommunications Service (UMTS) and 4G Long-Term Evolution (LTE), were mathematically calculated using specific models and included in the figures below to provide a wider insight and comparison. For 1000 nodes in simulation, latency achieves the highest value of 9.86 ms by using WiMAX and 0.21 ms when 802.11a is used, as shown in Figure 11. Furthermore, the actual average throughput per smart meter across all protocols was 17.78 bps (based on 8 bytes/hour per node × 1000 nodes). However, network traffic varied, as shown in Figure 12. The message overhead ratio was also considered, and the results showed that although WiMAX had the highest latency, it exhibited the lowest protocol overhead value, as shown in Figure 13, while 3G UMTS added the most overhead among the others. With regard to packet loss, its value can be neglected across all protocols because it was <1%, and thus, statistically insignificant. Finally, jitter was not calculated due to the low transmission frequency (1 packet per hour), making delay variation irrelevant.

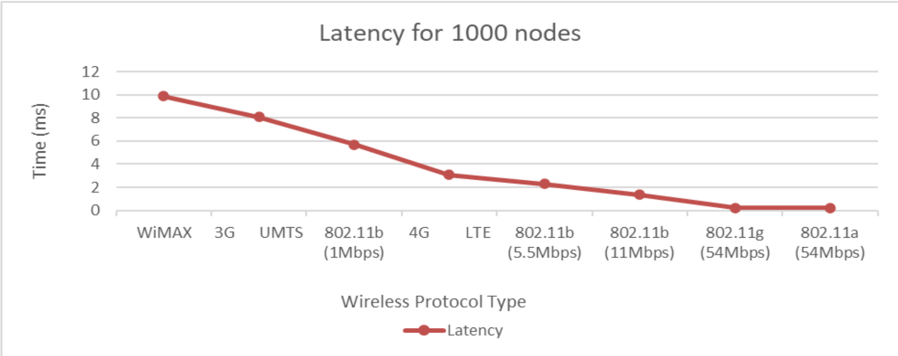


Fig. 11. Latency vs. wireless protocol type.

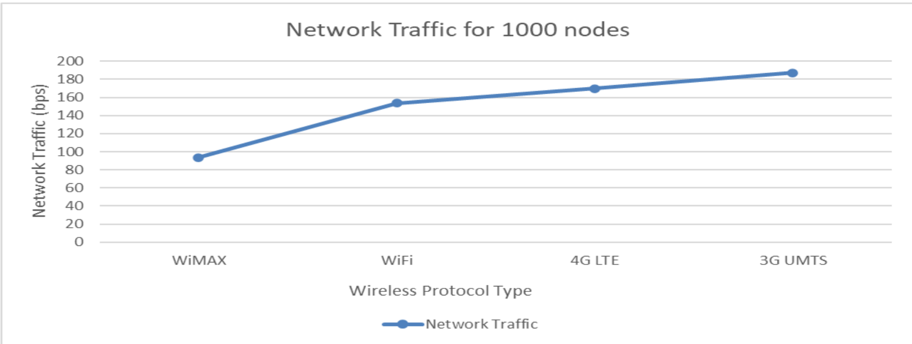


Fig. 12. Network traffic vs. wireless protocol type.

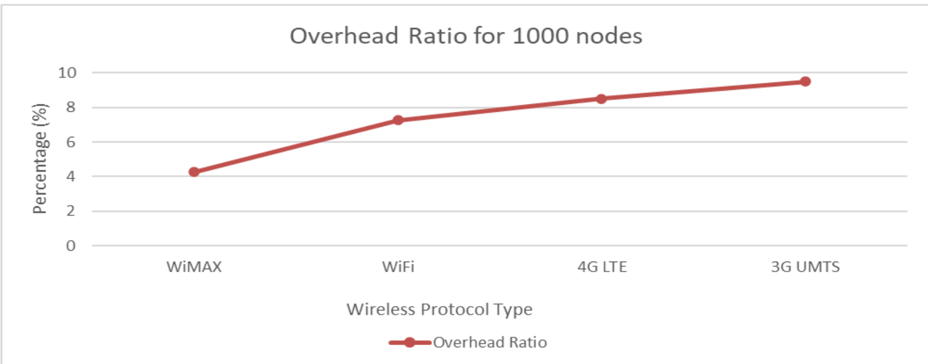


Fig. 13. Overhead ratio vs. wireless protocol type.

4.4. Network Discussion and Comparative Analysis of the Proposed System

The simulation confirms that Wi-Fi and WiMAX networks are technically capable of handling smart meter communication at low data rates (e.g., 8 bytes/h). Table 4 provides a comparative analysis.

Despite the vast difference in nominal data rates (1–54 Mbps), the actual required throughput was extremely low due to the infrequent transmission schedule. Consequently, all scenarios comfortably supported the load. However, factors, such as cost, infrastructure control, and power consumption, became more decisive in selecting the optimal method.

Table 4,
Comparative analysis of the studied network scenarios.

Network Type	Pros	Cons
Wi-Fi Mesh via Households (Scenario 1)	Achieved similar throughput (17.78 bps per node) with negligible packet loss, despite lower-cost infrastructure. Multi-hop transmission extends coverage without the need for centralized infrastructure.	Relies on users' home networks, introducing variability into reliability and security. Routing complexity and energy costs for relay nodes is also a challenge.
WiMAX (Scenario 2)	Provides wide-area coverage, reliable delivery, and consistent latency	Requires individual SIM cards and data subscriptions, resulting in high operational costs. LTE modules also consume more power, affecting battery life.
Dedicated Wi-Fi Infrastructure (Scenario 3)	Offers complete control over connectivity, security, and bandwidth management. Avoids dependence on customer infrastructure.	High deployment and maintenance costs. Not simulated, but expected to perform similarly or better than Scenario 1 under controlled conditions.

5. Security Concerns

In IoT-based SEMS, especially one that uses built-in wireless communication, keeping the system secure is crucial for protecting user data, maintaining privacy, and ensuring that service is always available [31]. The suggested system uses devices, such as PZEM-004T sensors, ESP32 microcontrollers, Wi-Fi networks, and cloud computing platforms (e.g., AWS and Firebase). To guarantee their safe and dependable operation, these components must be fixed for a number of inherent vulnerabilities.

5.1. Possible Threats

The suggested system involves a number of serious weaknesses in security.

- Unsecured communication channels: Unauthorized parties may intercept data sent over Wi-Fi between sensors, microcontrollers, and cloud databases if it is not sufficiently encrypted.
- Unauthorized access: Smart meters and control nodes may be hacked if weak or default passwords are used, possibly resulting in data breaches or unauthorized device manipulation.
- ML data poisoning: The reliability of the ML classifier that differentiates between normal and abnormal energy use can be reduced by adversarial data injections.
- Cloud vulnerabilities: You become dependent on the security protocols of third party systems, such as Firebase or AWS, when you store real-time consumption data on them. Data breaches or limitations on user functionality can arise from compromised or incorrectly configured cloud storage.
- Physical tampering: Attackers may physically access devices to remove firmware, credentials, or limit functionality if they are located in inaccessible areas.
- Device cloning or spoofing: Without authentication steps, an attacker can imitate a network node or duplicate a device. Attackers are able to introduce inaccurate data as a result.
- Firmware exploitation: The system may be exposed to remote attacks if ESP32 firmware is not updated or if out-of-date libraries are being used.

5.2. Suggested Security Model

With regard to the potential attacks and threats that the proposed system might encounter, the recommended countermeasures are detailed in Table 5.

Table 5,
Possible threats and corresponding countermeasures.

Threat Name	Threat Type	Countermeasure	Justification
Unsecured Communication	Man-in-the-Middle Attack	Apply TLS/SSL encryption between ESP32 and cloud services	Ensures data confidentiality and integrity
Unauthorized Access	Credential Theft	Enforce strong passwords and 2FA in the app interface	Prevents unauthorized login attempts and weak credential exploitation
ML Data Poisoning	Model Inaccuracy	Validate incoming data and use outlier detection filters	Ensures training and inference data remain reliable
Cloud Vulnerability	Data Breach	Configure secure access rules and data encryption in Firebase/AWS	Protects user data stored remotely
Physical Tampering	Hardware Intrusion	Deploy tamper-evident seals and secure enclosures	Detects or prevents unauthorized physical access
Device Spoofing	Identity Impersonation	Use unique device tokens and mutual authentication	Distinguishes legitimate devices from rogue ones
Firmware Exploitation	Outdated Firmware	Regularly update ESP32 firmware and use signed binaries	Blocks common firmware-based exploits

5.3. Recommended Practices

Best practices for IoT SEMS security:

- Encrypt all sensitive data when they are stored (in the database) and when they are sent over the network.
- Set up a secure boot on ESP32 to ensure that only verified firmware can run.
- Use an Intrusion Detection System (IDS) on cloud platforms to monitor traffic that looks suspicious.
- Only collect the data you need (e.g., energy in kWh only) to reduce exposure surface.
- Use MQTT over TLS instead of plain HTTP to keep communication between smart meters and the cloud separate.
- Encourage users to keep their login information private and to update the application.

The suggested SEMS will be more resilient to physical and cyber attacks by including multilayer security measures, providing a solid basis for smart city infrastructure.

6. Conclusions and Future Work

With its built-in wireless connectivity, the suggested SEMS provides a practical and efficient way to improve energy use in modern applications. To provide real-time monitoring, control, and data-driven decision-making, the suggested system integrates embedded microcontrollers with wireless communication technologies at a reasonable cost. This integration improves system reliability and saves energy. The amount of energy that will be used is calculated and estimated using various ML methods. The proposed system can develop and change to fit different situations, from homes and businesses to integrating renewable energy, due to its scalability and flexibility. Overall, this solution not only supports sustainable energy practices but also paves the way for smarter, more connected energy infrastructure. Network simulation results that used OPNET showed that the latency for the proposed system could be as low as 0.21 ms with approximately 20 bytes/s of network traffic when 1000 nodes are simulated. Boosting family classifiers achieved the highest accuracy, with a 98% prediction rate.

The major limitations of the proposed system are as follows: the proposed design is not stand-alone because it is cloud-based due to the prediction model of ML being located on a cloud server. In addition, the proposed hardware design depends on ESP32, which uses mostly Wi-Fi for communication, making it network-dependent. Furthermore, the simulation was conducted to measure network performance only, no real security tests were performed, making the system vulnerable to security threats. These limitations can be addressed in future research.

Future work can focus on adding more sensors for environmental monitoring (temperature, light, and motion). These sensors can help reduce energy consumption. The system is scalable, and thus, these sensors and more can be added to the system. Although this work utilizes cloud-based ML for prediction, edge AI integration can be adopted by deploying lightweight, on-device ML models (e.g., TinyML), which can enable faster inference, reduced latency, and offline decision-making in cases of connectivity issues. Edge-based intelligence will also enhance privacy by minimizing the transmission of raw data to the cloud. In addition, security and privacy reinforcement could explore blockchain-based authentication,

Federated learning for privacy-preserving model training and AI-driven IDS tailored to IoT energy networks can be adopted. Moreover, renewable energy integration is the future for SEMS deployment, because it can incorporate renewable sources (e.g., solar and wind) and local energy storage systems. Finally, conducting more real experiments can be helpful in gathering additional data. Comparing the percentage error between traditional and smart meters reading will be helpful to validate the proposed system.

Conflict of Interest

No conflict of interest has been declared for authorship, research, and/or publication.

Funding

No funding or financial assistance was received from any organizations or industries in conducting this research.

References

[1] R. M. Hagem, D. V. Thiel, S. G. O Keefe, N. Dahm, A. Stamm, and T. Fickenscher, "Smart optical wireless sensor for real time swimmers feedback," in SENSORS, 2012 IEEE, 2012, pp. 1-4.
[2] R. Hagem, D. Thiel, S. O'Keefe, and T. Fickenscher, "Real-time swimmers' feedback based on smart infrared (SSIR) optical wireless sensor," Electronics Letters, vol. 49, pp. 340-341, 2013.
[3] B. M. H. Alhafidh, R. M. Hagem, and A. I. Daood, "Face detection and recognition techniques analysis," in 2022 International Conference on Computer Science and Software Engineering (CSASE), 2022, pp. 265-270.
[4] B. M. H. Alhafidh, R. M. Hagem, M. Qassab, M. B. Mahmood, "Privileging and Prioritizing Processes of Sustainable Energy Resources for Residential Loads to Energize Green Cities," Al-Rafidain Engineering Journal (AREJ), vol. 30, 2025.
[5] M. B. Mahmood and J. M. Abdul-Jabbar, "Enhancing Industrial Internet of Things Availability and Reliability Using Dual-Communication Links Mechanism Based on the OPC UA Protocol," Architecture, vol. 20, p. 21.
[6] D. K. Ashwini Rathod, Gauravi Shetty, Prof. Shaista Khanam, "Smart Energy Meter," International Journal of Engineering Research & Technology (IJERT), vol. 9, pp. 281 - 285, 2020.
[7] N. Sulthana, N. Rashmi, N. Prakyathi, S. Bhavana, and K. S. Kumar, "Smart energy meter and monitoring system using IoT," International Journal of Engineering Research & Technology, vol. 8, pp. 50-53, 2020.
[8] K. A. Yasa, I. M. Purbhawa, I. M. S. Yasa, I. W. Teresna, A. Nugroho, and S. Winardi, "IoT-based Electrical Power Recording using ESP32 and PZEM-004T Microcontrollers," Journal of Computer Science and Technology Studies, vol. 5, pp. 62-68, 2023.
[9] A. T. Jambhulkar, A. M. Bhoyar, C. M. Shivankar, P. S. Balwad, and P. Nandankar, "IoT-Based Smart Energy Meter for Monitoring Home Appliances," International Journal, vol. 8, pp. 1-5, 2023.
[10] P. R. S. Kalyana Kiran Kumar M Srinivasa Rao, P Murali Kirshna, P Santhoshi, "Smart energy metering and load controlling by using internet of things," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, pp. 571 - 576, 2019.
[11] P. M. Niharika Banerjee, K. Surendhirababu, K. Venkatasubramani, "Enhanced Smart Energy Meter using IoT," International Journal of Novel Research and Development, vol. 7, 2022.
[12] C. D. A. Alan P. Nebrida, Clyed M. Madiam, Glenn John S. Ranche, Niel Carlo P. Nieves, "Arduino Based Smart Energy Meter," International Journal of Progressive Research in Science and Engineering, vol. 4, 2023.
[13] H. Z. Iqbal, M. Waseem, and T. Mahmood, "Automatic energy meter reading using smart energy meter," in Int. Conf. Eng. Emerging Technol, 2014.
[14] N. L. Muralidharan P, Aishwarya U, "IoT Based Smart Energy Meter for Power Monitoring System Using ESP8266," International Research Journal of Engineering and Technology (IRJET), vol. 9, 2023.
[15] P. S. Macheso and D. Thotho, "ESP32 Based Electric Energy Consumption Meter," International Journal of Computer Communication and Informatics, vol. 4, pp. 23-35, 2022.
[16] R. Malathi, "Smart Energy Metering System using Android Mobile Application," 2020.

[17] M. R. Islam, S. Sarker, M. S. Mazumder, and M. R. Ranim, "An IoT based real-time low cost smart energy meter monitoring system using android application," arXiv preprint arXiv:2001.10350, 2020.
[18] P. J. Siddhesh Barkale, Sanket Bagul, Shivnath Bhosale, "IoT Based Smart Energy Metering with Blynk Application," International Journal of Trend in Scientific Research and Development (IJTSRD), vol. 7, 2023.
[19] T. S. As Vishal Kumar, Abu Farhan, "IoT Based Smart Energy Meter," International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE), vol. 8, 2021.
[20] S. Gadekar, M. Pimple, S. Thopate, and A. Nikam, "Iot based smart energy meter using esp 32," in Proceedings of the 3rd International Conference on Communication & Information Processing (ICCIP), 2021.
[21] M. H. A. Dhafir A. Alneema, "Energy Management and Cost Reduction: A Predictive Approach to Monthly Electricity Billing in Mosul Residential Settings," Al-Rafidain Engineering Journal (AREJ), vol. 30, pp. 151-162, 2025.
[22] M. K. Hasan, S. R. Kabir, S. Islam, S. Abdullah, M. A. Khan, G. B. Brahim, et al., "Edge Smart Meter Based LSTM Federated Learning with AES Cryptographic Blockchain for Smart Grid AIoT Networks," IEEE Communications Standards Magazine, 2025.
[23] A. D. Kaif, K. S. Alam, S. K. Das, G. Chen, S. Islam, and S. Muyeen, "Blockchain-Integrated Cyber-Physical Smart Meter Design and Implementation for Secured Energy Trading in Virtual Power Plants," IEEE Transactions on Automation Science and Engineering, 2025.
[24] I. Fathi, "An IoT-Based Low-Cost Smart Greenhouse Monitoring System Using ESP8266 and Firebase for Real-Time Environmental Control," International Journal of Advanced Natural Sciences and Engineering Researches, vol. 9, pp. 268–286, 2025.
[25] A. S. Abdulaziz, A. Dawood, and A. Daood, "Speaker Identification and Verification Using Convolutional Neural Network CNN," Tikrit Journal of Engineering Sciences, vol. 32, pp. 1-13, 2025.
[26] S. R. Awad, A. I. Daood, and A. A. Dawood, "Scalable and Efficient Multi-Class Brain Tumor Classification with a Compact Hybrid Deep Learning Model for Real-Time Applications," ARO-The Scientific Journal of Koya University, vol. 13, pp. 162-174, 2025.
[27] R. M. Hagema, T. Haelsig, S. G. O'Keefe, A. Stamm, T. Fickenscher, and D. V. Thiel, "Second generation swimming feedback device using a wearable data processing system based on underwater visible light communication," Procedia Engineering, vol. 60, pp. 34-39, 2013.
[28] A. Daood, E. Ribeiro, and M. Bush, "Pollen Recognition Using Multi-Layer Feature Decomposition," in FLAIRS, 2016, pp. 26-31.
[29] R. M. Hagem, M. Qassab, and B. M. H. Alhafidh, "IoT-Based Low-Cost Wearable Interactive Wireless Embedded Communication System for Health and Sport Applications," Kufa Journal of Engineering, vol. 16, 2025.
[30] M. A. e. al., "Performance evaluation of OPNET simulation for smart grid networks," Int. J. Distrib. Sens. Netw., vol. 16, pp. 1–14, 2020.
[31] S. L. Qaddoori and Q. I. Ali, "An efficient security model for industrial internet of things (IIoT) system based on machine learning principles," Al-Rafidain Engineering Journal (AREJ), vol. 28, pp. 329-340, 2023.

نظام إدارة الطاقة الذكية القائم على الاتصال اللاسلكي المدمج

ربيع موفق حاجم[1]*، مصطفى سهام عبد الرحمن[2]، عمار ادريس داؤود[3]
[١،٢،٣] *قسم هندسة الحاسوب، كلية الهندسة، جامعة الموصل، الموصل، العراق*
*البريد الالكتروني:rabeehagem@uomosul.edu.iq

المستخلص

أدى الطلب المتزايد على الحلول الموفرة للطاقة إلى تطوير أنظمة إدارة الطاقة الذكية (SEMS) التي تعتمد على تقنيات إنترنت الأشياء (IoT). لذلك، تم في هذه الورقة تصميم نظام إدارة طاقة ذكي وفعّال، وقليل التكلفة (أقل من ٢٠ $) قائم على إنترنت الأشياء بهدف تحسين استهلاك الطاقة، وتقليل التكاليف، وتعزيز الاستدامة. يتكامل النظام المقترح مع حساسات منخفضة التكلفة، وعدادات ذكية، وتحليلات سحابية لمراقبة استهلاك الطاقة والتحكم فيه في الوقت الحقيقي. ومن خلال استخدام خوارزميات التعلم الآلي (ML) واتخاذ القرارات بناءً على البيانات، يمكن للنظام التنبؤ بأنماط الاستهلاك، وتقديم رؤى عملية للمستخدمين. تجعل تكلفة النظام المنخفضة مناسبًا للاستخدامات السكنية والتجارية والصناعية، مما يعزز من الحفاظ على الطاقة دون الحاجة إلى استثمارات كبيرة في البنية التحتية. وتُثبت النتائج التجريبية فعالية النظام في تقليل الهدر في الطاقة مع الحفاظ على راحة المستخدم. وتُبرز هذه الدراسة الإمكانات الكبيرة لتقنيات إنترنت الأشياء في إنشاء إطار عمل لإدارة الطاقة يكون مستدامًا ومتاحًا للجميع. كما تم تقديم خوارزميات تعلم آلي للتنبؤ باستهلاك الطاقة، باستخدام محاكي OPNET لمحاكاة ما يقارب ١٠٠٠ عقدة (عدادات ذكية)، بهدف المساهمة في حل قابل للتوسع نحو استخدام أكثر ذكاءً واستدامة للطاقة في مدن المستقبل الذكية. وأظهرت النتائج أن خوارزميات التصنيف من نوع Boosting حققت أعلى دقة، بنسبة توقع بلغت ٩٨٪. كما أظهرت نتائج المحاكاة الشبكية باستخدام OPNET أن زمن الاستجابة للنظام المقترح قد يصل إلى ٠,٢١ ميلي ثانية، مع مرور بيانات بمعدل يقارب ٢٠ بايت/ثانية عند محاكاة ١٠٠٠ عقدة.