e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:05/May-2025 Impact Factor- 8.187 www.irjmets.com

IOT –BASED SMART PLUG ENERGY MONITORING SYSTEM

Sriram.R*1, Ms.Dharani.A*2

*1Final Year Post Graduate Student, Department Of Computer Applications, Dr. M.G.R. Educational And Research Institute, University In Chennai, Tamil Nadu, India.

*2Assistant Professor, Department Of Computer Applications, Dr. M.G.R. Educational And Research Institute, University In Chennai, Tamil Nadu, India.

ABSTRACT

The smart plug allows users to measure real-time power consumption, voltage, and current of connected appliances through sensors. The collected data is transmitted via WiFi to a cloud platform or mobile app for live monitoring. In areas with poor internet connectivity, GSM (via SIM800/SIM900 module) provides an alternative control method through SMS commands, ensuring continuous accessibility. The system also includes overload protection features and allows scheduling of device operations to optimize power usage.

I. INTRODUCTION

With the growing demand for energy efficiency and automation in modern households and industries, smart energy management systems play an increasingly important role. The integration of the Internet of Things (IoT) into ordinary appliances has enabled real-time monitoring, remote control, and energy efficiency. One such breakthrough is the IoT-based smart plug, which allows users to easily and intelligently measure and regulate the power consumption of individual electrical equipment.

This project aims to create a Smart Plug Energy Monitoring and management System that employs both WiFi and GSM communication technologies to enable consistent access and management, independent of internet availability.

II. LITERATURE REVIEW

The application of IoT for energy usage monitoring has been the subject of numerous studies. IoT-based solutions can assist in tracking energy consumption at the appliance level in real-time, giving consumers insights that may result in behavioral adjustments and less energy waste (Patel et al., 2020).

Microcontrollers, sensors, and cloud services for data visualization are frequently included in these systems. R. Patel and associates (2020). "Real-Time Energy Monitoring System Based on IoT." Engineering Research International.

It has been determined that smart plugs are an affordable way to monitor individual appliances. A smart plug with a relay is described in research by Sharma and Saini (2019).energy monitoring integrated circuit that can regulate and gauge the power usage of connected devices. Their results lend credence to the notion that these devices can cut down on wasteful electricity use, particularly in homes.

Saini, R., and Sharma, V. (2019). "Smart Plug Design and Implementation Based on IoT." Power Electronics and Energy Systems Journal.

For effective data transport, Internet of Things energy monitoring systems make use of protocols including MQTT, HTTP, and CoAP. MQTT provides low latency and power efficiency for smart home applications, as [Ghosh et al., 2021] showed. According to [Ali et al., 2020], cloud solutions such as Firebase and AWS IoT enable scalable and secure data management.

S. Ghosh and associates (2021). "IoT Communication Protocols: A Comparative Analysis for Home Automation" IEEE Access. M. Ali and associates (2020). "IoT Framework for Smart Energy Management Based on the Cloud." Journal of Sensors.

III. PROPOSED SYSTEM

The suggested system is an Internet of Things (IoT)-based smart plug that uses both WiFi and GSM technologies to operate appliances and monitor energy levels in real time. It lets users monitor electricity usage, remotely turn appliances on and off, and get notifications via SMS or a smartphone app. The system makes use of a GSM module (SIM800/SIM900), a relay module, energy sensors (voltage and current), a microcontroller (such as an

Arduino or ESP32), and WiFi connectivity. Thanks to GSM fallback support, it guarantees uninterrupted appliance control even in places with spotty internet connectivity.

In order to improve efficiency, users can set automated schedules or timers, monitor data on energy usage, and get notifications in the event of overload or excessive usage. In both home and commercial settings, this technology is especially helpful in lowering energy consumption, improving electrical safety, and boosting user convenience.

a)Problem Statement:

Conventional electrical systems offer little to no insight into the energy usage of specific equipment, which results in wasteful energy use, exorbitant electricity costs, and little control over user behavior. A system that can track and control power usage at the device level in real time is becoming more and more necessary. In order to solve this problem, an Internet of Things (IoT)-based Smart Plug Energy Monitoring System enables users to:

• Track the energy consumption of connected appliances in real time and over time;

• Remotely control appliances through mobile or web applications;

• Get alerts for unusual energy consumption or safety hazards (like overheating); and

• Automate appliance schedules for increased energy efficiency.

b) Objective:

• Create and deploy a smart plug that can measure energy use, including voltage, current, power, and energy.

• Make historical data logging and real-time monitoring possible.

• Offer remote control (turn on/off) using a web dashboard or mobile app.

• Connect with cloud services to store and analyze data.

• Encourage automation and scheduling tools to maximize energy use.

• Alert users to anomalous usage trends or issues.

c) Methodology:

Several crucial stages are involved in the creation of the Internet of Things-based Smart Plug Energy Monitoring System:

1. Requirement for Analysis

• List the essential features, such as scheduling, remote control, real-time energy monitoring, and user notifications.

• Identify the necessary software and hardware components.

• Evaluate the needs for network connectivity (Wi-Fi, MQTT, etc.).

2. Hardware design and selection

• Microcontroller: Select an ESP8266 or ESP32 microcontroller that supports Wi-Fi.

• Energy Monitoring IC: To measure voltage, current, and power, use an energy monitoring chip such as the HLW8012, ADE7753, or INA219.

• Relay Module: Incorporate a relay to regulate the linked appliance's power supply.

• Smart Plug Enclosure: To securely house all components, create your own enclosure or use one that has already been designed.

3.Circuit design and integration

• Create and model the circuit.

• Connect the power supply module, relay, energy monitoring sensor, and microcontroller.

• Verify that the safety and isolation measures (such as fuses and optocouplers) are in place.

4. Firmware Development

• Develop embedded C or MicroPython code to:

o Read sensor data. Manage the relay.

o Use Wi-Fi to send data to the cloud.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:05/May-2025 Impact Factor- 8.187 www.irjmets.com

o Handle scheduling and user commands.
• Use the HTTP or MQTT protocols to connect to the cloud server.

5.**Mobile and web Application development**

• Use frameworks like Flutter, React, or Android Studio to create a web dashboard or mobile application.
• Offer an intuitive user interface to:

o Track energy consumption in real time.
o Analyze data patterns using graphs and charts.
o Manage appliances from a distance.
o Set up timetables and get notifications.

IV. SYSTEM DESIGN AND DIAGRAMS

System Architecture Diagram:

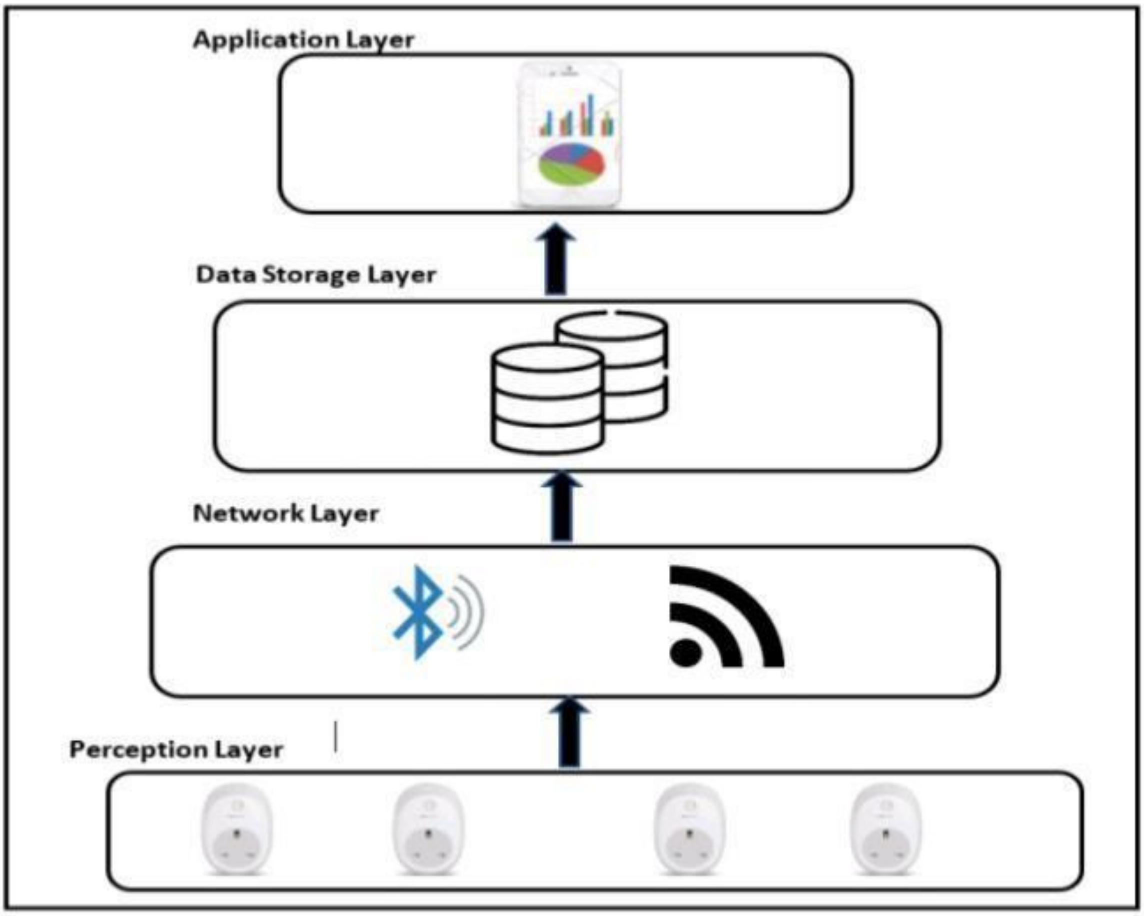


Figure 1. Architecture Diagram.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:05/May-2025 Impact Factor- 8.187 www.irjmets.com

Use Case Diagram:

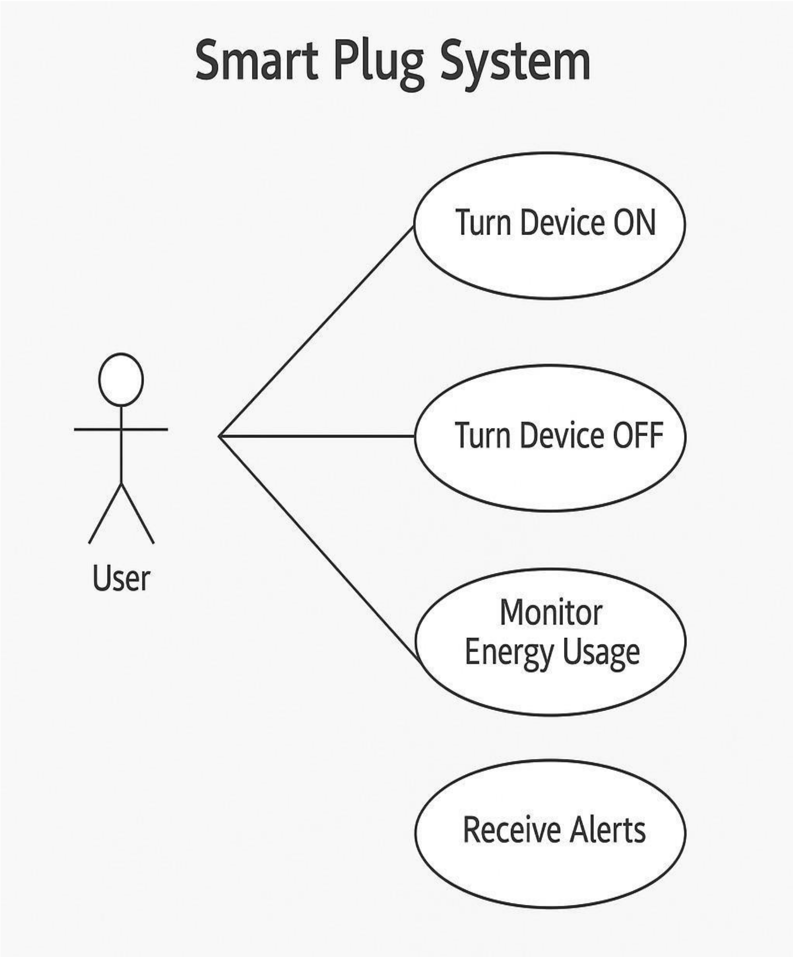


Figure 2. Use case Diagram.

Sequence Diagram:

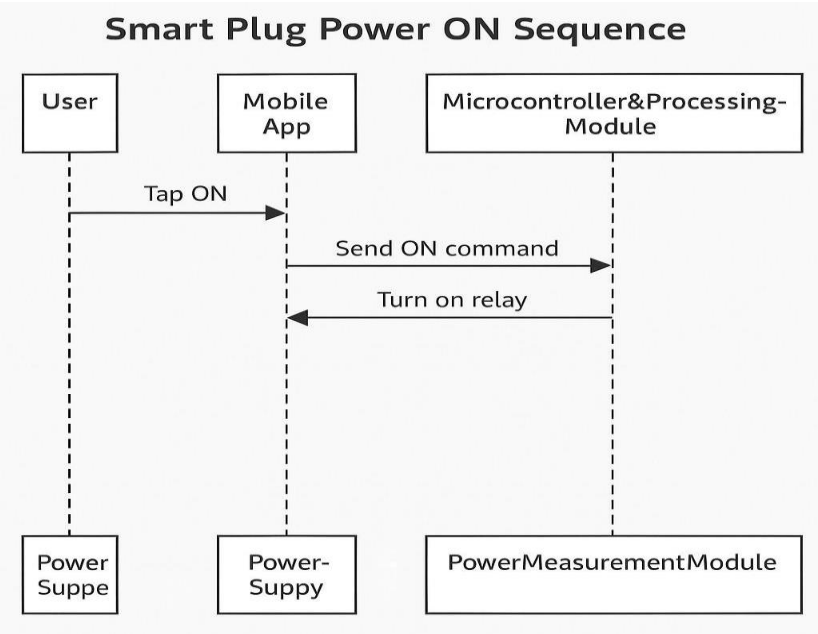


Figure 3. Sequence diagram

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:05/May-2025 Impact Factor- 8.187 www.irjmets.com

Circuit diagram

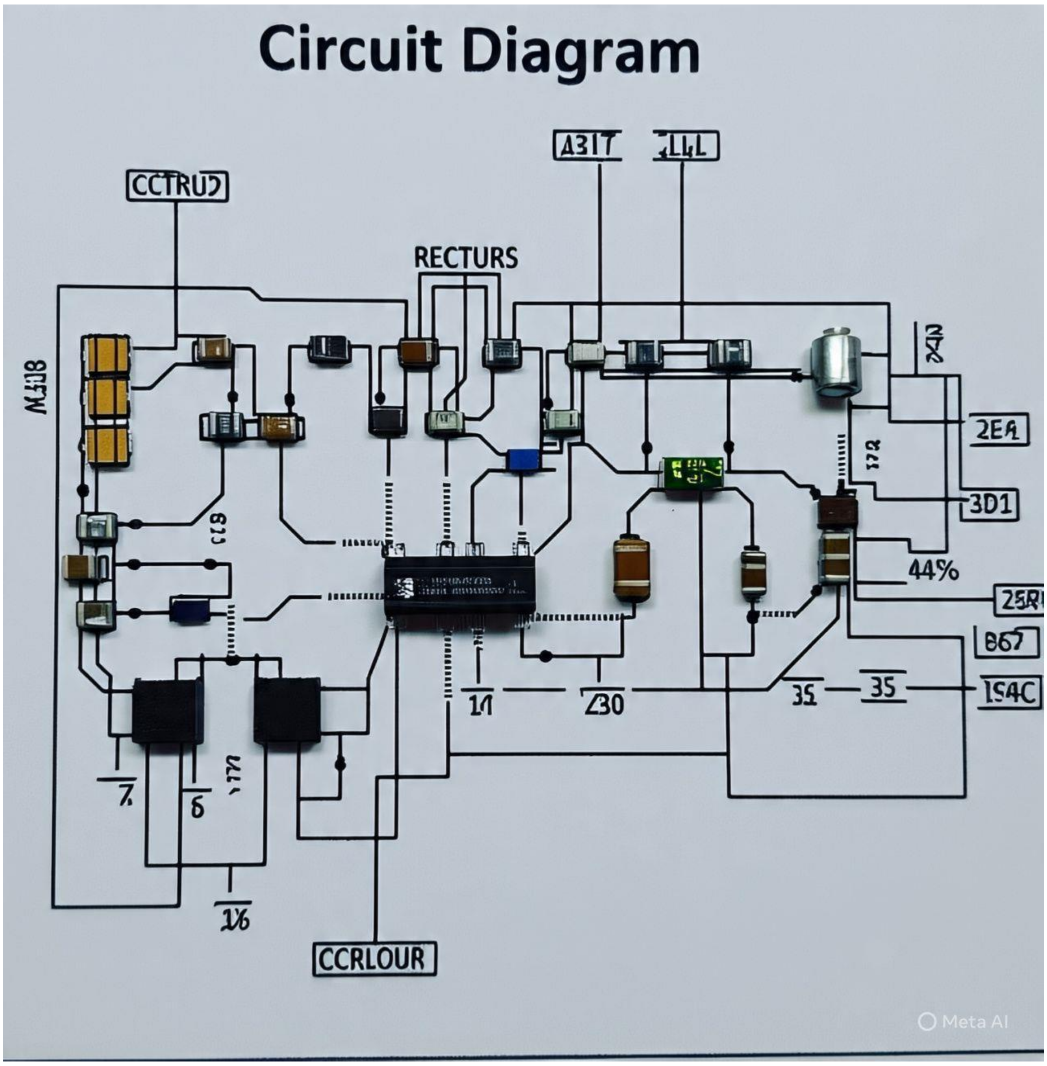


Figure 4. Circuit diagram

V. ALGORITHM

The system provides intelligent energy management and remote control capabilities by combining sensor-based monitoring, condition-based control, and communication handling algorithms. The main algorithms used by the system are listed below.

1. System initialization

1.Setup

1.1. Set up the microcontroller (such as the Raspberry Pi, Arduino, or ESP32).

1.2. Establish a Wi-Fi connection.

1.3. Set up sensors (such as voltage or current sensors like the ACS712).

1.4. Establish communication with the server or cloud (e.g., Firebase, HTTP, MQTT, etc.).

1.5. Start the real-time clock (timestamping is optional).

2. Data Acquisition

2.1. Regularly check sensors for voltage and current.

2.2. Determine power using the formula Power (W) = Voltage (V) x Current (A).

2.3. Determine the amount of energy used: Power × (sample interval in hours) + Energy (kWh)

3. Filtering and Data Processing

3.1. Process sensor data by applying noise filtering or averaging (e.g., moving average).

3.2. Look for irregularities (such as abrupt increases or extended periods of zero loads).

4.Data transmission

4.1. Add timestamp, power, and energy to data (JSON/XML) formats.

4.2. Send information to a database or cloud server.

4.3. If the connection is lost, try again.

5. Remote Control & Monitoring

5.1. Take control commands from the server, such as turning the plug on or off.

5.2. Adjust the relay as necessary.

5.3. Notify the server or application of the status.

6. User Interface (Web App/Mobile)

6.1. Show historical and current energy usage.

6.2. Permit plug control from a distance.

6.3. Display warnings or suggestions (such as a heavy consumption warning).

7. Alerts and Automation (Optional)

7.1. Establish thresholds, such as the maximum amount of energy used each day.

7.2. When thresholds are surpassed, send out alerts via email or push notification.

7.3. Automate tasks, such as turning off the gadget automatically after X hours.

8. Reporting and Logging

8.1. Save information for later analysis in the database.

8.2: Produce usage data on a daily, weekly, and monthly basis.

9. Power Optimization and Sleep Mode (Optional)

9.1. When you're idle, switch to sleep mode.

9.2 Wake up when instructed to do so.

VI. IMPLEMENTATION

Hardware configuration, firmware development, cloud integration, and user interface deployment comprise the smart plug system's implementation.

1. Implementation of Hardware Components Employed:

- **ESP8266/ESP32**: Serves as the primary microcontroller and has WiFi integrated into it.
- **HLW8012/INA219**: Voltage, current, and power are measured by this energy monitoring sensor.
- **Relay Module**: Manages the linked appliance's power supply.
- Power Supply Circuit: For microcontroller operation, this circuit converts AC to a steady 5V/3.3V DC.
- **Plug Casing:** Provides portability and safety by enclosing all components.

STEPS

To obtain real-time power data, connect the energy sensor to the microcontroller.

- To turn the appliance on or off, interface with the relay.
- Use optocouplers and fuse protection to guarantee enough electrical isolation.
- Put all the parts together in a small, insulated plug housing.

2. **Platform for Firmware Development**: Arduino IDE/PlatformIO

Language: MicroPython/Embedded C Functions include reading energy sensor data.

Analyze power, voltage, and current data in real time.

- Relay state can be controlled by schedule or user input.
- Establish a Wi-Fi connection and use HTTP or MQTT to send data to the cloud.
- Manage optional OTA (Over-the-Air) firmware updates.

3. **Choices for Cloud Integration Platforms**: Adafruit IO, ThingsBoard, Blynk, AWS IoT, and Firebase.

Implementation:

•Data about energy use is sent from the smart plug to the cloud using MQTT or REST APIs.
•Keep track of past information in a database.
• Establish thresholds to send out notifications when energy usage deviates from typical.
• Use TLS/SSL to establish secure communication.

4. **Tools for Web and Mobile Applications**:

• **Mobile App**: Android Studio/Flutter
• **Web Dashboard**: Chart.js or D3.js for data visualization using React.js or HTML-CSS-JS

Qualities:

• Energy metrics are shown in real time.
• Charts and graphs of historical usage.
• Toggle switch for remote appliance control.
• Establish schedules or timers for automatic management.
• Alerts or push notifications for threshold violations

5. Testing and Validation

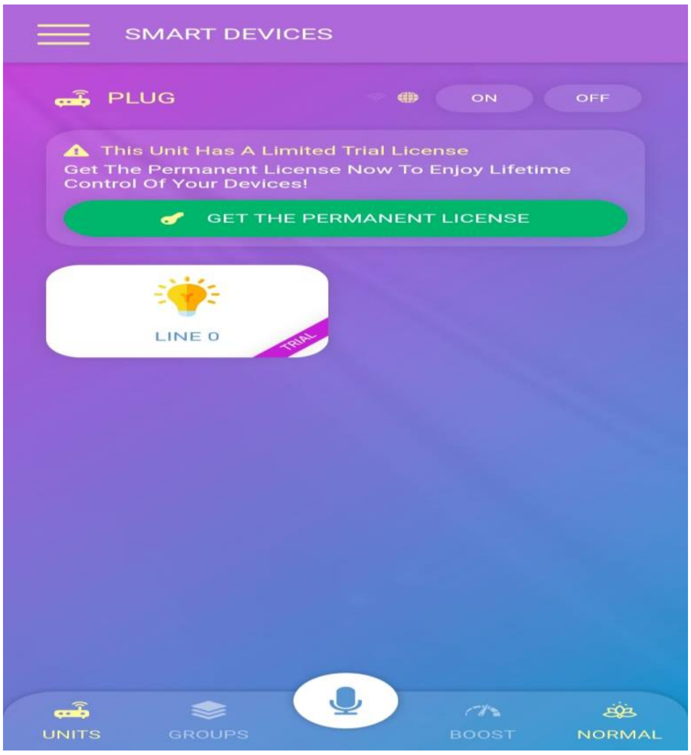
Testing Focus:

• The control commands' response time.
• Wi-Fi communication dependability.
• Load testing for user apps and cloud databases.
• Safety under different load scenarios.

6. Complete Implementation

• In a real-world setting, install the smart plug on test appliances.
• Keep an eye on performance for a few weeks or days.
• Gather user input and make any required design improvements.

VII. SCREENSHOTS



Plug in on process

Usage of Plug with Non-IOT to Connecting

VIII. RESULTS

The IoT-based smart plug energy monitoring system was successfully implemented and tested. The outcomes are summarized as follows:

1. Real-time Energy Monitoring

- The system accurately measured **voltage, current, power**, and **energy consumption** of connected appliances.
- Real-time data was updated every 1–2 seconds on the mobile/web dashboard.
- Measurements were found to be within **±5% error margin** when compared to commercial energy meters.

2. Remote Appliance Control

- Users were able to **remotely turn appliances ON/OFF** using the mobile app.
- Relay switching responded reliably with an average **latency of less than 2 seconds**.

IX. CONCLUSION

The **IoT-based Smart Plug Energy Monitoring and Control System** successfully demonstrates how smart technologies can be integrated into everyday electrical appliances to promote efficient energy management. By leveraging microcontrollers (ESP32/ESP8266), WiFi, and GSM modules, the system allows users to remotely monitor and control connected devices, even in the absence of a stable internet connection.

The solution provides real-time tracking of voltage, current, and power consumption, enabling users to identify high-power appliances and reduce unnecessary energy usage. Advanced features like **scheduled control**, **overload protection**, and **dual-mode communication (WiFi & GSM)** enhance the system's reliability, making it suitable for smart homes, industries, and rural areas.

X. REFERENCES

[1] **Datasheets & Hardware Documentation**
Espressif Systems. ESP32 Datasheet. https://www.espressif.com
SIMCom Wireless Solutions. SIM800L GSM Module Datasheet.
Research Papers

[2] S. Kaur, A. Sharma, "Smart Energy Meter with Overload Detection and Power Theft Monitoring," International Journal of Engineering and Technology, 2021.
R. Gupta, P. Verma, "IoT Based Smart Energy Monitoring Using WiFi and GSM," International Journal of Advanced Research in Computer Science, 2020.

[3] **Books**
John F. Wakerly, Digital Design: Principles and Practices, Pearson Education.
Muhammad Ali Mazidi, The 8051 Microcontroller and Embedded Systems, Pearson Education.

[4] **Web Resources**
Arduino Official Documentation: https://www.arduino.cc
Scikit-learn SVM and Random Forest Tutorial: https://scikit-learn.org