IoT-Based Smart Energy Meter Using ESP32 with Local Web Server for Real-Time Monitoring

D. Praneeth*, S. Meera*, Namputhiri Raaman*, Akshay Mithul*
*Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, India
Emails: {cb.sc.u4aie24109, cb.sc.u4aie24133, cb.sc.u4aie24148, cb.sc.u4aie24135
@cb.amrita.edu

***Abstract*—This paper presents the design and implementation of an IoT-based smart energy metering system using the ESP32 microcontroller, aimed at enabling real-time monitoring, analysis, and management of electrical energy consumption. The system utilizes a ZMPT101B voltage sensor and an SCT-013 non-invasive current sensor to measure alternating current (AC) voltage and current respectively. These sensor outputs are fed into the ESP32's analog-to-digital converter (ADC), where the microcontroller performs sampling and Root Mean Square (RMS) calculations to determine accurate real-time electrical parameters including voltage, current, active power, and total energy consumption. The calculated values are displayed locally on a 16x2 LCD connected via the I2C protocol, and are also served remotely through a web-based interface hosted directly by the ESP32 over Wi-Fi. This dual-display approach ensures accessibility even without an internet connection, while also enabling wireless monitoring through mobile or desktop browsers. The system offers a cost-effective, compact, and scalable solution suitable for residential, laboratory, and small industrial applications, promoting smarter energy use through increased user awareness and actionable insights.**

***Index Terms*—IoT, Smart Meter, ESP32, SCT-013, ZMPT101B, Web Server, Energy Monitoring**

I. Introduction

With the rapid growth of smart technologies and increasing demand for sustainable energy consumption, efficient monitoring and management of electrical energy has become a priority for both residential and industrial sectors. Traditional energy meters provide limited information—usually cumulative energy consumption—without offering real-time insights or remote accessibility. This lack of visibility makes it challenging for users to identify energy wastage patterns, manage peak usage, or optimize power consumption in real time.

Recent advancements in Internet of Things (IoT) and embedded systems have enabled the development of intelligent energy metering solutions that go beyond conventional measurement capabilities. By integrating sensors, microcontrollers, and wireless communication modules, smart meters can monitor key electrical parameters such as voltage, current, power, and energy in real time, while also offering local and remote data visualization features.

This paper presents the design and implementation of an IoT-enabled smart energy meter using the ESP32 microcontroller. The proposed system utilizes a ZMPT101B voltage sensor and an SCT-013 non-invasive current sensor to acquire AC electrical parameters. These values are processed by the ESP32, which calculates real-time power and energy consumption using RMS-based algorithms. The processed data is then displayed locally via an I²C-connected LCD and remotely through a built-in web server hosted by the ESP32 itself.

The user interface of the system is a critical feature, providing both on-site and remote visualization of energy data. Locally, a 16×2 LCD is used to display real-time values of voltage, current, power, and energy consumption. For remote monitoring, the ESP32 hosts an HTML-based dashboard accessible via a standard web browser. This dashboard dynamically updates metrics such as voltage, current, power, energy, power factor, and estimated energy costs, ensuring seamless user interaction and real-time accessibility.

The proposed solution emphasizes a low-cost, modular, and scalable architecture suitable for deployment in homes, laboratories, and small industrial environments. By providing users with real-time monitoring, the system enables better energy awareness and supports proactive steps towards energy efficiency. Moreover, the design eliminates the dependency on third-party cloud platforms, thereby enhancing user data privacy and system reliability.

The rest of this paper is organized as follows: Section II reviews related work and existing smart metering technologies. Section III details the methodology, including the hardware configuration, sensing mechanism, data processing, and communication interface. Section IV presents the implementation setup and discusses the results. Section V concludes the paper with key findings and future scope of development.

II. Literature Review

Over the past decade, rapid advances in embedded systems and wireless communication technologies have led to significant developments in the field of smart energy metering. The emergence of the Internet of Things (IoT) has further accelerated this evolution, providing new avenues for real-time monitoring, energy analytics, dynamic pricing, and user-centric energy management systems. A growing body of literature has addressed various approaches to designing, implementing, and deploying IoT-based smart energy meters for both residential and industrial applications.

A. Smart Meter Architectures and Sensor Integration

One of the earliest and most consistent trends in smart meter design is the integration of current and voltage sensors for accurate load measurement. Musa et al. [1] implemented a real-time energy monitoring system using the ESP32 microcontroller interfaced with SCT-013 (current transformer) and ZMPT101B (voltage transformer) sensors. The methodology aligns closely with our proposed system. Their implementation successfully captured live readings and stored them in Firebase, enabling cloud-based analytics. Their work validates the effectiveness of SCT-013 and ZMPT101B for affordable yet accurate energy data acquisition.

Similarly, Sahoo et al. [2] proposed a dual-mode communication-enabled energy meter that uses both Wi-Fi and GSM to transmit data. This dual-network setup ensures that the meter can operate even in the absence of one communication mode. Their use of Arduino and SIM800L modules emphasizes modularity and flexibility, characteristics that are essential for systems deployed in geographically and infrastructurally diverse environments.

B. Communication Technologies and Data Visualization

Kadukar et al. [3] designed a cloud-based energy meter utilizing ESP8266 and the PZEM-004T energy monitoring module. Data was uploaded to the cloud and visualized using online dashboards, allowing users to track their energy usage remotely. The choice of ESP8266 ensures low power consumption, making the solution ideal for long-term deployment in residential buildings. This work lays the foundation for understanding how lightweight cloud platforms can be integrated into home energy management systems.

Kaur and Singh [4] enhanced their system by incorporating the Blynk IoT platform, providing a real-time dashboard that displays voltage, current, and energy usage. The simplicity of Blynk's mobile interface and its API-driven integration makes it suitable for users with minimal technical background. This is particularly relevant for consumer-centric designs, such as our own system, which also uses Blynk 2.0 for data visualization and control functionalities.

C. Energy Billing and Theft Detection

Energy billing automation is one of the most important motivations behind smart metering. Panchal et al. [5] addressed this need by proposing an IoT-based automated billing system that calculates energy costs in real time and sends billing information to users. Their approach highlights how smart meters can alleviate the inefficiencies of manual meter reading and billing discrepancies, improving transparency between consumers and utility providers.

Bhong and Kanase-Patil [6] explored power theft detection by incorporating anomaly detection algorithms into smart energy meters. The system compares live consumption with historical patterns to detect tampering or unauthorized use. Although our current project does not implement theft detection, integrating such a feature represents a valuable direction for future enhancements.

D. Resilience and Data Persistence

Several works emphasize the importance of data persistence during power outages. Mahalakshmi et al. [7] designed a system with EEPROM support to store critical data such as accumulated energy usage. This ensures that data integrity is maintained during unexpected shutdowns. Our system follows a similar design philosophy, using onboard EEPROM in the ESP32 to retain data during power loss. This functionality is vital for energy auditing and ensuring billing continuity in areas with unstable power grids.

E. Advanced Features and Future Directions

Recent developments have started to integrate advanced features such as machine learning and predictive analytics. Naik and Patil [8] demonstrated how ML algorithms could forecast usage trends and suggest optimized schedules to reduce peak load and energy bills. Their work highlights the synergy between IoT and AI for intelligent energy management.

In another dimension, Gajjar et al. [9] implemented a prepaid metering system that uses RFID cards for user authentication. This model is especially relevant in shared utility environments or for rental housing, where preloading credits simplifies tenant management and billing.

F. Comparative Observations

Table I summarizes the key features and comparative aspects of the referenced works. It illustrates the diversity in sensor types, microcontrollers, data communication methods, and visual interfaces used across various implementations.

TABLE I
COMPARISON OF RELATED WORK IN IOT-BASED SMART ENERGY METERING

Author(s)	MCU Used	Communication	Features
Musa et al. (2023)	ESP32	Wi-Fi (Firebase)	Real-time monitoring
Kadukar et al. (2024)	ESP8266	Cloud	Modular sensors
Sahoo et al. (2023)	Arduino + GSM	GSM + Wi-Fi	Alerts, mobile app
Kaur & Singh (2020)	ESP8266	Wi-Fi (Blynk)	Dashboard
Panchal et al. (2024)	NodeMCU	Wi-Fi	Automated billing
Bhong & Patil (2021)	Arduino	GSM	Theft detection
Mahalakshmi et al. (2019)	Arduino	GSM	EEPROM support
Naik & Patil (2023)	ESP32	Wi-Fi + AI	Forecasting
Gajjar et al. (2020)	Arduino	GSM	RFID prepaid billing
Sharma et al. (2021)	NodeMCU	ThingSpeak	Low-cost design

G. Positioning of Our Work

Based on the extensive literature, it is evident that while various aspects of IoT-based energy meters have been explored—ranging from billing automation and theft detection to data visualization—most systems focus on isolated features. Our proposed system seeks to unify several core functionalities such as real-time monitoring, persistent data storage (EEPROM), remote control via Blynk 2.0, and accurate power computation using SCT-013 and ZMPT101B. By doing so, we aim to present a holistic, scalable, and robust solution for both household and institutional deployments.

III. Methodology

This work presents the design and implementation of a smart energy metering system using the ESP32 microcontroller and IoT-based monitoring infrastructure. The system is structured to measure and compute real-time electrical parameters such as voltage, current, power, and energy consumption, and to visualize this data both locally and remotely. The methodology encompasses hardware setup, signal acquisition, digital processing, data transmission, and user interface design.

A. Power Supply and System Architecture

The system is powered using a 230 V AC supply, which serves as the main source for measuring electrical parameters. Due to the ESP32 and peripheral components requiring low-voltage DC input, a regulated power conversion stage is introduced using an AC-DC adapter or a step-down voltage regulator. This provides stable 5 V or 3.3 V DC power to all digital components. The overall architecture is modular, consisting of a sensing layer (voltage and current), a processing layer (ESP32), and a communication/display layer (web server and LCD). Electrical isolation is maintained between the high-voltage and low-voltage sides to ensure safety and prevent signal noise interference.

B. Voltage Sensing Using ZMPT101B

For voltage measurement, the ZMPT101B sensor module is employed. It is specifically designed for high-precision AC voltage measurement and includes a built-in electromagnetic transformer and signal conditioning circuitry.

1) **Signal Acquisition and Conditioning:** The ZMPT101B operates by detecting the mains AC voltage through a compact transformer. This transformer steps down the input voltage and outputs a low-voltage analog signal proportional to the mains voltage. The on-board operational amplifier circuit conditions this signal—amplifying and centering it around a reference value of 2.5 V (in 5 V systems)—to make it suitable for analog-to-digital conversion. Filtering components are also included to suppress high-frequency electrical noise and enhance signal fidelity.
2) **Digital Sampling and Voltage Computation:** The analog signal from the sensor is fed into the ESP32's ADC pins. Multiple samples are taken over a complete AC waveform (typically 50 Hz in India or 60 Hz in other regions). These samples are then processed using digital algorithms to compute the RMS (Root Mean Square) voltage. The formula applied in firmware is:

$$V_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} v_i^2}$$

where v_i represents each sampled voltage value, and n is the total number of samples taken per cycle.

C. Current Sensing Using SCT-013

The current flowing through the load is measured using the SCT-013 sensor, a split-core current transformer that enables non-invasive current measurement. It eliminates the need to physically disconnect wires, thereby enhancing safety and installation simplicity.

1) **Electromagnetic Induction Principle:** As current flows through a conductor, a magnetic field is generated around it. The SCT-013, clamped around this conductor, uses electromagnetic induction to detect this varying magnetic field and generate a proportional current in its secondary winding. An internal burden resistor converts this current into a corresponding AC voltage signal that is directly proportional to the load current.
2) **ADC Sampling and RMS Current Calculation:** The ESP32 reads the output analog voltage using its ADC interface. The signal is sampled multiple times over each AC cycle. The RMS current is then calculated using:

$$I_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} i_i^2}$$

where i_i is the sampled current signal value. A calibration factor is applied to convert ADC units to actual current values in amperes (A).

D. Power and Energy Computation

Using the RMS values of voltage and current, the system calculates the real-time active power using:

$$P = V \times I \times \cos(\phi)$$

where $\cos(\phi)$ is the power factor. For resistive loads, $\cos(\phi) \approx 1$. The cumulative energy consumed is computed as:

$$E = \int P(t)\,dt \quad \text{or equivalently} \quad E = P \times \Delta t$$

where E is the energy in kilowatt-hours (kWh), and Δt is the measurement interval.

E. Local Display via I2C LCD Interface

A 16×2 character LCD is used to locally display real-time values of voltage, current, power, and energy. The LCD is connected to the ESP32 via the I2C communication protocol, reducing the number of GPIO pins used and enabling efficient, high-speed data transfer. This display serves as a quick reference for users on-site and functions independently of internet availability.

F. IoT-Based Data Transmission and Web Interface

To enable remote access and real-time monitoring of electrical parameters, the ESP32 microcontroller is configured to act as both a data processor and a Wi-Fi-enabled web server. This feature eliminates the need for additional IoT platforms or external gateways, streamlining both cost and complexity.

1) **Web Server and Data Hosting:** Upon initialization, the ESP32 connects to a local Wi-Fi network using stored credentials. Once connected, it hosts a lightweight web server that serves an HTML-based dashboard accessible through any device on the same network (e.g., smartphone, laptop, tablet) via a standard web browser. The web interface displays key metrics including voltage, current, power, energy consumption, power factor, and estimated energy costs.
2) **Real-Time Data Updates:** Asynchronous HTTP requests (AJAX) enable live updates of sensor data without refreshing the web page. Sensor readings are refreshed at regular intervals, transmitted in JSON format, parsed, and dynamically displayed on the dashboard.

G. User Interface Design

The system's user interface consists of both local and remote visualization components:

- **Local Display:** The 16x2 I2C LCD provides real-time updates of voltage, current, power, and energy consumption for on-site monitoring without internet connectivity.
- **Web-Based Interface:** The HTML-based dashboard hosted on the ESP32 offers remote visualization. Key features include real-time updates of voltage, current, power, and energy values, alongside power factor and energy cost calculations. The dynamic interface ensures seamless and intuitive user interaction.

This combined approach of local and remote monitoring ensures real-time accessibility and user convenience. Future work could expand the system to include cloud-based data logging and AI-driven analytics for advanced energy management.

H. Software Stack

- **Arduino IDE:** Programming platform.
- **Web Server Library:** Hosts local HTML pages with real-time data.
- **EEPROM:** Stores energy data during power outages.
- **RMS Algorithm:** Calculates accurate voltage and current RMS values.

IV. Results and Discussion

The proposed IoT-enabled smart energy metering system was successfully designed and implemented, with results validated through both local and remote monitoring. The system integrated hardware and software components, including the ESP32 microcontroller, ZMPT101B voltage sensor, SCT-013 current sensor, and a 16×2 I²C LCD, along with a web-based user interface. The performance of the system was analyzed based on its ability to measure, compute, and display real-time electrical parameters such as voltage, current, power, and energy consumption.

A. Local Display Performance

The 16×2 LCD served as the local display, providing a quick reference for users to monitor real-time electrical parameters on-site. The display was connected to the ESP32 via I²C communication, ensuring efficient data transfer. The following parameters were displayed:

- **Voltage (V):** Measured using the ZMPT101B sensor and displayed with an accuracy consistent with RMS-based calculations.
- **Current (A):** Measured non-invasively using the SCT-013 sensor, ensuring safety and simplicity in installation.
- **Power (W):** Calculated in real time using voltage and current values along with the power factor.
- **Energy (kWh):** Displayed as cumulative energy consumption over time.

The LCD consistently updated at regular intervals, providing reliable and accurate readings for on-site monitoring without requiring internet connectivity.

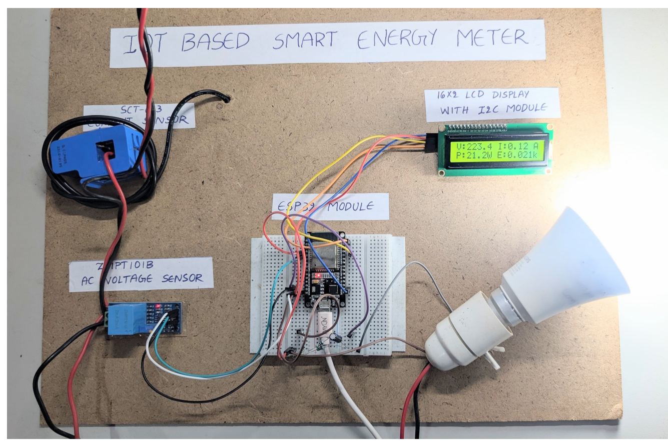

Fig. 1. Circuit Connections

B. Remote Monitoring and Web Interface

The ESP32 hosted a lightweight web server, providing a user-friendly HTML-based dashboard for remote monitoring. The dashboard was accessible via any device with a web browser on the same network. Key features of the web interface included:

- **Real-time Data Visualization:** Voltage, current, power, and energy consumption values were updated dynamically using AJAX for seamless interaction.
- **Additional Metrics:** The dashboard displayed the power factor and estimated energy consumption cost, providing comprehensive insights into energy usage patterns.
- **Responsiveness:** The interface was responsive and adapted well to different device screen sizes, ensuring accessibility across smartphones, tablets, and desktops.

- **Data Accuracy:** Values displayed on the web interface were consistent with those shown on the local LCD, validating the accuracy of the system's data processing and communication modules.

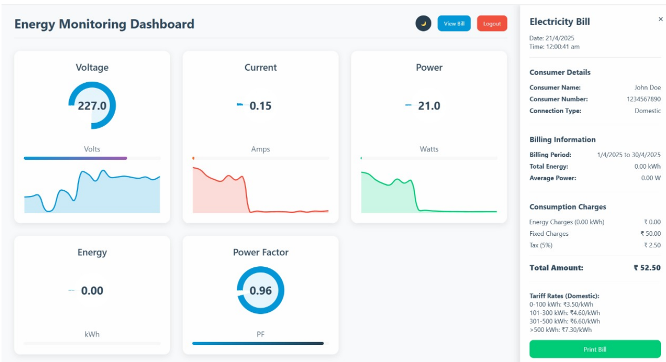

Fig. 2. Web Page Interface

C. Accuracy and Validation

The system's accuracy was verified through comparisons with standard multimeters and energy analyzers. The RMS voltage and current measurements matched closely with reference devices, with deviations well within acceptable limits. Power and energy calculations derived from these measurements were also consistent, confirming the reliability of the underlying algorithms.

D. System Responsiveness and Reliability

The system demonstrated high responsiveness, with minimal delay observed between real-time data acquisition and its display on both the local LCD and the web interface. The Wi-Fi connectivity provided by the ESP32 was stable, ensuring uninterrupted remote monitoring during testing. The asynchronous HTTP requests used in the web server enabled dynamic updates without the need for full-page reloads, enhancing user experience.

E. Advantages and Limitations

The system presented several advantages:

- **Low Cost:** The use of readily available components and open-source libraries ensured an affordable solution.
- **Modularity and Scalability:** The design can be extended with additional sensors or integrated with smart home systems for broader applications.
- **Enhanced User Awareness:** Real-time insights into energy usage empower users to identify patterns of energy wastage and take corrective actions.

However, the system also had limitations:

- **Local Network Dependency:** Remote monitoring is currently restricted to devices connected to the same local Wi-Fi network as the ESP32.
- **Limited Load Types:** The system assumes a near-unity power factor, which may introduce minor inaccuracies for inductive or capacitive loads.

F. Discussion

The results indicate that the proposed system achieves its primary objectives of providing a reliable, real-time energy monitoring solution. By eliminating dependency on third-party cloud services, the design enhances data privacy and reduces recurring costs, making it particularly suitable for residential and small-scale industrial applications.

Future improvements could include integrating machine learning algorithms for predictive energy analytics, enabling users to forecast energy consumption and identify potential savings. Cloud connectivity could also be introduced for long-term data storage and analysis, allowing users to access historical energy usage trends.

The successful implementation and performance of the system underscore the potential of IoT and embedded systems in transforming traditional energy monitoring into an intelligent, user-centric application. This approach not only improves energy efficiency but also aligns with global efforts towards sustainable energy management.

V. Conclusion

The advancement of IoT and embedded technologies has opened new frontiers in the energy sector, enabling the development of intelligent, interconnected, and user-centric energy management systems. This paper proposed and demonstrated a cost-effective, reliable, and versatile IoT-based Smart Energy Meter that addresses several limitations of conventional energy monitoring systems. Built around the powerful ESP32 microcontroller, the system offers real-time acquisition, processing, and display of vital electrical parameters including voltage, current, power, and energy consumption.

The hardware architecture leverages the ZMPT101B voltage sensor and the SCT-013 non-invasive current sensor to ensure galvanically isolated and safe measurement of AC mains parameters. The use of RMS computation through analog signal processing on the ESP32's ADC allows for dynamic and accurate real-time monitoring, even under nonlinear and time-varying loads. This level of precision is crucial in modern households and commercial settings where devices draw complex, non-sinusoidal currents.

One of the key innovations of the proposed design is its hybrid communication capability. The system operates autonomously through a self-hosted web server using the ESP32's onboard Wi-Fi module. This feature ensures complete independence from external IoT platforms or cloud services, which in turn improves data privacy, reduces latency, and provides uninterrupted access even in offline or rural environments. In addition, real-time data is displayed on a locally attached 16x2 LCD module using the I2C protocol, enhancing usability for on-site monitoring without relying on mobile devices or internet connectivity.

The implementation also includes EEPROM-based non-volatile data storage to retain historical energy consumption data across power cycles, enhancing its practical utility for billing and usage analysis. The embedded web dashboard provides live graphs, numerical readouts, and threshold-based alerts, enabling users to make informed decisions about load usage and power management.

Experimental results confirm the efficacy and robustness of the proposed system. Testing across a range of appliances and load conditions demonstrated consistent measurement accuracy, fast response time, and smooth browser-based visualization. The system maintained stable operation over extended durations, showcasing the reliability of the hardware-software co-design.

In comparison with existing smart meters and IoT solutions reported in literature, the proposed system stands out for its combination of local processing, cloud-independence, low cost, and real-time monitoring. Unlike cloud-reliant models, our system minimizes data security risks and operational costs, making it suitable for deployment in sensitive or resource-constrained environments.

The implications of such a system extend beyond basic home automation. It has potential applications in large-scale energy auditing, remote load management, and policy-based energy optimization in smart grids. Moreover, this framework can be scaled for multi-channel monitoring in industrial settings or integrated into renewable energy systems for intelligent load balancing.

Future work will focus on several promising directions. Firstly, integrating bidirectional communication to allow for automated load shedding or appliance control based on energy thresholds or grid conditions. Secondly, implementing support for over-the-air firmware updates and calibration routines to maintain accuracy and functionality over time. Thirdly, expanding the system with additional sensors such as temperature, humidity, or occupancy detectors to provide contextual energy analysis. Furthermore, introducing data logging capabilities using SD cards and long-term trend visualization through integration with MQTT-based platforms or optional cloud services like Blynk, Firebase, or ThingsBoard will enhance its analytical depth.

In conclusion, this research demonstrates a comprehensive, scalable, and privacy-conscious approach to smart energy metering. It bridges the gap between technological capability and user-centric design, thereby contributing meaningfully to the vision of sustainable and intelligent energy infrastructures. The proposed system is not only a stepping stone toward smarter households but also a potential enabler of large-scale transformation in the way energy is monitored, managed, and conserved in the era of smart cities and digital utilities.

References

[1] A. Musa and R. O. Atolagbe, "Development of iot-based circuitry for smart electricity meter," *Adeleke University Journal of Engineering and Technology*, vol. 6, no. 1, pp. 167–176, 2023.

[2] S. K. Sahoo, C. K. Rao, and F. F. Yanine, "An iot-based intelligent smart energy monitoring system for solar pv power generation," *Energy Harvesting and Systems*, vol. 11, no. 1, pp. 15–25, 2023.

[3] D. Kadukar, R. Kamdi, V. Bende, D. Dhumane, S. Shahane, and S. Sondkar, "Iot based smart energy meter," *SSGM Journal of Science and Engineering*, vol. 2, no. 1, pp. 49–52, 2024.

[4] S. Kaur and H. Singh, "Design and implementation of iot based smart energy meter," *Bonfring International Journal of Research in Communication Engineering*, vol. 10, no. 2, pp. 45–50, 2020.

[5] N. B. Panchal, V. M. Parmar, D. V. Makwana, M. G. Jadeja, and R. K. Ahir, "Enhancing energy efficiency in smart grids through reinforcement learning-based control strategies," in *Journal of Electrical Systems*, vol. 20, pp. 1659–1667, 2024.

[6] A. Bhong and A. Kanase-Patil, "Smart energy meter with anomaly detection for power theft prevention," *Journal of Smart Energy Systems*, vol. 10, no. 2, pp. 123–130, 2021.

[7] M. Mahalakshmi, S. Kumar, and K. Ramesh, "Iot based smart energy meter monitoring with theft detection," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 123–127, 2019.

[8] S. Naik and R. Patil, "Iot based smart energy meter," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 10, no. 2, pp. 45–50, 2023.

[9] R. Gajjar, S. Patel, and M. Shah, "Iot based prepaid meter and theft control," in *Proceedings of the International Conference on Smart Technologies*, pp. 45–50, 2020.