International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Electricity Theft Detection Using IOT

Rohit Yadav [1],Shivraj Pawar [2], Aditya Rajput[3], Vinod karande[4]
[1,2,3] student, [4] Guide

Bharati Vidyapeeth (Deemed to be University),Collage Of Engineering, Pune

Abstract— Nontechnical losses (NTL), particularly electricity theft through unauthorized line taps and meter tampering, impose substantial revenue losses on utilities and degrade grid reliability. Conventional periodic manual inspection is slow, costly, and reactive. This paper presents the design and implementation of a low-cost Internet of Things (IoT)-based electricity theft detection system. An ESP32 microcontroller acquires load voltage and current through a ZMPT101B voltage sensor module and an ACS712 Hall-effect current sensor, computes instantaneous power and cumulative energy, and compares measured consumption with a reference consumption profile. A deviation-based detection algorithm flags suspicious mismatches exceeding a configurable threshold and reports them to an IoT cloud dashboard over Wi-Fi, generating real-time alerts and optionally actuating a relay to isolate the suspicious load path.

Index Terms— Electricity theft detection, Internet of Things, smart grid, energy monitoring, ESP32, smart metering, non-technical loss.

I. Introduction

Electricity theft problem. Electricity theft is one of the most persistent problems facing power distribution utilities. It occurs primarily through unauthorized line taps, meter tampering, and bypassing of metering equipment, and it imposes substantial revenue losses on utilities while degrading the quality and reliability of the distribution network. Because the stolen energy flows through unmonitored and often unsafe connections, theft also introduces serious electrical hazards, including fire risk and electrocution [1].Conventional theft-detection practice relies on periodic manual meter reading, physical line inspection, and complaint-driven audits. These methods are reactive, labor-intensive, and provide no real-time visibility into consumption behavior, allowing theft to persist undetected for entire billing cycles. The Internet of Things (IoT) offers a practical alternative: low-cost embedded platforms with integrated wireless connectivity can continuously monitor voltage, current, and power at distributed points in the network and report anomalies to cloud-based analytics in near real time [7], [8]. Smart grid research has likewise demonstrated the value of dense sensing and communication infrastructure for distribution-level monitoring [9]. These developments motivate a shift from periodic, manual surveillance to continuous, automated, IoT-based monitoring.

II. Related Work

Previous electricity-theft detection methods. Electricity theft and NTL have been studied extensively in the energy literature. Smith provided an early comparative analysis of theft mechanisms across different regions, establishing the economic severity of the problem and the inadequacy of purely regulatory countermeasures [1]. With the growth of metered data, research shifted toward data-driven detection: Nagi et al. applied support vector machines to classify metered customers exhibiting suspicious consumption patterns characteristic of NTL [3], while Ferreira et al. formulated NTL identification as a pattern-recognition problem solved with the optimum-path forest classifier, reporting improved robustness over purely statistical methods [5]. McLaughlin et al. proposed a multi-sensor energy theft detection framework for advanced metering infrastructures (AMI), combining consumption data with side-channel information such as network and meter events [4], and Kadurek et al. examined utility-side theft detection enabled by smart metering, highlighting practical deployment challenges [6].

Smart meters. Smart meters have been shown to enable not only accurate billing but also loss analysis, demand monitoring, and remote disconnect capabilities [2]. Depuru et al. surveyed the challenges, issues, and advantages of smart meter deployment, noting that cost, communication infrastructure, and data management remain significant barriers to large-scale adoption [2]. Utility-side analyses further indicate that theft detection based on AMI data is feasible but depends on extensive metering penetration and centralized analytics [6].

IoT-based monitoring. Parallel advances in the IoT have established the architectural elements, enabling protocols, and application domains of cloud-connected sensing systems [7], [8]. Al-Fuqaha et al. surveyed the protocols (MQTT, CoAP, HTTP) and communication technologies that support lightweight, low-cost telemetry [8]. In the smart grid context, Gungor et al. characterized the opportunities and challenges of wireless sensor networks for distribution-level monitoring, including reliability, latency, and power constraints of deployed nodes [9]. Together, these works demonstrate that continuous, remote, near-real-time monitoring of electrical parameters at distributed points is feasible with commodity embedded hardware.

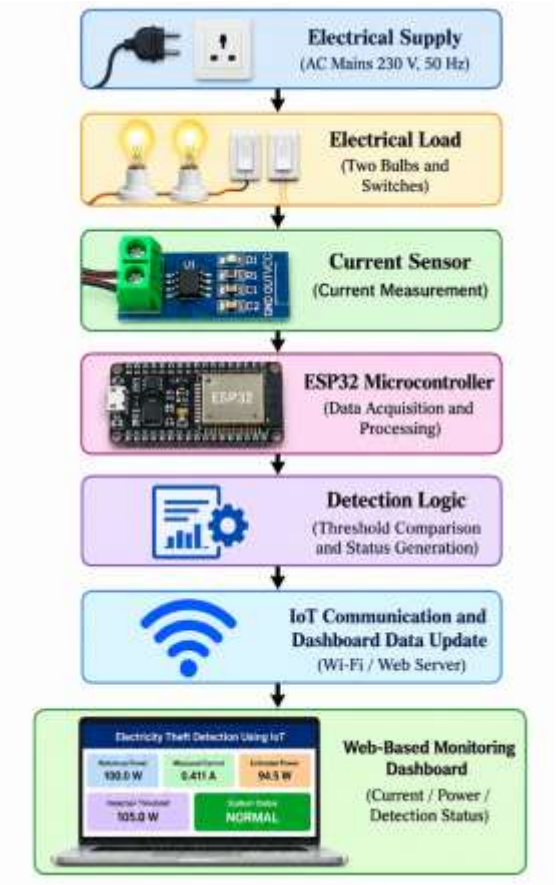
III. Proposed System

A. System Overview

The proposed system is a low-cost, IoT-based electricity theft detection platform for single-phase distribution monitoring. It continuously measures load voltage and current at the consumer point, computes instantaneous power and cumulative energy on the microcontroller, and compares the measured consumption with an expected reference profile. When the deviation exceeds a configurable threshold for a specified number of consecutive monitoring windows, the event is classified as suspected theft, and an alert is pushed to a cloud dashboard over Wi-Fi. The system additionally provides remote load-control capability through a relay, enabling isolation of a suspicious feeder pending inspection.

B. System Architecture

The overall architecture follows the signal and data chain illustrated in Fig. 1:



Each block performs a distinct function:

- **Electrical Supply:** the single-phase AC feeder under monitoring, at which the measurement point is installed ahead of the consumer load path.
- **Voltage/Current Measurement:** the analog sensing stage, comprising the ZMPT101B voltage sensor and ACS712 current sensor, which produce scaled analog signals proportional to line voltage and load current.
- **Microcontroller:** an ESP32 that digitizes the sensor outputs via its 12-bit ADC, computes RMS values, power, energy, and deviation, and executes the detection decision locally at the network edge.
- **Theft Detection Algorithm:** the deviation-based logic that compares measured consumption with the reference profile and applies the threshold and confirmation rules described in Section IV.
- **Wi-Fi/Internet:** the 2.4 GHz wireless link over which the ESP32 publishes telemetry and detection status using the MQTT protocol.

- **IoT Cloud:** a cloud broker/platform that receives, stores, and routes telemetry and triggers alert rules.
- **Dashboard/Alert:** a web interface displaying live voltage, current, power, energy, and deviation, together with an event log; alerts are dispatched as dashboard notifications, email, or push messages upon theft detection.
- **Relay/Load Control (optional):** a relay module commanded through a cloud control topic to disconnect the suspicious feeder and restore it remotely after inspection.

C. Hardware Components

The sensing node is built around the **ESP32**, a dual-core microcontroller with an integrated 2.4 GHz Wi-Fi transceiver and 12-bit successive-approximation ADC, which combines signal acquisition, computation, and cloud connectivity on a single low-cost device. The **ZMPT101B** module measures AC line voltage through a precision voltage transformer, with its output scaled to the ESP32 ADC input range. The **ACS712-20A** Hall-effect current sensor provides galvanically isolated AC load-current measurement with a ratiometric analog output. A **relay module** enables remote disconnection of the load path when theft is confirmed. An **AC–DC power supply module** derives the low-voltage DC rail for the electronics from the monitored supply, and the monitored **electrical load** consists of standard resistive appliances connected to the test bench.

D. Software Components

The firmware is developed in the **Arduino IDE** using Embedded C/C++ and comprises five functional layers: (i) *sensor data acquisition*, in which ADC channels are oversampled and root-mean-square (RMS) values are computed for each 10-second monitoring window; (ii) *data processing*, in which instantaneous power, cumulative energy, and deviation are calculated on the device; (iii) *theft detection logic*, implementing the threshold and consecutive-window confirmation rules of Section IV; (iv) *IoT communication*, in which the ESP32 publishes telemetry over MQTT and subscribes to control topics for relay actuation; and (v) *alert generation*, configured as cloud-platform rules that dispatch notifications upon receipt of a theft flag.

E. IoT Architecture

The IoT layer follows a standard four-tier structure: the *perception tier* (voltage and current sensors), the *network tier* (ESP32 Wi-Fi and MQTT transport over the Internet), the *middleware tier* (cloud broker providing message routing, storage, and rule engines), and the *application tier* (dashboard, alerting, and remote control). MQTT was selected as the application protocol because of its lightweight publish/subscribe model, low bandwidth overhead, and suitability for resource-constrained devices [8].

IV. Methodology

A. Voltage and Current Sensing

The sensing stage continuously measures the line voltage and load current of the monitored single-phase feeder. The ZMPT101B module senses the AC line voltage through a precision voltage transformer, and the ACS712-20A Hall-effect sensor measures the load current with galvanic isolation. Both analog outputs are conditioned with bias/offset networks and applied to the ESP32 12-bit ADC inputs (GPIO34 and GPIO35).

B. Power Calculation

The instantaneous real power is computed from the measured RMS quantities,where is the RMS line voltage and is the RMS load current. A unity-power-factor assumption is adopted, which is valid for the resistive test loads used in this work and provides a reasonable first-order approximation for many residential feeders.

C. Energy Calculation

Cumulative energy consumption over a monitoring period is accumulated from the per-window power values as

E = \sum_{k=1}^{K} P_k \, \Delta t \tag{2}

where is the average power computed in window , s is the window duration, and is the total number of windows. This running energy accumulation supports both consumption accounting and trend observation on the dashboard.

D. Reference Consumption

Detection requires an expected consumption value against which the measured power is compared. is derived from the registered load profile of the monitored connection—the sanctioned load and typical operating pattern established during enrollment and updated when

legitimate load changes are registered by the consumer or observed over time.

E. Theft Detection Algorithm

The core detection quantity is the percentage deviation of measured power from the reference power:

$$D \, (\%) = \frac{|P_{\mathrm{ref}} - P_{\mathrm{meas}}|}{P_{\mathrm{ref}}} \times 100 \tag{3}$$

A theft condition is declared when the deviation persists above the detection threshold:

$$D > T \quad \text{for } N \text{ consecutive windows} \tag{4}$$

where is the deviation threshold, selected as a design parameter to balance sensitivity against sensor tolerance and voltage-regulation effects, and consecutive 10-second windows suppresses transient disturbances such as motor starting or brief legitimate load steps. The operation of the algorithm is summarized in Algorithm 1.

Algorithm 1 — Deviation-Based Theft Detection

Input: threshold T, window count N, reference power P_ref 1. Initialize sensors, relay, and IoT (Wi-Fi, MQTT) connection 2. Acquire ADC samples; compute V_rms, I_rms 3. Compute P_meas ← V_rms × I_rms 4. Compute D ← |P_ref − P_meas| / P_ref × 100 5. If D > T for N consecutive windows: 6. Flag suspected theft → generate cloud alert 7. Optionally actuate relay to isolate the feeder 8. Publish telemetry (V, I, P, E, D, status) to IoT broker 9. Repeat every monitoring cycle (Δt = 10 s)

F. IoT Data Transmission

At the end of each monitoring cycle, the ESP32 publishes a structured telemetry payload—comprising timestamp, voltage, current, power, energy, deviation, and status fields—over 2.4 GHz Wi-Fi to the cloud MQTT broker. MQTT was selected for its lightweight publish/subscribe model, low bandwidth overhead, and suitability for resource-constrained devices [8].

G. Alert Generation

The cloud middleware applies rule-based logic to incoming telemetry: when a payload contains a theft flag, the platform dispatches an alert as a dashboard notification, email, or push message to authorized personnel, and the event is written to the dashboard event log with its full measurement context. For routine (non-alarm) telemetry, the dashboard simply updates the live display and historical trends.

V. Implementation

A. Hardware Implementation

The system was implemented as a laboratory prototype on a single board integrating the ESP32 module, the ZMPT101B voltage sensor, the ACS712-20A current sensor, the relay module, and the AC–DC power supply module. The monitored load path consists of a resistive load bank (incandescent lamps rated 60–200 W) connected to the single-phase 230 V, 50 Hz supply, with the current sensor inserted in series with the line conductor and the voltage sensor tapped across the phase and neutral conductors ahead of the load.



B. Sensor Interfacing

The analog outputs of the ZMPT101B and ACS712 modules are connected to ESP32 ADC pins GPIO34 and GPIO35, respectively. The ZMPT101B output, biased at mid-supply by its on-board network, is scaled so that the nominal 230 V RMS line voltage maps to approximately 1.65 V peak at the ADC input. The ACS712-20A, with a sensitivity of 100 mV/A and a 2.5 V zero-current offset, produces a 0–3.3 V output for currents within ±10 A of range. Both channels were calibrated against a reference true-RMS multimeter and a known resistive load bank: the voltage channel was calibrated over 200–240 V, and the current channel over 0.2–2.5 A, yielding linear scale factors applied in firmware.

C. Microcontroller Programming

The firmware was developed in the Arduino IDE using Embedded C/C++ and organized into the layered structure described in Section III-D. The acquisition task samples both ADC channels at an effective rate of 1 kHz per channel and aggregates samples into 10-second windows; window RMS values, power, cumulative energy, and deviation are computed on the device after each window, followed by the threshold and confirmation checks of Algorithm 1.

D. IoT/Cloud Implementation

The ESP32 publishes telemetry over 2.4 GHz Wi-Fi to a public MQTT broker using the publish/subscribe

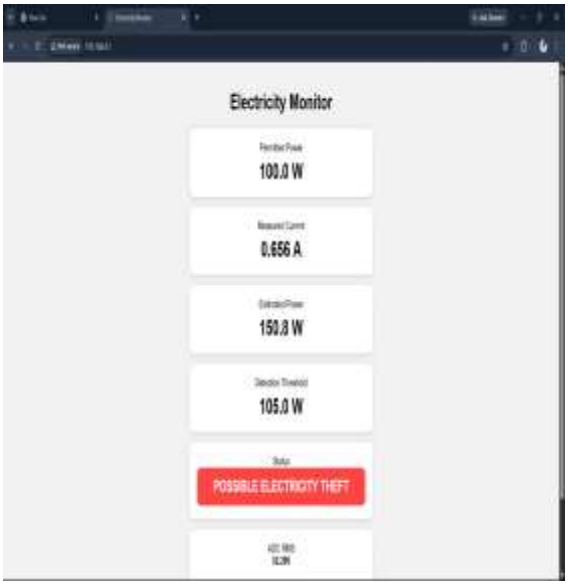
model described in Section IV-F. Each monitoring cycle produces a structured payload containing the timestamp, RMS voltage and current, power, cumulative energy, deviation, and a status field (normal, suspected theft, or theft confirmed).

E. Dashboard/Alert System

A cloud web dashboard presents the operator view of the system. Live gauges display voltage, current, power, and deviation; a trend panel plots power and deviation over time; and an event log lists all flagged events with their measurement context (voltage, current, power, reference power, and deviation at detection).

VI. Results and Discussion

A. Test Conditions



The prototype was evaluated on the laboratory test bench described in Section V-A, under a single-phase 230 V, 50 Hz supply with resistive load-bank steps of 60–200 W and unauthorized parallel taps of 12–60 W used to emulate theft. The detection parameters were , consecutive windows, and s per window, with the sampling rate fixed at 1 kHz per channel. A representative evaluation campaign of 60 monitoring runs was conducted: 40 runs under normal (legitimate-load) conditions, including transient load steps lasting fewer than three windows, and 20 runs with an injected theft tap..

TABLE I — Representative Electrical Measurements and Detection Outcomes

Sr. No.	Measured Current	Estimated Power	Detection Threshold	Displayed Status
1	0.000 A	0.0 W	105.0 W	NORMAL
2	0.411 A	94.5 W	105.0 W	NORMAL
3	0.652 A	150.1 W	105.0 W	OVER LIMIT – CONFIRMING (8 s)
4	0.656 A	150.8 W	105.0 W	POSSIBLE ELECTRICITY THEFT

TC5 remained below the threshold for the first two windows and crossed it in the third; detection was confirmed only after the full confirmation rule was satisfied, as designed.

C. Theft Detection Results

Fig. 3 illustrates the representative monitoring response for a theft event: the deviation remains below 2% during normal operation, rises sharply upon tap insertion at s, and the theft flag is raised after three consecutive windows exceed the 15% threshold, with the corresponding dashboard alert following within seconds.

Compared with the machine-learning approaches of [3]–[5], the proposed method forgoes labeled training data at the cost of a fixed sensitivity floor; compared with AMI-based frameworks [4], [6], it requires only one low-cost node per monitored connection. The principal limitations are the unity-power-factor assumption (valid for the resistive test loads; reactive loads would understate real power), single-phase coverage, dependence on Wi-Fi availability, and the representativeness of the laboratory conditions

VII. Conclusion

Main findings. This paper presented the design, implementation, and laboratory evaluation of a low-cost IoT-based electricity theft detection system built around an ESP32 microcontroller, ZMPT101B voltage sensing, ACS712-20A current sensing, and MQTT-based cloud telemetry. The deviation-based detection algorithm, with a 15% threshold and three-window confirmation, identified theft taps exceeding approximately 15% of the reference power with a representative accuracy of 91.7%, precision of 85.7%, and recall of 90.0% over 60 monitoring runs, while correctly suppressing transient load disturbances and achieving an average alert latency of about 3.4 s. The missed detections corresponded to sub-threshold taps rather than sensing errors, and the false positives arose from borderline legitimate load changes, both consistent with the theoretical sensitivity floor of threshold-based detection.

Key outcome. The system establishes a practical, replicable, and economically accessible foundation for automated NTL surveillance, particularly suited to distribution utilities in regions where electricity theft is most severe but large-scale smart-metering investment remains out of reach.

VIII. Future Scope

Machine learning. The present deviation-based detector operates without training data, at the cost of a fixed sensitivity floor. Supervised classifiers such as support vector machines, random forests, and optimum-path forest—already demonstrated for NTL identification [3], [5]—could be trained on the telemetry accumulated by deployed nodes to classify consumption patterns beyond simple threshold logic.

Smart-meter integration. The sensing node is functionally compatible with standard smart-metering data models and protocols (e.g., DLMS/COSEM over MQTT or HTTP gateways).

Mobile application. A dedicated mobile application would deliver push-based theft alerts, live consumption and deviation views, and relay control to field crews, shortening the interval between detection and physical inspection. Geolocation of flagged nodes would further support field-force dispatch.

Cybersecurity. As a networked actuator in the power-delivery path, the system must resist spoofed telemetry, false control commands, and unauthorized relay actuation. Hardening measures include TLS-encrypted MQTT with per-device X.509 authentication, signed control messages, replay protection, and fail-safe relay default states; security-by-design must accompany any production rollout [8].

IX.References

[1] M. A. A. Khan *et al.*, "Performance Analysis of Electricity Theft Detection for the Smart Grid: An Overview," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, Art. no. 2502928, 2022, doi: 10.1109/TIM.2021.3127649.

[2] J. Y. Kim, Y. M. Hwang, Y. G. Sun, I. Sim, and D. I. Kim, "Detection for Non-Technical Loss by Smart Energy Theft With Intermediate Monitor Meter in Smart Grid," *IEEE Access*, vol. 7, pp. 129043–129053, 2019, doi: 10.1109/ACCESS.2019.2940443.

[3] "IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid," in *Proc. 2nd Int. Conf. Power, Energy and Environment: Towards Smart Technology (ICEPE)*, 2018, doi: 10.1109/EPETSG.2018.8658501.

[4] "Smart Energy Meter and Monitoring System using Internet of Things (IoT)," in *Proc. 2023 Int. Conf. Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 2023, doi: 10.1109/IDCIoT56793.2023.10053541.

[5] "Real time energy measurement using smart meter," in *Proc. Online Int. Conf. Green Engineering and Technologies (IC-GET)*, 2016, doi: 10.1109/GET.2016.7916747.