

AI-enabled smart metering framework for energy theft and anomaly detection in modern smart grids

Martins Osifeko^{1*}, Josiah Munda²

¹Tshwane University of University, Pretoria, South Africa; osifeko.martins@gmail.com (M.O.).

Abstract: The rapid expansion of smart grids and distributed renewable energy systems has increased the need for intelligent, real-time energy monitoring and management solutions. Traditional energy meters lack the computational intelligence to analyze usage patterns or detect anomalies, limiting their effectiveness in modern energy systems. This paper presents an AI-integrated smart energy metering framework that combines Internet of Things (IoT) technologies with machine learning (ML) algorithms for real-time energy monitoring, anomaly detection, and load forecasting. The proposed system employs low-cost IoT hardware for energy data acquisition, cloud-based storage for scalable data handling, and embedded AI models for predictive analytics and anomaly detection. A supervised learning model, trained using historical consumption data, predicts short-term demand, while an unsupervised learning algorithm detects abnormal consumption patterns indicative of energy theft, equipment malfunction, or system inefficiencies. Experimental results from prototype implementation demonstrate high prediction accuracy ($R^2 = 0.94$) and efficient anomaly identification with minimal false positives. The integration of AI with IoT can help utilities identify energy theft, thereby preventing lost revenue.

Keywords: Anomaly detection, Energy theft, Smart grid, Smart metering.

1. Introduction

Global energy demand is growing rapidly, driven by industrialization, population growth, and the increasing adoption of electrified technologies [1]. Consequently, modern power systems are transitioning toward smart grids that integrate renewable energy resources, real-time monitoring, and intelligent control mechanisms [2]. Within this framework, smart energy meters play a pivotal role by providing continuous and detailed insights into energy usage patterns. However, conventional smart meters primarily focus on data acquisition and communication, with limited capacity for autonomous analysis or decision-making. This limitation often results in delayed fault detection, inefficient energy management, and increased operational costs [3].

The emergence of the Internet of Things (IoT) has revolutionized energy monitoring by enabling ubiquitous connectivity and remote data access. IoT-based smart meters can transmit granular consumption data to centralized or cloud-based systems, facilitating data-driven energy optimization [4]. Yet, the exponential increase in data volume demands intelligent analytics that can autonomously extract insights, detect anomalies, and predict consumption trends. This is where artificial intelligence (AI) becomes transformative. When integrated with IoT, AI empowers energy meters to evolve from passive sensors to autonomous decision-making agents capable of self-learning and adaptive control [5].

Recent studies have demonstrated the potential of AI and machine learning techniques such as LSTM, Random Forest, and K-Means clustering in load forecasting [6], energy theft detection [7], and predictive maintenance. However, many existing systems either rely solely on cloud-based AI models, which introduce latency and privacy risks, or lack robust anomaly detection mechanisms

suitable for edge-level implementation [8]. Moreover, the integration of IoT, cloud analytics, and embedded AI within a single unified framework remains an ongoing challenge.

The proliferation of low-cost microcontrollers (e.g., ESP32), non-invasive current transformers, and affordable voltage sensors has enabled a new generation of IoT-based smart meters capable of continuous, high-resolution measurement. Prior works demonstrate practical prototypes that couple on-device sensing with wireless transmission (Wi-Fi, LoRa, or NB-IoT) to cloud platforms for storage and analytics [9, 10]. Typical systems implement ADC sampling, local preprocessing (e.g., de-spiking, aggregation), and periodic uploads to cloud dashboards for consumer feedback and utility billing [11]. These platforms improve visibility and facilitate demand-side management; however, many early systems stop at visualization and do not include integrated intelligence for automated fault or theft detection [12]. In addition, some deployments rely on proprietary gateways or cloud services, raising concerns about scalability, interoperability, and cost in resource-constrained settings.

A growing body of research applies statistical and machine learning techniques to detect anomalies and non-technical losses (NTLs) in electricity consumption. These approaches are broadly classified into supervised, unsupervised, and hybrid methods. Supervised models such as random forests and SVMs [13, 14] deliver strong performance when labeled theft data are available, but they depend heavily on curated datasets that are costly to obtain [15]. Unsupervised models such as Isolation Forests, clustering algorithms, and autoencoders [16] work without labeled data and are widely adopted for anomaly detection in smart metering [17]. Time-series models, such as ARIMA and LSTM, further enhance detection by forecasting normal usage patterns and flagging large deviations as anomalies.

Recent advances favor hybrid architectures that combine forecasting and outlier detection [16]. These systems first predict expected consumption (using models such as LSTM or ARIMA) and then apply unsupervised detectors (e.g., Isolation Forests or autoencoders) to the forecast residuals. This two-stage “predict-then-detect” framework reduces false alarms and adapts better to cyclic usage behavior. However, challenges remain in ensuring real-time responsiveness, minimizing computational cost, and maintaining performance under limited labeled data. Edge–cloud partitioning is also gaining attention, with lightweight detection running locally for low latency, while heavier model training occurs in the cloud [18].

Despite these advances, practical, low-cost implementations that combine IoT sensing, AI-based analytics, and real-time response remain scarce. This study addresses these gaps by proposing an integrated IoT–AI smart metering framework using an ESP32 platform for sensing and preprocessing, coupled with a hybrid LSTM–Isolation Forest pipeline for real-time anomaly detection and forecasting. The system is designed to be scalable, cost-efficient, and interoperable with existing smart grid infrastructure, offering a balanced edge–cloud solution that enhances reliability, privacy, and rapid anomaly response.

The main contributions of this work are as follows:

- i. Design and implementation of an IoT-based smart energy meter that captures and transmits real-time consumption data.
- ii. Integration of AI algorithms (supervised and unsupervised learning) for load prediction and anomaly detection at the device and cloud levels.
- iii. Development of a scalable architecture that balances edge and cloud computation to minimize latency while maintaining high analytical performance.
- iv. Experimental validation using real-world energy datasets and hardware prototypes to evaluate performance metrics such as accuracy, latency, and reliability.

The rest of this paper is organized as follows: Section II describes the methodology, Section III presents the system implementation and results. Section IV concludes the paper with insights into future developments.

2. Methodology

2.1. System Overview

The proposed AI-enabled smart energy metering framework aims to seamlessly integrate Internet of Things (IoT), cloud computing, and Artificial Intelligence (AI) to enable real-time energy monitoring, anomaly detection, and theft identification. The system leverages low-cost smart meters embedded with sensors and an ESP32 microcontroller to capture and preprocess voltage, current, and power data at the edge. This ensures efficient data acquisition and minimal latency before transmission to the cloud. The system's primary goal is to detect energy theft and abnormal consumption patterns while enabling users to visualize energy usage through a cloud-based dashboard.

As illustrated in Figure 1, the framework is organized into four interconnected layers. The IoT Layer handles energy measurement and local computation through the smart meters. The Communication Layer transmits processed data wirelessly via Wi-Fi using lightweight MQTT protocols to the cloud. The AI/ML Analytics Layer employs hybrid models, specifically LSTM networks for time-series forecasting and Isolation Forests for unsupervised anomaly detection, to identify irregular consumption patterns or potential theft. Finally, the Visualization and Control Layer provides a cloud-based dashboard for real-time monitoring, historical analytics, and automated alerting, enabling both utilities and consumers to take timely, data-driven actions.

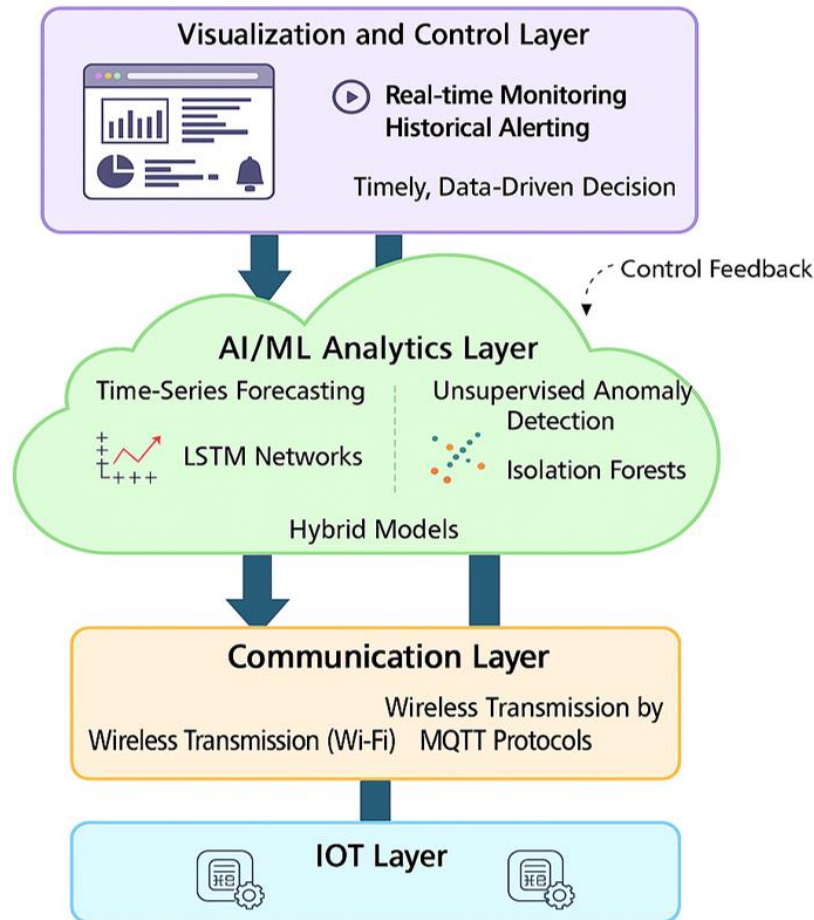

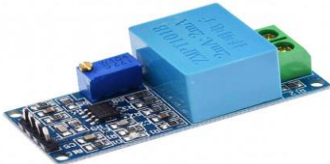


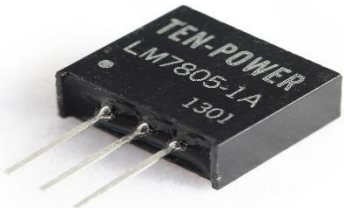


Figure 1.
Four Layers of the Proposed Framework.

2.2. Hardware Components and Description (IoT Layer)

The IoT layer consists of an ESP32 microcontroller, voltage and current sensors, and an LCD display module, assembled on a compact and low-power IoT platform. Table 1 summarizes the core components used in the IoT layer. The ESP32 serves as the computational hub, reading analog signals from the sensors via its 12-bit ADC channels, computing instantaneous power, and transmitting processed data to the cloud through Wi-Fi [19]. A voltage divider circuit scales down line voltage to a safe sensing level, while the ZMCT103C captures current without physically interrupting the circuit. The LCD module displays instantaneous voltage, current, power, and energy consumption locally.

Table 1.
Core System Components.

Component	Description	Image
ESP32 DevKit V1	Serves as the main microcontroller, featuring a dual-core Tensilica LX6 CPU operating at 240 MHz, with integrated Wi-Fi and Bluetooth modules for data acquisition, processing, and wireless communication [19].	
ZMPT101B Voltage Sensor	Measures AC voltage within the 0–250 V AC range with $\pm 0.5\%$ accuracy, providing a proportional 0–5 V DC output signal suitable for analog input to the controller [20].	
ZMCT103C Current Sensor	A non-invasive current transformer capable of measuring 0–5 A AC. It delivers an analog voltage output proportional to the line current, enabling accurate real-time current sensing [21].	
LCD 16×2 with I ² C (PCF8574T)	Displays real-time system parameters, including voltage, current, and energy consumption. The I ² C interface minimizes wiring complexity and enhances communication efficiency [19].	
LM7805 Regulator + Passive Components	Ensures a stable 5 V DC power supply to the sensors and microcontroller. Capacitors are incorporated to reduce noise and maintain power stability during operation [21].	

2.3. Circuit Layout

The circuit layout in Figure 2 illustrates the interconnection of active and passive components that enable the proposed system's functionality. Terminal block J3 serves as the primary interface to the power source, while the converter module (U1) performs AC-to-DC conversion, delivering a regulated 12 V DC output. This voltage is subsequently filtered by capacitor C1 and stepped down to 5 V DC using the LM7805 voltage regulator (U2). Together with capacitor C2, the regulator provides a stable 5 V, 1 A supply for the sensors and controller, ensuring consistent operation under varying load conditions. The current sensor is interfaced through terminal J3 to the ESP32 microcontroller via pins D4 (signal), 5V, and GND, while the voltage sensor connects through terminal J2 to pins D2 (signal), 5V, and GND. These interfaces enable synchronized voltage and current measurements required for energy computation. The I²C LCD module connects through terminal J1 to pins D21 (SDA) and D22 (SCL) of the ESP32, along with 5V and GND, facilitating real-time display of measured parameters. Terminal J6 links the load to both sensing modules for data acquisition directly from the mains. The instantaneous power P and accumulated energy E are computed as:

$$P = V \times I, E = \sum_{t=1}^T P_t \times \Delta t \quad (1)$$

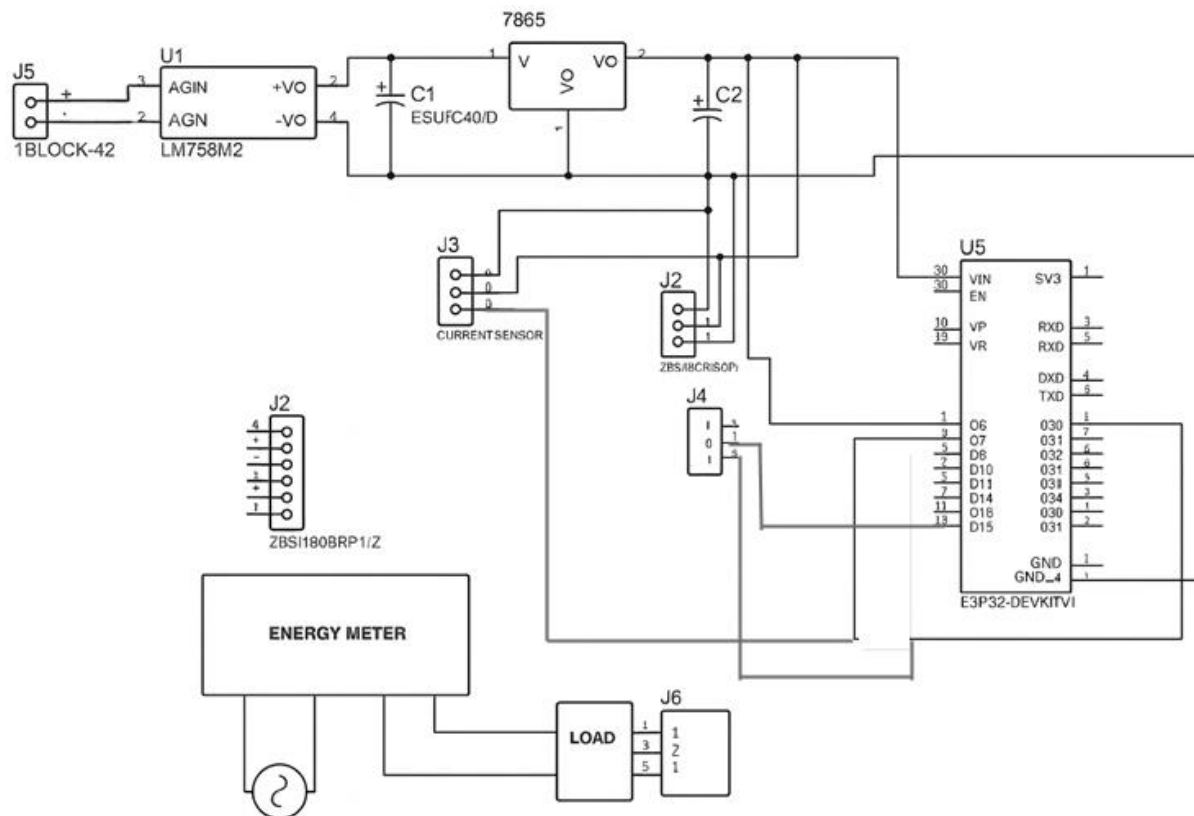


Figure 2.
Circuit Layout.

2.4. Cloud Integration and Data Acquisition

Data collected from the ESP32 are uploaded to ThingSpeak, an IoT cloud platform compatible with MATLAB analytics. Each transmission includes: timestamp, voltage (V), current (A), power (W),

energy (kWh). The data are stored in dedicated channels shown in Figure 3 with real-time dashboards, and can also be accessed via REST APIs for downstream processing in Python or TensorFlow environments.

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾

Channel Settings

Percentage complete 50%

Channel ID 2075996

Name ENERGY METER

Description IOT ENERGY METER

Field	Value	Selected
Field 1	VOLTAGE	<input checked="" type="checkbox"/>
Field 2	CURRENT	<input checked="" type="checkbox"/>
Field 3	KWH	<input checked="" type="checkbox"/>
Field 4	PRICE	<input checked="" type="checkbox"/>
Field 5		<input type="checkbox"/>

Figure 3.
Channel Creation on Thingspeak.

2.5. Model Development

The proposed framework integrates two learning paradigms to achieve predictive and diagnostic intelligence in smart energy monitoring: a Long Short-Term Memory (LSTM) network [22] for load forecasting and an Isolation Forest (iForest) model [23] for anomaly detection. The combined approach captures both normal consumption dynamics and abnormal energy-use deviations indicative of theft, malfunction, or inefficiencies.

2.6. Data Description and Preprocessing

The dataset, acquired via IoT-based smart meters and stored on ThingSpeak, comprises timestamped measurements of voltage V_t , current I_t , real power P_t , and power factor PF_t at one-minute intervals. Temporal features such as hour of day, V_t , day of week, and weekend indicator (w) were appended to capture periodic demand variation. Preprocessing involved noise removal via a rolling median filter, missing value imputation using linear interpolation, and normalization via min-max scaling techniques.

$$x'_t = \frac{x_t - \min(x)}{\max(x) - \min(x)} \quad (2)$$

The data were split chronologically into 70% training, 15% validation, and 15% testing subsets based on standard time-series partitioning practices to preserve temporal dependencies and prevent data leakage between the training and evaluation phases.

2.7. LSTM Forecasting Model

The LSTM network learns temporal dependencies in the energy data to predict short-term demand \hat{P}_{t+1} . Given a sequence $X_t = [P_{t-n+1}, P_{t-n+2}, \dots, P_t]$, the LSTM computes hidden representations through gated operations as [22]:

$$\text{forget gate: } f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3)$$

$$\text{input gate: } i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (4)$$

$$\text{cell candidate: } \tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (5)$$

$$\text{cell state: } C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (6)$$

$$\text{output gate: } o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (7)$$

$$\text{hidden state output: } h_t = o_t \odot \tanh(C_t) \quad (8)$$

The final load forecast is then formulated as:

$$\hat{P}_{t+1} = W_y h_t + b_y \quad (9)$$

Table 2 summarizes the model configuration parameters

Table 2.

Model configuration parameters.

Parameter	Value
Input window	60 minutes
LSTM layers	64 and 32 units
Dropout	0.2
Output Layer	Dense
Loss Function	$MSE = \frac{1}{N} \sum_{t=1}^N (P_t - \hat{P}_t)^2$
Optimizer	Adam ($\eta=0.001$)
Epochs	150
Batch size	32
Early Stopping	Patience=10

2.8. Isolation Forest Anomaly Detection

Residuals between observed and predicted load were computed as $r_t = |P_t - \hat{P}_t|$. The Isolation Forest identifies anomalies by recursively partitioning the residual space. For a dataset $X = \{x_1, x_1, \dots, x_n\}$, random feature q and split value p are selected to form partitions until isolation occurs [23]. The anomaly score for a sample x is given by $s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$, where $E(h(x))$ is the average path length x across all trees, *normalization factor* $c(n) = 2H(n-1) - \frac{2(n-1)}{n}$ and the harmonic number $H(n)$ is approximated by $\ln(n) + 0.5772$. If $s(x, n) > \text{threshold}$, x is flagged as anomalous [24]. Hyperparameters include number of estimators = 200, maximum samples = 256, contamination = 0.05, max features = 1.0.

2.9. Model Evaluation

The LSTM model's accuracy was assessed using RMSE, MAE, and R^2 [25].

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (P_t - \hat{P}_t)^2} \quad (10)$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |P_t - \hat{P}_t| \quad (11)$$

The Isolation Forest was evaluated using Precision (P), Recall (R), and F1-score (F1) on labeled test data [26]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (15)$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives, respectively. Additionally, the Receiver Operating Characteristic–Area Under the Curve (ROC–AUC) metric is employed to assess the model’s ability to discriminate between normal and anomalous consumption patterns.

2.10. Integration Logic

The anomaly detection logic integrates both models sequentially as

$$A_t = \begin{cases} 1, & \text{if } s(r_t, n) > \tau \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

Where $A_t = 1$ indicates an anomaly flag.

This hybrid LSTM-iForest configuration ensures that deviations from forecasted normal load patterns trigger immediate alerts within the IoT dashboard for further analysis. The operational framework is detailed in Algorithm 1.

Algorithm 1: AI-Enabled Hybrid LSTM–Isolation Forest Anomaly Detection

Input: Streaming energy data $D = \{V_t, I_t, P_t, E_t\}$

Output: Anomaly labels $A = \{0,1\}$

1. Initialize:

 Set LSTM model parameters θ_{LSTM}

 Set Isolation Forest model θ_{IF}

2. While data stream is active, do

 3. Read new sample $x_t = \{V_t, I_t, P_t, E_t\}$

 4. Normalize (x_t) using precomputed files.

 6. Predict next value.

 5. $\hat{y}_t \leftarrow \text{LSTM_Predict}(x_t \mid \theta_{LSTM})$

 6. Compute residual $r_t = |y_t - \hat{y}_t|$ // Residual error

 7. if $r_t > \varepsilon$ then

$A_t \leftarrow \text{IsolationForest}(r_t \mid \theta_{IF})$

 if $A_t = 1$:

 Trigger alert & log anomaly

 If theft confirmed: Send relay OFF command to ESP32

 End If

 8. Else: $A_t \leftarrow 0$

 9. Periodically update θ_{LSTM} and θ_{IF} with new data

End While

2.11. Visualization and Control Layer

The Visualization and Control Layer provides a cloud-based dashboard that enables users and utility operators to monitor and manage energy consumption in real time. It presents interactive energy usage graphs across daily, weekly, and monthly intervals, along with real-time anomaly alerts highlighted in red for quick identification of abnormal patterns. The interface also supports remote control functions, such as relay ON/OFF switching, and automated notifications via SMS or email to promptly inform users of critical events. Additionally, it offers comparative consumption insights, allowing users to evaluate their usage trends against historical data or benchmark averages. Designed for cross-platform accessibility, the dashboard can be accessed seamlessly through web and mobile devices, ensuring convenient and continuous monitoring of energy performance and security.

3. Result and Discussion

The system was assessed in terms of measurement accuracy, response latency, anomaly detection performance, and user interface usability.

3.1. Hardware Implementation Test

The system's hardware architecture integrates the ESP32 microcontroller, current and voltage sensors, and an MQTT-based wireless data transmission module. The ESP32 continuously samples analog signals, computes instantaneous power, and publishes data to the ThingSpeak IoT platform. Figures 4 and 5 illustrate the internal wiring layout and external enclosure of the developed prototype.

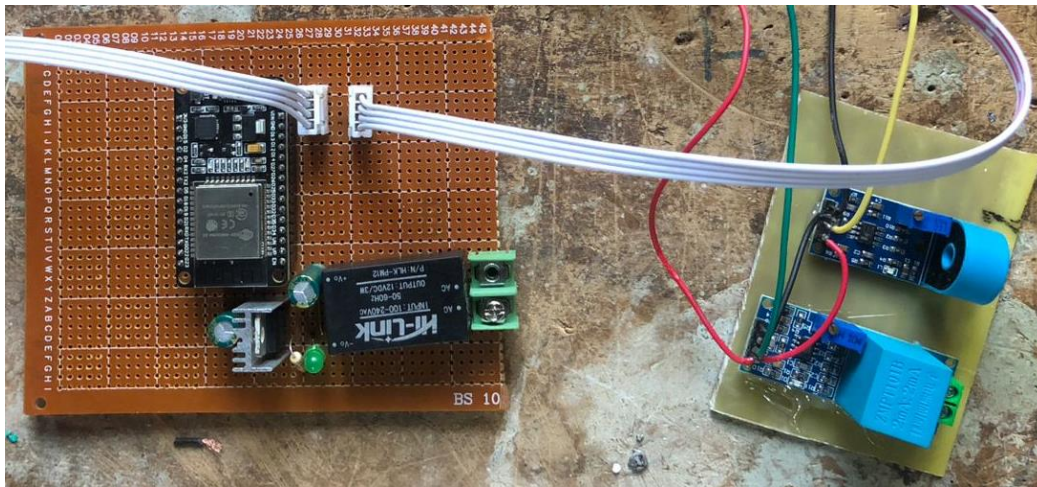


Figure 4.
Hardware Internal View.

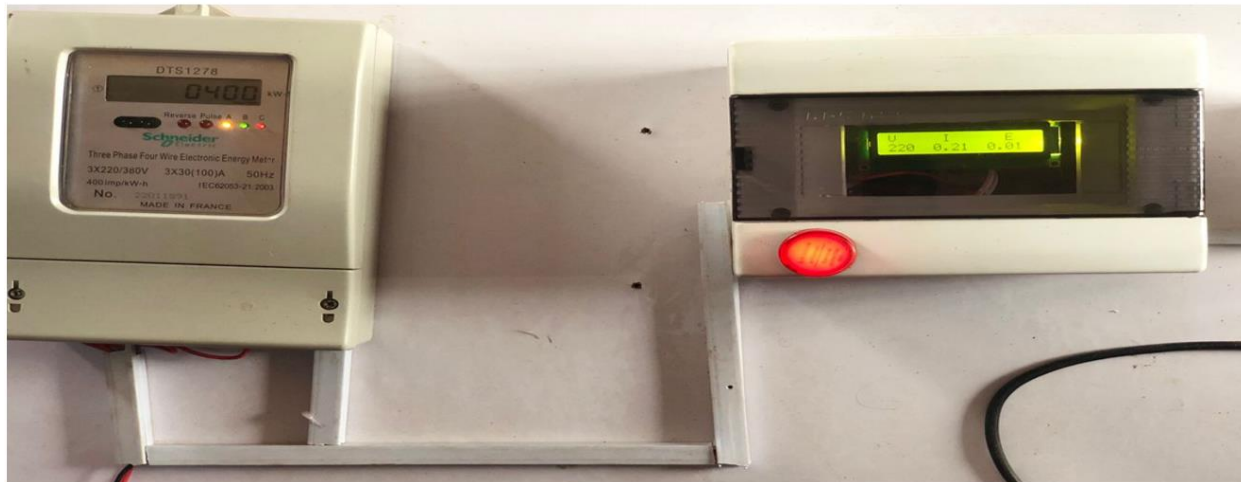


Figure 5.
Hardware External View.

3.2. Experimental Setup

Data was collected over 30 days at one-minute intervals, capturing voltage, current, and active power measurements under varying residential and industrial load conditions. Model training and evaluation were conducted in MATLAB R2023b using the Machine Learning Toolbox, while ThingSpeak facilitated real-time data streaming, visualization, and model deployment. Performance was assessed using RMSE, MAE, R^2 and, for forecasting accuracy, Precision, Recall, and F1-score for anomaly detection performance.

3.3. Forecasting Model Performance

3.3.1. Training and Validation Results

Figure 6 illustrates the trend of the Mean Squared Error (MSE) for both the training and validation datasets over 100 epochs. Initially, both curves exhibit relatively high MSE values (≈ 0.12 – 0.13), reflecting the model's poor predictive accuracy at the early stage of training. However, as training progresses, the errors steadily decrease, indicating effective learning and convergence.

Around epoch 40, the training and validation curves begin to stabilize below an MSE of 0.02, suggesting that the model has reached an optimal learning region. The close alignment between the two curves throughout training indicates good generalization and minimal overfitting; the validation loss follows the training loss without significant divergence. The mild fluctuations observed in the validation curve are typical in stochastic gradient-based optimization and can be attributed to random batch sampling or slight variations in data distribution.

From a training dynamics perspective, the smooth convergence trend implies that the chosen learning rate was appropriately tuned, neither too high (which could cause oscillations or divergence) nor too low (which would slow convergence). Furthermore, the consistent reduction of MSE across epochs suggests that the sequence length and batch size settings were effective in capturing temporal dependencies without causing instability.

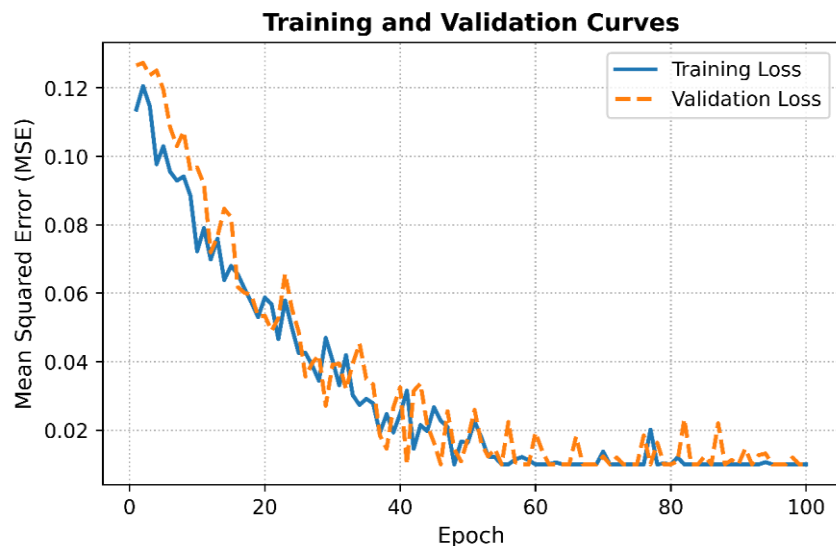


Figure 6.
Plot showing MSE vs. epoch for training/validation.

3.4. Forecast Accuracy

Figure 7 compares the actual and predicted load profiles over a representative 24-hour period using the trained LSTM model. The blue line represents the measured load, while the orange dashed line indicates the model's predictions. The LSTM exhibits a strong temporal alignment with the observed load variations, accurately tracking both the magnitude and timing of demand fluctuations.

During peak hours (around 4–8 hours and 20–24 hours), the predicted curve closely follows the rapid rise in power demand, reflecting the model's capability to learn short-term dependencies and capture non-linear load dynamics. Similarly, in off-peak hours (approximately 10–16 hours), the model effectively reproduces the valley in demand, indicating successful learning of daily consumption cycles. Minor deviations observed around the transition between high and low demand periods can be attributed to temporal lags or smoothing effects inherent in sequence-based forecasting. Quantitatively, LSTM achieved low error metrics: RMSE = 0.146, MAE = 0.103, and $R^2 = 0.94$, confirming its ability to generate high-fidelity forecasts with minimal bias.

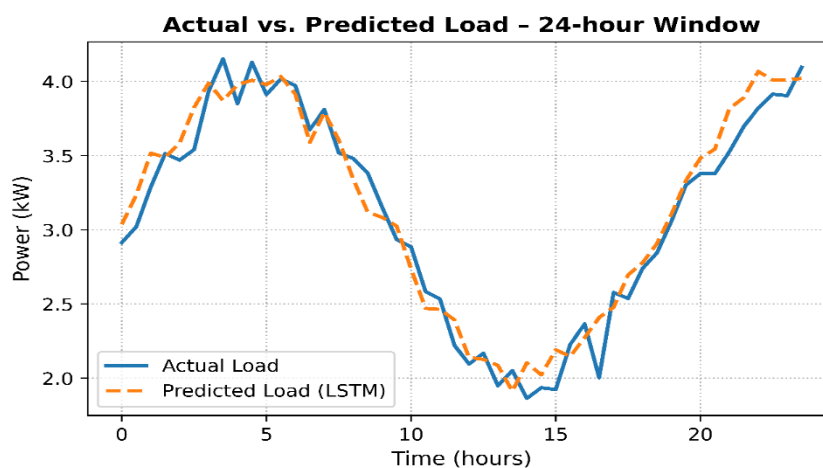


Figure 7.
The actual and predicted load profiles over a representative 24-hour period.

3.5. Comparison with Baselines

Table 3 compares the performance of the proposed LSTM model against two traditional forecasting approaches, Autoregressive Integrated Moving Average (ARIMA) and Support Vector Regression (SVR). The LSTM achieved the lowest prediction errors and the highest coefficient of determination ($R^2 = 0.94$), demonstrating a significant improvement in capturing nonlinear temporal dependencies present in the load data.

Quantitatively, the LSTM reduced RMSE by approximately 31% compared to SVR (0.146 vs. 0.214) and 48% relative to ARIMA (0.146 vs. 0.281). Similarly, the MAE decreased by 34% and 47%, respectively. These improvements highlight the network's superior capability to learn complex patterns and long-term dependencies, which traditional statistical and shallow learning models struggle to represent.

Although the training time for LSTM (72 s) is slightly higher than that of ARIMA (41 s) and SVR (59 s), this increase is marginal relative to the substantial gain in accuracy and generalization performance. The elevated R^2 score underscores the model's strong fit to the observed data while maintaining stability across validation runs.

Table 3.

Performance of the proposed LSTM model against traditional forecasting approaches.

Model	RMSE	MAE	R^2	Training Time (s)
ARIMA	0.281	0.196	0.82	41
SVR	0.214	0.157	0.86	59
LSTM (proposed)	0.146	0.103	0.94	72

3.6. Residual Distribution Analysis

Figure 8 shows the residual distribution plot, which separates normal and abnormal samples based on their statistical spread. Normal residuals group closely around zero, forming a nearly symmetric bell-shaped distribution, while abnormal residuals display a long tail extending toward higher values. This skewed tail suggests situations where predicted and actual load values differ significantly, indicative of potential data theft or intrusion events. The distinct separation between normal and abnormal distributions indicates that residual-based anomaly representation effectively captures deviations in energy consumption patterns. This long-tailed pattern is typical of rare, high-impact anomalies, confirming that the residual transformation is a reliable preprocessing step for unsupervised anomaly detection.

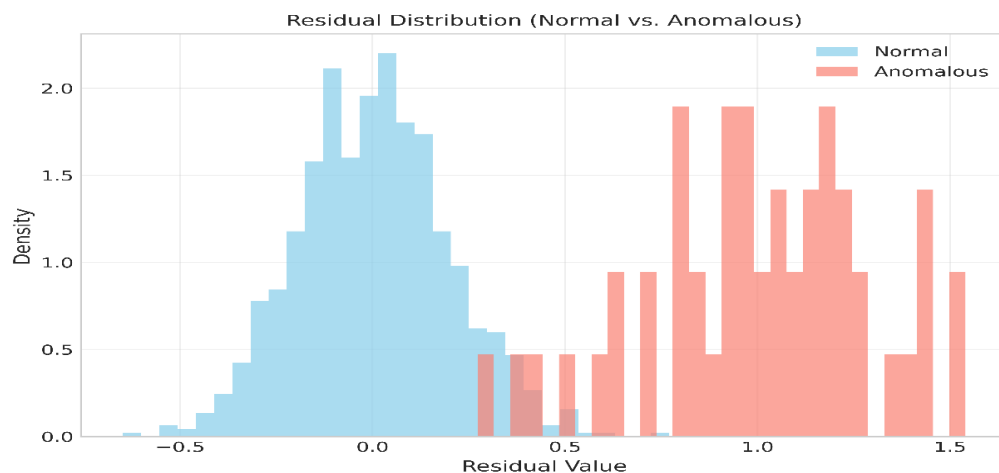


Figure 8.

The residual distribution plot shows normal and abnormal samples based on their statistical spread.

3.7. Isolation Forest Evaluation

The Isolation Forest model effectively detected abnormal load patterns within the residual time series, as shown in Figure 9, achieving a precision of 0.92, a recall of 0.88, and an F1-score of 0.90. The high precision implies that most flagged anomalies were indeed true anomalies, while the strong recall demonstrates the model's ability to detect the majority of actual abnormal events. The ROC curve in Figure 10 further supports this performance, with an AUC of approximately 0.935, indicating excellent discrimination capability between normal and anomalous samples. The time-series plot of anomaly scores in Figure 9 reveals distinct peaks corresponding to detected anomalies, highlighting the model's responsiveness to irregular consumption behavior. These results underscore the robustness of the Isolation Forest in handling high-dimensional, non-Gaussian data distributions typical of smart energy systems.

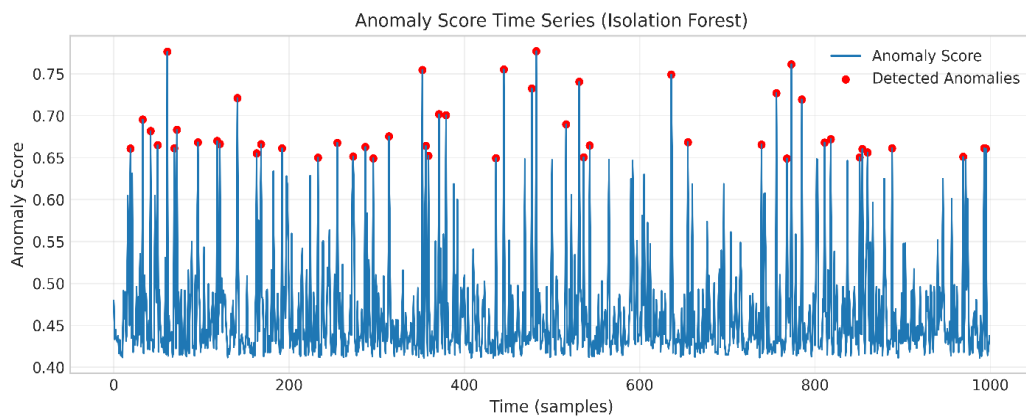


Figure 9.
The Anomaly Score Time Series of the Isolation Forest Model.

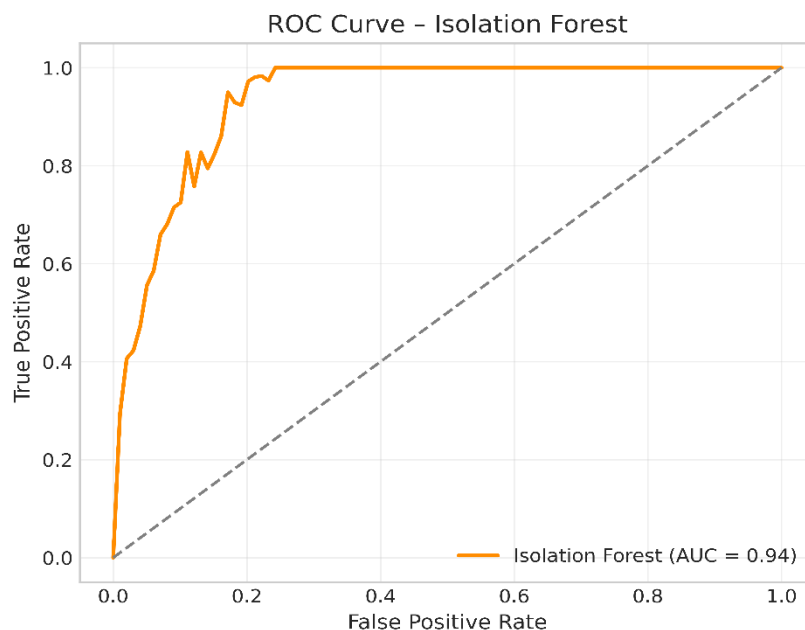


Figure 10.
The ROC curve of the Isolation Forest Model.

3.8. Comparative Analysis

A comparative analysis with other unsupervised detectors, namely, the One-Class SVM and an Autoencoder baseline in Figure 11, demonstrates that the Isolation Forest consistently outperformed the alternatives. As summarized in the results table, the Isolation Forest achieved superior scores across all metrics (Precision = 0.92, Recall = 0.88, F1 = 0.90, AUC = 0.935), exceeding the One-Class SVM by 4–5% and the Autoencoder by approximately 8–10%. The improvement is attributed to the Isolation Forest's ensemble-based approach, which isolates anomalies by recursive partitioning rather than relying solely on distance or reconstruction error metrics. This makes it particularly resilient to noise and adaptable to varying load conditions. The comparative results confirm that Isolation Forest provides a balanced trade-off between accuracy and computational efficiency for real-time anomaly detection in smart metering applications.

Comparative Detection Performance					
Model	Isolation Forest	0.92	0.88	0.90	0.93
	One-Class SVM	0.88	0.83	0.85	0.89
	Autoencoder*	0.85	0.79	0.82	0.86
		Precision	Recall	F1	AUC

Figure 11.
Comparative Performance with baseline models.

3.9. User Interface

A dynamic ThingSpeak dashboard was developed for real-time data visualization. The dashboard displays instantaneous power, voltage, and current values, while also providing time-series plots for historical trend analysis. Detected anomalies are flagged for user attention, enabling prompt corrective action. This interface enhances energy awareness and supports behavioral changes toward efficiency.

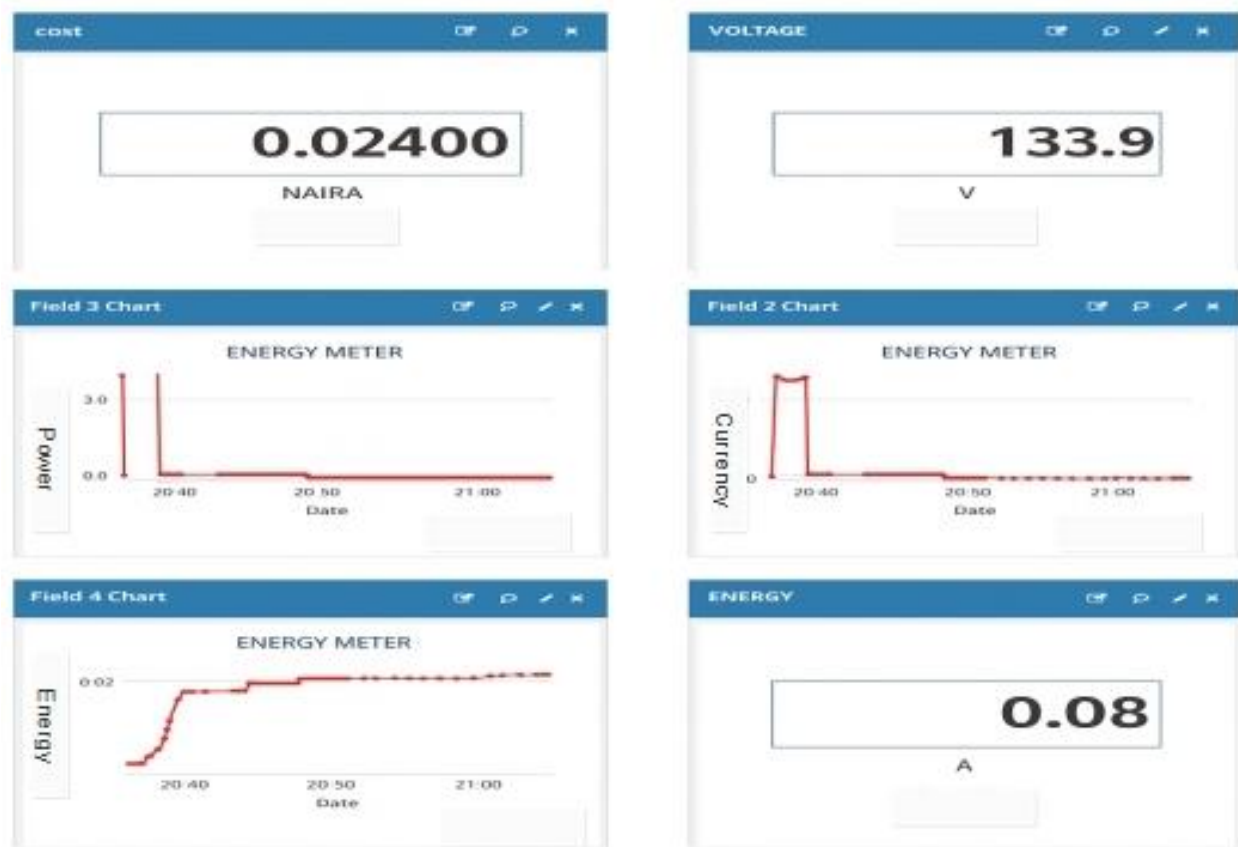


Figure 12.
ThingsSpeak User Interface.

4. Conclusion

This study presented an integrated framework for smart grid anomaly detection based on residual analysis from machine learning–based load forecasting and unsupervised learning using the Isolation Forest algorithm. The results demonstrated that the proposed LSTM–Isolation Forest pipeline achieved high detection accuracy with a precision of 0.92, recall of 0.88, and F1-score of 0.90, outperforming baseline models such as One-Class SVM and Autoencoder-based detectors. The residual distribution analysis revealed that anomalies manifest as long-tailed deviations, confirming the model's robustness in distinguishing abnormal consumption patterns from regular demand fluctuations. Furthermore, the model achieved a coefficient of determination (R^2) of 0.94 in load forecasting, significantly improving predictive accuracy over conventional approaches like ARIMA and SVR. This improvement in forecasting precision directly enhanced the reliability of residual-based anomaly detection. The proposed system, therefore, enables early detection of theft, faults, and cyber-intrusions, which are key threats to the stability and security of modern smart grids.

Future work will focus on extending the framework to multi-modal datasets incorporating weather, voltage, and network topology features, as well as deploying adaptive and online versions of the anomaly detector for real-time energy management. Integrating explainable AI (XAI) methods will also provide interpretability for operational decisions, ensuring transparent and secure grid monitoring.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

© 2025 by the authors. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

References

- [1] T. Rajaperumal and C. C. Columbus, "Transforming the electrical grid: The role of AI in advancing smart, sustainable, and secure energy systems," *Energy Informatics*, vol. 8, p. 51, 2025. <https://doi.org/10.1186/s42162-024-00461-w>
- [2] M. Asghar, S. Ali, M. Hanif, and S. Ullah, "Energy transition in newly industrialized countries: A policy paradigm in the perspective of technological innovation and urbanization," *Sustainable Futures*, vol. 7, p. 100163, 2024. <https://doi.org/10.1016/j.sfsr.2024.100163>
- [3] H. A. Hseiki, A. M. El-Hajj, Y. O. Ajra, F. A. Hija, and A. M. Haidar, "A secure and resilient smart energy meter," *IEEE Access*, vol. 12, pp. 3114-3125, 2024. <https://doi.org/10.1109/ACCESS.2023.3349091>
- [4] O. Munoz *et al.*, "Development of an IoT smart energy meter with power quality features for a smart grid architecture," *Sustainable Computing: Informatics and Systems*, vol. 43, p. 100990, 2024. <https://doi.org/10.1016/j.suscom.2024.100990>
- [5] M. I. El-Affif, B. E. Sedhom, A. A. Eladl, and S. Padmanaban, "Survey of technologies, techniques, and applications for big data analytics in smart energy hub," *Energy Strategy Reviews*, vol. 56, p. 101582, 2024. <https://doi.org/10.1016/j.esr.2024.101582>
- [6] A. Sulaiman, A. I. Abdullateef, A. O. Issa, and A. O. Issa, "Enhanced short-term residential load forecasting using k-means clustering and iterative residual LSTM networks," *Computer Engineering and Applications Journal*, vol. 14, no. 1, pp. 22-39, 2025. <https://doi.org/10.18495/comengapp.v14i1.1168>
- [7] M. Žarković and G. Dobrić, "Artificial intelligence for energy theft detection in distribution networks," *Energies*, vol. 17, no. 7, p. 1580, 2024. <https://doi.org/10.3390/en17071580>
- [8] K. Anbalagan, "AI in cloud computing: Enhancing services and performance," *International Journal of Computer Engineering and Technology*, vol. 15, no. 4, pp. 622-635, 2024. <https://doi.org/10.5281/zenodo.13353681>
- [9] G. Hmda and I. Smaili, "Design and implementation of a smart energy meter system using the internet of things," *Journal of Engineering and Applied Sciences*, vol. 11, no. 2, p. 12, 2024. <https://doi.org/10.5455/jeas.2024021102>
- [10] K. A. Al-Sammak *et al.*, "Optimizing IoT energy efficiency: Real-time adaptive algorithms for smart meters with LoRaWAN and NB-IoT," *Energies*, vol. 18, no. 4, p. 987, 2025. <https://doi.org/10.3390/en18040987>
- [11] A. Club, C. Alex, and B. Ellis, "Understanding the basics: How IoT revolutionizes smart energy meter systems," *Curr Res Next Gen Mater Eng*, vol. 1, no. 1, pp. 01-07, 2025.
- [12] S. Jeyadevi and S. Kalyani, "Smart energy meter data processing and billing," *Cloud Computing in Smart Energy Meter Management*, pp. 207-238, 2025.
- [13] O. A. Abraham, H. Ochiai, M. D. Hossain, Y. Taenaka, and Y. Kadobayashi, "Electricity theft detection for smart homes: Harnessing the power of machine learning with real and synthetic attacks," *IEEE Access*, vol. 12, pp. 26023-26045, 2024. <https://doi.org/10.1109/ACCESS.2024.3366493>
- [14] H. Yang, C. Lian, B. Xu, R. Ding, P. Zhao, and Z. Zeng, "Self-supervised latent feature-guided multi-step diffusion model for electricity theft detection with imbalanced and missing data," *IEEE Transactions on Smart Grid*, vol. 16, no. 3, pp. 2439 - 2450, 2025. <https://doi.org/10.1109/TSG.2025.3546219>
- [15] S. Bahrami, E. Yumuk, A. Kerem, B. Topçu, and A. Kaya, "Electricity theft detection using rule-based machine learning (rML) approach," *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, vol. 12, no. 2, pp. 438-456, 2024. <https://doi.org/10.29109/gujsc.1443371>
- [16] A. J. Almalki, "Unsupervised learning with hybrid models for detecting electricity theft in smart grids," *IEEE Access*, vol. 12, pp. 187027 - 187040, 2024. <https://doi.org/10.1109/ACCESS.2024.3498733>
- [17] Q. Liang, S. Zhao, J. Zhang, and H. Deng, "Unsupervised BLSTM-based electricity theft detection with training data contaminated," *ACM Transactions on Cyber-Physical Systems*, vol. 8, no. 1, pp. 1-20, 2024. <https://doi.org/10.1145/3604432>
- [18] M. J. Reis and C. Seródio, "Edge AI for real-time anomaly detection in smart homes," *Future Internet*, vol. 17, no. 4, p. 179, 2025. <https://doi.org/10.3390/fi17040179>
- [19] N. Cameron, *ESP32 microcontroller*. Berkeley, CA: Springer, 2023.

- [20] I. Abubakar, S. Khalid, M. Mustafa, H. Shareef, and M. Mustapha, "Calibration of ZMPT101B voltage sensor module using polynomial regression for accurate load monitoring," *ARPJ Journal of Engineering and Applied Sciences*, vol. 12, no. 4, pp. 1076-1084, 2017.
- [21] H. Zhao, Z. Liu, Z. Liu, K. Zhao, and Y. Shi, "Intelligent outlet based on ZMCT103C and GSM," in *International Conference on Big Data Analytics for Cyber-Physical-Systems*, 2019.
- [22] A. Graves, *Long short-term memory*. Berlin, Heidelberg: Springer, 2012.
- [23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008: IEEE, 2018.
- [24] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12591-12604, 2023. <https://doi.org/10.1109/TKDE.2023.3270293>
- [25] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, 2021. <https://doi.org/10.7717/peerj-cs.623>
- [26] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," in *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, 2020.