Explainable AI (XAI) for Intelligent Energy Management in IoT-Enabled Smart Grids

Faithpraise B. Otosi[1*], *Celestine A*. Udie[2], *Fina O*. Faithpraise[3]

[1]Business Management, Faculty of Administration and Management Sciences, University of Calabar, Nigeria
[2] Petroleum Engineering, Faculty of Engineering and Technology, University of Calabar, Nigeria
[3] Computer Engineering, Faculty of Engineering and Technology, University of Calabar, Nigeria

Abstract. Non-technical losses (NTLs) are a major problem in modern smart grids, damaging revenue and operational functionality. This study proposes an integrated IoT–edge–cloud framework to improve fraud detection, analyze electricity usage patterns, and enhance data reliability in distributed smart grids. The approach extracts multiple features from smart meter data and uses a hybrid machine learning method that combines classification and clustering. To test the system's robustness, the study included simulated fraud scenarios in difficult circumstances. Results show that the system achieved high detection accuracy (96.4%) and an AUC of 0.98. The framework also reduced false alarms by 86% compared to traditional rule-based methods, improving consistency and productivity. It supports near-real-time operation with response times around 125 ms and is scalable for larger smart grid environments. Behavioral segmentation further improved reliability by identifying differences in electricity consumption and reducing incorrect classifications. Overall, the study shows that combining data quality management, behavioral analysis, and distributed processing yields a more reliable and resilient solution for operational smart grid systems.

1 Introduction

Non-technical losses (NTLs), such as electricity theft, meter tampering, bypassing, and billing manipulation, continue to threaten revenue protection and grid reliability within modern power networks [1], [2]. Globally, utilities lose an estimated $101.2 billion annually due to electricity theft, fraud, and billing inefficiencies [3]. The World Bank further reports that utilities in developing countries experience significant quasi-fiscal deficits, arising from the gap between expected revenue in an efficient electricity sector (covering operational and capital costs) and actual cash recovered by utilities. Although advanced metering infrastructure (AMI) enables detailed monitoring, many utilities still rely on static, rule-based screening and threshold-based anomaly detection. These methods assume stable consumption patterns and cannot keep pace with evolving fraud tactics, diverse user behavior, and complex time-based trends in smart meter data [4]. As a result, they often generate too many false positives and miss hidden irregularities, reducing their dependability. As power systems become more digital, smart grids have evolved into distributed cyber-physical environments that produce high-frequency, multivariate time-series data. These data streams enable advanced analytics, such as behavioral profiling, segmentation, and demand forecasting [5]. Modern smart grids use* hierarchical IoT architectures, in which smart meters send data via networks such as cellular IoT, RF mesh, and LPWAN to edge and cloud platforms [6]. This shift creates major challenges on how to best distribute analytical intelligence across the sensing, edge, and cloud tiers.

Most existing studies conceptualize the system primarily as a communication pipeline rather than a fully integrated analytical platform. Although cloud-centric architectures with multi-agent orchestration layers support global optimization, distributed intelligence, and large-scale learning, they often incur orchestration latency, communication overhead, and coordination complexity that constrain deterministic real-time responsiveness. Edge-focused approaches lower latency but often rely on simpler models, which can reduce identification performance and miss broader context. Hybrid edge–cloud models try to handle these trade-offs [7-8]. However, despite the multi-agent orchestration layers' ability to offer a new pathway to address challenges of weak coordination, inefficient resource utilization, and limited resilience when communication is disrupted, current systems still face trade-offs due to security vulnerabilities.

At the same time, NTL detection methods have shifted from rule-based estimation to adaptive machine learning. Early approaches used statistical analysis and historical baselines [9]. Supervised learning methods treat NTL detection as a classification problem, using algorithms such as Support Vector Machines, Random Forests, and ensemble algorithms to improve performance [10]. However, these methods rely on tagged datasets that commonly exhibit class imbalance,

*Corresponding Author: ffaithpraise@unical.edu.ng

inspection bias, and noisy labels. Deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks can detect complex temporal patterns in consumption data, leading to better predictions [11]. However, these models are often difficult to interpret, which limits explainability and raises concerns about regulatory compliance and accountability. Unsupervised methods, such as autoencoders and isolation-based techniques, are useful when labeled data are scarce but often lack context, making their anomaly detection less reliable [12]. Reinforcement learning offers adaptive inspection strategies, yet remains largely experimental and underused in real-world systems [13]. Most studies rely on static datasets and standard evaluation metrics, such as accuracy, recall, and AUC, yet often overlook real-world factors, including streaming data, inference delays, scalability, changing data patterns, and infrastructure constraints [14]. This gap makes many proposed solutions not as practical for real deployment. Current research regularly treats fraud detection and behavioral analytics as separate goals. Fraud detection focuses on classification, while behavioral analytics focuses on clustering and usage patterns, but does not connect these data to fraud decisions [15]. This separation reduces context and can lead to the misclassification of normal lifestyle changes as fraud. NTL detection systems also operate under regulations that may result in inspections, fines, or service cuts. However, many high-performing models lack clear and auditable decision-making processes, thereby limiting transparency, trust, and regulatory compliance. In real-world deployments, systems frequently encounter challenges such as packet loss, latency, bandwidth limitations, synchronization issues, and unstable network connections. Despite these practical constraints, many existing studies assume ideal conditions characterized by perfect data availability and uninterrupted communication.

1.1 Novelty statement

This study presents a unified IoT–edge–cloud architecture for non-technical loss (NTL) detection in smart grids, in which IoT devices enable fine-grained data collection, edge nodes perform near-real-time inference and filtering, and cloud resources support large-scale model training and cross-regional optimization.

Main contributions of this work include: (i) a joint modelling strategy with a shared feature space for cross-layer representation learning across IoT, edge, and cloud data streams; (ii) a hierarchical edge–cloud architecture tailored for non-technical loss (NTL) detection in smart grids; (iii) system-level evaluation of latency and scalability under varying workload conditions; (iv) a data-continuity mechanism designed to mitigate missing rates, and deployment feasibility in resource-constrained environments. or disrupted smart meter readings for robust classification; and (v) a comprehensive evaluation

2 System model

2.1.1 System architecture overview

The proposed framework is implemented as a hierarchical IoT–edge–cloud architecture, as illustrated in **Fig. 1**, enabling distributed processing, scalable analytics, and adaptive decision-making. The system is formally represented as a directed graph where:

$$G=(V,E) \quad (1)$$

where the node set V = {S, E, C, A} represents the sensing (smart meters), edge, cloud, and application layers, respectively, and E denotes bidirectional communication links.

To support adaptive operation, the framework adopts a closed-loop learning mechanism defined as:

$$\Theta_{t+1}= U(\theta_t,D_{feedback}) \quad (2)$$

where θ represents model parameters, and $D_{feedback}$ denotes feedback derived from application-level outcomes (e.g., inspection results and billing corrections). Updated models are periodically deployed to edge nodes, enabling real-time inference as consumption and fraud conditions evolve.

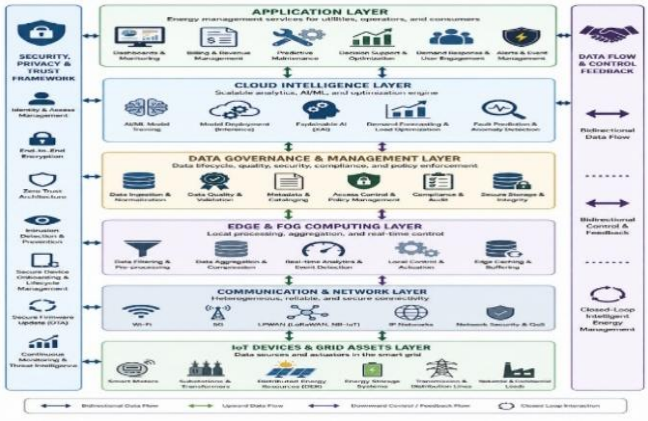

Fig. 1. Hierarchical IoT–edge–cloud architecture.

The model of Figure 1 integrates sensing technologies, edge computing, cloud intelligence, data governance, communication infrastructure, and security mechanisms to enable efficient, transparent, and secure energy management [15].

The system is structured as a layered IoT–cloud smart grid architecture with integrated analytics, governance, and security controls. At the foundational level, the Communication and network layer provides heterogeneous bidirectional connectivity across the system using technologies such as Wi-Fi, 5G, and LPWAN, supporting real-time, reliable, and QoS-aware data exchange between distributed devices, edge nodes, and cloud resources.

Above this, the Edge and Fog computing layer performs localized data pre-processing, filtering, aggregation, and low-latency decision-making close to data sources such as smart meters, substations, and distributed energy resources. This layer reduces latency, optimizes bandwidth usage, and supports near-real-time grid control functions.

Processed and raw data are then managed through a Data Governance and Management layer, which

operates as a cross-cutting control plane responsible for data lifecycle management, metadata control, quality assurance, lineage tracking, access policies, and regulatory compliance enforcement across storage, processing, and analytics pipelines.

The Cloud intelligence layer serves as the system's scalable compute and analytics core, supporting both offline model training and large-scale optimization tasks. Machine learning and Explainable AI models are developed, validated, and deployed (via MLOps pipelines) to enable demand forecasting, fault detection, load optimization, and predictive maintenance, while ensuring interpretability of AI-driven decisions through transparent explanations.

The Application layer delivers domain-facing services to utilities, operators, and consumers, including energy dashboards, billing systems, demand-response coordination, predictive-maintenance interfaces, and intelligent-grid monitoring and control applications. This layer also translates AI outputs into actionable operational decisions and, where required, automated control signals.

A security, privacy, and trust framework serves as a horizontal, cross-cutting architecture that spans all layers. It enforces end-to-end cybersecurity and trust mechanisms, including identity and access management, encryption, intrusion detection, zero-trust enforcement, secure device onboarding, and continuous threat monitoring for both IoT devices and cloud services.

Collectively, the architecture forms a closed-loop intelligent energy management system in which bidirectional data flow and control feedback enable continuous learning, adaptive optimization, and real-time grid responsiveness.

2.1.2 Mathematical formulation and data representation

The mathematical variables and symbols used in this study are summarized in Table 1**.** Key variables are referenced here without repeating full definitions to maintain clarity and conciseness.

Table 1. Abbreviations and mathematical symbols

Symbol / Term	Définition
D	Dataset of labeled samples
N	Number of customers
x_i	Feature vector of customer i
y_i	Class label (0 = normal, 1 = fraud)
d	Feature dimensionality
T	Number of temporal observations
$P_{i,t}$	Power consumption at time ttt
$f_{i,j}$	Engineered feature
k	Number of engineered features
$\hat{y}_i$	Predicted fraud probability
f_θ	Predictive model
α,β	Fraud manipulation parameters
Δ	Temporal shift
μ_i	Mean load
L_w	Loss function
w_0,w_1	Class weights

The mathematical formulation defines the relationship between input consumption data, engineered features, and model estimates. The dataset structure is summarized in Table 1, while the data processing pipeline that generates these features is illustrated in Figure 2.

Let the labeled dataset be defined as:

$$D = \{(x_i,y_i)\}_i^N = 1 \quad (3)$$

where x_i and y_i follow the notation in Table 1.

Each feature vector is constructed from time-series consumption and engineered descriptors:

$$x_i = [P_{i,t1}, P_{i,t2}, \ldots\ldots, P_{i,tT}, f_{i,1}, \ldots\ldots, f_{i,k}] \quad (4)$$

The transformation of raw data into structured features follows the multi-stage preprocessing workflow shown in Figure 2, which ensures data consistency before model training.

The predictive model is defined as a mapping:

$$f_\theta: R^d \rightarrow [0,1] \quad (5)$$

such that the predicted fraud probability is

$$\hat{y}_i = f_\theta(x_i) \quad (6)$$

The model parameters are learned by solving the optimization problem:

$$\theta^* = arg_\theta min L_w(\theta;D) \quad (7)$$

Use a two-column format, and set the spacing between the columns at 8 mm. Do not add any page numbers.

2.2 Smart meter data Structure preprocessing

The dataset characteristics are summarized in **Table 2,** while the preprocessing and data continuity workflow are illustrated in Figure 2**.**

Table 2. Dataset summary

Parameter	Value
Number of Customers	500
Sampling rate	30 minutes
Observation Period	12 months
Total Measurements	≈ 8.75 millions
Feature Count	12
Fraud Prevalence	15%
Data Format	Structured time-series

As shown in Figure 2, the preprocessing pipeline consists of the validation, missing-data handling, buffering, and normalization stages.

The preprocessing operation is defined as:

$$x_i' = P(x_i) = N(I(V(xi))) \quad (8)$$

where:

V = validation and anomaly filtering,

I = missing data imputation,

N =normalization.

Missing data handling is integrated into the workflow shown in Figure 2, in which short-term gaps are interpolated, and long-term outages are replaced with historical averages.

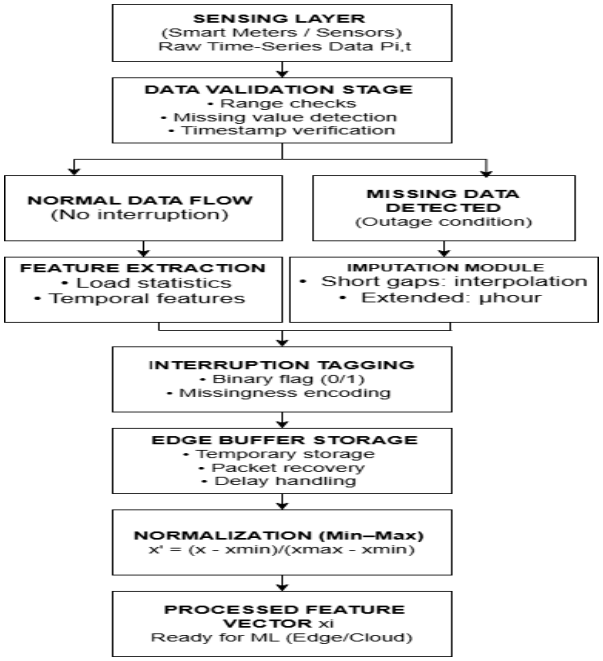

Fig. 2. Multi-layer data continuity workflow

2.2.1 Normalization

Each feature is scaled using min–max normalization:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (9)$$

Data interruptions are handled under two outage scenarios :

Missing data handling

Short-duration gaps (≤ 2 intervals):

$$p_{i,t} = \frac{P_{i,t-1} + P_{i,t+1}}{2} \quad (10)$$

For extended outages:

$$P_{i,t} = \mu_{hour} \quad (11)$$

where μ_{hour} is the mean consumption for the corresponding hour across historical data.

Additionally, interruption flags are incorporated into the feature space to preserve outage information for downstream learning.

2.3 Fraud injection and attack modelling

To evaluate robustness, synthetic fraud scenarios are injected into a subset of customer profiles using parameterized attack models. Representative attack patterns are illustrated in Figure 3, while their configuration is summarized in Table 3

Scaling Attack (Meter Tampering)

$$P_{i,t}^{fraud} = \alpha P_{i,t}, \quad \alpha \sim U(a,b), \quad 0<\alpha<1 \quad (12)$$

Temporal Shift Attack (Clock Manipulation)

$$P_{i,t}^{fraud} = P_{i,t+\Delta}, \quad \Delta \sim D \quad (13)$$

Load Flattening :

$$P_{i,t}^{fraud} = \mu_i + \beta\,(P_{i,t} - \mu_i), \quad 0 < \beta < 1 \quad (14)$$

where:

- μ_i is the mean load for customer i,
- α, β, and Δ are stochastic manipulation parameters.

Table 3. Fraud injection configuration

Fraud Type	Manipulation Strategy	Prevalence %
Scaling Attack	20-50% reduction	5%
Random Suppression	Random removal of peaks	3%
Temporal Shift	Peak hour displacement	2%
Load Flattening	Variance reduction	2%
Total Fraud Ratio	-	12%

As illustrated in Figure 3, different attack types emulate realistic manipulation strategies, including meter tampering, time shifting, and consumption smoothing.

These patterns introduce both structured and irregular anomalies, permitting the model to learn strong representations of fraud behavior as shown in Figure 3.

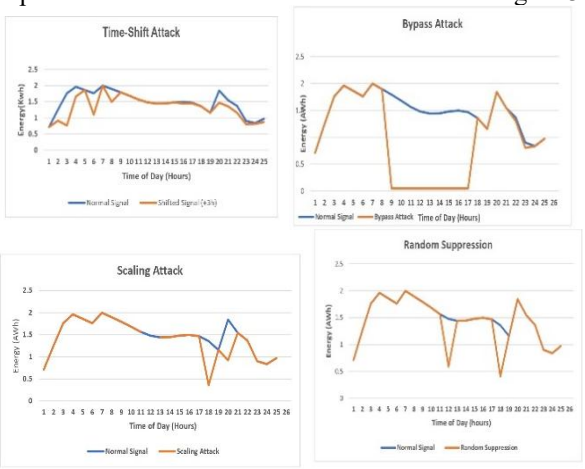

Fig. 3. Representative synthetic fraud injection scenarios illustrating scaling, temporal shift, and load manipulation strategies

2.4 Behavioral clustering

To capture latent consumption patterns, K-means clustering is applied to the feature space. The clustering objective is:

$$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (15)$$

where:

- C_k is the set of points assigned to cluster k,
- μ_k is the centroid of cluster k,
- K=5 is selected empirically.

Centroid updates follow:

$$\mu_k = \frac{1}{C_k} \sum_{x_i \in C_k} x_i \quad (16)$$

This behavioral segmentation complements supervised learning by distinguishing legitimate variability in consumption from fraudulent anomalies.

The supervised model functions on preprocessed data generated by the pipeline illustrated in Fig. 2 and learns to discriminate between normal and fraudulent consumption patterns, as shown in Fig. 3. Fraud detection is formulated as a weighted binary classification problem using the loss function.

The model predicts:

$$L_w = -\frac{1}{N} \sum_{i=1}^{N} [w_{y1} y_i log(\hat{y}_i) + w_0 (1 - y_i) log(1 - \hat{y}_i)] + \lambda \|\theta\|^2 \quad (17)$$

where:

- w_1, w_0 are class weights,
- λ is a regularization parameter.

Class weights are defined as:

$$w_1 = \frac{N}{2N_{fraud}}, \quad w_0 = \frac{N}{2N_{normal}} \tag{18}$$

The training configuration is summarized in Table 4**.**

Table 4. Model training configuration

Parameter	Value
Train / Validation / Test Split	70% / 15% / 15%
Batch Size	128
Learning Rate	0.001
Optimizer	Adam
Epochs	50
Class Weighting	Balanced inverse frequency
Early Stopping Patience	10 epochs

This formulation mitigates class imbalance while improving generalization.

2.5 Edge-cloud-task allocation

The distributed deployment is modeled as an optimization problem to minimize system latency:

$$minL_{total} = L_{edge} + L_{transmission} + L_{cloud} \tag{19}$$

subject to:

$$C_{edge} \leq C_{max}, \qquad E_{edge} \leq E_{budget} \tag{20}$$

where:

- C_{edge} is computational capacity,
- E_{edge} is energy consumption.

Edge nodes perform real-time inference and preprocessing, while the cloud handles resource-intensive training and global optimization.
Model effectiveness is evaluated using both predictive and operational metrics.

Predictive Metrics

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{21}$$

$$Precision = \frac{TP}{TP+FP} \tag{22}$$

$$Recall = \frac{TP}{TP+FN} \tag{23}$$

$$F1 = 2\frac{Precision.Recall}{Precision+recall} \tag{24}$$

$$AUC = \int_0^1 TPR(FPR)dFPR \tag{25}$$

Equations 21 to 26 [15]
Operational latency is measured as :

$$Latency = t_{decision} - t_{arrival} \tag{26}$$

These metrics enable a comprehensive evaluation of detection accuracy, robustness, and system-level efficiency.

3 Results and discussion

This section presents a detailed evaluation of the 8proposed IoT–edge–cloud fraud detection system, covering predictive performance, behavioral insights, system scalability, and real-time capability. The results are discussed, along with their consequences, to provide a clear overall analysis. The predictive performance of the evaluated models is summarized in Table 5 and Figure 4**,** using the metrics defined in Section 2.

Table 5. Fraud detection performance across models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC
Rule-based	78.4	65.2	59.8	62.3	0.71
SVM	90.5	88.1	85.6	86.8	0.92
Random Forest	95.8	94.5	92.8	93.6	0.97
Neural Network	**96.4**	**95.1**	**93.9**	**94.5**	**0.98**
Autoencoder	92.7	90.3	88.4	89.3	0.94

As illustrated in Figure 4, a clear performance differences between the rule-based method and learning-based models across all metrics. The rule-based approach performs poorly, particularly on recall and F1-score, suggesting it struggles to detect hidden or evolving fraud patterns.

Machine learning models show considerable improvements. SVM offers strong baseline results, while Gradient Boosting and Neural Network models attain the best and most uniform performance. The Neural Network performs best overall (Accuracy = 96.4%, F1 = 94.5%), showing its ability to detect complex, time-based consumption patterns.
The Autoencoder also performs well but is slightly behind the supervised models, suggesting that labeled learning is better at detecting structured fraud. Overall, Figure 4 shows that more complex models lead to better detection results.

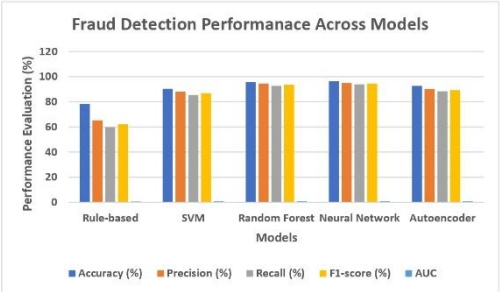

Fig. 4. Performance comparison of rule-based and learning-based models across classification metrics (accuracy, precision, recall, and F1-score)

Behavioral clustering results are summarized in Table 6, corresponding to the K-means formulation in Section 2.5.

Table 6. Consumer behavioral segmentation

Cluster	Description	Share (%)	Avg Daily (kWh)	Peak Period	Flexibility
C1	Low usage	28	4.2	Ev	High
C2	Fr	41	11.5	Night	Me

C3	C/d	19	35.7	Day	Low
C4	I/M	12	Va	Mixed	Unc

Note: Ev =Evening; Me =Medium; Va=Variable; Unc=Uncertain. Fr = Family residential,
C/d = Commercial/ daytime, I/m = Irregular/ mixed

The clustering results show that most of the data comes from residential consumers (C1 and C2). C3 represents structured commercial use, whereas C4 shows irregular, highly variable patterns.
Figure 5 further illustrates the time-based consumption patterns of these clusters, using 24-hour load profiles from Table 7. Residential clusters display predictable daily cycles, whereas irregular users show much greater variability.

Table 7. 24-hour load profile dataset (kwh)

Hour	Low Usage	Family Peak	Commercial	Irregular
0	1.10	1.20	4.00	2.10
6	1.28	2.40	4.50	2.20
12	1.15	2.60	3.90	2.20
18	0.85	1.40	3.60	2.30
23	1.05	1.20	4.00	2.00

Cluster C4 overlaps with fraud-like patterns described in Section 2.4. In Figure 5, irregular consumption in this cluster resembles manipulated load profiles. This shows why it is important to combine clustering with supervised detection to reduce false positives.

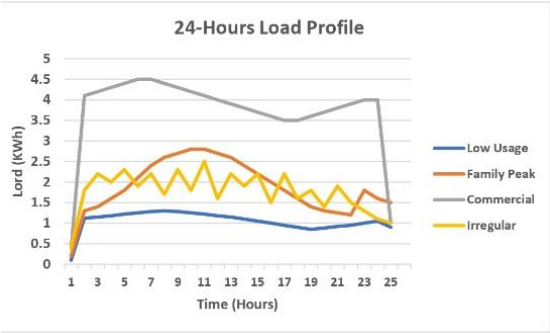

Fig. 5. Representative daily load profiles

The operational impact of the proposed framework is presented in Table 8**.**

Table 8. Operational impact comparison

Metric	Rule-Based	Proposed Framework	Improvement
False Positive Rate (%)	34.8	4.9	↓ 86%
Recall (%)	59.8	93.9	+34.1 pts
Inspection Load	High	Reduced	Significant

The large drop in false positives shows better discrimination. This improvement comes from combining supervised learning and behavioral clustering.

The robustness of the framework under data disruption is evaluated using the continuity model (Section 2.3). Results are summarized in Table 9, and the impact on classification performance is illustrated in Figure 6.

Table 9. Data continuity validation performance

Condition	Recovery Rate (%)	Relative MAD (%)	F1-score Change
With buffering	98.7	< 2.4	Baseline
Without buffering	N/A	N/A	-6.1%

N/A: indicates that recovery and imputation metrics are not applicable when buffering mechanisms are absent.
As shown in Figure 6**,** the absence of data continuity mechanisms leads to a noticeable degradation in classification performance. Buffered recovery maintains stability, confirming the importance of preprocessing and data integrity in the overall system

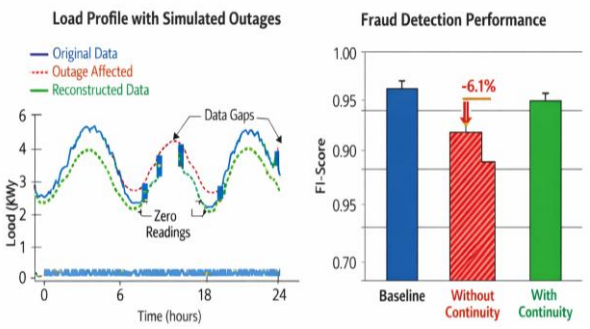

Fig. 6. Impact of data continuity mechanisms on classification performance under simulated outages

The latency distribution across system components is presented in Table 10**.**

Table 10. Latency breakdown

Component	Latency (ms)
Edge Processing	18
Transmission	42
Cloud Processing	65
Total	**125 ms**

The results show that most of the total latency is due to transmission and cloud processing.
A comparative evaluation of deployment strategies is provided in Table 11, while the latency–accuracy trade-off is summarized in Table 12.

Table 11. Comparative performance of edge, cloud, and hybrid architectures

Metric	Rule-Based	Proposed Framework	Improvement
Latency (ms)	45	180	125
Accuracy (%)	91.2	96.4	96.4

Scalability	Limited	High	High
Resilience	Medium	Low	High

The superior performance of the hybrid architecture is achieved through an edge–cloud collaborative mechanism in which lightweight preprocessing, buffering, and preliminary anomaly screening are performed at the edge, while computationally intensive learning and model retraining are executed in the cloud. This reduces communication overhead and latency while preserving high detection accuracy. Compared with lightweight edge-only schemes, the hybrid approach maintains the same cloud-level accuracy (96.4%) while improving resilience through local continuity handling and distributed decision support. Unlike cloud-only systems, it is less vulnerable to connectivity disruptions and achieves better operational stability, scalability, and near–real-time responsiveness.

Table 12. Latency–accuracy trade-off

Configuration	Latency (ms)	Accuracy (%)
Edge-only	45	91.2
Hybrid	125	96.4
Cloud-only	180	96.4

The combined architecture provides the best balance between latency and predictive performance as illustrated in Figure 7.

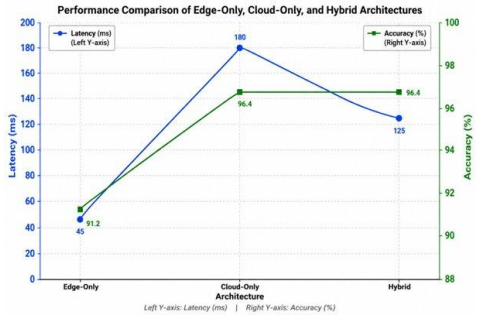

Fig. 7. Comparative performance of trade-off

System operation under different network situations is shown in Table 13, while the architectural flow and task distribution are illustrated in Figure 8.

Table 13. Latency under network conditions

Scenario	Edge	Transmission	Cloud	Total
Best-case	15	20	50	85 ms
Nominal	18	42	65	125 ms
Worst-case	25	120	80	225 ms

As illustrated in Figure 8, the edge layer reduces transmission overhead through local preprocessing, while the cloud performs computationally intensive tasks.

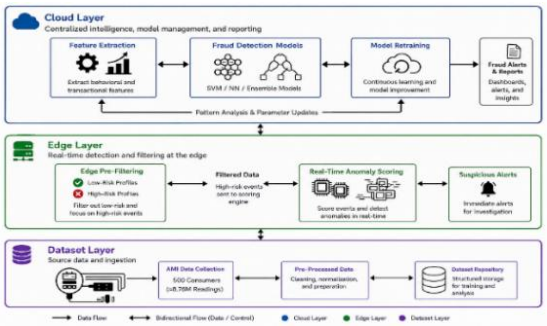

Fig. 8. Scalable fraud detection pipeline illustrating edge–cloud task distribution and computational flow.

3.1 Integrated system insights

Overall, the results show that system performance depends on how model intelligence, behavioral context, and architecture work together. Trends in Figures 4–7 confirm that:

- Advanced models improve detection accuracy.
- Behavioral clustering enhances interpretability and reduces false alarms.
- Edge–cloud integration optimizes latency and scalability.

Latency analysis also shows that transmission remains the main bottleneck, underscoring the need for edge intelligence in real-time systems.

4 Conclusion and policy implications

4.1 Conclusion

This study introduced an integrated IoT–edge–cloud framework for intelligent energy management, with a focus on robust fraud detection and behavioral analysis in smart grids. By combining supervised learning, unsupervised clustering, and distributed system design, the approach overcomes key limits of traditional rule-based systems, especially for nonlinear, changing, and adversarial consumption patterns.

The results show that advanced learning models outperform traditional methods, achieving high predictive accuracy (Accuracy = 96.4%, F1-score = 94.5%, AUC = 0.98). Adding behavioral clustering provides the system with more context, helping it distinguish between normal usage changes and fraud. This reduces false positives by up to 86%, improving efficiency and reducing unnecessary inspection costs. From a systems perspective, the edge–cloud architecture enables scalable, near-real-time deployment. Latency analysis shows that the hybrid setup balances computational efficiency and predictive performance, with total latency around 125 ms. Adding data continuity features makes the system robust in real-world conditions, maintaining high classification performance even when data are missing or interrupted. Overall, the proposed framework shows that smart grid energy management requires both accurate predictive models and robust context and system design. Bringing these elements together provides a comprehensive solution for modern energy systems with large data volumes, evolving patterns, and security challenges.

4.2 Policy and practical implications

The proposed framework has major technical, operational, and societal benefits for IoT-enabled smart grids. Technically, it offers a unified system that combines fraud detection, behavioral intelligence, and edge–cloud computing for scalable, real-time analytics. Operationally, it reduces false positives by up to 86% and enables faster, more efficient decision-making, reducing inspection costs and improving reliability. Economically and in policy, it helps protect revenue, supports data-driven grid management, and meets modern regulations. More broadly, it encourages sustainable energy use and increases transparency in automated decisions through explainable AI, helping build smarter, more resilient grids.

We sincerely acknowledge and appreciate all individuals and organizations whose support, guidance, and contributions helped make this work a reality.

References

1. S. Steven, Causes of non-technical losses and reduction measures in power distribution utilities: A Review. IJERT. **15** (2), (2026). https://www.ijert.org/ ISSN: 2278-0181
2. G. U. Alao, B. O. Enobakhare, O. A. Okpoko, M. I. Adeoba, E. Nii-Okai, A. O. Oladunni, and V. U. Onyia, Artificial intelligence applications in smart grids and modern power systems: A comprehensive review, Cl in MPS, **1**, p. 23 (2026), doi: 10.58613/cimps112
3. R. Nyaupane, K. Bhatta, P. Giri, R. Sharma, A. Joshi, S. Chitrakar, A. Thapa, B. Adhikary, B. Baral, and M. Pokharel, IoT-enabled smart metering to enhance energy management for day-ahead electricity price forecasting, American Journal of Modern Energy, **11**, pp. 15–25 (2025), doi: 10.11648/j.ajme.20251101.12.
4. A. Mohamed, I. Ismail, and M. AlDaraawi, IoT-driven intelligent energy management: Leveraging smart monitoring applications and artificial neural networks for sustainable practices, Computers, **14**, p. 269, (2025), doi: 10.3390/computers14070269.
5. M. Hashim, L. Khan, N. Javaid, Z. Ullah, and A. Javed, Stacked machine learning models for non-technical loss detection in smart grids: A comparative analysis, Energy Reports, **12**, pp. 1235–1253, (2024), doi: 10.1016/j.egyr.2024.06.015.
6. S. Gresoi, G. Stamatescu, and I. Făgărășan, Advanced methodology for fraud detection in energy using machine learning algorithms, Applied Sci, **15**, p. 3361, (2025), doi: 10.3390/app15063361.
7. Y. Zhao, Design of an IoT-based intelligent construction energy monitoring and management system, CIB Conferences, **1**, p. 358, (2025), doi: 10.7771/3067-4883.1696.
8. H. M. Tsai, C. W. Chen, and Y. T. Li, Application of smart grid anomaly detection based on machine learning, IOP Conference Series: Earth and Environmental Science, **1386**, p. 012024, (2023).
9. M. Burgos, J. Morato, and P. Vizcaino, A review of smart grid anomaly detection approaches pertaining to artificial intelligence, Applied Sci, **14**, p. 1194, (2024), doi: 10.3390/app14031194.
10. Y. Wang, Q. Chen, T. Hong, and C. Kang, Review of smart meter data analytics: Applications, methodologies, and challenges, IEEE Transactions on Smart Grid, **10**, pp. 3125–3148, (2019), doi: 10.1109/TSG.2018.2818167.
11. M. A. Belay, S. S. Blakseth, A. Rasheed, and P. S. Rossi, Unsupervised anomaly detection for IoT-based multivariate time series: Existing solutions, performance analysis, and future directions, Sensors, **23**, p. 2844, (2023), doi: 10.3390/s23052844.
12. V. Nik, Adaptive reinforcement learning for energy management, in Proc. Building Simulation: 19th Conf. IBPSA, (2025), doi: 10.26868/25222708.2025.1247.
13. F. O. Faithpraise, E. O. Obeisung, A. E. Asiya, O. A. Ilori, and C. R. Chatwin, Automated fraud elimination via electrical signal monitoring and recording device, JETE and App. Sci., **9**, pp. 99–104, (2018).
14. S. Darby, Smart metering: What potential for householder engagement? Building Research & Information, **38**, pp. 442–457, (2010), doi: 10.1080/09613218.2010.492660.
15. K. C. Okafor, O. M. Longe, M. O. Ezeja, I. I. Ayogu, K. Anoh, and B. Adebisi, HKC-MBLT: Enhancing SaaS Transaction Security with Hybrid Cryptography and Memory-Based Lightweight Tokenization for IoT-Enabled Cyber-Physical Systems, Cogent Engineering, **12**(1), pp. 1–33, Taylor & Francis, (2025), DOI:https://doi.org/10.1080/23311916.2025.2583538