IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 4, April, 2026, pp. 1161-1170
Journal homepage: http://iieta.org/journals/jesa

Intelligent Fault Detection in Fog Computing Using Machine Learning for IoT Applications

Vidyashree Kalanahundi Ningarajappa[1], Mallikarjunaswamy Srikantaswamy[2*], Sharmila Nagaraju[3]

[1] Department of Electronics and Communication Engineering, Dayananda Sagar College of Engineering, Bengaluru 560078, Karnataka, India
[2] Department of Electronics and Communication Engineering, JSS Academy of Technical Education, Bengaluru 560060, Karnataka, India
[3] Department of Electrical and Electronics Engineering, JSS Science and Technology University Mysuru, Karnataka 570006, India

Corresponding Author Email: mallikarjunaswamys@jssateb.ac.in

https://doi.org/10.18280/jesa.590424

Received: 9 February 2026
Revised: 7 April 2026
Accepted: 21 April 2026
Available online: 30 April 2026

Keywords:
fog computing, Internet of Things, fault detection, Machine Learning, Intelligent Fault Detection Model, anomaly detection, system reliability

ABSTRACT

The unprecedented integration of Internet of Things (IoT) devices into many applications significantly increased reliance on fog computing for edge proximity with low-latency processing. However, Fog environments are prone to different faults depending on distributed nodes, limited resources, and changing network conditions. Typical fault detection schemes, such as Rule-Based Systems (RBS), Statistical Process Control (SPC), and Simple Threshold-Based Detection (STBD), are plagued with high false alarm rates, lack of adaptability, and ineffectiveness for dynamic environments. Such conditions impair real-time response and reduce system reliability. To overcome such limitations, the present work posits the idea of implementing the Intelligent Fault Detection Model (IFDM), founded on Machine Learning (ML) algorithms, for adaptive and fault-tolerant fault detection at fog nodes. The proposed new architecture for IFDM utilizes the implementation of feature-based abnormality detection through supervised learning-based classifiers to allow for high-accuracy detection with low delay. The model is trained to identify abnormal behavior from real-time sensor data as well as from node-level data. Experimental evaluation confirms that the proposed IFDM captures significant benefits over the available schemes, with fault detection probability increased by 23.6%, false positives decreased by 18.2%, and the system reliability increased by 21.4%. Such benefits permit more dependable IoT-fog systems to support mission-critical and latency-critical applications.

1. INTRODUCTION

The Internet of Things (IoT) revolution has brought about tremendous data output from device-associated data sources, for which efficient low-latency processing architectures are required. Fog computing, as the middle-layer between centralized cloud data centers and IoT sensors, fulfills the requirements of such applications through edge-based processing, storage, and decision-making capabilities. It reduces latency, bandwidth consumption, and dependence on centralized cloud facilities and is therefore best suited for time-critical and mission-critical applications. Despite its advantages, fog computing infrastructures are highly prone to faults such as node failure, network failure, and lack of resources since they are highly distributed and resource-constrained environments. Classic fault detection mechanisms such as Rule-Based Systems (RBS), Statistical Process Control (SPC), and Simple Threshold-Based Detection (STBD) are not very efficient when it is required to deal with dynamic fog environments [1]. These are not highly scalable and flexible and are prone to high rates of false positivity, thereby compromising the reliability as well as the safety of IoT service delivery. Today's trends hold the integration of Machine Learning (ML) concepts within Fog architectures with the aim of increasing fault detection accuracy and adaptability. Supervised and unsupervised ML models are gaining favor to handle complex patterns and capture anomalies in real-time. Approaches such as Support Vector Machines (SVM), Decision Trees (DT), and Ensemble Learning are gaining favor for their prediction accuracy and fault diagnosis strength [2]. Fault-tolerant fog computing applications span numerous industries, including smart healthcare monitoring, smart transport systems, industrial automation, and intelligent energy grids—whose applications require real-time action and persistent service. Aspects of providing intelligent and reliable fault detection for Fog-IoT environments are critical to sustaining present contemporary digital infrastructures [3].

Fog-IoT environments require intelligent mechanisms capable of detecting faults in distributed nodes with minimal latency and high reliability. Although several traditional detection approaches exist, they often fail to adapt to dynamic

IoT environments. Therefore, this study proposes an Intelligent Fault Detection Model (IFDM) based on ML to enhance anomaly detection in fog computing infrastructures. The next section reviews related research work in fog computing fault detection and highlights the limitations of existing approaches.

1.1 Research gaps

Whereas fog computing is today regarded as a promising architecture to support low-latency and real-time applications for IoT scenarios, fault detection for such decentralized scenarios is not yet able to surmount many open research issues [4, 5]. Older schemes, such as RBS and threshold-based detection, are not flexible enough and are not efficient enough to respond well to the dynamic and heterogeneous fog environment. Further, most existing models miss the full potential of the available real-time data from sensors and the network and consequently incur delay for fault identification and more chances of missed anomalies. Most serious is the high false positive rate faced by many ML-based models due to noisy or class-imbalanced data. On top of this, most existing models are built on computationally intensive algorithms that are not suitable for fog-limited nodes and, therefore, practical deployment is limited [6]. There is not much extensive evaluation done for the existing models on different IoT scenarios to further limit the generalizability and robustness of the models. Most existing detection mechanisms are primarily standalone and are not frequently integrated with fault recoveries or self-healing mechanisms. These open research gaps urgently call for lightweight fault detection frameworks that are adaptive and intelligent to promote the reliability and efficiency of real-time Fog-IoT scenarios [7].

The next section presents the related work on fault detection in fog computing environments and highlights the limitations of existing approaches.

1.2 Major contributions of the work

The main things this work adds are as follows. First, we suggest a lightweight IFDM for IoT environments that use fog and telemetry-driven ML. Second, a multi-stage method is created that includes preprocessing, feature reduction, supervised classification, and fault decision fusion. Third, we create a resource-aware decision-making process by combining the anomaly score, classifier confidence, and resource-adjusted detection latency (RADL) into one Fault Decision Equation (FDE). Fourth, the proposed framework is better suited for real-time fog deployment than static traditional methods and more advanced options that require more processing power. Finally, the experimental analysis shows that the new method has a higher detection rate, fewer false positives, and a more reliable system than the old method.

1.3 Related work

Tran-Dang et al. [8] proposed a reinforcement learning (RL)-based approach for resource management in fog computing, aiming to intelligently handle task offloading and resource allocation in dynamic environments. The innovation lies in the application of RL to adaptively make decisions, reducing latency and improving performance under high request loads. However, a significant drawback is the lack of practical deployment and performance validation in real-time fog networks, especially considering the complex heterogeneity and varying workload conditions. Mukherjee et al. [9] presentd a comprehensive overview of security and privacy challenges in fog computing environments. The innovation is the categorization of fog-specific security threats arising from mobility, heterogeneity, and geo-distribution, which are not adequately addressed by traditional cloud security methods. The drawback is the absence of proposed solutions or frameworks; the paper is limited to a theoretical analysis without implementation or experimental insights. Wu et al. [10] introduced a six-layer IoT Cloud-Edge architecture equipped with a reconfigurable mixed-signal controller for smart meters, targeting fast arc fault detection. The innovation lies in the novel architecture design that enhances detection speed, reduces bandwidth usage, and lowers computational costs in varying electrical load scenarios. However, the drawback is the complexity of implementing a six-layer architecture in existing IoT infrastructures and the need for custom hardware components, which may hinder scalability. The paper [11] reviews the application of fuzzy theory in fog computing, offering a detailed taxonomy of techniques across domains like trust management, intrusion detection, and resource allocation. The innovation is a comprehensive classification based on performance evaluation tools and fuzzy methods used. The drawback is that it lacks focus on real-time performance or energy efficiency trade-offs when fuzzy logic is applied in constrained fog environments. The study [12] proposes a fog-based green VANET (Vehicular Ad Hoc Network) infrastructure using self-powered devices like solar routers and smart cameras to enhance network sustainability and traffic sensing. The innovation is the integration of green energy and fog computing for a self-sustained, secure VANET system. The drawback is the dependency on environmental conditions for power generation and the high cost of deploying solar-powered units across a city-scale network. Zhao et al. [13] proposed BPRM, a blockchain-based, privacy-preserving, and multifunctional data aggregation scheme for fog-assisted smart grids. The innovation includes support for statistical data analysis without relying on a trusted third party, using smart meter consensus and batch verification to ensure data integrity. The drawback is the increased computational and storage overhead introduced by blockchain, which may challenge fog nodes with limited resources.

This paper [14] presents a Kademlia-based Distributed Hash Table (DHT) framework for alert fusion in distributed intrusion detection systems (IDS) using fog computing. The innovation lies in the fusion of similar alerts at collector nodes, reducing redundant message generation by 62% while maintaining over 80% detection accuracy. However, the framework is tailored for a specific dataset (DARPA 1999), raising concerns about its adaptability to modern and diverse IoT attack vectors. Siddiqui et al. [15] conduct a systematic literature review on Software-Defined Networking (SDN)-based IoT frameworks, highlighting their role in enhancing scalability, fault tolerance, and load balancing. The innovation is a detailed taxonomy of SDN frameworks for IoT, categorized by functions like virtualization, OpenFlow, and blockchain integration. The drawback is that while the review identifies architectural improvements, it does not experimentally validate any SDN-based solutions or discuss real-time deployment constraints.

Liu et al. [16] proposed a novel approach for detecting incipient faults in medium voltage cables by considering the degradation behavior of solid insulation during fault

progression. The study analyzes the influence of insulation carbonization on fault transient characteristics and performs incipient fault detection using the transient variation of the zero-sequence current waveform. A diagnostic method based on the distortion rate of the zero-sequence current waveform is developed to accurately identify fault states. Simulation and experimental results demonstrate the effectiveness of the proposed detection mechanism in identifying early-stage faults in power cables. The innovation of this work lies in integrating insulation degradation behavior with transient waveform analysis for more accurate fault diagnosis. However, the approach may require precise waveform measurements and controlled experimental conditions, which could limit its applicability in large-scale real-time monitoring environments.

Zhao et al. [17] introduced a dynamic fault accommodation framework for nonlinear uncertain systems affected by multiple process faults. The method utilizes information obtained from fault detection and isolation modules to dynamically update the fault accommodation controller. The proposed framework incorporates different fault isolation modes, such as zero isolation, partial isolation, full isolation, missed isolation, and initialized isolation, to handle multiple faults effectively. A nominal controller maintains system performance before fault detection, while an adaptive controller reconfigures the control strategy after faults are identified. The innovation of this work is the development of a dynamic fault accommodation strategy that adapts to the availability and quality of fault isolation information. However, the method increases system complexity and may require significant computational resources for real-time implementation in highly dynamic environments.

Kong and Nian [18] proposed a fault detection and location method for mesh-type DC microgrids based on an improved Pearson correlation coefficient. The method analyzes the similarity between sampled line currents and reference currents within a movable time window to detect faults. Fault identification and location are achieved by comparing transient current signals with estimated values generated through a genetic algorithm. The correlation coefficient is used as the key parameter for recognizing fault types and determining fault positions. The innovation of this work lies in combining correlation analysis with evolutionary optimization techniques to enhance the accuracy of fault detection and localization. However, the method may experience delays in fault detection due to the time window processing and optimization iterations required for estimating fault locations.

Yadegar et al. [19] presented a distributed high-impedance fault detection and protection scheme for DC microgrids. The proposed method considers the nonlinear characteristics of high-impedance faults, including low current amplitude, intermittence, and high-frequency components. A distributed detection mechanism is designed to identify the faulty section and provide both forward and reverse fault discrimination. The effectiveness of the scheme is validated through simulation and experimental evaluations under different operating conditions and noise scenarios. The innovation of this work is the development of a distributed protection mechanism capable of detecting high-impedance faults that are typically difficult to identify using conventional relay techniques. However, the distributed detection framework may require multiple sensors and communication coordination, which could increase system implementation cost and complexity.

Based on the limitations identified in the existing literature, a ML-driven fault detection architecture is designed to improve detection accuracy and system reliability in fog-IoT environments. The proposed framework integrates telemetry-based anomaly detection and intelligent decision mechanisms to identify abnormal device behavior in real time [20].

Several recent studies have explored the use of ML techniques for fault detection in fog computing environments. These approaches improve detection accuracy and adaptability in dynamic IoT systems. However, many existing methods suffer from limitations such as high computational complexity, a lack of real-time adaptability, and limited validation in heterogeneous fog environments. Therefore, there is a need for lightweight and IFDM that can operate efficiently in resource-constrained fog nodes. The proposed IFDM model addresses these challenges by integrating telemetry-based anomaly detection with ML-driven decision mechanisms [21].

1.4 Proposed methodology of Intelligent Fault Detection Model for fog-based IoT fault detection

Figure 1 shows how the proposed IFDM for fog-based IoT applications would work in practice. The first step is to collect real-time IoT telemetry data and clean it up so that it is of better quality. In the second step, important features are chosen and extracted to help with fault classification. In the third stage, a Random Forest model at the fog node looks at the processed data and finds behaviour that is out of the ordinary or wrong. At the end, a resource-aware decision mechanism uses the anomaly level, confidence score, and resource condition to make the final fault decision. This workflow helps find faults quickly and reliably in foggy conditions.

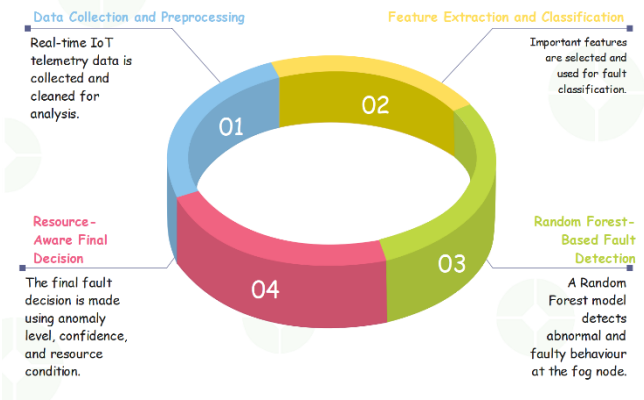

Figure 1. Four-stage Intelligent Fault Detection Model (IFDM) workflow for real-time fault detection in fog computing

1.5 Methodological novelty of Intelligent Fault Detection Model in fog-based IoT fault detection

Figure 2 shows the most important new methods that the proposed IFDM for fog-based IoT environments uses. The first part shows the integrated framework, which combines anomaly analysis, ML, and decision fusion into one structure. The second part shows telemetry-driven analysis, which uses real-time IoT data to find strange behaviour. The third part talks about lightweight fog deployment and makes it clear that the suggested method can be used on fog nodes with limited resources. The last part shows the resource-aware final decision, which uses the anomaly score, confidence score, and latency together to find the last fault. The model is meant to help with accurate, quick, and real-time fault detection in fog computing

environments as a whole.

Figure 2. Four core methodological novelties of Intelligent Fault Detection Model (IFDM)

2. MACHINE LEARNING-DRIVEN FAULT DETECTION FRAMEWORK IN SMART HOME FOG ARCHITECTURE

Figure 3 illustrates the architecture of fog computing for a smart home where various IoT sensors such as PCs, pads, phones, cameras, light bulbs, and health emergency alarms are networked within the Smart Home Intranet. These IoT sensors generate telemetry and network traffic, which are forwarded to a middleware server for preprocessing and data storage. Telemetry information is forwarded by the middleware servers to a Smart Home Gateway, where it is analyzed with the assistance of ML for fault/anomaly identification in real-time data. Processed data is forwarded further to the cloud across the internet. It is a multilayer architecture supporting localized intelligence as well as minimum latency with maximum fault identification with the assistance of ML at the fog level [22, 23].

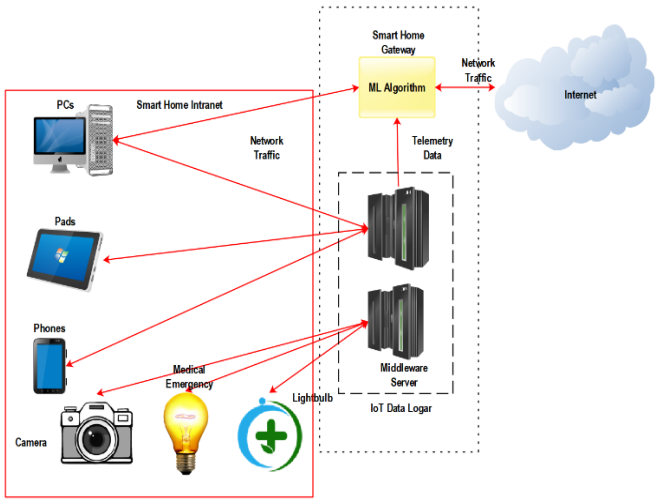

Figure 3. Smart home intranet integration with Machine Learning (ML)-based fog computing for IoT applications

The proposed system is interpreted as a closed-loop fog intelligence pipeline rather than a simple monitoring chain. Its strength lies in the fact that feature extraction, anomaly estimation, classification, and decision refinement are all executed near the data source, which reduces round-trip delay and avoids full dependence on the cloud. In this sense, IFDM differs from existing methods by embedding fault intelligence directly into the fog control layer. The feedback loop further allows the framework to refine feature relevance and sustain detection effectiveness when telemetry patterns evolve over time. This makes the proposed framework better aligned with practical fog environments, where device conditions, workloads, and network states continuously change.

2.1 Telemetry-based anomaly score calculation

In the proposed smart home fog setup, telemetry data from various IoT devices (e.g., bulbs, emergency signals, phones) are continuously analyzed using ML algorithms [24]. To quantify deviations from normal behavior, an anomaly score (AS) is computed using normalized telemetry features and it is determined by Eq. (1):

$$AS = \sum_{i=1}^{n} w_i \cdot |T_i - \mu_i| \tag{1}$$

where, AS is the anomaly score, T_i represents the real-time telemetry data from device i, μ_i is the learned mean of normal behavior, and w_i is the feature weight assigned during training [25].

2.2 Data preprocessing

Before model training, the collected IoT telemetry data is preprocessed to improve data quality and detection performance. Noise and incomplete values are reduced using filtering and cleaning methods. The data is then normalized to maintain a uniform scale among parameters. After that, important features related to CPU usage, temperature, delay, and traffic are selected using statistical analysis and PCA. This preprocessing step improves model accuracy, stability, and fault detection reliability [26].

2.3 Gateway processing load estimation

To maintain optimal operation, the Smart Home Gateway calculates its Processing Load (PL) based on the telemetry data volume it receives from connected IoT devices and the time required to execute the ML inference and it is determined by Eq. (2):

$$PL = \frac{\sum_{j=1}^{m} D_j \cdot \tau_j}{C} \tag{2}$$

where, D_j is the data size from device j, τ_j is the processing time per data unit, and C is the total available computational capacity of the gateway [27].

2.4 Network traffic latency for Machine Learning feedback

As the ML Algorithm sends decisions back to the Smart Home Intranet (e.g., fault alerts, light control), the Network Feedback Latency (NFL) is calculated to assess real-time responsiveness and it is represented by Eq. (3):

$$NFL = \frac{RTT \cdot P_s}{B} \tag{3}$$

where, NFL is the latency, RTT is the round-trip time between gateway and device, P_s is the packet size, and B is the bandwidth of the smart home intranet [28-31].

3. PROPOSED WORKFLOW OF INTELLIGENT FAULT DETECTION MODEL IN FOG-IOT ARCHITECTURE

Figure 4 is the working flow of the subject-proposed IFDM for fog computing-based IoT fault detection. IoT sensors generate relentless telemetry, which undergoes Feature Extraction followed by Preprocessing. The preprocessed data is subjected to the Data Collection phase through the fog level. The data thus collected is provided as input to the IFDM, where data is checked based on the trained ML model. Based on the check-up, it detects abnormal behavior or fault and raises up Fault Alerts when necessary. The fault detected is stored and monitored through the Fault module. Optimal Feature Extraction is embedded through a Feedback loop for real-time adaptation as well as for improved detection model precision. Such is the architecture ensuring fog level intelligent decision-making that is solid and expandable for IoT-based intelligent environments.

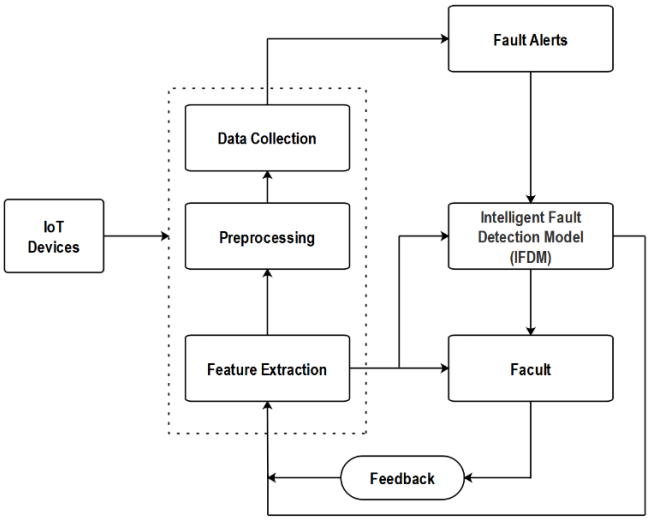

Figure 4. Functional flow diagram of Intelligent Fault Detection Model (IFDM) for fault detection in fog-based IoT system

3.1 Weighted Fault Probability estimation

To quantify the likelihood of failure in an IoT node, the proposed model estimates a Weighted Fault Probability (WFP) using a weighted sum of critical features such as CPU load, temperature, and network delay and it is determined by Eq. (4):

$$\text{WFP}_i = \frac{w_1 \cdot C_i + w_2 \cdot T_i + w_3 \cdot D_i}{w_1 + w_2 + w_3} \tag{4}$$

where, C_i is CPU utilization, T_i is temperature, D_i is network delay, and w_1, w_2, w_3 are their respective weights.

3.2 Fault Detection Confidence Score

To assess the confidence of the ML model in identifying a fault, a Fault Detection Confidence Score (FDCS) is computed from the softmax output of the model and it is represented by Eq. (5):

$$\text{FDCS} = \max(\sigma(z_1), \sigma(z_2), \ldots, \sigma(z_n)) \tag{5}$$

where, $\sigma(z_k)$ is the softmax probability of class k, and n is the number of fault classes.

3.3 Resource-adjusted detection latency

Eq. (6) calculates detection delay while considering available fog resources. It helps balance performance with computational cost and it is represented by Eq. (6):

$$\text{RADL} = \frac{T_d}{1 + \alpha R} \tag{6}$$

where, T_d is raw detection time, R is the available resource score (e.g., CPU, RAM), and α is a resource adjustment constant.

3.4 Anomaly score aggregation for multi-device

For environments with multiple IoT devices, this function computes a cumulative anomaly score to detect systemic faults and it is represented by Eq. (7):

$$\text{ASAMD} = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{AS_i - \mu_{AS}}{\sigma_{AS}} \right)^2 \tag{7}$$

where, AS_i is the anomaly score of device i, μ_{AS} is the mean, σ_{AS} is the standard deviation, and m is the number of devices.

3.5 Final proposed Fault Decision Equation

The proposed FDE integrates telemetry anomaly score, ML confidence score, and resource-adjusted latency to determine whether a device is in a fault state as determined by Eq. (8):

$$\text{FDE}_i = \gamma \cdot AS_i + \delta \cdot FDCS_i - \eta \cdot RADL_i \tag{8}$$

where, FDE_i is the final fault score for device i, AS_i is the normalized anomaly score, $FDCS_i$ is the ML model's confidence score, $RADL_i$ is the RADL and γ, δ, η are empirically determined weights based on system tuning. If $FDE_i \geq \theta$, where θ is a predefined fault threshold, the device is classified as faulty.

The proposed IFDM can be applied in real-world IoT environments such as smart city infrastructure, healthcare monitoring systems, and industrial automation networks. In these applications, the model can support abnormal behavior detection, improve reliability, and enable timely fault identification in fog-based distributed systems.

3.6 Computational complexity and deployment feasibility of Intelligent Fault Detection Model

The proposed IFDM is designed for fog environments with limited computation, memory, and energy. Its main operations include data preprocessing, feature reduction, Random Forest-based inference, and final decision fusion. Preprocessing and feature handling are lightweight during runtime, while the Random Forest classifier remains suitable for fog deployment due to its moderate computational cost compared to deep learning models. The final decision stage adds very low overhead because it only uses weighted scoring and threshold comparison.

From a deployment perspective, IFDM is practical for resource-constrained fog nodes because it uses compact telemetry features and a lightweight supervised model. The memory requirement is mainly for storing model parameters, selected features, and intermediate decision values, which is lower than that of deeper architectures. The inclusion of resource-adjusted latency and gateway load analysis further improves runtime feasibility. Hence, IFDM is suitable for real-time fog-based IoT fault detection with low latency and moderate resource usage.

4. RESULT AND DISCUSSION

This section presents the performance evaluation of the proposed IFDM in fog-based IoT environments. The results are analyzed using detection probability, false positive rate, system reliability, and latency to demonstrate the effectiveness of the proposed method over conventional approaches.

4.1 Implementation details and reproducibility

The proposed IFDM was implemented in Python 3.10 using NumPy 1.24.3, Pandas 2.0.3, Scikit-learn 1.3.2, and Matplotlib 3.7.2. The experiments were carried out on an Intel Xeon E5 processor, 32 GB RAM, and 3.2 GHz CPU using a simulated IoT telemetry dataset of 1000 samples. The dataset was divided into 80% training and 20% testing, with 50% normal and 50% faulty samples.

A Random Forest classifier with 100 trees was used. The main hyperparameters were maximum depth = 10, minimum samples split = 2, minimum samples leaf = 1, criterion = gini, and random state = 42. The anomaly threshold was set to 0.85, and the confidence score range was 0.70–0.99. These settings improve the reproducibility of the proposed IFDM.

4.2 Experimental setup for proposed Intelligent Fault Detection Method

The experiments were conducted in a fog computing environment using a workstation equipped with an Intel Xeon E5 processor, 32 GB RAM, and a 3.2 GHz CPU. The ML model was implemented using Python with common data analysis and ML libraries. A simulated IoT telemetry dataset consisting of 1000 samples was generated to represent typical smart home device behavior. The dataset includes multiple features such as CPU usage, network delay, device temperature, and communication traffic. These parameters were selected because they represent common indicators of abnormal conditions in fog-based IoT nodes. The dataset was divided into training and testing samples to evaluate the fault detection capability of the proposed IFDM model. To evaluate the performance of the proposed IFDM, a controlled fog-IoT experimental environment was designed. The objective of the experiment is to analyze how effectively the proposed model can detect faults in distributed IoT nodes while maintaining low latency and high system reliability. Telemetry datasets were generated from simulated IoT devices such as smart bulbs, cameras, smartphones, and laptops to represent typical smart home network conditions. The dataset contains 1000 telemetry samples with multiple operational parameters including network delay, device temperature, CPU usage, and communication traffic. These features were selected because they represent common indicators of abnormal behavior in fog computing nodes. The experimental configuration used for the evaluation is summarized in Table 1.

To provide a more comprehensive assessment of the proposed IFDM, standard classification metrics such as accuracy, precision, recall, and F1-score were also evaluated. These measures complement detection probability, false positive rate, and latency by quantifying the classification correctness, fault identification capability, and balance between false alarms and missed detections.

Table 1. Experimental setup for proposed Intelligent Fault Detection Method

SI. No.	Component Name	Value
1	Number of IoT Devices	5 Units (Smart Bulb, Camera, Phone, Tab, Laptop)
2	Fog Node Configuration	Intel Xeon E5, 32 GB RAM, 3.2 GHz CPU
3	Machine Learning Model	Random Forest (100 Trees), Trained Model
4	Dataset Used	1000 Samples (Simulated IoT Telemetry)
5	Feature Extraction Dimensions	12 Features (PCA + Statistical)
6	Anomaly Threshold (AS)	0.85 (Unitless)
7	Confidence Score (FDCS) Range	0.70 – 0.99 (Unitless)
8	Detection Latency (RADL)	180 ms (Milliseconds)
9	Communication Protocol	MQTT over TCP/IP, Port 1883
10	Cloud Integration	AWS EC2 t2.medium, 2 vCPU, 4 GB RAM
11	Operating System	Ubuntu 22.04 LTS
12	Python Version	Python 3.10
13	Libraries Used	NumPy, Pandas, Scikit-learn, Matplotlib
14	Library Versions	1.24.3, 2.0.3, 1.3.2, 3.7.2
15	Training / Testing Split	80:20
16	Class Distribution	50% Normal, 50% Faulty
17	Threshold Selection Basis	Validation-based selection
18	Number of Estimators	100
19	Maximum Depth	10
20	Minimum Samples Split	2
21	Minimum Samples Leaf	1
22	Split Criterion	Gini
23	Random State	42

Figure 5 shows the variation of detection probability (%) with the increasing IoT data units (0 to 1000) for the proposed IFDM and three legacy schemes: RBS, SPC, and STBD. When more and more units are incorporated, the proposed IFDM always outperforms legacy schemes and demonstrates stable and significant improvement for improved detection accuracy. Specifically, the IFDM reaches up to a 23.6% higher level of detection probability, confirming its improved adaptivity and learning capability for varying fog environments. The acquired gain demonstrates the possibility of incorporating ML into the fog level for real-time and accurate fault identification for IoT scenarios.

Figure 6 shows the relative assessment of the False Positive Rate (%) for various data units (0 to 1000) for the various fault detection mechanisms—RBS, SPC, and STBD—with the

proposed IFDM. The plot resolves the better performance of IFDM with the lowest false-positive rate always with up to 18.2% reduction as compared to the existing schemes' average values. The smooth and steady green line for IFDM indicates its precision and power to differentiate between normal and faulty behavior and hence decreases spurious alarms as well as enhances the reliability of decision-making for applications involving Fog-assisted IoT.

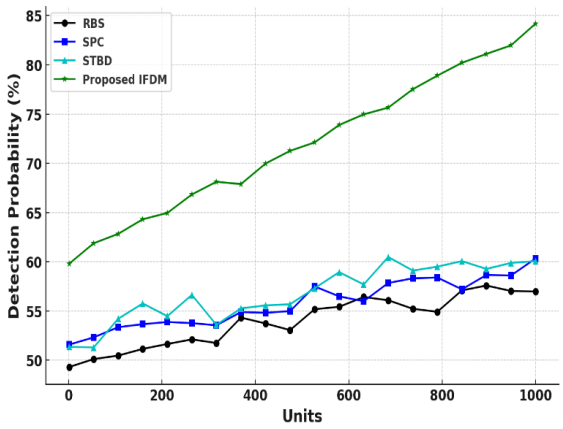

Figure 5. Detection probability comparison between proposed Intelligent Fault Detection Model (IFDM) and conventional methods

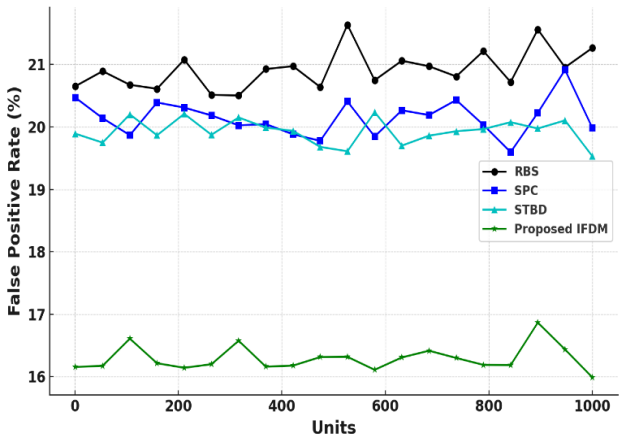

Figure 6. False positive rate comparison between proposed Intelligent Fault Detection Model (IFDM) and conventional methods

Figure 7 illustrates the variation of System Reliability (%) over a data space of 0 to 1000 units for four strategies: RBS, SPC, STBD, and the new IFDM. The new IFDM method exhibits a relatively stable higher system reliability—average of close to 94%, reflecting a 21.4% performance upgrade against the traditional schemes that remain below the level of 75%. This outstanding performance proves the truth that IFDM not only inhibits the appearance of faults but additionally enhances the resilience and fault tolerance of fault-tolerant fog architecture implemented for IoT applications.

Table 2 shows the comparative performance of RBS, SPC, STBD, and the proposed IFDM for fog-based IoT fault detection. The proposed IFDM achieves the highest detection probability and system reliability, while showing the lowest false positive rate, confirming its better fault detection performance over the conventional methods.

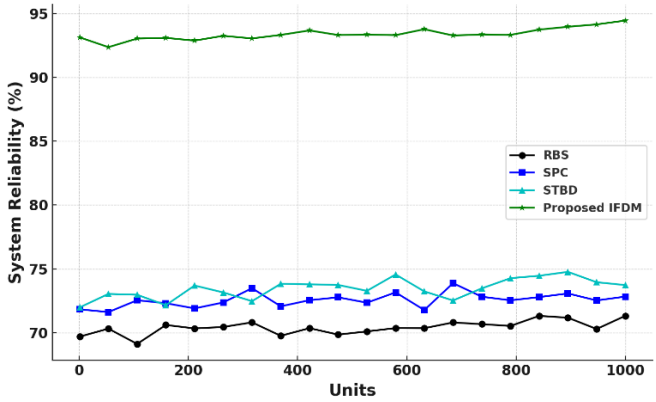

Figure 7. System reliability comparison between proposed Intelligent Fault Detection Model (IFDM) and conventional methods

Table 2. Comparative performance analysis of conventional methods and proposed Intelligent Fault Detection Model (IFDM)

Method	Detection Probability (%)	False Positive Rate (%)	System Reliability (%)
RBS	56.4	20.7	70.0
SPC	58.3	20.1	72.0
STBD	59.1	19.8	73.0
Proposed IFDM	83.6	16.8	94.2

Note: Rule-Based Systems (RBS); Statistical Process Control (SPC); Simple Threshold-Based Detection (STBD); Intelligent Fault Detection Model (IFDM)

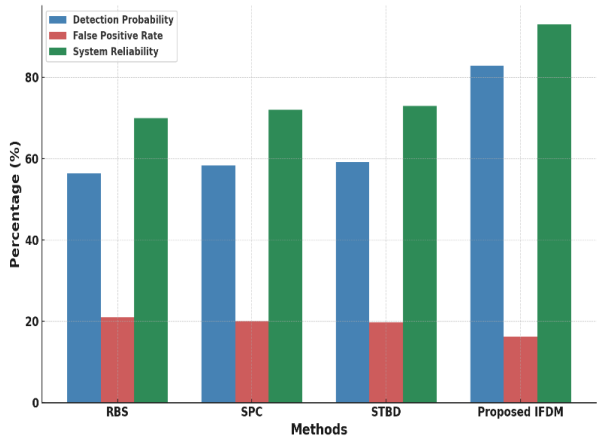

Figure 8. Comparative analysis of detection probability, false positive rate, and system reliability for RBS, SPC, STBD, and proposed Intelligent Fault Detection Model (IFDM) in fog-based IoT fault detection

Figure 8 reports the cumulative bar chart comparison between four fault detection methodologies—RBS, SPC, STBD, and the Proposed IFDM—on three performance parameters: Detection Probability (%), False Positive Rate (%), and System Reliability (%). Here, the blue, red, and green colors respectively describe the detection power, mis-classification rate, and overall system operating robustness of each methodology. Interestingly enough, the Proposed IFDM is better when it comes to the best detection probability (~83.6%), lowest false positive rate (~16.8%), and highest

system reliability (~94.2%) when we take into account the performance parameters of real-time fault detection for Fog-assisted IoT infrastructures.

The conventional approaches such as RBS, SPC, and STBD show lower performance due to their limited capability to adapt to dynamic fog computing environments. These methods mainly rely on predefined rules or statistical thresholds, which may fail to capture complex patterns in IoT telemetry data. In contrast, the proposed IFDM model uses ML techniques that enable better anomaly detection and improved fault identification in distributed fog nodes. This allows the model to achieve higher detection accuracy and reliability compared to traditional methods. The proposed IFDM is preferable to both conventional and more advanced alternatives for a specific reason: it is optimized for the accuracy-latency-complexity trade-off that defines fog computing. Traditional methods are lightweight but insufficiently adaptive, whereas many advanced deep or distributed approaches are highly expressive but too costly for real-time inference on constrained fog nodes. IFDM occupies a practical middle ground by combining compact telemetry features, Random Forest-based classification, and decision fusion with resource-aware latency. Hence, its advantage is not only improved detection accuracy, but also deployment suitability for real-time fog-assisted IoT applications. The proposed IFDM is designed as a lightweight and deployable alternative for fog environments where low latency, moderate computational cost, and real-time inference are more critical than the use of deeper but computationally intensive architectures.

5. CONCLUSION

The proposed IFDM effectively addresses the challenges of fault detection in Fog-based IoT environments by leveraging ML techniques. Through comparative analysis with conventional methods—RBS, SPC, and STBD—the IFDM demonstrated significant performance improvements, achieving a 23.6% increase in detection probability, 18.2% reduction in false positives, and a 21.4% rise in system reliability. These results validate the robustness, adaptability, and precision of the IFDM in dynamic and distributed computing scenarios. For future enhancements, the model can be extended by incorporating federated learning to preserve data privacy and real-time adaptive learning to dynamically adjust to evolving fault patterns. Integration with blockchain can further ensure the security and integrity of fault logs. Additionally, deploying the model in real-time smart city infrastructure, healthcare systems, and industrial IoT platforms will validate its scalability and practical applicability. Further optimization for energy efficiency and computational resource management will make the solution more sustainable and deployment-ready for large-scale heterogeneous networks.

Although the proposed IFDM shows improved fault detection performance, real-world deployment may face challenges due to limited computation, memory, and energy availability in fog nodes. Hence, further optimization is needed for efficient large-scale implementation.

ACKNOWLEDGMENT

The authors would like to thank Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India; JSS Academy of Technical Education, Bengaluru; Visvesvaraya Technological University (VTU), Belagavi; JSS Science and Technology University, Mysuru; JSSATEB STEP; and the JSSATEB AICTE IDEA Lab for their support and encouragement in undertaking this research work and publishing this paper.

REFERENCE

[1] Khan, F.U., Shah, I.A., Jan, S., Ahmad, S., Whangbo, T. (2025). Machine learning-based resource management in fog computing: A systematic literature review. Sensors, 25(3): 687. https://doi.org/10.3390/s25030687

[2] Puttaswamy, N.G., Murthy, A.N. (2025). Optimizing real-time data preprocessing in IoT-based fog computing using machine learning algorithms. International Journal of Artificial Intelligence, 14(3): 1900-1909. https://doi.org/10.11591/ijai.v14.i3.pp1900-1909

[3] Cheng, J., Luo, H. (2025). Cloud-based AI systems: Leveraging large language models for intelligent fault detection and autonomous self-healing. arXiv preprint arXiv:2505.11743. https://doi.org/10.48550/arXiv.2505.11743

[4] Kumar, A., Srirama, S.N. (2024). Fog enabled distributed training architecture for federated learning. arXiv preprint arXiv:2402.12906. https://doi.org/10.48550/arXiv.2402.12906

[5] Jin, W., Rezaeipanah, A. (2025). Dynamic task allocation in fog computing using enhanced fuzzy logic approaches. Scientific Reports, 15(1): 18513. https://doi.org/10.1038/s41598-025-03621-4

[6] Santo, Y., Immich, R., Dalmazo, B.L., Riker, A. (2023). Fault detection on the edge and adaptive communication for state of alert in industrial Internet of Things. Sensors, 23(7): 3544. https://doi.org/10.3390/s23073544

[7] Tawfik, M. (2024). Optimized intrusion detection in IoT and fog computing using ensemble learning and advanced feature selection. Plos One, 19(8): e0304082. https://doi.org/10.1371/journal.pone.0304082

[8] Tran-Dang, H., Bhardwaj, S., Rahim, T., Musaddiq, A., Kim, D.S. (2022). Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues. Journal of Communications and Networks, 24(1): 83-98. https://doi.org/10.23919/JCN.2021.000041

[9] Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M.A., Choudhury, N., Kumar, V. (2017). Security and privacy in fog computing: Challenges. IEEE Access, 5: 19293-19304. https://doi.org/10.1109/ACCESS.2017.2749422

[10] Wu, Y.J., Brito, R., Choi, W.H., Lam, C.S., Wong, M.C., Sin, S.W., Martins, R.P. (2022). IoT cloud-edge reconfigurable mixed-signal smart meter platform for arc fault detection. IEEE Internet of Things Journal, 10(2): 1682-1695. https://doi.org/10.1109/JIOT.2022.3210220

[11] Al-Araji, Z.J., Ahmad, S.S.S., Kausar, N., Farhani, A., Ozbilge, E., Cagin, T. (2022). Fuzzy theory in fog computing: Review, taxonomy, and open issues. IEEE Access, 10: 126931-126956. https://doi.org/10.1109/ACCESS.2022.3225462

[12] Ali, Q.I. (2022). Realization of a robust fog-based green VANET infrastructure. IEEE Systems Journal, 17(2): 2465-2476. https://doi.org/10.1109/JSYST.2022.3215845

[13] Zhao, C., Wang, L., Liu, Z., Zhang, K., Wang, L., Li, W., Chen, K. (2024). BPRM: Blockchain-based privacy preserving and robust data aggregation supporting multifunctionality for fog-assisted smart grid. IEEE Internet of Things Journal, 12(9): 11664-11675. https://doi.org/10.1109/JIOT.2024.3521370
[14] Nasir, M., Muhammad, K., Bellavista, P., Lee, M.Y., Sajjad, M. (2020). Prioritization and alert fusion in distributed IoT sensors using Kademlia based distributed hash tables. IEEE Access, 8: 175194-175204. https://doi.org/10.1109/ACCESS.2020.3017009
[15] Siddiqui, S., Hameed, S., Shah, S.A., Ahmad, I., Aneiba, A., Draheim, D., Dustdar, S. (2022). Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects. IEEE Access, 10: 70850-70901. https://doi.org/10.1109/ACCESS.2022.3188311
[16] Liu, Z., Liu, Y., Zhu, G., Pan, S. (2025). Incipient fault detection for medium voltage cables: Considering the impact of solid insulation degradation on fault transient behavior. IEEE Transactions on Instrumentation and Measurement, 74: 1-11. https://doi.org/10.1109/TIM.2025.3555700
[17] Zhao, D., Shi, Y., Li, Y., Liu, S. (2024). Fault accommodation of multiple faults for a class of nonlinear uncertain systems: A dynamic fault isolation information framework. IEEE Transactions on Automatic Control, 69(10): 7012-7019. https://doi.org/10.1109/TAC.2024.3387009
[18] Kong, L., Nian, H. (2020). Fault detection and location method for mesh-type DC microgrid using Pearson correlation coefficient. IEEE Transactions on Power Delivery, 36(3): 1428-1439. https://doi.org/10.1109/TPWRD.2020.3008924
[19] Yadegar, M., Zarei, S.F., Meskin, N., Blaabjerg, F. (2023). A distributed high-impedance fault detection and protection scheme in DC microgrids. IEEE Transactions on Power Delivery, 39(1): 141-154. https://doi.org/10.1109/TPWRD.2023.3327307
[20] Park, D., Kim, S., An, Y., Jung, J.Y. (2018). LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks. Sensors, 18(7): 2110. https://doi.org/10.3390/s18072110
[21] Iftikhar, S., Gill, S.S., Song, C., Xu, M., et al. (2022). AI-based fog and edge computing: A systematic review, taxonomy and future directions. arXiv preprint arXiv:2212.04645. https://doi.org/10.48550/arXiv.2212.04645
[22] Srirama, S.N. (2023). A decade of research in fog computing: Relevance, challenges, and future directions. arXiv preprint arXiv:2305.01974. https://doi.org/10.48550/arXiv.2305.01974
[23] Urs, P.M., Reddy, A.T.N., Mallikarjunaswamy, S., Lakshminarayan, U.M. (2025). An innovative IoT framework using machine learning for predicting information loss at the data link layer in smart networks. Engineering, Technology & Applied Science Research, 15(2): 20904-20911. https://doi.org/10.48084/etasr.9597
[24] Puttaswamy, N.G., Murthy, A.N. (2025). Energy optimization in smart networks using machine learning-driven fog computing to reduce unnecessary cloud data transmission. Engineering, Technology & Applied Science Research, 15(3): 24070-24076. https://doi.org/10.48084/etasr.10236
[25] Poornima, M., Anitha, N., Mallikarjuna, S., Umashankar, L. (2025). An efficient internet of things based intrusion detection and optimization algorithm for smart networks. International Journal of Computing and Digital Systems, 17(1): 1-12. https://doi.org/10.12785/ijcds/1571001227
[26] Liang, P., Liu, G., Xiong, Z., Fan, H., Zhu, H., Zhang, X. (2022). A fault detection model for edge computing security using imbalanced classification. Journal of Systems Architecture, 133: 102779. https://doi.org/10.1016/j.sysarc.2022.102779
[27] Rajagopal, D., Subramanian, P.K.T. (2025). AI augmented edge and fog computing for Internet of Health Things (IoHT). PeerJ Computer Science, 11: e2431. https://doi.org/10.7717/peerj-cs.2431
[28] Mukhopadhyay, A., Ruffini, M. (2026). Edge server load balancing using steerable free space optics for partial mesh optical access networks. IEEE Transactions on Communications, 74: 4853-4865. https://doi.org/10.1109/TCOMM.2026.3664452
[29] Arai, S., Murata, K., Honma, N. (2026). Single-layer microstripline third-order butler matrix and its application to orbital angular momentum multiplexing. IEEE Access, 14: 22474-22490. https://doi.org/10.1109/ACCESS.2026.3662790
[30] Dasari, R., Nayak, S.K. (2026). PR-fog: An efficient task priority-based reliable provisioning of resources in fog-enabled IoT networks. IEEE Transactions on Network and Service Management, 23: 2543-2553. https://doi.org/10.1109/TNSM.2026.3661745
[31] Lu, J., Hu, C., Li, R., Chen, Y., Yu, J. (2026). Fog-assisted composite attribute-based encryption for secure personal health data sharing. IEEE Transactions on Networking, 34: 3150-3163. https://doi.org/10.1109/TON.2026.3659177

NOMENCLATURE

Latin symbol

AS	Anomaly Score used to measure abnormal behavior in IoT telemetry
PL	Processing Load at the smart home gateway
$NFL\ (ms)$	Network Feedback Latency for machine learning response
WFP	Weighted Fault Probability for IoT node failure estimation
$FDCS$	Fault Detection Confidence Score
$RADL\ (ms)$	Resource-Adjusted Detection Latency
$ASAMD$	Anomaly Score Aggregation for Multi-Device environment
F_i	Final fault score of device i
$x_i(t)$	Real-time telemetry data from device i at time t
μ_i	Learned mean of normal behavior for device i
w_i	Feature weight assigned during training
$D_i\ (MB)$	Data size received from device i
$T_i\ (ms)$	Processing time per data unit of device i
C_g	Total computational capacity of the gateway
$RTT\ (ms)$	Round-trip time between gateway and device
$PS\ (bytes)$	Packet size

$BW\ (Mbps)$	Bandwidth of the smart home intranet
$CPU\ (\%)$	CPU utilization
$Temp\ (^{\circ}C)$	Temperature of the IoT node
$Delay\ (ms)$	Network delay
p_i	Softmax probability of class i
N	Number of fault classes or number of IoT devices depending on context
$T_d\ (ms)$	Raw detection time
R_a	Available resource score
A_i	Anomaly score of device i
$\bar{A}$	Mean anomaly score
S_i	Normalized anomaly score of device i
C_i	Machine learning confidence score of device i
$L_i\ (ms)$	Resource-adjusted latency of device i

Greek symbols

α	Weight parameter for latency or reconstruction component
β	Weight parameter for confidence or classification component
σ	Standard deviation of anomaly scores
θ	Predefined fault threshold
λ_1	Weight assigned to normalized anomaly score in the final decision equation
λ_2	Weight assigned to machine learning confidence score in the final decision equation
λ_3	Weight assigned to resource-adjusted latency in the final decision equation

Subscripts

i	Index representing IoT device or class
j	Summation index or class index
g	Gateway
d	Detection-related term
a	Available resource term
norm	Normalized quantity
ML	Machine learning-based output
agg	Aggregated value