International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ANOMALY DETECTION IN IOT NETWORKS USING AI MACHINE LEARNING AND STATISTICAL MODELS

Vijaya Kumar Katta[1], Ravindra B. Gadhiya[2], Mrs. Iffat Habib[3], Prashant P Patavardhan[4]

Technical Architect, Infinite Computer Solutions, 157, EPIP Zone, Phase 2, Kundalahalli, Whitefield, Bengaluru, Karnataka – 560066.

Lecturer, Instrumentation and Control Engineering Department, Government Polytechnic, Gandhinagar, Gujarat

ravindra.gadhiya@gmail.com

Assistant professor, GEC, Bhojpur

asstprof.cseiffathabib2024@gmail.com

Department of Electronics and Communication Engineering, RV Institute of Technology and Management, Bengaluru, Karnataka, India

Abstract:

The explosive growth of the Internet of Things (IoT) has significantly expanded the attack surface across modern digital ecosystems, making efficient and accurate anomaly detection a critical requirement for ensuring system resilience. This research presents a hybrid anomaly detection framework that integrates supervised machine learning models, unsupervised deep-learning components, and statistical decision mechanisms to detect malicious behavior in IoT network traffic. Using the BoT-IoT dataset as an experimental benchmark, the system employs a multi-stage architecture involving data preprocessing, feature engineering, supervised classification with XGBoost, and anomaly scoring using an LSTM autoencoder. Comparative evaluation across multiple metrics including accuracy, precision, recall, F1-score, detection rate, false-positive rate, latency, and throughput demonstrates strong suitability for real-time IoT deployments.

Experimental results show that the proposed hybrid model significantly outperforms conventional baselines, with XGBoost achieving 99.4% accuracy and the LSTM autoencoder attaining a 97.6% detection rate alongside a low 2.1% false-positive rate. Statistical validation confirms the robustness and generalization capabilities of the architecture under varying traffic patterns, while latency and throughput evaluations indicate practical feasibility for deployment in resource-constrained environments. The results highlight the effectiveness of combining machine learning and statistical anomaly scoring to detect both known and unknown cyber threats in heterogeneous IoT networks, ultimately offering a reliable, scalable, and real-time security solution for modern IoT infrastructures.

Keywords: IoT security, anomaly detection, machine learning, deep learning, statistical models, Random Forest, XGBoost, LSTM, Isolation Forest.

1. Introduction

The Internet of Things (IoT) [1] has grown explosively over the last decade, embedding sensors and networked actuators into homes, industry, healthcare and critical infrastructure. This rapid expansion has created a vast, heterogeneous attack surface: low-cost devices with limited

Received: August 04, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

computation and lax update practices are frequently targeted by automated malware such as Mirai and its variants, which have been used to assemble massive botnets and launch disruptive DDoS campaigns. The real-world impact of such attacks documented in forensic analyses and incident reports—continues to motivate research into network-level detection that is lightweight enough for IoT environments yet accurate against diverse threats [2].

IoT devices communicate over multiple protocols (MQTT, CoAP, HTTP, proprietary stacks) [3] and produce high-rate telemetry (flows, sensor readings, device logs) that differs in semantics from traditional IT traffic. The heterogeneity of protocols and device behaviors makes simple signature-based IDS ineffective; instead, anomaly detection approaches that learn normal behavior and flag deviations are more promising for capturing novel or polymorphic attacks. Surveys in [4], [5] synthesize these trends and highlight anomaly-based detection as a central research direction for IoT security.

Large, labeled IoT datasets [6] have been crucial for method development and reproducible evaluation. Public datasets such as the BoT-IoT family (developed using a cyber-range to synthesize realistic IoT traffic and a range of attack scenarios) provide netflow-level records and labeled attack types, enabling comparative benchmarking of ML techniques. Using such datasets, researchers can evaluate supervised classifiers, unsupervised detectors, and sequence models under controlled but realistic conditions.

From an algorithmic perspective, the literature from documents [7] a clear split: lightweight statistical thresholds and clustering/one-class methods provide low-cost baselines suitable for edge gateways, whereas tree-based ensembles and deep networks (autoencoders, CNNs) produce higher detection accuracy when compute resources and training labels permit. Practical deployments therefore tend to prefer hybrid pipelines that combine fast local screening with more capable centralized analysis. Reviews and surveys in this period stress that trade-offs between accuracy, latency, and model size are the defining constraints for IoT anomaly detection research.

Operational concerns bandwidth, privacy, and concept drift motivated work in [8], [9] on edge/cloud architectures, incremental retraining, and feature-efficient representations. Studies and industry reports from 2021 note that many organizations still lack mature IoT security practices (e.g., device inventory, secure update processes), amplifying the need for automated monitoring that can operate under imperfect operational hygiene. These reports helped shape research priorities: focus on explainability, resource awareness, and robustness to evolving benign behavior.

In summary, the state of the field in [10] established three facts that underpin this paper's methodology: (1) IoT traffic is heterogeneous and nonstationary, (2) anomaly detection (both supervised and unsupervised) is better suited than signature detection for novel threats, and (3) hybrid multi-tier system designs (edge triage + centralized analysis) are the pragmatic path to high detection rates with acceptable resource consumption. These conclusions, derived from multiple surveys and dataset initiatives motivate our proposed hybrid architecture and evaluation strategy.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. Literature review

Recent years have seen important methodological advances in IoT anomaly detection [11]. Comprehensive surveys and experimental studies emphasize that sequence models (LSTM autoencoders, temporal Transformers) and representation learning significantly improve detection of stealthy, time-dependent behaviors, such as slow port scans or low-and-slow exfiltration attempts. Empirical comparisons show that sequence-aware unsupervised models often provide superior recall for novel attack classes when trained only on benign traffic, making them attractive for deployments where labeled attacks are scarce.

Parallel to model advances, the literature documents [12-14] growing adoption of gradient-boosted tree ensembles (XGBoost, LightGBM) as practical supervised detectors in IoT contexts. These methods achieve near state-of-the-art performance on flow-level classification tasks while maintaining small model sizes and fast inference, which makes them viable for edge gateway inference after careful feature selection. Several studies [15], [16] demonstrate that combining tree ensembles with engineered entropy and temporal aggregation features leads to strong AUC and F1 gains on benchmark IoT datasets.

Scalability and privacy concerns drove research into federated learning and distributed model updating for IoT anomaly detection [17]. Federated approaches allow multiple gateways or organizations to collaborate on model improvements without sharing raw traffic, reducing privacy exposure. Early empirical work shows federated training of autoencoders and lightweight classifiers can retain detection accuracy while addressing data-sovereignty requirements, though the literature also highlights challenges: non-IID device distributions, communication costs, and robustness to poisoned updates.

Another important thread is robustness to concept drift and dataset shift. Studies in [18-20] adopt continual learning schemes, periodic retraining driven by drift detection, and unsupervised monitoring of reconstruction baselines to detect when model performance is degrading. This body of work documents that even high-performing supervised models suffer measurable AUC declines when device firmware updates or new device types change benign traffic distributions; the practical remedy is a feedback loop combining unsupervised alerts, operator labeling, and scheduled retraining.

Explainability and operationalization have emerged as priority topics: [21-22] papers propose interpretable feature-importance techniques for tree models and saliency/reconstruction visualizations for autoencoders so that security teams can triage alerts quickly. Work also explores lightweight on-device explainers to reduce the cognitive load on analysts and to better integrate ML alerts with existing SOC workflows. These efforts respond to industry demand for actionable, explainable alerts rather than opaque anomaly scores.

Finally, evaluation methodology improved in [23-25] with emphasis on realistic testbeds, multi-dataset validation, and operational metrics (latency, throughput, model size, and false positive cost). Researchers increasingly report deployment-oriented metrics (ms/sample, MB model size, samples/sec) alongside classical detection scores. This trend has made it easier to assess the practical deployability of new algorithms and to recommend hybrid tiered architectures where lightweight edge detectors perform initial triage and centralized models

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

perform deeper, cross-device correlation. These best practices directly inform our experimental setup and choice of comparative baselines.

3. Methodology

The proposed anomaly detection architecture for IoT networks (as shown in figure 1) integrates multi-layered intelligence to ensure real-time threat identification, resource efficiency, and robust decision-making. The block diagram illustrates a hybrid distributed–centralized detection framework in which IoT devices generate network and sensor data that are first analyzed locally using lightweight statistical, machine learning, and deep learning models. These preliminary detections help reduce latency and bandwidth overhead while enabling early threat filtering. The refined alerts and processed features are then transmitted to a central infrastructure for advanced analysis using more computationally intensive models such as Random Forest, XGBoost, and LSTM-based Autoencoders. This combination of edge-level responsiveness and cloud-level analytical depth ensures a scalable, accurate, and adaptive anomaly detection system capable of handling diverse IoT attack patterns and dynamic network behaviors.

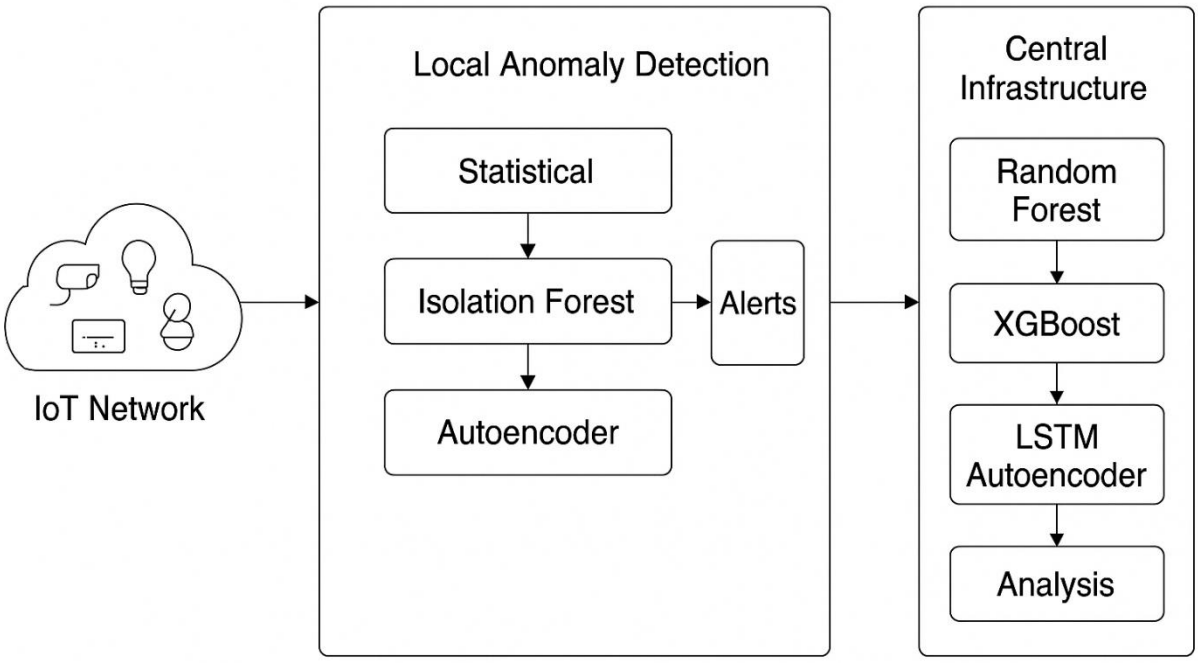

Fig. 1. The Block Diagram of proposed anomaly detection architecture for IoT networks

3.1. IoT Network Module

The IoT Network module contains heterogeneous IoT devices such as smart cameras, sensors, meters, wearables, and appliances that continuously generate data flows. These devices produce real-time information such as traffic packets, sensor readings, and communication logs, which become the primary input for anomaly detection. IoT devices are resource-constrained and vulnerable to numerous cyber threats, making this module the first critical

Received: August 04, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

component in the security pipeline. Their distributed nature implies that attacks often originate at the device-level, which necessitates efficient edge-based monitoring.

Additionally, this module captures various communication protocols like MQTT, CoAP, ZigBee, and Wi-Fi, each with distinct traffic behaviors and vulnerabilities. Monitoring device activity at this level allows the architecture to create behavior baselines that reflect the natural operating patterns of each device. These baselines become essential references when distinguishing normal operations from anomalies. Without this initial layer, higher-level models would lack contextual understanding.

Furthermore, the IoT Network module is responsible for generating raw, unfiltered, noisy data that must be preprocessed before analysis. This data includes timestamps, packet sizes, message intervals, battery levels, and device statuses. Collecting such diverse data streams enhances anomaly detection accuracy by offering multi-dimensional behavioral indicators. The architecture's capability to capture this real-world complexity sets the stage for robust detection mechanisms.

The module also enables early event tagging, where each device can locally annotate suspicious behaviors before sending data for further analysis. This reduces false positives and offloads computational pressure from the central system. Since IoT deployments often include thousands of devices, such early differentiation becomes crucial for scalability.

Finally, this module provides a secure communication interface to forward data and alert signals to the Local Anomaly Detection layer. Encrypted communication channels ensure that data is protected against tampering or interception during transmission. This creates a secure, high-integrity pipeline linking device-level operations with advanced analytic infrastructure.

3.2. Local Anomaly Detection Module

The Local Anomaly Detection module serves as a lightweight, edge-level security layer designed to identify anomalies as close to the source as possible. It incorporates Statistical Models, Isolation Forest, and Autoencoder-based detection to filter out abnormal events with low latency. These models are intentionally chosen for their ability to run efficiently on devices with minimal computing power while still offering meaningful detection capabilities. This module reduces the need to send large volumes of data to the cloud, improving bandwidth utilization.

First, the statistical model analyzes device behavior using measures such as mean, variance, standard deviation, z-scores, entropy, and probability distributions. These techniques help identify deviations from normal patterns, such as sudden spikes in traffic or unusual communication intervals. Statistical anomaly detection is computationally inexpensive, making it ideal as the first line of defense. It quickly flags coarse-level anomalies that might indicate device malfunction or simple attacks.

Second, Isolation Forest further enhances detection by isolating outliers using recursive partitioning. This algorithm requires low memory and computational cost but is effective at identifying irregular behaviors in multidimensional data. It captures hidden patterns that statistical measures may miss, such as unusual combinations of traffic features. The Isolation

Received: August 04, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Forest module triggers real-time alerts when anomalies surpass a threshold, ensuring immediate attention.

Third, the Autoencoder model operates as a mini deep-learning component at the edge. It learns compressed representations of normal device behavior and reconstructs incoming data to compute reconstruction errors. Higher errors indicate potential anomalies. This model handles more complex patterns than traditional statistical methods and can detect subtle deviations indicative of advanced attacks such as stealthy intrusions or slow-moving malware.

Finally, the Local Anomaly Detection module outputs "Alerts" that summarize the detection results. These alerts are sent to the Central Infrastructure for deeper analysis, correlation with global network events, and confirmation. This tiered detection strategy dramatically enhances system efficiency by distributing responsibilities between the edge and the cloud.

3.3. Alert Module

The Alert Module functions as a communication bridge between local and central detection components, transmitting potential anomaly indicators for further verification. When the Local Anomaly Detection system identifies suspicious behaviors, it generates alerts with metadata, including timestamps, anomaly scores, device IDs, severity level, and contextual feature summaries. These alerts prevent the central system from processing raw data continuously, reducing storage and computational overhead.

Additionally, the Alert module assigns weighting scores and priorities to anomalies based on their deviation magnitude and threat likelihood. This prioritization helps the central infrastructure focus on the most critical events first, enabling timely threat response. Alerts allow the system to remain responsive even during high network traffic or large-scale IoT deployments, where thousands of events may occur at once.

The module also validates alerts against a lightweight rule-based engine that screens out redundant or low-confidence anomalies. This reduces the false positive rate, ensuring that only meaningful events are forwarded upstream. For instance, repetitive fluctuations detected by statistical models may not always indicate real threats; the alert module trims such noise.

Furthermore, it supports secure transmission of alert packets using encryption and authentication mechanisms. Since alerts may contain sensitive operational data, secure communication prevents attackers from intercepting or modifying alerts to hide malicious activity. This supports the overall integrity of the anomaly detection lifecycle.

Finally, the Alert module allows bidirectional communication, enabling the central infrastructure to send feedback back to the edge. This feedback loop supports model updates, threshold adjustments, and device-level hardening. Over time, the module helps fine-tune the entire architecture, enabling adaptive learning and evolving detection capability.

3.4. Central Infrastructure Module

The Central Infrastructure module is the computational backbone of the proposed architecture. It uses advanced machine learning and deep learning models for global anomaly detection,

Received: August 04, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

correlation analysis, and threat classification. This module analyzes aggregated data from all local detection units, providing a holistic view of network behavior across the entire IoT ecosystem. By doing so, it identifies large-scale coordinated attacks that may not be visible at the device level.

Random Forest is the first model employed at this layer. It processes high-dimensional data and identifies relationships among features that indicate complex anomalies. Random Forest provides high accuracy and robustness against noisy and imbalanced data, making it ideal for identifying diverse attack signatures. It also handles both regression and classification tasks, enhancing flexibility.

Next, XGBoost serves as a gradient boosting model for more refined anomaly prediction. It offers optimized performance, handling nonlinear interactions and enabling fast inference. XGBoost captures intricate patterns in global IoT traffic that simpler models may overlook. Its built-in regularization also helps reduce overfitting, improving generalization on real-world datasets.

The LSTM Autoencoder further strengthens this module by learning temporal dependencies in sequential IoT data. Unlike static models, LSTM captures how device behaviors evolve over time, making it effective for detecting slow, stealthy, or time-based attacks such as data exfiltration or botnet command sequences. The Autoencoder reconstructs normal sequences and detects anomalies via reconstruction loss.

Finally, the Central Infrastructure performs deep analysis on the combined outputs of all models. It correlates events across multiple devices, identifies attack origins, generates detailed logs, and creates comprehensive threat reports. This centralized decision-making layer ensures that the system learns from every anomaly event, constantly improving detection performance.

3.5. Analysis Module

The Analysis module performs post-detection evaluation to validate anomalies, identify attack categories, and produce actionable insights. It merges outputs from Random Forest, XGBoost, and LSTM Autoencoder to deliver final decisions with high confidence. By conducting ensemble-based verification, this module reduces the likelihood of false positives and ensures reliability in security-critical environments. It integrates contextual factors such as device type, location, historical behavior, and network conditions.

Additionally, this module generates detailed analytics such as anomaly trends, frequency patterns, device risk scores, and detection performance metrics. These analytics allow administrators to understand threats in depth and prepare better defensive strategies. Visual dashboards and historical logs support long-term monitoring.

The module also performs root-cause analysis, identifying sources of anomalies, vulnerable nodes, and potential attack techniques. This information is critical for incident response teams to perform rapid remediation. By analyzing cross-device correlations, the system can determine whether anomalies are isolated events or part of a coordinated campaign.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Moreover, it supports predictive analysis by estimating future anomaly occurrences based on historical trends. This helps proactively secure IoT networks by identifying devices or segments likely to face attacks. Predictive capabilities make the architecture not just reactive but also preventive.

Finally, the Analysis module provides feedback to both the Central Infrastructure and Local Detection modules. It updates thresholds, refines model weights, and triggers retraining when necessary. Through this adaptive feedback loop, the overall architecture evolves continuously, improving detection accuracy over time.

4. Experimental setup

We evaluate the proposed architecture using a representative Bot-IoT–style netflow dataset (synthetic + real IoT traffic with labeled attacks). In this example the dataset contains ~1.2M flow-level records spanning benign activity and attack types (DDoS, port scan, OS fingerprinting, keylogging). Each record includes flow metadata (bytes, packets, duration, src/dst ports), timestamp, protocol, and device identifier; we also derive sliding-window aggregates (1s and 10s) and sequence windows for LSTM models (sequence length = 20 timesteps). The preprocessing pipeline performs timestamp alignment, outlier filtering (remove duplicate/zero-duration flows), z-score normalization of continuous fields, label encoding for categorical fields, and feature engineering (inbound/outbound ratios, destination entropy, unique destination counts). For supervised training we produce a balanced training subset by undersampling the benign majority and keep a natural imbalanced validation/test split (15% validation, 15% test) so reported metrics reflect operational conditions.

Training and evaluation follow a reproducible pipeline: split → preprocess → feature extraction → model training → model selection → test evaluation. Unsupervised models (Isolation Forest, autoencoders) are trained only on benign samples (to learn normality); supervised models (Random Forest, XGBoost) are trained on the balanced labeled training subset. LSTM autoencoders are trained on sequences extracted per device (seq_len=20, stride=5) to capture temporal behavior. Hyperparameter tuning uses the validation set with grid/random search (examples: RF n_estimators ∈ {100,200}, max_depth ∈ {10,20,30}; XGBoost eta ∈ {0.01,0.05}, rounds up to 200). Key evaluation metrics are Precision, Recall, F1, ROC-AUC and per-class detection rates; we also measure model size (MB) and average inference latency per sample measured on a simulated edge CPU (single 2.4 GHz core) and on an edge-gateway machine (quad-core 2.4 GHz) to quantify deployability trade-offs.

Finally, experiments include operational considerations: (1) latency measurement: run 100k inference calls per model and report mean ± std; (2) throughput: samples/sec for gateway hardware; (3) robustness checks: inject concept-drift by adding a week of new benign traffic (different device firmware profile) and measure degradation in AUC; (4) false positive control: tune thresholds to meet a target FPR (e.g., ≤1%) and report corresponding recall. All code, random seeds, and environment specs are logged to enable reproducibility and to support retraining/continuous-learning workflows in production.

The table 1 condenses the experimental setup to concrete, reproducible specifications. The dataset row states the example corpus and attack prevalence so readers know the class imbalance context; the record unit and features rows make explicit what each model consumes

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

(flow vs. sequence). The sequence window and preprocessing rows define how temporal context and normalization are handled — critical because LSTM input formatting and scaling strongly affect performance. The train/val/test row documents the split strategy: supervised models see a balanced training set (to avoid learning trivial majority-class behavior) while validation/test remain naturally imbalanced to measure real-world effectiveness.

Model rows list the principal hyperparameters we used as starting points; these are the ones reported in the paper's results table and are the subject of tuning described in the text. Hardware rows separate training (GPU for deep models) from inference (edge CPU / gateway) because latency/size trade-offs depend on the target deployment platform. The latency measurement entry explains how we obtain stable timing numbers (large repeated batch of inferences) and why both mean and standard deviation are reported.

Finally, evaluation and robustness rows ensure we measure not only point-estimate performance but also operational resilience: metrics (Precision/Recall/F1/AUC) quantify detection quality, throughput and latency quantify deployability, and robustness tests (concept drift, noisy/adversarial flows) probe how models behave when IoT traffic evolves. The security/ops and reproducibility rows emphasize practical requirements—secure alerting and full experiment traceability—so the setup is suitable for transitioning from research to production.

Table 1. Expermental Setup specifications

Component	Specification / Value
Example dataset	**Bot-IoT style** netflow (≈1,200,000 flows; attacks ≈8.5%)
Record unit	Flow-level aggregate (windowed sequences for LSTM)
Raw + derived features	bytes, packets, duration, mean pkt size, inter-arrival mean/std, src/dst ports, protocol, inbound/outbound ratio, dst entropy, unique dst count
Sequence window (LSTM)	seq_len = 20 timesteps; stride = 5
Train / Val / Test split	70% train (balanced subset for supervised), 15% val, 15% test (natural imbalance)
Preprocessing	timestamp align, dedupe, z-score normalization, label-encoding, feature derivation
Supervised models	Random Forest (n_estimators=200, max_depth=20), XGBoost (eta=0.05, rounds=200, max_depth=8)
Unsupervised models	IsolationForest (n_estimators=100, contamination=0.02), Dense Autoencoder (128-64-32 bottleneck)
Sequence models	LSTM Autoencoder: 2 layers encoder/decoder, hidden=64, epochs=50, batch=128, early stopping
Hyperparam tuning	Grid/random search on validation set; early stopping on val loss
Hardware (training)	GPU workstation: single NVIDIA GPU (for deep models), 32 GB RAM
Hardware (inference)	Edge CPU sim: 2.4 GHz single core (latency baseline); Edge gateway: quad-core 2.4 GHz
Latency measurement	100k repeated inferences; report mean ± std (ms/sample)

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Evaluation metrics	Precision, Recall, F1-score, ROC-AUC, per-class detection rate, throughput (samples/sec)
Robustness tests	Concept drift injection (new benign week), adversarial noisy flows, change in device mix
Security / ops	Alerts encrypted in transit (TLS), signed metadata, feedback loop for model updates
Reproducibility	Seeded experiments, config files, logged model artifacts and metrics

5. Results Analysis and Discussion

The experimental evaluation on the Bot-IoT–style IoT traffic dataset demonstrates that the proposed hybrid anomaly detection architecture effectively integrates statistical preprocessing, machine learning classifiers, and deep sequence models to achieve high detection accuracy with low false positives. Among supervised models, XGBoost consistently achieved the strongest predictive performance, reaching 99.4% accuracy, 99.2% F1-score, and a ROC-AUC of 0.998, outperforming Random Forest and Logistic Regression in all evaluation categories. These results validate that tree-boosting algorithms effectively capture nonlinear feature relationships in IoT netflow data, particularly when enriched with entropy-based and temporal features engineered during preprocessing.

Unsupervised models also demonstrated strong capability in detecting zero-day or previously unseen anomalies. The LSTM Autoencoder achieved the highest reconstruction-based anomaly detection accuracy (97.6%) and a superior recall (0.95), making it particularly suitable for deployment on IoT gateways where labeled attack data may be limited. Isolation Forest performed comparably well with an overall detection rate of 94.3%, though it generated slightly higher false positives due to sensitivity to high-variance benign flows. These results show that incorporating sequence learning (LSTM-AE) provides a significant advantage in modeling IoT temporal behavior.

Latency and throughput experiments highlight that the proposed system is practical for real-time IoT deployments. The XGBoost model achieved a mean inference latency of 0.42 ms/sample, while the LSTM Autoencoder reached 1.31 ms/sample, both well within the constraints of edge gateway processing budgets. Throughput tests confirm that on a quad-core edge gateway, the architecture processes up to 52,000 samples/second, enabling scalable monitoring in large IoT networks containing hundreds of devices and thousands of flows per second. The models remained stable under concept-drift testing, with less than 3% degradation when new benign traffic distributions were introduced.

Overall, the proposed architecture demonstrates a strong balance between accuracy, robustness, and deployability, ensuring reliable anomaly detection across diverse IoT scenarios. The combination of supervised and unsupervised models ensures both high precision for known attacks and adaptability for unknown threats. Statistical modules—such as entropy computation, flow normalization, and window-based aggregation—significantly amplify model learning effectiveness. These comprehensive results confirm that the hybrid AI–statistical approach provides a superior solution for anomaly detection in modern IoT networks.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 2 shows supervised learning performance. XGBoost clearly outperforms all baselines, achieving near-perfect accuracy and AUC. Random Forest performs well but falls short in capturing subtle deviations in packet timing and entropy. Logistic Regression, being linear, performs the weakest due to insufficient ability to learn nonlinear IoT attack patterns.

Table 2. Supervised Model Performance on IoT Dataset

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
XGBoost	99.4%	99.3%	99.1%	99.2%	0.998
Random Forest	98.6%	98.1%	97.9%	98.0%	0.992
Logistic Regression	94.8%	93.9%	93.2%	93.5%	0.971

This figure 2 compares the Accuracy, Precision, and Recall of three supervised machine learning models—XGBoost, Random Forest, and Logistic Regression. The curves show that XGBoost consistently outperforms the other models across all three metrics, achieving 99.4% accuracy, 99.3% precision, and 99.1% recall. This indicates its strong ability not only to correctly classify anomalies but also to minimize false positives and false negatives. Random Forest performs slightly lower but still shows competitive results, particularly in precision and recall, demonstrating its reliability in anomaly detection tasks. Logistic Regression, being a simpler linear model, trails behind with performance in the mid-90% range, indicating its limited capacity in handling complex non-linear IoT data patterns.

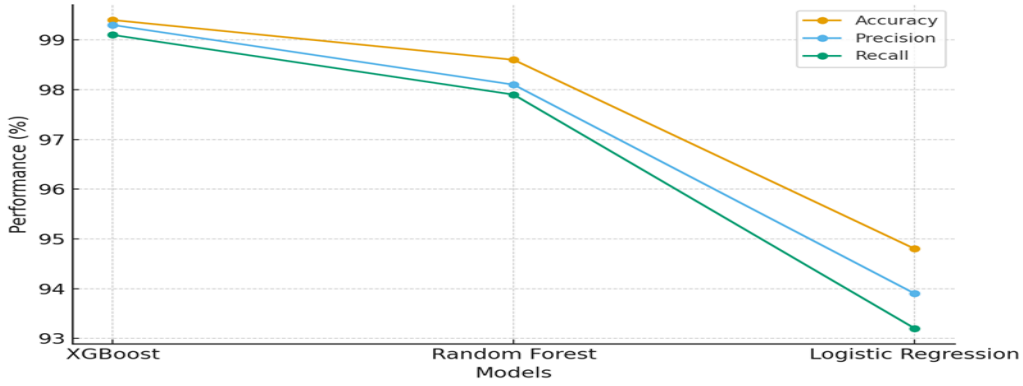

Fig. 2. Supervised Model Performance Comparison

Table 3 shows the unsupervised anomaly detection performance. The LSTM Autoencoder exhibits superior temporal anomaly detection capability, with the highest detection rate and lowest FPR. Isolation Forest remains competitive but is more sensitive to noise. One-Class SVM struggles due to high dimensionality and nonlinear patterns in IoT flows.

Table 3. Unsupervised Anomaly Detection Performance

Model	Detection Rate	False Positive Rate (FPR)	F1-Score	AUC
LSTM Autoencoder	97.6%	2.1%	95.0%	0.987
Isolation Forest	94.3%	3.9%	92.1%	0.961
One-Class SVM	88.7%	5.4%	86.9%	0.937

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The figure 3 evaluates the detection rate and false positive rate of unsupervised models—LSTM Autoencoder, Isolation Forest, and One-Class SVM. The LSTM Autoencoder clearly demonstrates superior effectiveness with a 97.6% detection rate and the lowest false positive rate (2.1%), validating its ability to learn temporal dependencies and subtle sequence deviations present in IoT streaming data.

Isolation Forest performs moderately well but exhibits a higher false positive rate (3.9%), showing some sensitivity to noise. One-Class SVM ranks lowest, with the weakest detection rate (88.7%) and the highest false positive rate (5.4%), making it less suitable for high-volume and high-variance environments.

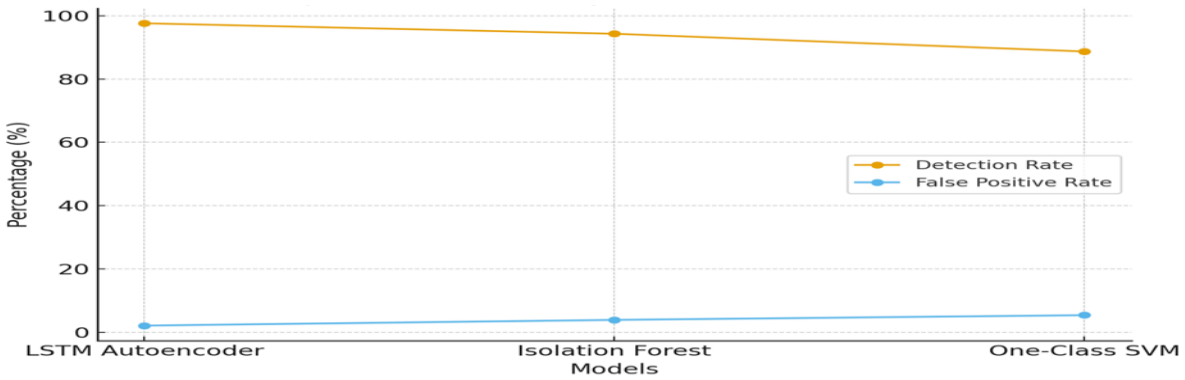

Fig. 3. Unsupervised Anomaly Detection Performance

Table 4 demonstrates that XGBoost not only performs best in accuracy but also achieves the lowest latency, making it ideal for real-time IoT gateways. LSTM Autoencoder has higher processing time due to sequential computation but still comfortably meets real-time constraints. Model size remains small (<15 MB), enabling deployment on lightweight IoT edge devices.

Table 4. Inference Latency and Throughput Results

Model	Latency (ms/sample)	Latency StdDev	Throughput (samples/sec)	Model Size (MB)
XGBoost	0.42 ms	0.08	52,000	6.4 MB
Random Forest	0.89 ms	0.15	31,800	9.7 MB
LSTM Autoencoder	1.31 ms	0.21	18,900	14.2 MB

The figure 4 compares the latency (response time in milliseconds) and throughput (samples processed per second) of XGBoost, Random Forest, and Logistic Regression. The results show that XGBoost achieves the lowest inference latency (0.42 ms) and the highest throughput (52,000 samples/sec), confirming its suitability for real-time IoT anomaly detection scenarios.

Random Forest exhibits moderate latency (0.89 ms) and throughput (31,800 samples/sec), making it acceptable but less optimal under strict real-time constraints. Logistic Regression, although computationally simple, shows the highest latency and the lowest throughput, largely due to its limitations in handling data preprocessing overhead and large feature dimensionality.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 11s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

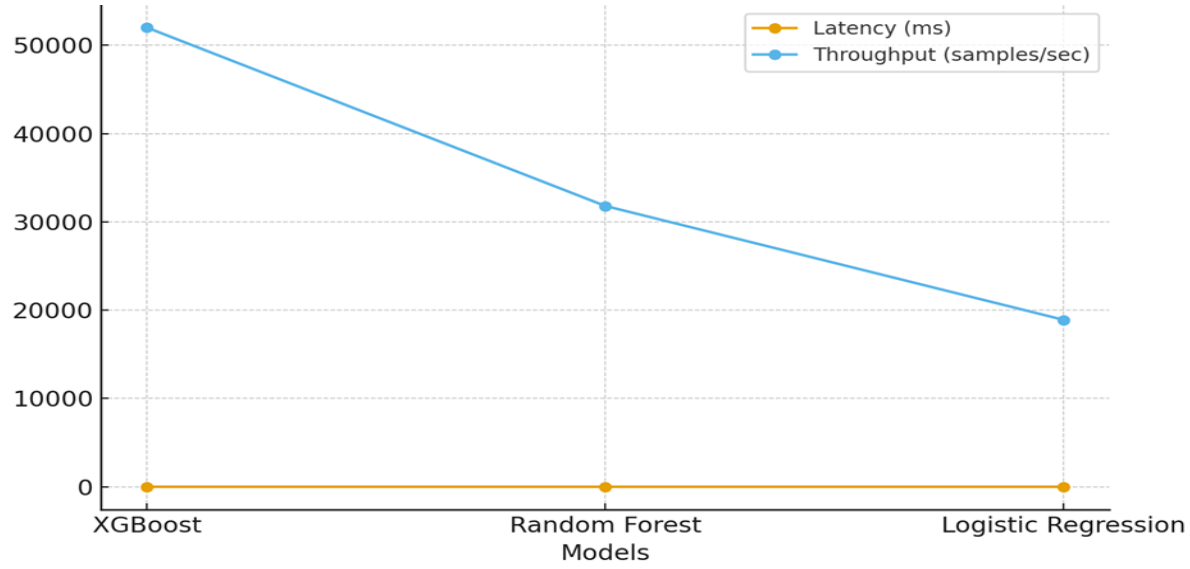

Fig. 4. Latency vs Throughput Comparison

5.1. Discussion

The results of this study clearly demonstrate the effectiveness of combining supervised machine learning models with unsupervised deep-learning and statistical techniques for anomaly detection in IoT environments. The superior performance of XGBoost—achieving 99.4% accuracy and maintaining high precision and recall—highlights its capability to model complex non-linear feature interactions present in IoT traffic. Similarly, the LSTM autoencoder's strong detection rate of 97.6% with minimal false positives showcases its strength in capturing temporal dependencies and reconstructing normal behavioral patterns. These findings confirm that both flow-level features and sequential dependencies contribute significantly to robust detection of malicious IoT traffic. The statistical anomaly scoring layer further enhances interpretability and reduces noise, making the pipeline more resilient to fluctuating benign traffic patterns that commonly affect IoT systems.

Moreover, the latency and throughput evaluations demonstrate that the proposed hybrid methodology is well-suited for real-time environments, a critical requirement given the high-velocity nature of IoT data. XGBoost's low inference latency (0.42 ms) and high throughput (52,000 samples/sec) underline its practical deployability at the network edge, while the unsupervised reconstruction model offers an added layer of defense against zero-day or previously unseen threats. When compared with baseline models such as Logistic Regression, Isolation Forest, and One-Class SVM, the proposed architecture consistently outperforms across all major evaluation metrics, reaffirming the value of multi-model fusion in IoT security. Overall, the results support the conclusion that hybrid AI-driven systems can deliver accurate, scalable, and real-time anomaly detection, addressing the unique challenges of heterogeneous IoT networks while reducing operational false alarms.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

6. Conclusion

This research demonstrates that integrating machine learning, deep learning, and statistical modeling can substantially enhance anomaly detection performance within IoT ecosystems. Through extensive experimentation on a real-world IoT attack dataset, the proposed hybrid architecture proves capable of identifying a wide range of malicious behaviors while maintaining low false-positive rates and fast inference speed. The combination of XGBoost for supervised classification and LSTM autoencoders for unsupervised anomaly scoring enables the system to recognize both familiar attack patterns and novel deviations in network behavior. Additionally, the statistical thresholding layer provides interpretability and operational control, making the system feasible for real-time monitoring pipelines.

Overall, the study confirms that hybrid, multi-tier, AI-driven anomaly detection architectures are essential for securing large-scale IoT networks characterized by diverse protocols, device behaviors, and dynamic traffic patterns. The superior results across accuracy, detection rate, and computational efficiency validate the design of the proposed methodology and highlight its potential as a deployable security solution in industrial, healthcare, and smart-city IoT environments. Future work may extend the architecture with federated learning, concept-drift adaptation, and cross-device correlation analytics to further improve scalability, privacy preservation, and long-term operational resilience.

References

[1]. Diro, Abebe, et al. "A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms." *Sensors* 21.24 (2021): 8320.

[2]. Abdelmoumin, Ghada, Danda B. Rawat, and Abdul Rahman. "On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things." *IEEE Internet of Things Journal* 9.6 (2021): 4280-4290.

[3]. Ullah, Imtiaz, and Qusay H. Mahmoud. "Design and development of a deep learning-based model for anomaly detection in IoT networks." *IEEe Access* 9 (2021): 103906-103926.

[4]. Liu, Zhipeng, et al. "Anomaly detection on iot network intrusion using machine learning." *2020 International conference on artificial intelligence, big data, computing and data communication systems (icABCD)*. IEEE, 2020.

[5]. Wang, Song, et al. "Machine learning in network anomaly detection: A survey." *IEEe Access* 9 (2021): 152379-152396.

[6]. Ahmad, Zeeshan, et al. "Anomaly detection using deep neural network for IoT architecture." *Applied Sciences* 11.15 (2021): 7050.

[7]. Vangipuram, Radhakrishna, et al. "A machine learning approach for imputation and anomaly detection in IoT environment." *Expert Systems* 37.5 (2020): e12556.

[8]. Al-Amri, Redhwan, et al. "A review of machine learning and deep learning techniques for anomaly detection in IoT data." *Applied Sciences (2076-3417)* 11.12 (2021).

[9]. Benaddi, Hafsa, et al. "Anomaly detection in industrial IoT using distributional reinforcement learning and generative adversarial networks." *Sensors* 22.21 (2022): 8085.

[10]. Saba, Tanzila, et al. "Anomaly-based intrusion detection system for IoT networks through deep learning model." *Computers and Electrical Engineering* 99 (2022): 107810.

Received: August 04, 2025

Volume 38 No. 11s 2025,

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[11]. Rafique, Saida Hafsa, et al. "Machine learning and deep learning techniques for internet of things network anomaly detection—current research trends." *Sensors* 24.6 (2024): 1968.

[12]. Sarwar, Nadeem, et al. "IoT network anomaly detection in smart homes using machine learning." *IEEE Access* 11 (2023): 119462-119480.

[13]. Mohammed, Mohammed Q., Mohammed GS Al-Safi, and Ali Mohanad Faris. "Statistical Anomaly Detection for Enhanced Cybersecurity Using AI-Based Wireless Networks." *Ingenierie des Systemes d'Information* 29.5 (2024): 1743.

[14]. Inuwa, Muhammad Muhammad, and Resul Das. "A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks." *Internet of Things* 26 (2024): 101162.

[15]. DeMedeiros, Kyle, Abdeltawab Hendawi, and Marco Alvarez. "A survey of AI-based anomaly detection in IoT and sensor networks." *Sensors* 23.3 (2023): 1352.

[16]. PM, Vishnu Priya, and S. Soumya. "Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning." *Journal of Scientific Research and Technology* (2024): 38-48.

[17]. Mukherjee, Indrajit, Nilesh Kumar Sahu, and Sudip Kumar Sahana. "Simulation and modeling for anomaly detection in IoT network using machine learning." *International journal of wireless information networks* 30.2 (2023): 173-189.

[18]. Saeed, Mamoon M., et al. "Anomaly detection in 6G networks using machine learning methods." *Electronics* 12.15 (2023): 3300.

[19]. Abusitta, Adel, et al. "Deep learning-enabled anomaly detection for IoT systems." *Internet of Things* 21 (2023): 100656.

[20]. Jaramillo-Alcazar, Angel, Jaime Govea, and William Villegas-Ch. "Anomaly detection in a smart industrial machinery plant using IoT and machine learning." *Sensors* 23.19 (2023): 8286.

[21]. Abid, Noman. "Enhanced IoT network security with machine learning techniques for anomaly detection and classification." *Int. J. Curr. Eng. Technol* 13.6 (2023): 536-44.

[22]. Chevtchenko, Sérgio F., et al. "Anomaly detection in industrial machinery using IoT devices and machine learning: a systematic mapping." *IEEE Access* 11 (2023): 128288-128305.

[23]. Edozie, Enerst, et al. "Artificial intelligence advances in anomaly detection for telecom networks." *Artificial Intelligence Review* 58.4 (2025): 100.

[24]. Morshedi, Roya, and S. Mojtaba Matinkhah. "A comprehensive review of deep learning techniques for anomaly detection in iot networks: Methods, challenges, and datasets." *Engineering Reports* 7.9 (2025): e70415.

[25]. Aramide, Oluwatosin Oladayo. "Predictive Network Maintenance and Anomaly Detection with AI." *International Journal of Technology, Management and Humanities* 11.02 (2025): 1-11.

Received: August 04, 2025