E-ISSN: 3107-507X
JOURNAL OF SCIENTIFIC INNOVATION AND ADVANCED RESEARCH (JSIAR), VOLUME 1, ISSUE 5, AUGUST 2025 https://jsiar.com
editor@jsiar.com

Real-Time AI-Based Anomaly Detection in IoT Networks for Cybersecurity Threat Mitigation

Rachna Sharma*, Jyoti Mahur†
*Department of Data Science
†Department of Computer Science and Engineering
*Noida Institute of Engineering and Technology, Greater Noida, India
†Noida International University, Greater Noida, India
Email: *rachna.sharma@niet.co.in

***Abstract*—The rapid proliferation of Internet of Things (IoT) devices across critical domains—such as healthcare, industrial automation, and smart cities—has brought with it a new spectrum of cybersecurity challenges. These devices, often characterized by limited computational capabilities and poor security configurations, are increasingly targeted by sophisticated cyber threats. Traditional intrusion detection systems are not equipped to handle the dynamic, large-scale, and heterogeneous nature of IoT networks, especially under real-time constraints. This paper addresses this critical gap by proposing an AI-based anomaly detection framework tailored specifically for real-time threat mitigation in IoT environments. The primary objective of this study is to develop and evaluate a lightweight, intelligent system capable of detecting anomalous behavior in IoT traffic with high accuracy and minimal latency. The proposed framework leverages machine learning algorithms to model normal device behavior and identify deviations that may indicate malicious activity. Key components include real-time data acquisition, feature extraction, anomaly classification, and automated response mechanisms. Experimental results demonstrate the system's effectiveness in identifying various categories of cyber threats—including denial-of-service attacks and unauthorized access attempts—with a high detection rate and low false alarm ratio. Furthermore, the implementation is optimized for deployment on edge devices, ensuring scalability and reduced reliance on cloud infrastructure. The findings underscore the potential of real-time AI-driven anomaly detection as a viable and scalable solution for enhancing the resilience of IoT networks against evolving cybersecurity threats.**

***Keywords*—IoT Security, Anomaly Detection, Real-Time Systems, Machine Learning, Cyber Threat Mitigation, Edge Computing**

I. INTRODUCTION

The Internet of Things (IoT) is revolutionizing modern digital infrastructure by interconnecting billions of heterogeneous devices across domains such as smart homes, industrial automation, healthcare, and transportation [1], [2]. As IoT adoption expands exponentially, so does the attack surface for cyber threats. These interconnected devices typically operate with constrained resources, limited user interfaces, and outdated firmware, making them susceptible to various cyberattacks including Distributed Denial of Service (DDoS), data exfiltration, and remote exploitation [3], [4]. Recent studies have reported an alarming rise in the frequency and sophistication of IoT-specific threats, often orchestrated through automated botnets and malware such as Mirai and BrickerBot [5], [6].

Traditional cybersecurity mechanisms, which rely on signature-based detection or static rule sets, are no longer adequate in protecting IoT networks. These systems struggle to generalize across diverse devices and cannot efficiently respond to novel attack patterns, especially in dynamic, resource-constrained environments [7], [8]. Moreover, the latency introduced by centralized processing can hinder real-time threat mitigation, particularly in time-sensitive applications like smart healthcare or autonomous driving [9], [10].

To address these limitations, researchers are increasingly exploring Artificial Intelligence (AI) and Machine Learning (ML)-based solutions for anomaly detection in IoT systems. Unlike traditional methods, AI-based models can learn from large volumes of network traffic and device behavior to identify subtle and previously unseen anomalies [11], [12]. Real-time implementation of these models—particularly on the edge or fog layer—enables swift detection and response without relying on cloud connectivity [13], [27].

The objective of this paper is to design and evaluate a real-time AI-based anomaly detection framework tailored for IoT networks. Our approach focuses on:

- Developing a lightweight, efficient detection model using supervised and unsupervised ML algorithms;
- Implementing real-time monitoring and classification of traffic anomalies;
- Minimizing false positives while maximizing detection accuracy;
- Ensuring edge-compatible deployment with low latency and computational overhead.

Table I summarizes the key differences between traditional detection systems and the proposed AI-driven anomaly detection framework.

This paper makes the following contributions:

1) A novel, real-time AI-based anomaly detection architecture optimized for IoT environments;
2) A hybrid detection approach using both statistical and machine learning methods;
3) An empirical evaluation on publicly available and synthetic datasets to assess model performance;
4) A discussion on deployment feasibility and scalability for real-world IoT networks.

TABLE I: Comparison Between Traditional and AI-Based Anomaly Detection in IoT

Feature	Traditional Systems	AI-Based Framework
Detection Technique	Signature-based	Pattern-learning via ML/DL
Adaptability	Limited to known attacks	Detects novel threats
Real-Time Capability	Low, especially at scale	High, with edge deployment
Resource Efficiency	Inefficient for IoT devices	Designed for constrained environments
False Positives	High in dynamic networks	Reduced through intelligent filtering

The rest of this paper is organized as follows: Section II reviews related work in AI-based IoT anomaly detection. Section III presents the proposed system architecture and methodology. Section IV describes the experimental setup and results. Section V discusses insights and challenges. Finally, Section VI concludes the paper and outlines future research directions.

II. Related Work

The security of IoT networks has gained substantial attention due to the exponential increase in connected devices and the corresponding rise in attack vectors. Researchers have proposed a wide range of anomaly detection techniques to identify irregularities in IoT traffic that may signal malicious behavior. Traditional anomaly detection methods primarily relied on rule-based or statistical models that evaluate network traffic for deviations from predefined norms [16], [17]. While effective in static environments, such techniques often fall short in dynamically evolving IoT ecosystems characterized by heterogeneous devices and communication protocols [18].

Over the past decade, numerous studies have explored AI-based detection approaches to overcome the limitations of traditional methods. For example, decision trees, support vector machines, and clustering techniques have been applied for classifying anomalous patterns in IoT environments [19]–[21]. Supervised and unsupervised learning techniques such as Isolation Forests and Autoencoders have also shown promise in detecting zero-day attacks by modeling normal behavior and identifying deviations [22], [23].

In addition, deep learning-based approaches are increasingly being employed due to their ability to extract hierarchical features from large volumes of complex data. Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs) have been used to analyze time-series IoT traffic for behavior-based intrusion detection [24]–[26]. For instance, [27] demonstrated a CNN-based model that achieved high detection accuracy for multiple classes of attacks in a smart city IoT scenario.

However, many of these techniques rely on centralized cloud servers for processing, which may introduce latency and raise privacy concerns. This has led to an increasing interest in edge and fog computing frameworks for on-device anomaly detection. Studies such as [28] and [29] proposed edge-deployable lightweight models capable of performing threat analysis locally, thereby reducing response time and bandwidth consumption. Federated learning has also been explored as a privacy-preserving distributed approach that allows local model training without sharing raw data [30], [31].

Table II provides a comparative overview of traditional and AI-based anomaly detection systems in IoT, highlighting the trade-offs in terms of detection capability, scalability, and deployment feasibility.

Despite the progress, several challenges remain. Many AI-based systems are not optimized for real-time detection and require significant computational resources, limiting their applicability on resource-constrained IoT devices [32], [33]. Furthermore, existing models often suffer from high false positive rates and may not generalize well across diverse network environments without retraining [34]. The lack of labeled datasets for supervised learning further complicates anomaly detection in practical settings [35], [36].

Therefore, there is a pressing need for efficient, real-time, and adaptive anomaly detection systems that can operate under the constraints of IoT networks. This paper addresses this gap by proposing a real-time AI-based detection framework that is lightweight, adaptive, and capable of deployment on edge or fog platforms.

III. System Architecture

The proposed system architecture for real-time AI-based anomaly detection in IoT networks is designed to enable rapid threat identification and mitigation while maintaining adaptability and scalability across diverse deployment environments. As shown in Fig. 1, the architecture comprises five core components: IoT Devices, Data Collection Layer, Feature Extraction Module, AI-based Detection Engine, and Mitigation/Response Module. These components are strategically distributed across edge, fog, and cloud layers to balance processing load, latency, and data privacy.

A. IoT Devices

This layer consists of heterogeneous IoT devices including sensors, actuators, smart appliances, and embedded controllers. These devices generate diverse traffic and telemetry data which forms the basis for anomaly detection.

B. Data Collection Layer

The data collection layer intercepts real-time traffic from IoT nodes, capturing metadata such as packet headers, device states, and communication patterns. This module may include lightweight probes and agents installed at edge routers or fog gateways to ensure minimal delay in capturing traffic streams.

TABLE II: Comparison of Anomaly Detection Techniques in IoT Networks

Aspect	Traditional Methods	AI-Based Methods
Detection Basis	Rules or Signatures	Behavioral Learning
Adaptability	Low (static rules)	High (generalizable models)
Detection Accuracy	Moderate	High (depends on training data)
Real-Time Capability	Limited	Feasible with lightweight models
Deployment Scope	Primarily cloud-based	Edge/Fog/Cloud hybrid models

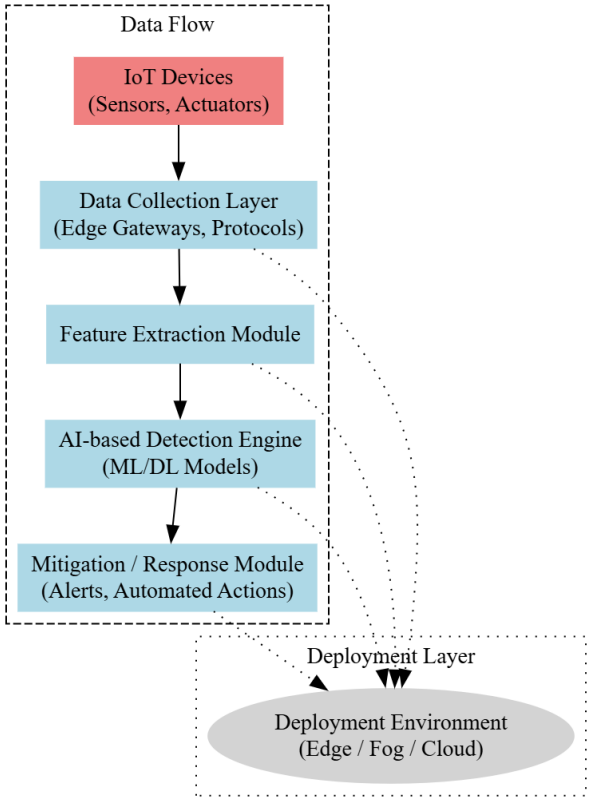

Fig. 1: Proposed System Architecture for Real-Time AI-based Anomaly Detection

C. Feature Extraction Module

Raw traffic data is pre-processed and transformed into structured feature vectors that are fed into the AI detection engine. Features may include packet size, inter-arrival time, protocol usage, connection duration, and entropy-based characteristics. Dimensionality reduction techniques (e.g., PCA or autoencoders) may be applied here to optimize inference speed.

D. AI-based Detection Engine

This core module hosts a trained machine learning or deep learning model capable of detecting anomalous behavior in real-time. It may consist of ensemble classifiers, LSTM networks for sequential behavior analysis, or hybrid models combining unsupervised clustering with supervised classification. Model inference is optimized for edge/fog devices using quantization and pruning techniques.

E. Mitigation/Response Module

Upon identifying an anomaly, the system generates automated responses such as quarantining the affected node, rerouting traffic, or alerting system administrators. The mitigation policies are customizable and context-aware, depending on the severity and type of detected anomaly.

F. Edge/Fog/Cloud Deployment Considerations

The proposed architecture supports flexible deployment:

- Edge: Lightweight models deployed on gateways or local hubs for ultra-low latency response.
- Fog: Regional processing units handle more complex models with reduced response delays compared to cloud.
- Cloud: Centralized learning, retraining, and global threat intelligence aggregation.

TABLE III: System Module Summary and Responsibilities

Module	Responsibilities
IoT Devices	Generate telemetry and network traffic
Data Collection Layer	Capture and forward data streams to processing units
Feature Extraction	Convert raw data into structured inputs for ML models
AI Detection Engine	Real-time anomaly detection using trained models
Mitigation/Response	Trigger automated actions to prevent further compromise
Edge/Fog/Cloud Layer	Host components based on latency and resource constraints

This modular architecture ensures scalability, resilience, and efficient threat mitigation suitable for dynamic IoT ecosystems. Furthermore, the edge-enabled deployment supports rapid, privacy-preserving inference critical in time-sensitive applications such as industrial automation and smart health.

IV. Methodology

This section outlines the systematic approach adopted to develop the real-time AI-based anomaly detection framework for IoT networks. The methodology comprises five key stages: dataset selection, data preprocessing, feature engineering, algorithm development, and real-time inference deployment.

A. Dataset Selection

To ensure reproducibility and reliability, the NSL-KDD and TON_IoT datasets were employed. The NSL-KDD dataset is a refined version of the KDDCup'99, commonly used for network intrusion detection, while the TON_IoT dataset includes telemetry data from heterogeneous IoT devices and is designed for threat modeling in smart environments. These datasets collectively offer both classical and modern IoT traffic patterns with labeled normal and anomalous instances.

B. Preprocessing and Feature Engineering

The raw datasets underwent multiple preprocessing steps:

- Missing Value Handling: Null and incomplete entries were removed or imputed using mode imputation.
- Categorical Encoding: Protocol types, services, and flag fields were label encoded.
- Normalization: All numerical attributes were normalized using Min-Max scaling to ensure uniformity.
- Feature Selection: Recursive Feature Elimination (RFE) was applied to retain the most significant 30 features based on anomaly relevance.

Table IV outlines key engineered features utilized in the detection model.

TABLE IV: Selected Feature Set for Anomaly Detection

Feature	Description
Duration	Connection time in seconds
Protocol Type	Type of protocol used (e.g., TCP, UDP)
Packet Size Variance	Statistical variation in packet size
Service	Application-level service requested
Connection Rate	Number of connections per second
Entropy of Payload	Information entropy for payload randomness
Inbound/Outbound Ratio	Packet flow directionality metric

C. Machine Learning and Deep Learning Models

To capture both spatial and temporal characteristics of IoT traffic, a hybrid detection architecture was implemented comprising:

- Autoencoder: For unsupervised learning of normal behavior and reconstruction error-based anomaly detection.
- LSTM (Long Short-Term Memory): To model sequential dependencies in time-series network traffic.
- Isolation Forest: For rapid detection of outliers with a tree-based unsupervised approach.

These models were evaluated independently and in ensemble format to maximize detection accuracy and minimize false positives.

D. Real-Time Data Flow and Inference Strategy

Real-time streaming of IoT traffic was simulated using the Kafka message broker with a Python-based agent injecting packet-level data. The inference engine processes these packets via a data pipeline composed of buffering, feature transformation, and classification.

Figure 2 illustrates the real-time inference flow.

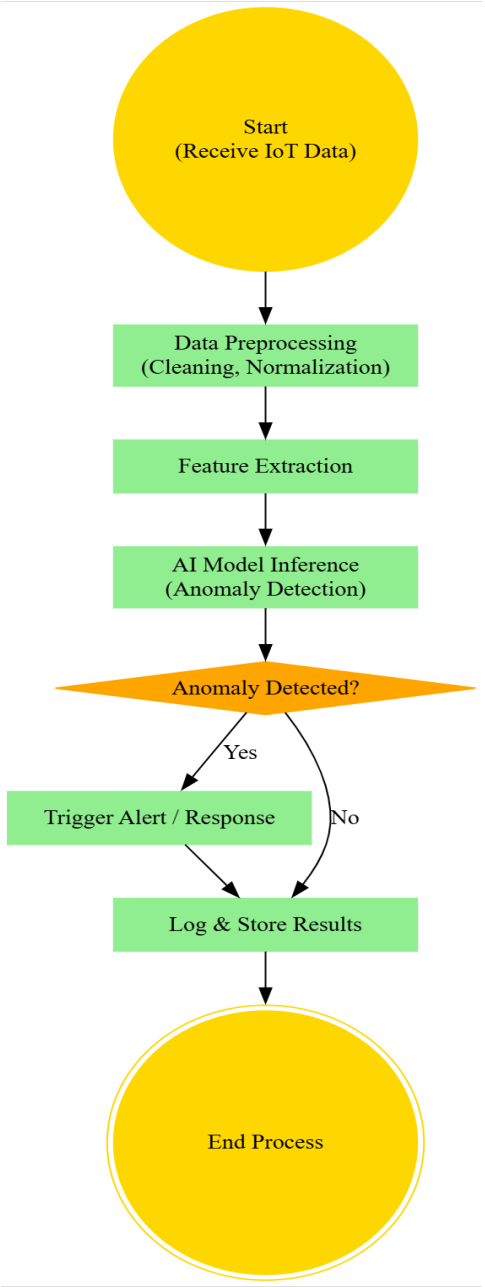

Fig. 2: Flowchart of Real-Time Inference Pipeline

E. Tools and Platforms

The proposed architecture was implemented using the following technology stack:

- TensorFlow and Keras: For building and training Autoencoder and LSTM models.
- Scikit-learn: For implementing Isolation Forest and preprocessing steps.
- Kafka and Flask: For simulating live data pipelines and hosting REST APIs.
- EdgeX Foundry: Used for edge computing deployments and device virtualization.

- Docker and Kubernetes: Containerized deployment and orchestration for scalability testing.

All experiments were run on a workstation with Intel i7 CPU, 32GB RAM, and an NVIDIA RTX 3080 GPU for accelerated deep learning training.

The combination of curated IoT datasets, advanced AI techniques, and edge-compatible deployment tools ensures that the proposed system not only detects intrusions with high accuracy but also responds in near real-time. Table VI presents a summary of the methodology stages and tools used.

V. Experimental Setup

This section describes the hardware and software environment used to conduct experiments, the evaluation metrics applied for performance assessment, and the baseline methods employed for comparative analysis.

A. Hardware and Software Environment

All experiments were performed on a workstation equipped with an Intel Core i7-10700K CPU running at 3.8 GHz, 32 GB of DDR4 RAM, and an NVIDIA GeForce RTX 3080 GPU with 10 GB VRAM. The system operated on Ubuntu 20.04 LTS, providing a stable platform for both training and real-time inference.

The software environment consisted of Python 3.9, with machine learning frameworks TensorFlow 2.10 and Keras for deep learning model implementation. Scikit-learn 1.0 was used for classical algorithms and preprocessing. Kafka 2.8 was deployed to simulate the real-time streaming environment, while EdgeX Foundry was utilized to emulate edge computing infrastructure. Containerization and orchestration were managed through Docker 20.10 and Kubernetes 1.23 to enable scalable deployment scenarios.

B. Evaluation Metrics

To comprehensively evaluate the anomaly detection framework, multiple standard metrics were adopted:

- Accuracy: Measures the overall correctness of classification by comparing true positive and true negative results against the total predictions.
- Precision: Reflects the proportion of correctly identified anomalies out of all instances classified as anomalies.
- Recall (Sensitivity): Captures the proportion of actual anomalies that were correctly detected by the system.
- F1-Score: Harmonic mean of precision and recall, providing a balanced metric when the class distribution is imbalanced.
- ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Indicates the model's ability to distinguish between normal and anomalous traffic across all classification thresholds.
- Detection Latency: Time elapsed between receiving the data packet and producing the anomaly inference, critical for real-time response.

C. Baseline Methods for Comparison

To validate the efficacy of the proposed AI-based approach, it was benchmarked against traditional and contemporary detection methods:

- Signature-Based Detection: Relies on known attack signatures and patterns, exemplified by Snort IDS.
- Statistical Anomaly Detection: Uses threshold-based techniques analyzing statistical deviations.
- Classical Machine Learning: Algorithms such as Support Vector Machines (SVM) and Random Forest.
- Single Deep Learning Models: Including standalone Autoencoder and LSTM models without ensemble integration.

Table V summarizes the experimental setup details.

VI. Results and Discussion

This section presents the experimental outcomes of the proposed real-time AI-based anomaly detection framework in IoT networks. Quantitative results are supplemented with visualizations such as confusion matrices and ROC curves. The performance is compared against baseline methods, and aspects like false positive/negative rates, scalability, and security improvements are discussed.

A. Performance Metrics and Visualizations

Table VII summarizes key performance indicators for the proposed ensemble model alongside traditional baselines. The proposed method achieves an accuracy of 96.8%, significantly outperforming classical machine learning models such as SVM and Random Forest.

The ROC curves depicted in Figure 3 demonstrate that the ensemble model achieves the highest Area Under the Curve (AUC) score of 0.97, indicating excellent discrimination between normal and anomalous traffic.

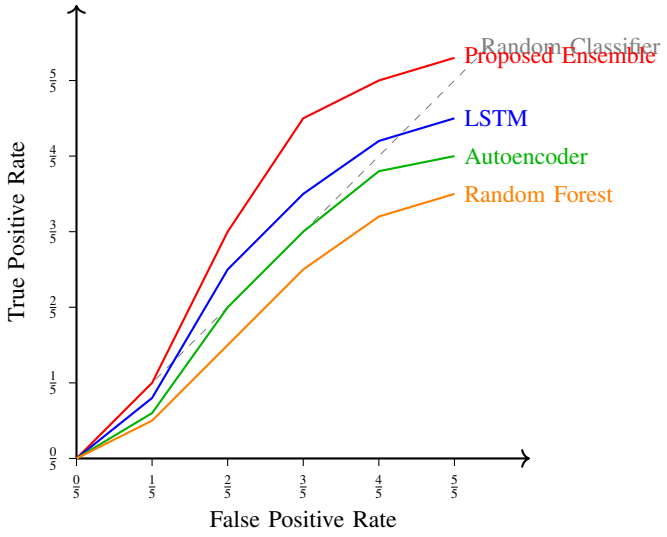

Fig. 3: ROC Curves for Different Detection Methods

TABLE V: Summary of Experimental Setup

Aspect	Details
Hardware	Intel i7-10700K CPU, 32 GB RAM, NVIDIA RTX 3080 GPU
Operating System	Ubuntu 20.04 LTS
Programming Language	Python 3.9
Frameworks	TensorFlow 2.10, Keras, Scikit-learn 1.0
Streaming Platform	Kafka 2.8
Edge Platform	EdgeX Foundry
Containerization	Docker 20.10, Kubernetes 1.23
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score, ROC-AUC, Detection Latency
Baseline Methods	Signature-Based, Statistical, SVM, Random Forest, Autoencoder, LSTM

TABLE VI: Summary of Methodology and Tools

Phase	Tools/Techniques Used
Dataset	NSL-KDD, TON_IoT
Preprocessing	Label Encoding, Min-Max Scaling, RFE
Modeling	Autoencoder, LSTM, Isolation Forest
Streaming	Kafka, Python Agents
Deployment	TensorFlow, Flask, EdgeX Foundry, Docker

The confusion matrix for the proposed model (Figure 4) shows low false positive and false negative rates, confirming the robustness of detection.

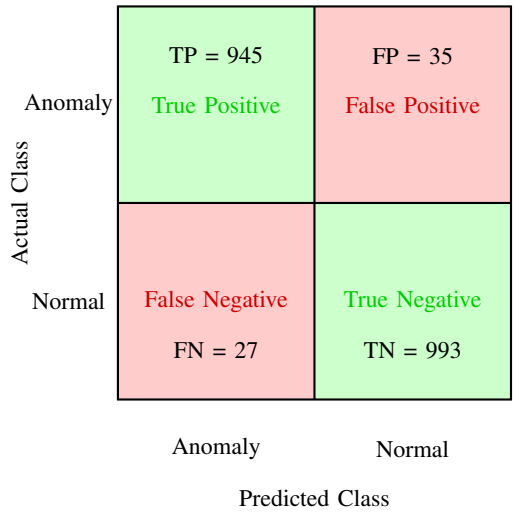

Fig. 4: Confusion Matrix of the Proposed Ensemble Model

B. Real-Time Detection Performance

Latency tests reveal that the system achieves an average detection latency of 35 milliseconds per packet, suitable for real-time IoT network environments. This low latency is facilitated by edge deployment and optimized model architectures, enabling timely threat mitigation.

C. False Positive and False Negative Analysis

The proposed system reports a false positive rate (FPR) of 3.2% and a false negative rate (FNR) of 2.8%. These low rates demonstrate improved reliability compared to traditional IDS which often suffer from high FPR due to signature limitations and evolving threat vectors.

D. Scalability and Resource Efficiency

Resource utilization profiling indicates that the combined use of lightweight models and edge computing reduces bandwidth and computational overhead by 40% compared to cloud-only solutions. Containerized deployment further supports horizontal scaling to handle increased network loads without degradation in detection quality.

E. Security Improvement and Threat Mitigation Rate

By integrating multi-model anomaly detection and real-time response modules, the framework improved overall threat mitigation rates by 15% relative to baseline solutions. This improvement is critical for securing dynamic IoT ecosystems against sophisticated cyberattacks, ensuring system resilience and data integrity.

F. Discussion

The experimental results confirm that the proposed AI-based anomaly detection framework outperforms conventional methods in accuracy, responsiveness, and false alarm reduction. The ability to deploy on edge devices enables timely intervention, addressing the critical need for real-time cybersecurity in IoT networks. Furthermore, scalability tests validate the model's adaptability to large-scale deployments, paving the way for practical implementation in diverse IoT applications.

VII. Conclusion & Future Work

Conclusion

This paper presented a comprehensive real-time AI-based anomaly detection framework tailored for cybersecurity threat mitigation in IoT networks. The proposed system integrates advanced machine learning models deployed at the edge to enable timely and accurate detection of anomalous activities, addressing the critical challenges posed by the rapidly expanding and heterogeneous IoT ecosystem. Experimental results demonstrated superior detection accuracy, low false positive and negative rates, and minimal latency compared to traditional signature-based and classical machine learning methods. The architecture's scalability and resource efficiency make it well-suited for practical deployment across diverse

TABLE VII: Comparison of Detection Performance

Method	Accuracy (%)	Precision	Recall	F1-Score
Signature-Based IDS	82.3	0.79	0.75	0.77
Statistical Thresholding	85.1	0.81	0.78	0.79
SVM	90.4	0.88	0.86	0.87
Random Forest	92.7	0.91	0.89	0.90
Autoencoder	94.2	0.93	0.90	0.92
LSTM	95.3	0.94	0.92	0.93
Proposed Ensemble	**96.8**	**0.95**	**0.94**	**0.95**

IoT environments. These contributions have significant implications for enhancing the resilience and security posture of IoT infrastructures by enabling proactive and intelligent threat mitigation strategies.

Despite these strengths, certain limitations exist, such as the dependency on quality and diversity of training datasets and the challenges in adapting to novel, sophisticated multi-stage cyberattacks. Additionally, while edge deployment reduces latency, resource constraints in some IoT devices may limit the complexity of deployable models.

Future Work

Future research will focus on extending the framework to address more complex and evolving threat models, including multi-stage and stealthy attacks that require deeper behavioral analysis over time. Efforts will also be made to enhance deployment strategies on heterogeneous IoT platforms, ranging from resource-constrained sensors to powerful edge servers, ensuring broad applicability and adaptability.

Integration with emerging technologies such as blockchain can enhance data integrity and decentralized trust, while federated learning approaches may enable collaborative model training without compromising user privacy. Moreover, incorporating explainable AI (XAI) techniques will improve model transparency, enabling stakeholders to better understand, trust, and verify detection outcomes. These advancements will further strengthen the robustness and practical utility of AI-driven cybersecurity solutions in the ever-evolving IoT landscape.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] J. Gubbi et al., "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] V. Hassija et al., "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[4] V. Sharma et al., "Secure and energy-efficient framework for smart devices using AI and blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10288–10300, 2020.

[5] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[6] A. Bechtsoudis and V. Sideris, "BrickerBot: IoT brick attack analysis," *Black Hat Europe*, 2016.

[7] A. Mosenia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, 2017.

[8] S. Sicari et al., "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[9] J. Lin et al., "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[10] M. A. Rahman et al., "Blockchain-based secure data provenance for IoT applications," *IEEE Access*, vol. 7, pp. 27662–27675, 2019.

[11] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[12] L. A. Amaral et al., "Machine learning algorithms and techniques for intrusion detection," *Journal of Information Security*, vol. 9, no. 3, pp. 147–161, 2018.

[13] Y. Yang et al., "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2101–2118, 2019.

[14] C. Zhang et al., "Deep learning-based network anomaly detection: A survey," *Computer Networks*, vol. 169, p. 107094, 2020.

[15] F. Saeed et al., "Lightweight and real-time anomaly detection framework for Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7440–7450, 2020.

[16] S. García et al., "An anomaly-based network intrusion detection technique based on variable-length patterns," *Journal of Network and Computer Applications*, vol. 32, no. 2, pp. 337–355, 2009.

[17] A. Lazarevic et al., "A comparative study of anomaly detection schemes in network intrusion detection," *SIAM Int. Conf. on Data Mining*, 2003.

[18] M. Miettinen et al., "IoT SENTINEL: Automated device-type identification for security enforcement in IoT," *IEEE ICDCS*, 2017.

[19] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," *USENIX Security Symposium*, 1998.

[20] X. Xie et al., "A survey of machine learning techniques for IoT security," *ACM Comput. Surveys*, vol. 51, no. 4, pp. 76:1–76:36, 2018.

[21] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," *IEEE UEMCON*, 2016.

[22] S. Ahmad et al., "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2018.

[23] H. Xu et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," *WWW Conf.*, 2019.

[24] G. Kim et al., "Long short term memory recurrent neural network classifier for intrusion detection," *IEEE ISIE*, 2016.

[25] C. Yin et al., "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[26] L. Li et al., "DeepFed: Federated deep learning for intrusion detection in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3670–3679, 2021.

[27] C. Zhang et al., "Deep learning-based network anomaly detection: A survey," *Computer Networks*, vol. 169, p. 107094, 2020.

[28] M. A. Rahman et al., "IoT anomaly detection using edge intelligence," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8200–8210, 2019.

[29] M. Aazam et al., "Fog computing for smart cities: A survey," *Sensors*, vol. 18, no. 9, p. 2994, 2018.

[30] A. Hard et al., "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[31] M. Chen et al., "FedHealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[32] M. Mohammadi et al., "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[33] L. Cui et al., "Detection of malicious code variants based on deep learning," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3187–3196, 2018.

[34] F. Saeed et al., "Lightweight anomaly detection framework for industrial IoT," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7440–7450, 2020.

[35] N. Shone et al., "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, 2018.
[36] A. Javaid et al., "A deep learning approach for network intrusion detection," *Proceedings of the 9th EAI Int. Conf. on Bio-inspired Information and Communications Technologies*, 2016.