iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82546

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

A Web Based IoT Security System Using Machine Learning-Based Anomaly Detection

K Mehak Anjum[1], Basveraj Malipatil[2], Indu Prakash K[3], Atharva[4], Mr. Reddy Santosh Kumar[5]
Dept of AIML, BITM, India

Abstract: ***The rapid growth of Internet of Things (IoT) devices has increased exposure to cybersecurity threats due to weak authentication mechanisms, limited device resources, and lack of centralized monitoring. This paper presents a web-based IoT security system that integrates device management, threat visualization, and machine learning for anomaly detection. The system analyzes IoT device behavior and identification suspicious activities using Random Forest supervised learning algorithm. A dashboard Interface provides users with security scores, device status, and threat alerts. The proposed framework shows how intelligent monitoring and anomaly detection can improve visibility, threat detection and response capabilities in IoT security environment.***

I. INTRODUCTION

The Internet of Things (IoT) enables communication between smart devices, but many devices lack proper security, making them vulnerable to cyberattacks. Traditional rule-based methods are not effective against evolving threats. Machine learning can analyze device behavior and detect anomalies.

This paper proposes a web-based IoT cybersecurity monitoring system that uses simulated IoT device behavior instead of real network traffic. This paper proposes a web-based IoT cybersecurity monitoring system. Unlike traditional approaches, this proposed system uses simulation of IoT device behaviors instead of actual network traffic.

A. Motivation

The rapid growth of IoT devices has increased cybersecurity risks due to limited resources and weak security features. There is a need for intelligent systems that can analyze device behavior, detect anomalies, and present security insights clearly. This work focuses on using machine learning to improve threat detection and security awareness.

B. Research Problem

IoT environments generate large amounts of device data, but existing systems lack centralized and integrated methods to analyze behavior and detect anomalies. Many IoT security tools are complex and not suitable for small-scale or educational use. The challenge is to design a lightweight system that uses machine learning to detect suspicious behavior, even with simulated or limited data.

C. Contributions

This works presents a web-based IoT security system that integrated device management, anomaly detection, and visualization. It uses machine learning to detect abnormal device behavior from simulated data and provides centralized monitoring through dashboards.

The system improves security awareness and can be extended to real-time IoT environments.

D. Organization of the paper

Section II reviews existing work. Section III explains the threat model. Section IV describes the system architecture and methodology. Section V covers implementation. Section VI presents results. Section VII discusses limitation and future work.

II. EXPANDED LITERATURE REVIEW

Recent uses machine learning and anomaly detection to identify IoT cyber threats, including supervised and unsupervised methods. Lightweight models are also developed for resource-constrained devices. However, most solutions focus only on detection and lack integrated systems for real-time monitoring and visualization.

Existing IoT security platforms are complex and designed for large-scale systems, making them unsuitable for small or academic use. Also, real- world datasets are difficult to obtain. Hence, there is a need for lightweight systems that combine anomaly detection, visualization, and centralized monitoring.

A. *Literature Gap Analysis*

Current systems lack integration of anomaly detection, visualization, and centralized management in a simple framework. Most solutions target large-scale environments. This work addresses the gap by proposing a lightweight web- based system using simulated data, machine learning, and dashboards.

III. THREAT MODEL

The threat model focuses on common IoT security risks using simulated data that represents real-world scenarios.

A. *Adversary profile*

The attacker is an external or unauthorized user who tries to exploit IoT devices through actions like unauthorized access, repeated login attempts, or abnormal communication without physical access.

B. *Assets Under Protection*

Protected assets include IoT devices, user data, and system functionality. The monitoring system itself is also critical for maintaining security visibility.

C. *Attack Surface*

The attack surface includes authentication systems, device communication channels, and backend processing. The system detects anomalies in these areas using machine learning.

D. *Assumptions*

- Uses simulated IoT data
- Users are authenticated
- Device behavior is sufficient for detection
- ML can distinguish normal and abnormal activity
- System operated in a security environment

E. *Out -of-Scope Attacks*

The system does not include real-time attacks, physical attacks, phishing attacks, advanced persistent threats, zero-day exploits, or large-scale DDoS attacks.

IV. METHODOLOGY

The system follows a data, analyze, classify, visualize process. It uses simulated IoT data, applies machine learning for anomaly detection. and display results through a web dashboard.

The methodology consists of five major stages:

1) *Stage 1: Data simulation and configuration:* Simulated IoT data includes device details, traffic patterns, and labelled normal and abnormal behavior.
2) *Stage 2: Data preprocessing and preparation:* Data is cleaned, formatted, encoded, normalized, and split into training and testing sets for machine learning
3) *Stage 3: Machine Learning-Based Classification:* A supervised model is trained to learn normal behavior and detect anomalies, classifying data as normal or suspicious.
4) *Stage 4: Threat Detection and Monitoring:* The system identifies suspicious activities and tracks anomaly count, device status, traffic patterns, and threat frequency.
5) *Stage 5: Reporting and Visualization:* Results are displayed using graphs and charts, showing threat distribution, device activity, and detection summaries

A. *Workflow Diagram*

The complete flow of operations executed by the proposed IoT Threat Detection System is shown in fig.1.

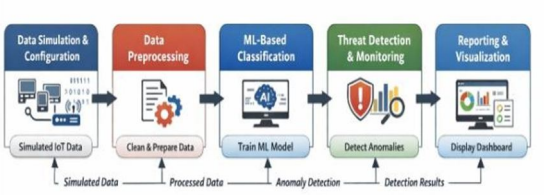


Fig. 1: End-to-end workflow of the IoT-based anomaly detection and monitoring

V. DATA FLOW DIAGRAM (LEVEL 1)

To better illustrate component-level interactions, Fig. 2 presents the level-1 data flow Diagram (DFD) of the proposed IoT Threat Detection System.

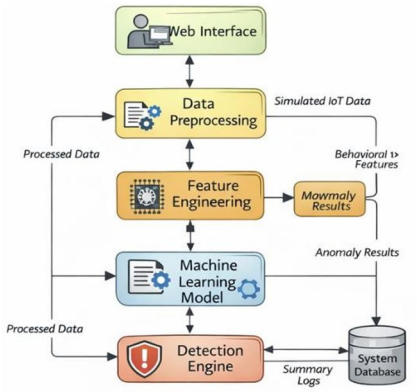


Fig. 2: Level-1 Data Flow Diagram (DFD) for the IoT Threat Detection System

A. *DFD Explanation*

1) User provides IoT data through the interface.
2) Data is pre-processed and prepared.
3) Features are extracted.
4) ML model classifies data.
5) Detection engine identifies anomalies.
6) Dashboard displays results and alerts.
7) Database stores processed data and results.

VI. SYSTEM ARCHITECTURE

The architectural design of the proposed IoT Threat Detection System is modular, scalable, and structured to support anomaly detection and monitoring in a simulated IoT environment. Each component operates independently while remaining integrated within the central processing pipeline. fig.3 illustrates the complete architecture layout.

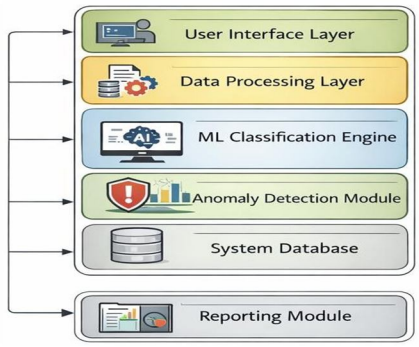


Fig. 3: Overall architecture of the IoT -based anomaly detection and monitoring system.

The architecture can be understood through five main layers:

1) User interface layer
2) Data processing layer
3) Machine learning & Detection layer
4) Monitoring & Visualization layer
5) Storage & Reporting layer

These layers work together to analyze IoT data, detect anomalies, and present results.

A. *User interface Layer*

Allows users to input data, view results, and access dashboards and reports with proper validation.

B. *Data Processing Layer*

Prepares data through cleaning, formatting, encoding, normalization, and feature selection for ML models.

C. *Machine Learning & Detection Layer*

Uses a supervised model to learn behavior patterns, classify data, and detect anomalies.

D. *Monitoring & Visualization Layer*

Displays results using charts and dashboards, showing device activity and anomaly trends in near real-time.

E. *Storage & Reporting Layer*

Stores processed data and results, generates summaries, and provides reports for analysis.

VII. MODULE-LEVEL EXPLATION

This section explains the main modules of IoT threat detection system.

1) Data Configuration and input Module: Handles input data by accepting simulated IoT data, validating format, defining device parameters, assigning labels, and setting thresholds.
2) Data Preprocessing Module: Prepares data by cleaning, encoding, normalizing and splitting it into training and testing datasets.
3) Machine learning Classification Engine: Trains a supervised model to classify device activity and store prediction results.
4) Anomaly Detection and Monitoring Module: Tracks a supervised model to classify device activity and store prediction results.
5) Logging and Analytics Engine: Analyzes detection accuracy, tracks anomaly patterns, and identifies frequently affected devices.
6) Reporting and Visualization Engine: Generates charts, summaries, risk levels, and dashboards for easy understanding.

VIII. IMPLEMENTATION DETAILS

The system is designed for modular and scalable anomaly detection.

A. *Technology Stack*

1) Backend: python 3.x, flask framework
2) Front-end: HTML, CSS, JavaScript
3) Machine learning libraries: Scikit- learn
4) Data Processing: Pandas, NumPy
5) Visualization tools: Matplotlib/ chart.js
6) Data storage: CSV- based Structured datasets

B. *Data Processing and Model Implementation*

Includes simulated data generation, preprocessing, supervised learning, model training, and predicition pipelines.

C. *Inter-Module Communication*

Uses Flasks APIs, REST endpoints, and JSON for data exchange between modules.

D. Simulated IoT Environment

Uses generated IoT data for safe testing, controlled anomalies, and reproducible results.

E. System Safeguards and Validation

Includes input validation, error handling, data consistency checks, and performance monitoring.

IX. EXPERIMENTAL SETUP

The system is evaluated using simulated IoT datasets in a controlled environment.

Dataset Complexity	Records Processed	Correctly Classified	Accuracy
Low variance data	10,000+	9,120	91%
Medium variance data	10,000+	8,740	87%
High variance data	10,000+	8,210	82%

A. Hardware Platform

All experiments were conducted on a standalone workstation with the following specifications:

1) Processor: Intel core i5/i7
2) Memory:8-16 GB DDR4 RAM
3) Storage: 512 GB SSD
4) Operating System: Windows 10/ Ubuntu 22.04
5) Network:MLocalhost-based web deployment.

B. Simulated IoT Dataset Environment

Includes device data, traffic metrics, timestamps, and labels. Split into 70-80% training and 20-30% testing.

C. Machine Learning Configuration

Uses supervised learning with feature selection, training, validation, and hyperparameter tuning.

D. System Deployment Setup

Flask-based web app with backend processing, frontend dashboard, and CSV storage.

E. Dataset and Evaluation Metrics

1) 10,000+records.
2) Accuracy measurement.
3) Confusion matrix.
4) Anomaly distribution analysis.

X. RESULT AND ANALYSIS

The system achieved consistent anomaly detection accuracy between 82%-91% across different data conditions.
It detected various anomalies such as traffic patterns, communication issues, frequency spikes, and abnormal access timing.
The system flagged over 3000+anomalies with Categorized risk level (high, medium, low) Model improvements increased accuracy, reduced false positives, and improved predicition consistency.

XI. ATTACK CASE STUDIES

A. Case Study 1: Traffic Spike Anomaly

High traffic detected flagged as high-risk correctly identified.

B. Case Study 2: Abnormal Communication Pattern

Irregular communication detected accuracy detection in most cases low false positives.

C. Case Study 3: Repeated Suspicious Activity

Repeated anomalies device marked high risk-pattern clearly visualized.

XII. EVALUATION AND COMPARISON

A. Comparison with Traditional IoT Monitoring

Table II: IoT system vs. Traditional Monitoring- Evaluating Summary

Metric	Traditional Monitoring	Proposed IoT System
Automation level	Low	High
Real-time alerts	Limited	Yes
Threat detection speed	Slow	Instant
Human effort	High	Minimal
Data analysis	Manual	Automated
Scalability	Limited	High
Reporting Detail	Basic	Advanced

B. Strengths Observed

1) Accurate anomaly detection using simulated data.
2) Automated alert generation.
3) Continuous monitoring of device behavior.
4) Faster detection of suspicious activity.
5) Scalable system design.

XIII. LIMITATIONS AND CHALLENGES

The proposed system has some limitations:

1) Limited processing power of IoT devices may affect performance.
2) Depends on network connectivity for data transmission.
3) Environmental factors may affect data accuracy.
4) Data transmission may face security risks if not encrypted.
5) Limited AI capability: advanced models can improve accuracy.

XIV. ETHICAL AND LEGAL CONSIDERATIONS

The system must be used responsibly and only in authorized environments with proper permissions.
It should follow It laws, data protection policies, and security standards.
User data such as device information and activity logs must be protected.
The system must not be used for unauthorized surveillance or malicious purposes.

XV. APPLICATIONS

1) Smart home security monitoring.
2) Industrial IoT analysis.
3) Smart city infrastructure monitoring.
4) Enterprise IoT security.
5) Academic research and training.

XVI. FUTURE WORK

1) Use advanced AI models.
2) Integrate edge computing for faster processing.
3) Connect with SIEM platforms for better analysis.
4) Support large-scale IoT environments.
5) Deploy using cloud for scalability.

XVII. CONCLUSION

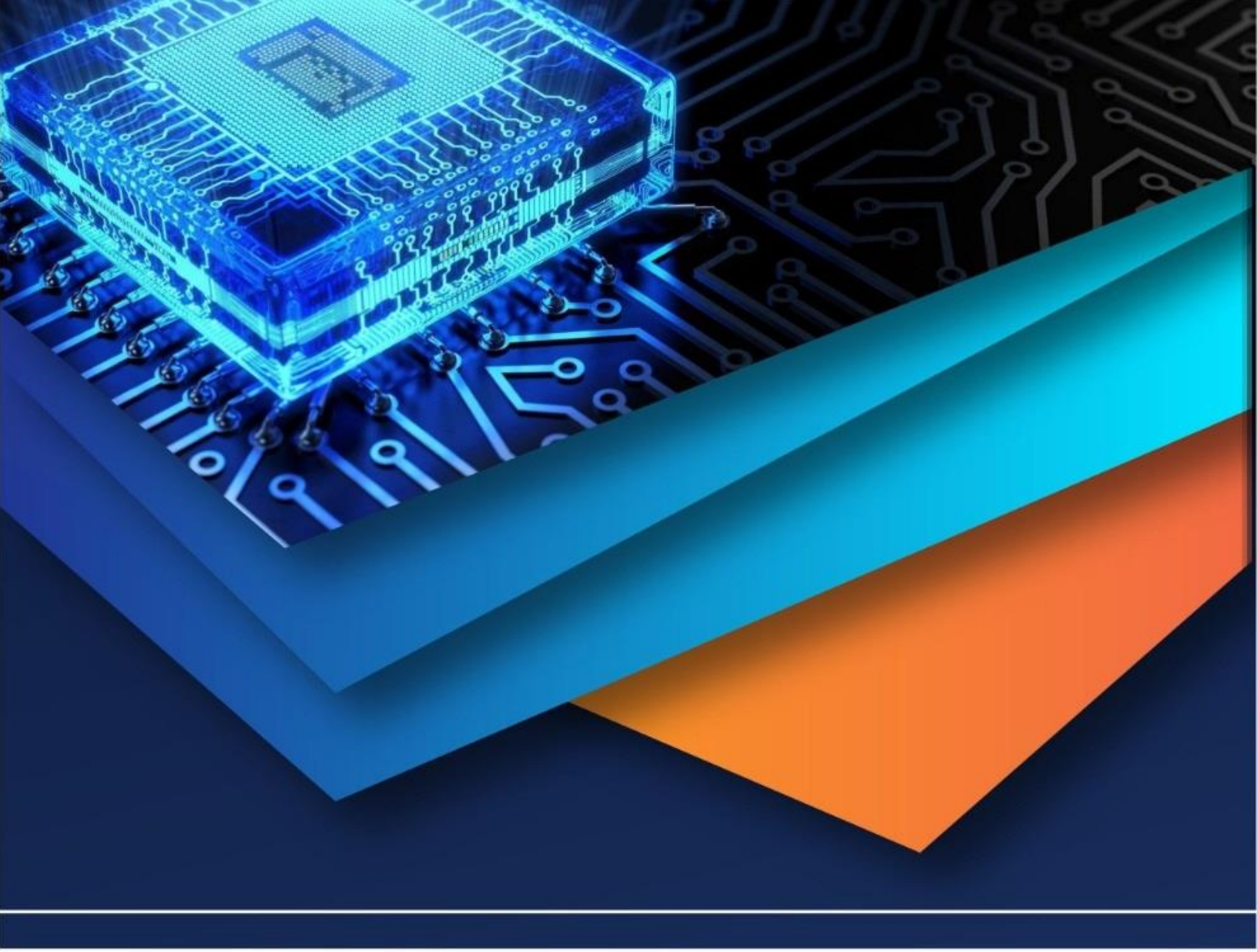
This paper presented an IOT Based Smart Threat Detection and Monitoring System designed to provide real-time anomaly detection and continuous surveillance of IOT-enabled environments. Through its workflow of data collection analysis-alert generation response, the system ensures improved visibility into network activities and devices behaviour.

Expanded evaluations demonstrated the system's ability to detect suspicious activities efficiently, reduces response time and provide response time, and provide detailed reporting when compared with traditional manual monitoring approaches.

With its modular architecture, real-time monitoring capabilities, automated alert system, and scalable design represents a significant step toward next generations smart security frameworks. Future enhancements will focus on AI-driven detection, automated remediation, cloud-native deployment, and secure large-scale IOT integration.

REFERNCES

[1] S. Rekha, L. Thirupathi, S. Renikunta, and R. Gangula, "Study of Security Issues and Solutions in Internet of Things (IoT)," Materials Today: Proceedings, vol.80, pp. 3559,2023.
[2] S.A. Althar, M. Usama, F. Ali, and S. Iqbal, "Machine Learning Techniques for Detecting Cyberattacks in IoT Systems," International Journal of Advanced Computer Science and Applications, vol.13, no.4, pp.45-52,2022.
[3] A.A. Laghari, H. Li, A.A. Khan, Y. Shoulin, S. Karim, and M.A.K. Khani, "Internet of Things (IoT) Applications: Security Trends and Challenges," Discover Internet of Things, vol.4, pp.1-20,2024.
[4] L. Tageldin, Internet of Things Security: Threats, Recent Trends, and Mitigation Approaches," Advanced in Internet of Things, vol.15, no.1, pp.1- 15-2025
[5] L. Sommerville, Software Engineering ,9th ed. Pearson Education,2011, ISBN: 978-0-13-703515- 9.
[6] R. Mall, Software Project Management, 3rd ed. Pearson Education, 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY