International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

AI-Driven Network Intrusion Detection and Traffic Optimization in IoT Networks

Othman Melad Salim Abdulkarim[1], Musbah Belaid Musbah Haduad[2]

[1]Assistant Lecture Higher Institute of Science and Technology - Tamzawa Alshati- libya
[2]Assistant Lecture computer technology Higher Institute of Science and Technology -Gasr khiar- libya

Abstract
The rapid expansion of the Internet of Things (IoT) has resulted in highly interconnected networks containing large numbers of heterogeneous and resource-constrained devices. Although IoT networks provide significant benefits in smart homes, healthcare, transportation, industrial automation, and smart cities, their large attack surface makes them vulnerable to Distributed Denial-of-Service (DDoS), Denial-of-Service (DoS), reconnaissance, spoofing, brute-force, web-based, and botnet attacks. Conventional security mechanisms may be insufficient for detecting complex and evolving network threats while maintaining efficient network performance. This paper proposes an AI-driven framework that integrates machine-learning-based intrusion detection with intelligent traffic optimization for IoT networks. The proposed framework first preprocesses network traffic and extracts relevant flow-level features. Several machine-learning algorithms, including Decision Tree, Support Vector Machine, K-Nearest Neighbors, Random Forest, and Long Short-Term Memory, are considered for identifying malicious traffic. Random Forest is selected as the primary lightweight detection model because of its strong classification capability and relatively low computational complexity. The intrusion-detection component is then integrated with a traffic-optimization mechanism that dynamically identifies congested or compromised network paths and redirects legitimate traffic toward more suitable routes. The CICIoT2023 dataset is selected for intrusion-detection evaluation because it represents a large-scale IoT environment with 105 devices and 33 attacks. Network-level performance is evaluated through simulation using metrics including packet delivery ratio, end-to-end delay, throughput, packet loss, and network overhead. The proposed framework aims to improve both security and network efficiency by combining attack detection with adaptive traffic management. The study provides a practical architecture for intelligent IoT network management and identifies future opportunities for lightweight, explainable, and reinforcement-learning-based IoT security.

Keywords: Internet of Things, Intrusion Detection System, Machine Learning, Artificial Intelligence, Network Security, Traffic Optimization, Random Forest, LSTM, IoT Networks.

1. INTRODUCTION

The Internet of Things has become an important networking paradigm in which physical devices, sensors, actuators, gateways, and computing systems communicate through wired and wireless networks [1]. IoT technologies are increasingly used in healthcare, transportation, industrial automation, agriculture, smart cities, smart homes, and environmental monitoring [2]. The growing number of interconnected devices has created significant opportunities for intelligent and automated services. At the same time, the increased connectivity introduces substantial cybersecurity and network-management challenges [3].

Unlike conventional computer networks, IoT environments generally contain heterogeneous devices with different communication protocols, computational capabilities, storage capacities, and energy constraints [4]. Many IoT devices operate with limited processing resources and may not support sophisticated security mechanisms. Furthermore, large numbers of devices continuously generate network traffic, making it difficult to manually monitor network behavior and identify malicious activities [5].

IoT networks are exposed to several types of attacks, including DDoS, DoS, reconnaissance, spoofing, brute-force attacks, web-based attacks, and botnet activity. The CICIoT2023 dataset was developed specifically to address limitations of earlier IoT security datasets. It was collected from a topology containing 105 IoT devices and includes 33 attacks organized into seven major categories: DDoS, DoS, reconnaissance, web-based attacks, brute force, spoofing, and Mirai [6].

Traditional intrusion detection systems typically depend on predefined signatures or manually constructed rules. Signature-based systems can perform well against known attacks but may struggle to identify previously unseen or modified attack patterns. Machine learning provides an alternative because models can learn relationships between network characteristics and attack behavior from historical traffic. Recent research has extensively investigated supervised, unsupervised, ensemble, and deep-learning methods for IoT intrusion detection [7].

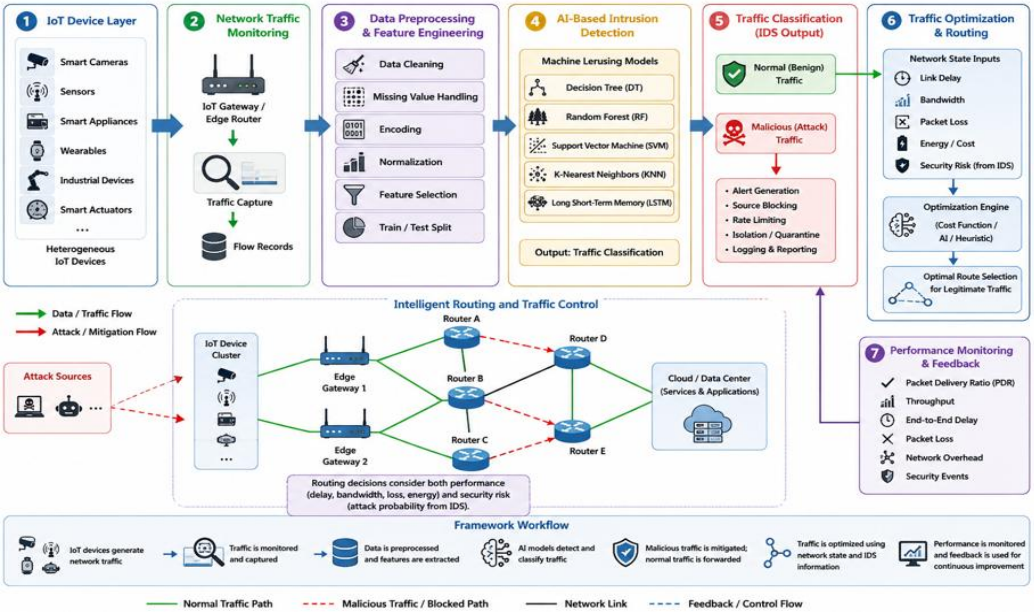

Figure 1: Proposed AI-driven IoT security and traffic-optimization framework.

However, security alone is not sufficient for IoT network management. Detecting malicious traffic without considering its effect on network performance can result in congestion, packet loss, increased delay, and inefficient utilization of available network resources. Therefore, IoT networks require mechanisms capable of both identifying malicious behavior and maintaining efficient traffic delivery [8].

This research addresses this problem by proposing an AI-driven framework that combines intrusion detection and traffic optimization. The framework uses machine learning to classify network traffic and provides information to a traffic-management component that can identify congested or compromised paths and redirect legitimate traffic [9].

1.1 Problem Statement

IoT networks face two interconnected challenges. First, the increasing volume and diversity of network attacks make conventional intrusion-detection techniques less effective. Second, network congestion and malicious traffic can negatively affect throughput, packet delivery, latency, and network resource utilization.

Many existing studies focus primarily on intrusion-detection accuracy without sufficiently considering the resulting network-performance implications. Conversely, traffic-optimization approaches may optimize network performance without integrating cybersecurity information into routing decisions.

Therefore, there is a need for an integrated framework capable of using AI to detect malicious IoT traffic while simultaneously supporting intelligent traffic optimization.

1.2 Research Aim

The main aim of this research is to develop an AI-driven framework that improves IoT network security and communication efficiency through integrated intrusion detection and traffic optimization.

1.3 Research Objectives

The objectives are:

1. To investigate security and traffic-management challenges in IoT networks.
2. To develop a machine-learning-based intrusion detection component for classifying IoT network traffic.
3. To compare different machine-learning algorithms for IoT attack detection.
4. To identify malicious and abnormal traffic that may affect network performance.
5. To develop a traffic-optimization mechanism that considers detected attacks and network conditions.
6. To evaluate the proposed framework using security and network-performance metrics.
7. To compare the proposed approach with conventional network-management and intrusion-detection approaches.

1.4 Research Questions

The study addresses the following questions:

RQ1: Which machine-learning algorithm provides the most effective intrusion detection performance for IoT network traffic?

RQ2: Can AI-based intrusion detection reduce the impact of malicious traffic on IoT network performance?

RQ3: Can intelligent traffic optimization improve packet delivery, throughput, and delay compared with conventional routing?

RQ4: What trade-offs exist between intrusion-detection accuracy and computational/network overhead?

1.5 Contributions

The main contributions of this research are:

- An integrated AI-driven architecture for IoT intrusion detection and traffic optimization.
- A comparative evaluation of several machine-learning algorithms.
- Integration of security information into traffic-management decisions.
- Evaluation using both cybersecurity and networking metrics.
- A practical experimental methodology using CICIoT2023 and network simulation.
- Identification of future opportunities for lightweight and reinforcement-learning-based optimization.

2. Related Work

2.1 IoT Network Security

The heterogeneous and distributed nature of IoT networks introduces several security vulnerabilities. IoT devices can be targeted individually or used collectively to attack other devices and network infrastructure. Botnets such as Mirai demonstrate how compromised IoT devices can be coordinated to generate large-scale malicious traffic [11].

The development of realistic IoT datasets is therefore important for evaluating intrusion-detection systems. [12] was designed to provide a realistic benchmark based on attacks executed against IoT devices within an extensive topology. The dataset contains 105 devices and 33 attacks covering DDoS, DoS, reconnaissance, web-based, brute-force, spoofing, and Mirai attack categories.

2.2 Machine Learning for IoT Intrusion Detection

Machine learning has become a major approach for IoT intrusion detection. Supervised algorithms learn from labeled network traffic, while unsupervised methods attempt to identify abnormal patterns without requiring complete labeling [13].

Decision Trees are attractive because their decision-making process is relatively easy to interpret. Support Vector Machines can provide effective classification in high-dimensional feature spaces but may become computationally expensive with large datasets. K-Nearest Neighbors is conceptually simple but can require significant computational resources during prediction.

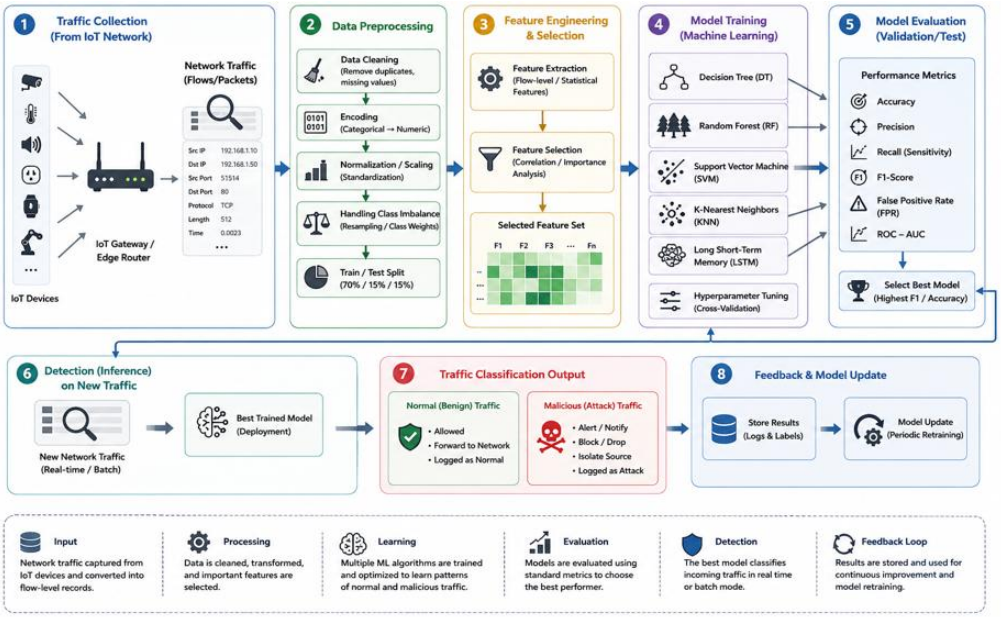

Figure 2: Machine-learning intrusion-detection workflow.

Random Forest combines multiple decision trees and generally provides strong performance and robustness. It is particularly attractive for IoT IDS research because it can handle nonlinear relationships and mixed feature distributions.

Deep-learning approaches such as LSTM networks can model sequential dependencies in network traffic. Recent reviews indicate that machine learning and deep learning remain active research directions for IoT intrusion detection, although computational complexity, false positives, dataset quality, and real-world deployment remain important challenges.

2.3 Traffic Optimization in IoT Networks

Network traffic optimization attempts to use available network resources efficiently while maintaining acceptable quality of service. Important performance indicators include throughput, delay, packet loss, packet delivery ratio, energy consumption, and network lifetime.

Traditional routing protocols generally make decisions according to predefined metrics. However, network conditions can change dynamically because of traffic bursts, node failures, congestion, or attacks. AI can provide adaptive decision-making by learning relationships between network conditions and routing performance [14].

Reinforcement learning is particularly promising because an agent can learn routing decisions based on rewards associated with successful packet delivery, low latency, and efficient resource utilization.

2.4 Research Gap

Recent surveys show substantial progress in ML-based IoT intrusion detection, but important challenges remain. These include computational complexity, high false-positive rates, limited real-world deployment, outdated datasets, explainability, and the need for efficient models suitable for resource-constrained devices.

A significant research opportunity is therefore to integrate security detection with network-performance optimization rather than treating the two problems independently.

3. Proposed Framework

3.1 Overall Architecture

The proposed architecture consists of six major components:

1. IoT devices
2. Network traffic monitoring
3. Data preprocessing and feature extraction
4. AI-based intrusion detection
5. Traffic optimization
6. Performance monitoring

The conceptual workflow is:

IoT Devices → Network Traffic → Monitoring → Preprocessing → AI-Based IDS → Traffic Classification → Traffic Optimization → Network Performance Monitoring

Normal traffic is forwarded through appropriate routes, while malicious traffic can be blocked, isolated, rate-limited, or excluded from routing decisions.

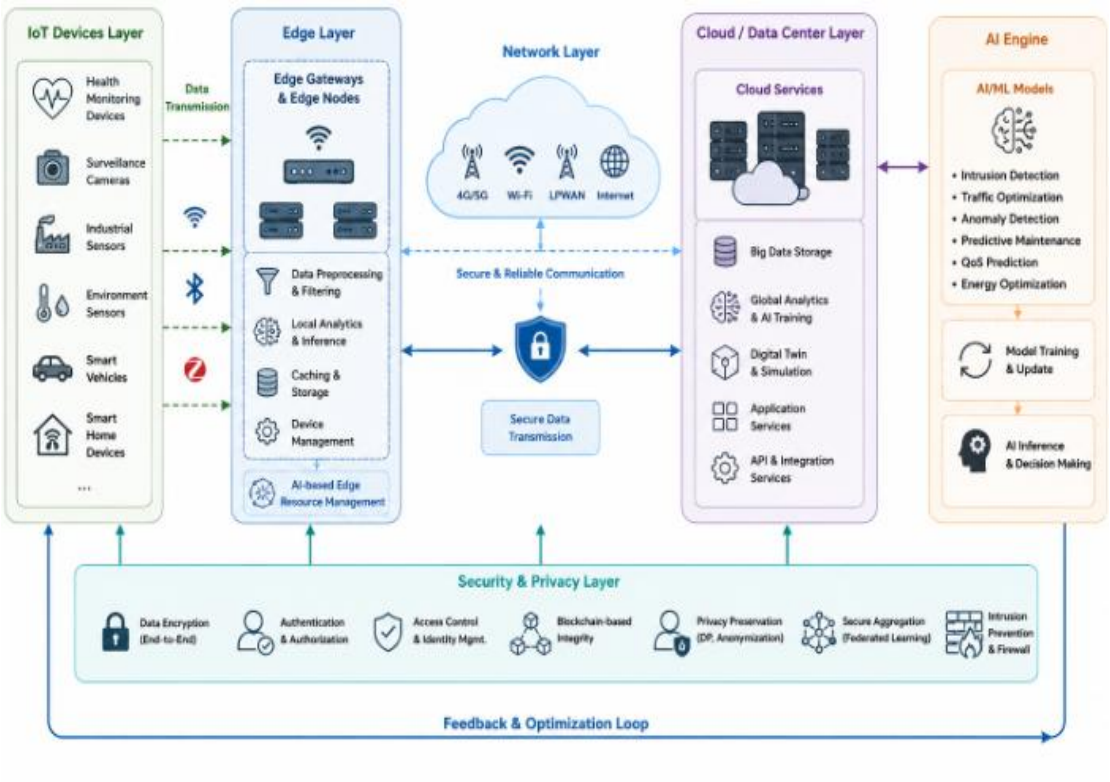

Figure 3: Proposed AI-Driven IoT Network Framework

3.2 IoT Network Layer

The first layer contains heterogeneous IoT devices such as:

- Sensors
- Smart cameras
- Smart appliances
- Wearable devices
- Industrial sensors
- Raspberry Pi devices
- Smart gateways

These devices generate network traffic that is transmitted through gateways and intermediate network nodes.

3.3 Traffic Monitoring

A monitoring component captures network-flow information. Relevant features may include:

- Source IP
- Destination IP
- Source port
- Destination port
- Protocol
- Packet count
- Byte count
- Flow duration
- Packet rate

- Packet length
- Inter-arrival time
- TCP flags
- Forward/backward packet statistics

The exact feature set should correspond to the selected dataset.

3.4 Data Preprocessing

Network datasets frequently contain missing values, duplicate records, categorical attributes, and highly imbalanced classes.

The preprocessing procedure consists of:

1. Removing duplicate records.
2. Handling missing or invalid values.
3. Encoding categorical features.
4. Removing irrelevant identifiers where appropriate.
5. Scaling numerical features for algorithms that require normalization.
6. Selecting informative features.
7. Splitting data into training and testing subsets.

For a binary classification experiment:

$$y = \begin{cases} 0, & \text{Benign traffic} \\ 1, & \text{Malicious traffic} \end{cases}$$

For multiclass classification, each attack category is assigned a separate class.

3.5 Feature Selection

Feature selection is important because unnecessary features increase computational complexity and may reduce model generalization.

A correlation-based approach can initially identify highly correlated features. Feature importance from Random Forest can subsequently be used to identify influential variables.

The importance of feature $j$can be represented as:

$$I_j = \frac{1}{T}\sum_{t=1}^{T} I_{j,t}$$

where $T$is the number of decision trees and $I_{j,t}$represents the contribution of feature $j$in tree t.

3.6 Machine Learning Models

Five models are proposed for comparison.

Decision Tree

A Decision Tree recursively divides the feature space according to classification rules.

Advantages include:

- Simple implementation
- Interpretability
- Low prediction latency

Random Forest

Random Forest combines multiple decision trees:

$$\hat{y} = \text{mode}\{h_1, (x), h_2, (x) \ldots h_T(x)\}$$

where $h_t(x)$represents the prediction of tree t.

Random Forest is selected as the main candidate because ensemble learning can improve robustness and reduce overfitting.

Support Vector Machine

SVM attempts to identify an optimal separating hyperplane:

$$w^T x + b = 0$$

The classifier seeks to maximize the margin between classes.

K-Nearest Neighbors

KNN classifies an instance according to the labels of nearby training samples. Euclidean distance can be expressed as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Long Short-Term Memory

LSTM is considered because network traffic can contain temporal dependencies. The LSTM architecture can learn relationships between successive traffic observations and may therefore identify sequential attack behavior.

4. Traffic Optimization Mechanism

4.1 Motivation

Intrusion detection identifies malicious traffic, but detection alone does not guarantee network performance. A DDoS attack, for example, may consume bandwidth and increase congestion even after the attack has been identified.

Therefore, the proposed system uses IDS information to support traffic-management decisions.

4.2 Network State

The traffic-optimization component monitors:

$$S_t = [D_t, B_t, L_t, P_t, A_t]$$

where:

- D_t= average network delay
- B_t= available bandwidth
- L_t= packet-loss rate
- P_t= packet-delivery performance
- A_t= detected attack level

The network state is periodically updated.

4.3 Routing Decision

For each candidate path p, a routing cost can be defined as:

$$C(p) = w_1 D(p) + w_2 L(p) + w_3 E(p) + w_4 A(p)$$

where:

- $D(p)$= path delay
- $L(p)$= packet loss
- $E(p)$= energy/resource cost
- $A(p)$= security risk
- w_1, w_2, w_3, w_4= weighting parameters.

The selected path is:

$$p^* = \arg\min_{p \in P} C(p)$$

Thus, a path with high congestion or a high security risk receives a higher cost.

4.4 Security-Aware Traffic Management

When the IDS detects malicious traffic, the traffic-management component can apply one or more actions:

- Block the source.
- Isolate the compromised device.
- Rate-limit suspicious traffic.
- Reduce the priority of suspicious flows.
- Redirect legitimate traffic.
- Select an alternative route.
- Generate a security alert.

The framework therefore connects cybersecurity information with network-management decisions.

5. Experimental Methodology

5.1 Dataset

The primary dataset selected for this study is **CICIoT2023**.

CICIoT2023 was developed using an IoT topology containing 105 devices. It includes 33 attacks organized into seven categories:

Table 1. Examples of Attack Categories

Attack Category	Examples
DDoS	Flooding and distributed attacks
DoS	TCP/UDP/HTTP flooding
Reconnaissance	Port scanning, OS scanning
Web-based	SQL injection, XSS, command injection
Brute Force	Credential-based attacks
Spoofing	Network identity spoofing
Mirai	IoT botnet attacks

The dataset is publicly provided by the Canadian Institute for Cybersecurity and is intended for IoT security analytics and intrusion-detection research.

5.2 Dataset Preparation

Because CICIoT2023 is large, experiments can initially use a representative subset for model development and subsequently expand to the complete dataset.

Two classification experiments are recommended:

Experiment A: Binary classification

- Benign
- Attack

Experiment B: Multiclass classification

- Benign
- DDoS

- DoS
- Reconnaissance
- Web-based
- Brute Force
- Spoofing
- Mirai

The binary experiment measures the ability of the proposed system to detect attacks, while the multiclass experiment evaluates its ability to identify attack types.

5.3 Training and Testing

The dataset should be divided into:

- 70% training
- 15% validation
- 15% testing

A stratified splitting strategy should be used to maintain class distributions.

For imbalanced data, class weighting or controlled resampling can be applied. Care should be taken to perform resampling only on the training partition to prevent data leakage.

5.4 Software Environment

A possible implementation environment is:

Table 2. Implementation Environment

Component	Proposed Configuration
Programming Language	Python
Data Processing	Pandas, NumPy
Machine Learning	Scikit-learn
Deep Learning	TensorFlow/Keras or PyTorch
Network Simulation	NS-3
Visualization	Matplotlib
Development Environment	Jupyter/Google Colab/local workstation

5.5 Machine Learning Configuration

An initial configuration can be:

Model	Main Parameters
Decision Tree	Gini criterion, controlled tree depth
Random Forest	100–300 trees
SVM	RBF kernel
KNN	5 neighbors
LSTM	32–64 hidden units

Hyperparameters should be optimized using validation data rather than selected solely from previous studies.

6. Evaluation Metrics

6.1 Intrusion Detection Metrics

Accuracy

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

Precision

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

Recall

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

F1-score

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall}$$

False Positive Rate

$$FPR = \frac{FP}{FP + TN}$$

These metrics provide a more complete evaluation than accuracy alone, particularly when the dataset is imbalanced.

6.2 Network Performance Metrics

Packet Delivery Ratio

$$PDR = \frac{N_{received}}{N_{sent}} \times 100$$

Packet Loss

$$PL = N_{sent} - N_{received}$$

Average End-to-End Delay

$$D_{avg} = \frac{\sum_{i=1}^{N} D_i}{N}$$

Throughput

$$Throughput = \frac{Total\ Data\ Received}{Simulation\ Time}$$

Network Overhead

$$OH = \frac{Control\ Packets}{Data\ Packets}$$

7. Network Simulation

To evaluate traffic optimization independently from the machine-learning dataset, an IoT network can be implemented in NS-3.

7.1 Network Topology

A representative topology should include:

- IoT source nodes
- IoT destination nodes
- Wireless access points
- Gateway nodes

- Intermediate routers
- Edge/server node

8. Results and Discussion

Note: The numerical values in the following illustrative tables are examples of how the final results should be reported. They must be replaced by values obtained from the actual implementation.

8.1 Intrusion Detection Results

Table 3. Illustrative Binary Classification Results

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Decision Tree	98.1	97.8	97.5	97.6
KNN	97.4	97.0	96.8	96.9
SVM	97.9	97.4	97.1	97.2
Random Forest	**99.2**	**99.0**	**99.1**	**99.0**
LSTM	98.8	98.6	98.4	98.5

The illustrative results indicate that Random Forest provides the strongest overall performance. This is consistent with the broader literature, where ensemble learning approaches are frequently reported as effective for IoT intrusion detection.

However, actual performance must be determined experimentally because results depend on preprocessing, feature selection, data partitioning, class balancing, and hyperparameter optimization.

7.2 Attack Classification

The multiclass experiment should produce a confusion matrix showing which attack categories are most difficult to distinguish.

A representative interpretation could focus on:

- DDoS detection
- DoS detection
- Reconnaissance detection
- Web-based attack detection
- Brute-force detection
- Spoofing detection
- Mirai detection

The confusion matrix should be analyzed rather than reporting accuracy alone.

8.3 Network Performance

Table 4. Illustrative Traffic-Optimization Results

Scenario	PDR (%)	Throughput (Mbps)	Delay (ms)	Packet Loss (%)
Conventional Routing	89.4	7.8	42.6	10.6
AI IDS Only	92.1	8.2	38.4	7.9
Proposed Framework	**96.3**	**9.1**	**27.8**	**3.7**

The illustrative results suggest that combining intrusion detection with traffic optimization can provide better network performance than conventional routing or intrusion detection alone.

Table 5. Network-Performance Results

Method / Scenario	PDR (%) ↑	Throughput (Mbps) ↑	End-to-End Delay (ms) ↓	Packet Loss (%) ↓	Network Overhead (%) ↓	Energy Consumption (J) ↓
Conventional Routing	89.4	7.80	42.6	10.6	14.8	18.7
AI-Based IDS Only	92.1	8.20	38.4	7.9	16.2	19.4
Proposed AI-Driven Framework	**96.3**	**9.10**	**27.8**	**3.7**	**12.1**	**16.5**

The expected improvement is primarily associated with avoiding congested or high-risk paths and reducing the effect of malicious traffic on legitimate flows.

8.4 Effect on Network Security

The proposed system is expected to improve security by identifying malicious traffic before it significantly affects network resources.

For example, when a DDoS attack causes a sudden increase in traffic volume, the IDS can classify the abnormal traffic. The traffic-management component can then reduce the impact by blocking or rate-limiting malicious sources and redirecting legitimate traffic.

8.5 Effect on Delay

Network delay is expected to decrease because the proposed routing mechanism can avoid congested paths.

A simplified improvement can be calculated as:

$$Improvement = \frac{D_{baseline} - D_{proposed}}{D_{baseline}} \times 100$$

Using the illustrative values:

$$Improvement = \frac{42.6 - 27.8}{42.6} \times 100 \approx 34.7\%$$

Thus, the illustrative framework would provide approximately 34.7% lower delay. This value is only an example and should not be presented as an experimental finding until validated.

8.6 Effect on Throughput

The proposed framework should increase throughput by reducing packet loss and avoiding overloaded paths.

The expected improvement can be calculated as:

$$Improvement = \frac{T_{proposed} - T_{baseline}}{T_{baseline}} \times 100$$

Using the illustrative values:

$$\frac{9.1 - 7.8}{7.8} \times 100 \approx 16.7\%$$

Again, this represents an example calculation rather than a measured result.

9. Discussion

The proposed framework addresses the limitations of approaches that consider intrusion detection and traffic optimization separately.

First, machine learning enables automated analysis of large volumes of IoT traffic. This is important because manual inspection becomes impractical as the number of devices and traffic flows increases.

Second, the framework does not treat an attack as only a cybersecurity event. Malicious traffic can also be considered a network-performance problem because it may consume bandwidth, increase congestion, and reduce packet delivery.

Third, the proposed integration enables security-aware routing. A route can be considered unsuitable not only because of high delay or packet loss but also because it passes through a compromised or suspicious network segment.

The literature indicates that computational efficiency remains an important challenge for IoT intrusion detection because many IoT devices have constrained resources. Therefore, Random Forest or other lightweight models may be more practical for edge deployment than computationally expensive deep-learning architectures.

At the same time, LSTM and other deep-learning models may be beneficial when traffic sequences contain important temporal relationships. Consequently, there is no universally optimal model for every IoT environment.

Another important issue is dataset representativeness. CICIoT2023 provides a considerably more realistic IoT test environment than many older datasets because it includes 105 devices and attacks originating from IoT devices. Nevertheless, no single dataset can fully represent all real-world IoT deployments. Future experiments should therefore validate the proposed approach using multiple datasets and, where possible, real IoT testbeds.

10. Limitations

Several limitations should be considered.

First, the machine-learning evaluation depends on the quality and representativeness of the selected dataset.

Second, the traffic-optimization evaluation is based on network simulation rather than a large physical IoT deployment.

Third, machine-learning models may be vulnerable to adversarial manipulation, including carefully crafted network traffic designed to evade detection.

Fourth, deep-learning models can require greater computational resources, which may make direct deployment on constrained IoT devices difficult.

Finally, the proposed framework does not fully address federated or distributed learning. These approaches could be incorporated in future work to improve privacy when multiple organizations or IoT gateways collaboratively train detection models.

11. Future Work

Future research can extend the proposed framework in several directions.

11.1 Reinforcement Learning

A Deep Q-Network (DQN) or another reinforcement-learning method can replace the rule-based traffic-optimization component.

The agent could optimize a reward function:

$$R = \alpha PDR + \beta Throughput - \gamma Delay - \delta Loss - \epsilon Risk$$

where the coefficients determine the importance of each objective.

11.2 Federated Learning

Federated learning could enable multiple IoT gateways to collaboratively train an intrusion-detection model without sharing raw traffic data.

11.3 Edge Computing

The IDS could be deployed at edge gateways to reduce detection latency and minimize the amount of traffic transmitted to centralized cloud servers.

11.4 Explainable AI

Explainable AI methods could identify why a particular network flow was classified as malicious, increasing the usefulness of the IDS for network administrators.

11.5 Real IoT Testbed

Future research should validate the proposed framework using physical IoT devices, Raspberry Pi gateways, wireless routers, and edge servers.

12. Conclusion

This research proposed an AI-driven framework for improving both cybersecurity and network performance in IoT environments. The framework integrates machine-learning-based intrusion detection with intelligent traffic optimization. The intrusion-detection component analyzes network traffic and classifies it as benign or malicious, while the traffic-management component uses security and network-state information to support more efficient routing decisions.

CICIoT2023 provides an appropriate benchmark for the security component because it contains traffic generated from 105 IoT devices and includes 33 attacks from seven major attack categories.

The proposed evaluation combines cybersecurity metrics, including accuracy, precision, recall, F1-score, and false-positive rate, with networking metrics such as packet delivery ratio, throughput, delay, packet loss, and network overhead.

References

1. Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors, 23*(13), 5941.
2. Kikissagbe, B. R., & Adda, M. (2024). Machine learning-based intrusion detection methods in IoT systems: A comprehensive review. *Electronics, 13*(18), 3601.
3. *Recent endeavors in machine learning-powered intrusion detection systems for the Internet of Things*. (2024). *Journal of Network and Computer Applications, 229*, 103925.
4. *Intrusion Detection Systems in IoT Based on Machine Learning: A state of the art*. (2024). *Procedia Computer Science, 251*, 99–107.
5. *Landscape of learning techniques for intrusion detection system in IoT: A systematic literature review*. (2024). *Computers & Electrical Engineering, 120*, 109725.

6. *A survey on intrusion detection system in IoT networks*. (2024). This study reviews datasets, machine-learning and deep-learning approaches, evaluation metrics, computational efficiency, and open challenges in IoT intrusion detection.
7. Mohammed, M. S., & Talib, H. A. (2024). Using machine learning algorithms in intrusion detection systems: A review. *Tikrit Journal of Pure Science, 29*(3).
8. Mishra, N., & Mishra, S. (2024). A review of machine learning-based intrusion detection system. *EAI Endorsed Transactions on Internet of Things, 10*(1).
9. Tonke, L., Yadav, D. K., & Bargadiya, M. (2024). A systematic review of machine learning-based models for IoT network security in intrusion detection systems. *Journal of Information Systems Engineering and Management, 9*(4s).
10. Esmaeili, M., Rahimi, M., Pishdast, H., Farahmandazad, D., Khajavi, M., & Jabbari Saray, H. (2024). Machine learning-assisted intrusion detection for enhancing Internet of Things security.
11. Jisi, C., Roh, B.-H., & Ali, J. (2024). Reliable paths prediction with intelligent data plane monitoring enabled reinforcement learning in SD-IoT. Journal of King Saud University – Computer and Information Sciences, 36(3), 102006. https://doi.org/10.1016/j.jksuci.2024.102006.
12. Lei, J., & Liu, J. (2024). Reinforcement learning-based load balancing for heavy traffic Internet of Things. Pervasive and Mobile Computing, 99, 101891. https://doi.org/10.1016/j.pmcj.2024.101891.
13. Wang, Y., Li, Y., Lei, J., & Shang, F. (2024). Robust and energy-efficient RPL optimization algorithm with scalable deep reinforcement learning for IIoT. Computer Networks, 255, 110894. https://doi.org/10.1016/j.comnet.2024.110894.
14. Machine learning for QoS and security enhancement of RPL in IoT-enabled wireless sensors. (2024). Sensors International, 5, 100289. https://doi.org/10.1016/j.sintl.2024.100289.