



Explainable Artificial Intelligence for Intrusion Detection in Internet of Things Networks: A Lightweight and Adaptive Security Framework

Dr. Nikhil Verma

Department of Information Technology and Cybersecurity
Central Institute of Computing Sciences, Bhopal, Madhya Pradesh, India

Email: nikhil.verma@cicsbhopal.edu.in

Abstract

The Internet of Things (IoT) has become an important component of modern digital infrastructure, connecting sensors, industrial equipment, smart appliances, vehicles, healthcare devices, and urban infrastructure. Although IoT provides substantial operational benefits, its large-scale deployment also increases the potential attack surface of interconnected systems. Conventional security mechanisms often struggle with the heterogeneous architecture, resource limitations, dynamic behavior, and continuously changing traffic characteristics of IoT networks. Machine learning and deep learning have consequently attracted significant attention for developing intelligent intrusion detection systems capable of identifying anomalous and malicious activities. However, highly complex artificial intelligence models can be difficult for security analysts to interpret, while computational requirements may be unsuitable for resource-constrained IoT environments.

This paper presents a conceptual framework for an explainable and lightweight artificial intelligence-based intrusion detection system for IoT networks. The proposed framework combines data preprocessing, feature optimization, machine learning classification, anomaly detection, explainability, adaptive model updating, and edge-based deployment. Particular attention is given to the relationship between detection performance and model interpretability. The study examines common IoT security threats, including denial-of-service attacks, distributed denial-of-service attacks, probing, credential attacks, malicious traffic, botnet behavior, and anomalous communication patterns. It also discusses the limitations of commonly used benchmark datasets and emphasizes the importance of cross-environment validation.

The proposed framework is designed around five principles: lightweight computation, adaptive learning, explainable decisions, layered deployment, and continuous evaluation. Rather than presenting fabricated experimental results, the paper establishes an experimental methodology and



evaluation framework that can be implemented using publicly available IoT security datasets and real-world network environments. The study concludes that explainability should be treated as an operational component of intelligent intrusion detection rather than as an optional visualization feature. Future research should focus on TinyML-based IDS, federated explainable learning, adversarial robustness, continual learning, and autonomous security response.

Keywords: Internet of Things, Intrusion Detection System, Explainable Artificial Intelligence, Machine Learning, Deep Learning, Cybersecurity, Anomaly Detection, Edge Computing, TinyML, IoT Security

1. Introduction

The Internet of Things represents a broad technological ecosystem in which physical objects are equipped with sensing, computation, communication, and control capabilities. IoT technology is now used in smart homes, industrial automation, transportation, agriculture, energy management, healthcare, logistics, and smart-city infrastructure.

The increasing number of connected devices has created an environment in which large amounts of information move continuously across networks. While this connectivity improves automation and efficiency, it also creates opportunities for attackers.

IoT devices can differ significantly in processor capability, memory, operating system, communication protocol, firmware, and security configuration. Some devices operate continuously with minimal human supervision. Others may remain unpatched for long periods. These characteristics make IoT environments particularly challenging to protect.

Intrusion Detection Systems provide an additional layer of cybersecurity by monitoring network or system activity and identifying potentially malicious behavior. Traditional signature-based IDS approaches are effective against known patterns but may struggle with previously unseen attacks.

Machine learning provides an alternative by learning behavioral patterns from data. However, a second challenge emerges: many sophisticated machine learning models are difficult to interpret.

A security analyst may receive an alert stating that traffic has been classified as malicious without knowing why the model reached that conclusion. In operational environments, such limited interpretability can reduce confidence in automated decisions.

Explainable Artificial Intelligence (XAI) addresses this issue by providing information about the factors influencing model predictions.

This paper investigates how lightweight machine learning and explainability can be integrated into IoT intrusion detection.

2. Background

An IoT security architecture typically contains several layers.

These may include:

- sensing devices,
- communication networks,
- gateways,
- edge computing infrastructure,
- cloud platforms, and
- application services.

An attacker may target any of these layers.

Examples include compromising individual devices, exploiting vulnerable services, manipulating network traffic, stealing credentials, launching denial-of-service attacks, or using compromised IoT devices as components of botnets.

Recent research increasingly combines machine learning, deep learning, federated learning, edge computing, explainability, and lightweight models for IoT intrusion detection. Recent surveys also emphasize computational complexity, false positives, dataset limitations, and real-world validation as continuing challenges.

3. Problem Statement

A practical IoT intrusion detection system must satisfy several requirements simultaneously.

It should:

1. Detect malicious traffic accurately.
2. Identify previously unseen behavior where possible.
3. Operate with limited computational resources.
4. Produce alerts quickly.

5. Minimize false alarms.
6. Explain important decisions.
7. Adapt to changing network behavior.
8. Resist manipulation.
9. Support heterogeneous devices.
10. Scale to large deployments.

Conventional systems often optimize only one or two of these characteristics.

For example, a deep neural network may provide strong classification capability but require computational resources unavailable on low-power devices.

Conversely, an extremely lightweight model may be computationally efficient but insufficiently expressive for complicated attack patterns.

The research problem is therefore to design an architecture that balances detection capability, interpretability, computational efficiency, and adaptability.

4. Research Objectives

The objectives of this study are:

- To examine security challenges in IoT networks.
- To analyze machine learning approaches for intrusion detection.
- To investigate explainability requirements for IoT security.
- To develop a conceptual lightweight XAI-based IDS architecture.
- To identify suitable data-processing techniques.
- To analyze edge deployment requirements.
- To develop an evaluation methodology.
- To identify limitations of current approaches.
- To propose future research directions.

5. Research Questions

The study addresses the following questions:

RQ1: What characteristics make intrusion detection difficult in IoT environments?

RQ2: How can machine learning improve IoT intrusion detection?

RQ3: Why is explainability important for automated security decisions?

RQ4: How can an IDS be designed for resource-constrained devices?

RQ5: How can adaptive learning improve detection against evolving threats?

RQ6: What metrics should be used to evaluate an explainable IoT IDS?

6. Research Methodology

This research follows a conceptual design and analytical methodology.

The methodology includes:

1. Identification of IoT security requirements.
2. Analysis of intrusion detection approaches.
3. Examination of machine learning techniques.
4. Examination of explainability techniques.
5. Development of a conceptual architecture.
6. Definition of an experimental evaluation strategy.
7. Identification of limitations and future research opportunities.

The paper does not present invented experimental measurements. The proposed evaluation methodology can be implemented using appropriate IoT intrusion datasets and controlled network environments.

7. IoT Threat Landscape

IoT networks can experience several categories of attacks.



Major categories include:

- denial-of-service,
- distributed denial-of-service,
- scanning and probing,
- credential attacks,
- malware activity,
- botnet communication,
- spoofing,
- man-in-the-middle attacks,
- unauthorized access,
- protocol exploitation,
- data manipulation.

Different attacks generate different traffic patterns.

For example, a denial-of-service attack may produce unusual traffic volumes, whereas credential attacks may involve repeated authentication attempts.

A machine learning system can potentially identify such behavioral differences.

8. Intrusion Detection Systems

An IDS monitors activity and determines whether observed behavior should be classified as normal or suspicious.

IDS architectures are commonly categorized into:

Host-Based IDS

A host-based system monitors activity on individual devices.

Network-Based IDS

A network-based system analyzes communication traffic.

Hybrid IDS

A hybrid approach combines host and network information.

For IoT environments, hybrid architectures can provide broader visibility but also increase computational and deployment requirements.

9. Signature-Based Detection

Signature-based systems identify known attack patterns.

Their major advantage is relatively strong performance against previously identified attacks.

However, they have limitations.

A signature must generally be created for an attack pattern before it can be reliably identified.

New attacks, polymorphic malware, and previously unknown behavioral patterns can therefore create challenges.

10. Anomaly-Based Detection

Anomaly-based IDS attempts to learn normal behavior and identify deviations.

This approach is attractive for IoT environments because many IoT devices perform relatively predictable functions.

For example, an environmental sensor may normally communicate with a limited set of services at regular intervals.

A sudden change in communication behavior could indicate compromise.

However, legitimate changes can also appear anomalous.

Therefore, anomaly detection can generate false positives if normal operational variation is not adequately represented during training.

11. Machine Learning for IoT IDS

Machine learning can process large collections of network features and learn relationships between features and attack categories.

Potential algorithms include:

- decision trees,
- random forests,
- support vector machines,
- logistic regression,
- k-nearest neighbors,
- gradient boosting,
- naive Bayes,
- neural networks.

The appropriate algorithm depends on dataset size, feature complexity, deployment constraints, and required latency.

12. Deep Learning for IoT IDS

Deep learning can automatically learn complex representations from large datasets.

Common architectures include:

- multilayer perceptrons,
- convolutional neural networks,
- recurrent neural networks,
- long short-term memory networks,
- autoencoders,
- transformers.

Deep learning may be useful for complicated traffic patterns but can increase computational requirements.

Consequently, a deep learning model designed for cloud deployment may need substantial optimization before being deployed at the edge.

13. Feature Engineering

Feature engineering remains important even when machine learning is used.

Network features can include:

- packet counts,
- packet sizes,
- connection duration,
- source and destination information,
- protocol characteristics,
- port information,
- request frequency,
- error rates,
- flow statistics.

The objective is to identify features that contain useful information while eliminating irrelevant or redundant variables.

14. Feature Selection

Large feature sets may increase computational requirements and create unnecessary complexity.

Feature-selection techniques can identify a smaller subset of informative variables.

Possible approaches include:

- correlation analysis,
- mutual information,

- recursive feature elimination,
- tree-based importance,
- statistical tests,
- dimensionality reduction.

Feature selection can improve inference speed and may also simplify explanations.

15. Data Preprocessing

IoT security datasets may contain:

- missing values,
- duplicate records,
- inconsistent formats,
- extreme values,
- categorical variables,
- class imbalance.

Preprocessing should therefore be carefully designed.

Typical steps include:

1. Data cleaning.
2. Missing-value treatment.
3. Encoding categorical attributes.
4. Feature scaling.
5. Outlier analysis.
6. Duplicate removal.
7. Feature selection.

The preprocessing pipeline used during training must remain consistent with the production environment.

16. Class Imbalance

Intrusion datasets frequently contain unequal numbers of samples across classes.

Normal traffic may represent the majority of observations, while some attack categories may be relatively rare.

A classifier trained without considering this imbalance may achieve high overall accuracy while performing poorly on rare attacks.

Possible approaches include:

- class weighting,
- controlled oversampling,
- undersampling,
- synthetic data generation,
- threshold adjustment.

Evaluation should therefore include class-specific metrics.

17. Importance of Precision

Precision measures how many observations classified as malicious are actually malicious.

A system with low precision may produce many false alerts.

In operational security environments, excessive false positives can overwhelm analysts.

Therefore, an IDS should not be evaluated solely on its ability to identify attacks.

It must also minimize unnecessary alerts.

18. Importance of Recall

Recall measures the ability to detect actual malicious observations.

High recall is particularly important for critical security applications.

However, maximizing recall without considering precision can increase false positives.

This demonstrates the need for balanced evaluation.

19. F1-Score

The F1-score combines precision and recall into a single metric.

It is particularly useful when classes are imbalanced.

However, even F1-score should not be considered sufficient on its own.

A practical IDS should also be evaluated using:

- detection latency,
- computational cost,
- memory usage,
- energy consumption,
- robustness,
- explainability.

20. Explainable Artificial Intelligence

Explainable AI refers to methods that provide human-understandable information about machine learning decisions.

For intrusion detection, an explanation might identify the network characteristics that contributed strongly to an alert.

For example, the explanation could indicate that an unusual connection rate, abnormal packet behavior, and unexpected communication pattern contributed to the classification.

This information can help analysts investigate alerts more effectively.

21. Importance of Explainability in Cybersecurity

Security analysts must often make decisions based on automated alerts.

An unexplained alert may create uncertainty.

An explanation can help answer:

- Why was this traffic considered suspicious?
- Which features influenced the decision?
- Was the classification caused by a single feature?
- Is the decision consistent with known attack behavior?
- Should the alert be investigated immediately?

Explainability therefore has practical operational value.

22. Local and Global Explanations

Explainability can be divided broadly into two categories.

Local Explanation

Explains one particular prediction.

For example:

Why was this network connection classified as malicious?

Global Explanation

Explains general model behavior.

For example:

Which features are generally most important for detecting attacks?

Both perspectives are useful for security operations.

23. Model-Agnostic Explainability

Model-agnostic methods attempt to explain different types of machine learning models without requiring the explanation technique to be specifically built into the classifier.

Examples include:

- LIME,
- SHAP,
- permutation-based importance.

These approaches can be applied to different model types.

However, their computational requirements may become significant when explanations must be generated for large numbers of real-time events.

24. Lightweight Explainability

A production IoT IDS should not generate an explanation that consumes more resources than the detection process itself.

Therefore, explainability should be designed with resource constraints in mind.

Possible strategies include:

- explanation only for high-risk alerts,
- precomputed global feature importance,
- simplified local explanations,
- edge-based explanation generation,
- analyst-triggered detailed explanations.

This selective approach can reduce overhead.

25. Proposed Architecture

The proposed framework consists of seven major components:

IoT Devices → Traffic Collection → Edge Preprocessing → Lightweight Detection → Explanation Engine → Security Dashboard → Adaptive Model Manager

Each component has a specific responsibility.



Component 1: Traffic Collection

Collects relevant network or device telemetry.

Component 2: Edge Preprocessing

Cleans and transforms the collected information.

Component 3: Detection Engine

Classifies activity as normal or suspicious.

Component 4: Explanation Engine

Generates interpretable information about important decisions.

Component 5: Security Dashboard

Displays alerts and explanations to analysts.

Component 6: Adaptive Model Manager

Monitors model performance and identifies potential degradation.

Component 7: Central Coordination

Manages model updates and long-term analysis.

26. Edge-Based Deployment

The proposed IDS places initial detection close to the data source.

Edge deployment can reduce:

- communication latency,
- bandwidth requirements,
- cloud dependency.

An edge gateway can monitor traffic from a group of IoT devices and execute a lightweight model. Only important alerts or aggregated information may need to be transmitted to a central platform.

27. Cloud-Based Coordination

Cloud infrastructure can still provide valuable services.

These include:

- long-term model training,
- historical analysis,
- model version management,
- centralized dashboards,
- cross-site security analysis,
- large-scale data storage.

The proposed architecture therefore does not eliminate cloud computing.

Instead, it distributes responsibilities between edge and cloud environments.

28. Adaptive Learning

IoT environments change over time.

New devices may be introduced. Communication patterns may change. Firmware updates may alter traffic characteristics. Attackers may also modify their techniques.

A static model can therefore experience performance degradation.

Adaptive learning enables periodic model updates.

However, continuous retraining must be controlled carefully because malicious data may otherwise influence the model.

29. Concept Drift

Concept drift occurs when the statistical characteristics of data change over time.

In an IoT environment, concept drift can result from:

- new device configurations,
- seasonal changes,
- software updates,
- changes in user behavior,
- network architecture changes,
- emerging attacks.

An adaptive IDS should monitor changes and determine when model updating is necessary.

30. Continual Learning

Continual learning allows a model to incorporate new information over time.

A major challenge is catastrophic forgetting, in which learning new patterns causes the model to lose previously learned knowledge.

Future IoT IDS research should therefore investigate continual learning techniques that preserve historical detection capability while incorporating new attack patterns.

31. Federated Learning Integration

Different organizations may operate IoT systems containing sensitive network information.

They may be unwilling to share raw traffic data.

Federated learning can allow multiple organizations or edge nodes to collaboratively improve models without directly centralizing raw datasets.

The combination of federated learning and explainability is promising, but explanations themselves must also be examined for privacy leakage.

32. Adversarial Machine Learning

Attackers may deliberately manipulate machine learning systems.

Possible techniques include:

- adversarial examples,
- poisoned training data,
- manipulated features,
- model evasion,
- backdoor behavior.

An IDS should therefore be tested against adversarial conditions rather than only conventional attack datasets.

33. Model Poisoning

If attackers gain control of training participants, they may attempt to influence the learned model.

This is particularly relevant to distributed and federated architectures.

Possible defenses include:

- robust aggregation,
- participant reputation,
- anomaly detection,
- update validation,
- secure participant authentication.

34. Explainability as a Security Tool

Explainability can potentially help detect model failures.

Suppose an IDS begins classifying normal traffic as malicious because one irrelevant feature has unexpectedly changed.

Feature-level explanations may help analysts identify this problem.

Therefore, XAI can support not only user trust but also model monitoring and debugging.

35. Benchmark Datasets

IoT IDS research commonly uses benchmark datasets.

Examples include:

- CICIDS2017,
- UNSW-NB15,
- TON_IoT,
- Bot-IoT,
- IoT-23,
- NSL-KDD.

Each dataset has different characteristics, attack categories, feature structures, and collection environments.

Recent literature emphasizes that benchmark performance does not automatically demonstrate real-world deployment reliability, particularly when datasets do not adequately represent current heterogeneous IoT environments.

36. Dataset Limitations

A major problem in machine learning-based security research is dependence on historical datasets.

A model may achieve strong results on a particular benchmark because the training and testing distributions are similar.

Real networks may be substantially different.

Potential problems include:

- outdated attack patterns,
- artificial traffic generation,

- duplicated records,
- unrealistic class distributions,
- limited device diversity,
- leakage between training and testing data.

Consequently, cross-dataset evaluation is strongly recommended.

37. Cross-Dataset Validation

A stronger experimental design trains a model using one dataset and tests it against another compatible environment.

This can provide evidence about generalization.

For example, researchers can investigate whether a model trained on one IoT traffic collection maintains useful performance when evaluated against a different dataset.

Such experiments can reveal whether a model has learned general behavioral characteristics or merely memorized dataset-specific patterns.

38. Proposed Experimental Design

A future implementation of the proposed framework can follow these stages:

Stage 1

Select one or more IoT intrusion datasets.

Stage 2

Clean and preprocess the datasets.

Stage 3

Divide data into training, validation, and testing subsets.



Stage 4

Perform feature selection.

Stage 5

Train lightweight baseline models.

Stage 6

Compare selected machine learning and deep learning models.

Stage 7

Add explainability mechanisms.

Stage 8

Measure computational overhead.

Stage 9

Test robustness against attack variations.

Stage 10

Evaluate cross-dataset generalization.

39. Baseline Algorithms

The experimental study can compare:

Algorithm	Expected Advantage	Major Concern
Decision Tree	Simple and interpretable	May overfit
Random Forest	Robust and effective	Larger model
Support Vector Machine	Strong classification capability	Computational cost
Gradient Boosting	High predictive performance	Training complexity



Logistic Regression	Lightweight	Limited nonlinear modeling
Neural Network	Flexible nonlinear learning	Reduced interpretability
Autoencoder	Useful for anomaly detection	Threshold selection

These models should be evaluated under identical data-processing conditions.

40. Detection Latency

In cybersecurity, a highly accurate model may still be ineffective if detection takes too long.

Latency can be divided into:

- data acquisition latency,
- preprocessing latency,
- inference latency,
- explanation latency,
- alert transmission latency.

Edge deployment can reduce some communication delays.

41. Computational Efficiency

A lightweight IDS should be evaluated using:

- CPU utilization,
- memory consumption,
- model size,
- inference time,
- energy consumption.

These measurements are particularly important when detection occurs on gateways or embedded devices.

42. Explainability Evaluation

Explainability should itself be evaluated.

Useful dimensions include:

- explanation consistency,
- feature relevance,
- computational cost,
- stability,
- analyst usefulness,
- comprehensibility.

A technically sophisticated explanation may not necessarily be useful to a security analyst.

Therefore, human-centered evaluation should complement mathematical explanation quality.

43. False Positive Management

False positives are one of the major operational challenges of IDS.

An organization receiving thousands of incorrect alerts may eventually ignore the system.

The proposed framework therefore includes risk scoring.

Instead of presenting every anomaly with equal priority, alerts can be classified into categories such as:

- informational,
- low risk,
- medium risk,
- high risk,
- critical.

This allows analysts to focus attention on the most important events.

44. Risk-Based Alert Prioritization

Alert prioritization can consider:

- confidence of the classifier,
- asset importance,
- attack category,
- historical behavior,
- frequency,
- communication destination,
- device vulnerability.

For example, unusual behavior from a critical industrial controller may deserve higher priority than a similar event from a low-risk sensor.

45. Security Dashboard

A practical system should present information in a concise form.

A dashboard could display:

- device identifier,
- event time,
- predicted attack category,
- confidence level,
- important features,
- historical activity,
- risk score,
- recommended investigation action.

This converts raw model predictions into actionable security information.

46. Human-in-the-Loop Security

Automated IDS should not necessarily replace security analysts.

A human-in-the-loop approach allows analysts to:

- confirm alerts,
- reject false positives,
- provide feedback,
- request explanations,
- initiate additional analysis.

Feedback can potentially be used for controlled future model improvement.

47. Privacy Considerations

Network monitoring can itself produce sensitive information.

Therefore, an intelligent IDS should protect:

- device identities,
- network addresses,
- communication metadata,
- organizational information,
- user behavior.

Data retention policies should also be considered.

Privacy protection is especially important when security monitoring data is shared between organizations.

48. Security of the IDS Itself

An IDS can become an attack target.

An attacker who compromises the detection system could:

- suppress alerts,
- manipulate classifications,
- modify models,
- steal security data,
- disable monitoring.

Therefore, the IDS must itself be protected using:

- secure authentication,
- access control,
- encrypted communication,
- secure software updates,
- model integrity verification,
- logging and auditing.

49. Model Version Management

Machine learning models require controlled lifecycle management.

Every deployed model should have identifiable:

- version,
- training dataset,
- training period,
- feature configuration,
- algorithm,
- validation results.

This makes it possible to investigate changes in detection performance.

50. Proposed Adaptive Security Cycle

The proposed system follows an iterative cycle:

Monitor → Detect → Explain → Investigate → Learn → Validate → Deploy

Monitoring generates observations.

Detection identifies suspicious behavior.

Explanation provides context.

Investigation validates the alert.

Learning incorporates approved information.

Validation checks model performance.

Deployment distributes an updated model.

This cycle supports continuous security improvement.

51. Advantages of the Proposed Framework

The proposed framework offers several conceptual advantages.

Lightweight Operation

Feature selection and edge processing reduce unnecessary computation.

Explainable Decisions

Analysts receive information about important decision factors.

Adaptive Detection

The system can be updated as network behavior changes.

Distributed Deployment

Detection can occur near the source of network traffic.

Flexible Model Selection

Different models can be used according to device capability.

Human Oversight

Security analysts remain part of the decision process.

52. Limitations

The framework also has limitations.

First, explainability does not guarantee that the underlying model is correct.

Second, adaptive learning may introduce new security risks if malicious data is incorporated.

Third, lightweight models may not capture every complex attack pattern.

Fourth, edge deployment requires appropriate hardware and software support.

Fifth, benchmark datasets cannot fully represent every real-world IoT environment.

Sixth, sophisticated attacks may deliberately attempt to deceive both the classifier and explanation mechanism.

53. Research Gap

Several important gaps remain.

First, many studies emphasize predictive accuracy while providing limited analysis of computational requirements.

Second, explainability is often evaluated independently of real security operations.

Third, many experiments use a single benchmark dataset.

Fourth, adversarial robustness is not consistently incorporated into evaluation.

Fifth, the relationship between explanation quality and analyst decision-making requires additional empirical investigation.

Sixth, real-world edge deployment remains less explored than laboratory evaluation.

54. Future Research Direction: TinyML-Based IDS

TinyML provides an opportunity to perform machine learning directly on highly constrained devices.

Future research can investigate extremely compact intrusion detection models that operate with:

- low memory,
- low power,
- limited computation,
- intermittent connectivity.

Model compression, quantization, pruning, and efficient architectures may be useful.

Recent IoT IDS research increasingly identifies lightweight and resource-efficient detection as an important direction for constrained environments.

55. Future Research Direction: Federated XAI

Federated learning can provide distributed training, while explainability can make resulting models more understandable.

A federated XAI framework could provide:

- local explanations,
- global explanations,
- privacy-preserving feature importance,
- distributed model monitoring.

However, explanation information itself may reveal sensitive characteristics.

56. Future Research Direction: Self-Adaptive IDS

Future IDS systems may automatically adjust detection strategies based on environmental changes.

A self-adaptive system could change:

- detection thresholds,
- client participation,
- feature sets,
- model versions,
- alert priorities.

Such autonomy must be accompanied by safeguards against unsafe automatic decisions.

57. Future Research Direction: Digital Twins

Digital twins can provide virtual representations of physical IoT environments.

A security-focused digital twin could be used to simulate:

- device behavior,
- network traffic,
- attack scenarios,
- model responses.

This may allow researchers to test IDS models without disrupting operational infrastructure.

58. Future Research Direction: Generative AI

Generative AI may support cybersecurity through:

- synthetic attack generation,
- security-log summarization,
- analyst assistance,
- explanation generation,
- threat intelligence analysis.

However, generated information must be validated carefully.

Generative models can produce incorrect or misleading outputs, which could be dangerous in security operations.

59. Future Research Direction: Autonomous Response

Detection is only one part of cybersecurity.

Future systems may integrate detection with controlled response actions.

Possible responses include:

- isolating compromised devices,
- blocking suspicious communication,
- increasing monitoring,
- requesting human confirmation,
- applying temporary network restrictions.

Automatic response should be carefully constrained because false positives can cause operational disruption.

60. Future Research Direction: Adversarially Robust XAI

Attackers may attempt to manipulate both predictions and explanations.

Future research should therefore investigate whether explanations remain reliable under adversarial conditions.

A trustworthy XAI-based IDS should ideally remain useful when:

- traffic features are manipulated,
- attack patterns evolve,
- training data is poisoned,
- models are targeted by evasion techniques.

61. Discussion

The integration of artificial intelligence and IoT security creates significant opportunities but also introduces new responsibilities.

A machine learning IDS should not be viewed simply as a classifier.

It is part of a larger security ecosystem.

The quality of an IDS depends on:

- data quality,
- feature design,
- model selection,
- deployment architecture,
- explanation mechanisms,
- security controls,
- update processes,
- human interaction.

Recent research similarly emphasizes the movement toward lightweight, adaptive, explainable, edge-based, and privacy-conscious IoT security systems.

62. Practical Deployment Recommendations

Organizations planning to deploy AI-based IoT IDS should consider the following recommendations:

1. Begin with a clearly defined security objective.
2. Understand normal device behavior.
3. Avoid relying on a single benchmark dataset.
4. Use feature selection to reduce unnecessary computation.
5. Evaluate false positives carefully.
6. Deploy lightweight detection near the edge when appropriate.

7. Provide explanations for high-risk decisions.
8. Protect the IDS itself.
9. Maintain model version records.
10. Continuously evaluate model performance.
11. Test against adversarial scenarios.
12. Keep human oversight for high-impact decisions.

63. Proposed Evaluation Matrix

A comprehensive experimental evaluation should measure six dimensions.

Dimension	Example Metrics
Detection	Accuracy, precision, recall, F1-score
Speed	Inference latency, response time
Resource Use	CPU, memory, energy
Explainability	Stability, relevance, analyst usefulness
Security	Attack resilience, poisoning resistance
Generalization	Cross-dataset and cross-environment performance

This multidimensional evaluation is more informative than reporting accuracy alone.

64. Expected Contributions

The conceptual contributions of this study are:

1. A lightweight architecture for AI-based IoT intrusion detection.
2. Integration of explainability into the detection workflow.
3. An edge-oriented deployment model.
4. An adaptive security cycle.
5. A multidimensional evaluation framework.
6. Identification of important research gaps.

7. A roadmap for future explainable and resource-efficient IoT security.

65. Conclusion

The rapid expansion of IoT networks has created a strong requirement for intelligent and adaptive cybersecurity mechanisms. Traditional security techniques remain important, but the scale, heterogeneity, and dynamic behavior of IoT systems create conditions in which machine learning-based intrusion detection can provide valuable additional capabilities.

However, predictive performance alone is insufficient. A practical IoT intrusion detection system must also consider computational limitations, detection latency, false positives, adaptability, security of the model itself, and the ability of human analysts to understand automated decisions.

This paper proposed a conceptual framework combining lightweight machine learning, edge computing, explainable artificial intelligence, adaptive learning, and human-centered security operations. The architecture moves initial processing and detection closer to IoT devices while allowing cloud infrastructure to support long-term training, management, and analysis.

Explainability represents a particularly important component. When an IDS identifies suspicious behavior, security analysts should have access to meaningful information about the factors influencing that decision. Such information can support investigation, debugging, trust, and model monitoring.

Future research should focus on TinyML-based intrusion detection, federated explainable learning, continual learning, adversarial robustness, digital twins, generative AI, and controlled autonomous response.

Ultimately, the next generation of IoT intrusion detection systems should not simply answer the question **“Is this traffic malicious?”** They should also help answer **“Why was it considered malicious, how confident is the system, what evidence supports the decision, and what should happen next?”**

References

1. Rahman, M. M., Al Shakil, S., & Mustakim, M. R. (2025). A survey on intrusion detection system in IoT networks. *Cyber Security and Applications*, 3, 100082.

2. Mallidi, S. K. R., & Ramisetty, R. R. (2025). Advancements in training and deployment strategies for AI-based intrusion detection systems in IoT: A systematic literature review. *Discover Internet of Things*, 5, 8.
3. Hozouri, A., Mirzaei, A., & Effatparvar, M. (2025). A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. *Discover Artificial Intelligence*, 5, 314.
4. Alhamdi, M. J. M., Lopez-Guede, J. M., AlQaryouti, J., Rahebi, J., Zulueta, E., & Fernandez-Gamiz, U. (2025). AI-based malware detection in IoT networks within smart cities: A survey. *Computer Communications*, 233, 108055.
5. Nguyen, D. C., Ding, M., Pathirana, P. N., et al. (2021). Federated learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1622–1658.
6. Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115, 619–640.
7. Ferrag, M. A., Maglaras, L., Moschogiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50, 102419.
8. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. *2015 Military Communications and Information Systems Conference*, 1–6.
9. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 108–116.
10. Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, 100, 779–796.
11. Meidan, Y., Bohadana, M., Mathov, Y., et al. (2018). N-BaloT: Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3), 12–22.
12. Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. *2018 IEEE Security and Privacy Workshops*, 29–35.

13. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
14. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?” Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.
15. Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries*, 1(1).
16. Shapira, A., & others. (2025). Machine learning-based security solutions for IoT networks: A comprehensive survey. *Sensors*, 25(11), 3341.
17. Alrashoud, M., et al. (2025). Unveiling IoT ecosystem security: A review of intelligent IDS, trends, challenges, and future directions. *Computers & Electrical Engineering*, 128, 110626.
18. Yeh, K. H., et al. (2025). Securing the future: AI-driven cybersecurity in the age of autonomous IoT. *Frontiers in the Internet of Things*, 4.
19. Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *International Conference on Learning Representations*.
20. Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2), 1–210.

Add in-text citations throughoutCondense repetitive sections and paragraphs