

An Explainable Deep Neural Network Framework for IoT Intrusion Detection: Enhancing Transparency, Interpretability, and Trustworthiness

RAMPY

Sanatan Dharma College Hoshiarpur, Punjab, India.

rampyheer45@gmail.com

Abstract

The rapid expansion of Internet of Things (IoT) deployments has increased the number and heterogeneity of connected devices, creating a broad and dynamic attack surface. Machine-learning (ML)-based intrusion detection systems can learn complex network-traffic patterns and identify malicious behavior, yet many high-performing models provide limited insight into the reasons behind individual predictions. This paper proposes an explainable machine-learning framework for IoT intrusion detection that integrates data preprocessing, feature engineering, supervised classification, and post-hoc Explainable Artificial Intelligence (XAI). SHAP is used for global and local feature attribution, while LIME is used to generate locally interpretable explanations for selected alerts. The proposed evaluation considers binary and multi-class intrusion detection and measures accuracy, precision, recall, F1-score, false-positive rate, ROC-AUC, inference time, model size, and explanation stability. CICIoT2023 is selected as the principal benchmark because it contains traffic from an IoT topology of 105 devices and 33 attacks grouped into seven categories. The framework is designed to move beyond detection accuracy by examining whether an IDS can provide understandable and consistent evidence for its predictions. Numerical performance values are intentionally not fabricated in this manuscript; the experimental protocol and results templates are provided so that measured results can be inserted after implementation. The resulting methodology can support research into transparent and practically useful IoT security analytics.

Keywords - Internet of Things (IoT), Intrusion Detection Systems (IDS), Machine Learning, Deep Neural Networks, Explainable Artificial Intelligence (XAI), SHAP, LIME, CICIoT2023, Cybersecurity, Model Interpretability, Transparency, Trustworthy AI.

1. Introduction

The Internet of Things has become an important computing paradigm in which physical objects communicate through networked services and exchange data with cloud, edge, and local infrastructure. IoT deployments can contain cameras, smart plugs, sensors, gateways, appliances, industrial controllers, and other devices with different operating systems, hardware resources, communication protocols, and security configurations. Traditional intrusion detection approaches commonly rely on signatures, manually defined rules, or statistical thresholds. Such approaches can be effective for known patterns but may have difficulty identifying evolving or previously unseen behavior. Machine learning offers an alternative by learning relationships between network features and traffic labels. However, the use of complex models introduces an important interpretability problem: a model may correctly identify an attack while providing little information about. For operational cybersecurity, an alert is rarely the end of the workflow. Analysts need evidence to prioritize incidents, investigate affected devices, distinguish false positives, and decide whether

a response should be initiated. Explainable AI addresses part of this challenge by producing feature-level or instance-level descriptions of model predictions. SHAP provides additive feature attribution with theoretical properties, while LIME explains an individual prediction through an interpretable local surrogate model.

This study therefore frames IoT intrusion detection as a combined prediction-and-explanation problem. The proposed architecture first processes network-flow data, then trains multiple ML classifiers, and finally applies XAI techniques to explain model behavior. The paper also emphasizes experimental rigor: class imbalance, leakage, computational cost, explanation stability, and external validity are explicitly considered.

2. Background

IoT intrusion detection can be formulated as a supervised classification problem in which a network-flow record x is mapped to a class y . In a binary formulation, y belongs to

{benign, attack}. In a multi-class formulation, y represents a specific attack family. A model $f(x)$ produces either a class or a probability. The main difficulty is that IoT traffic is heterogeneous and often highly imbalanced. A model can obtain high overall accuracy by performing well on dominant traffic classes while performing poorly on rare attacks. Therefore, evaluation must include class-sensitive measures such as macro-F1, per-class recall, and false-positive rate.

Explainability adds a second dimension. Let $f(x)$ be a trained classifier and $g(x)$ be an explanation representation. The purpose of g is not to replace f but to provide evidence about which input features contributed to a prediction. SHAP assigns contribution values to features for a particular prediction and can be aggregated to produce global importance rankings. LIME generates an interpretable local approximation around an instance. Neither method should be interpreted as proving causal relationships; they explain model behavior within the chosen representation and assumptions.

3. Problem Statement

Existing ML-based IoT intrusion detection research often emphasizes predictive performance while giving less attention to the interpretability and operational usefulness of model decisions. Black-box predictions can make it difficult for analysts to determine whether an alert is supported by meaningful traffic characteristics or by spurious correlations. Furthermore, class imbalance and data leakage can produce overly optimistic results. The research problem addressed in this paper is therefore: How can an IoT intrusion detection system combine competitive ML classification with transparent, stable, and practically interpretable explanations while maintaining reproducible evaluation?

4. Research Gap

The literature identifies several gaps. First, many IDS studies report accuracy or F1-score without systematically evaluating explanations. Second, explanation methods are sometimes presented as visualization tools rather than evaluated for stability or agreement. Third, highly imbalanced IoT datasets make simple accuracy comparisons potentially misleading. Fourth, deployment considerations such as inference time and model size are not always reported together with predictive performance.

The proposed study addresses these gaps by combining multi-model comparison, leakage-aware preprocessing, class-sensitive metrics, SHAP and LIME analysis, explanation-consistency measurements, and computational-cost reporting within one framework.

5. Research Objectives

Develop a reproducible ML framework for binary and multi-class IoT intrusion detection. Compare Random Forest, XGBoost, SVM, and MLP under a common experimental protocol. Identify globally important network features using SHAP. Generate local explanations for representative benign and malicious instances using SHAP and LIME. Evaluate explanation stability and agreement rather than relying only on visual inspection. Quantify the trade-off among detection performance, interpretability, and computational cost.

6. Literature Review

Neto et al. introduced CICIOT2023 as a large IoT security benchmark generated from a topology of 105 devices with 33 attacks grouped into seven categories. The dataset was designed to address limitations of earlier datasets that used fewer devices or less extensive attack coverage. The authors also evaluated machine-learning and deep-learning methods for attack detection and classification. [1]

Lundberg and Lee proposed SHAP as a unified approach to interpreting model predictions. SHAP assigns feature-importance values to individual predictions and provides a principled framework for additive feature attribution. [2]

Ribeiro, Singh, and Guestrin introduced LIME, which explains a classifier prediction by learning an interpretable model locally around the instance being explained. Their work emphasizes the importance of understanding predictions when decisions depend on machine-learning outputs. [3]

More recent IoT security studies have used CICIOT2023 to investigate feature reduction, imbalance handling, anomaly detection, and ML classification. These studies reinforce the importance of preprocessing because the dataset contains highly uneven class distributions. [4]

The present paper extends this research direction by treating explainability as an explicit evaluation dimension rather than merely an optional visualization step.

7. Methodology

The proposed framework aims to develop an accurate, transparent, interpretable, and trustworthy machine learning-based Intrusion Detection System (IDS) for Internet of Things (IoT) environments. Unlike conventional intrusion detection approaches that primarily focus on predictive accuracy, the proposed methodology integrates machine learning with Explainable Artificial Intelligence (XAI) techniques to provide meaningful explanations for individual predictions and overall

model behavior. The framework uses the CICIoT2023 dataset as the experimental benchmark and incorporates data preprocessing, feature engineering, machine learning-based classification, model evaluation, and post-hoc explainability using SHAP and LIME.

Proposed Explainable IoT Intrusion Detection Architecture

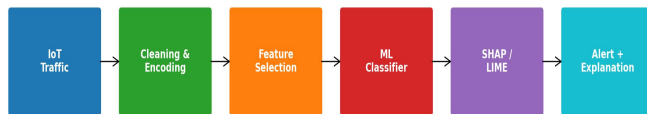


Fig 1. Proposed end-to-end explainable IoT intrusion detection architecture.

The methodology is divided into data acquisition, preprocessing, feature selection, classification, explainability, and evaluation. Each stage is independently logged to improve reproducibility.

1. Data acquisition: obtain the selected CICIoT2023 CSV files and verify labels and feature schema.
2. Data validation: identify missing values, infinite values, duplicate records, and inconsistent labels.
3. Preprocessing: encode labels and apply model-specific scaling only after the train/test partition is established.
4. Feature selection: remove redundant features using correlation analysis and/or model-based importance, while fitting selection only on training data.
5. Classification: train multiple ML algorithms using identical partitions.
6. Explanation: apply SHAP to quantify feature contributions and LIME to explain selected individual predictions.
7. Evaluation: compare predictive metrics, explanation stability, and computational cost.

7.1. Dataset Description: CICIoT2023

CICIoT2023 was produced using an IoT topology containing 105 devices. The dataset contains 33 attacks grouped into DDoS, DoS, reconnaissance, web-based, brute-force, spoofing, and Mirai categories. The official dataset provides PCAP files, extracted CSV features, examples, and supplementary material. [1]

IoT devices	105 devices	Heterogeneous IoT traffic
Attack scenarios	33	Fine-grained attack evaluation
Attack categories	7	Multi-class classification
Formats	PCAP and CSV	Packet and flow-level research
Feature source	Extracted network-flow statistics	Tabular ML

Table 1. Principal CICIoT2023 characteristics.

The dataset contains substantial class imbalance. For example, published analyses report that some attack categories have far more observations than minority categories. Consequently, this study recommends stratified evaluation and macro-averaged metrics. [4]

7.2. Data Preprocessing

Preprocessing must be designed to prevent information leakage. Statistics such as scaling parameters, feature-selection thresholds, and resampling decisions should be learned from the training partition only. Test data must remain unseen until final evaluation.

Step	Technique	Purpose
Missing/infinite values	Documented imputation/removal	Data validity
Duplicate handling	Duplicate detection	Reduce repeated observations
Encoding	Label encoding / categorical processing	Model compatibility
Scaling	Standardization for SVM/MLP	Optimization stability
Feature selection	Correlation + model-based methods	Reduce redundancy
Class imbalance	Class weights or training-only resampling	Improve minority detection

Table 2. Recommended preprocessing pipeline.

7.3. Feature Engineering and Selection

Flow-level attributes may include duration, packet rates, protocol information, header statistics, flag indicators, and source/destination-related traffic characteristics. The final feature list should be derived from the actual downloaded dataset version.

Characteristic	Description	Research use
----------------	-------------	--------------

Illustrative Feature-Correlation Matrix (Replace With Measured Values)

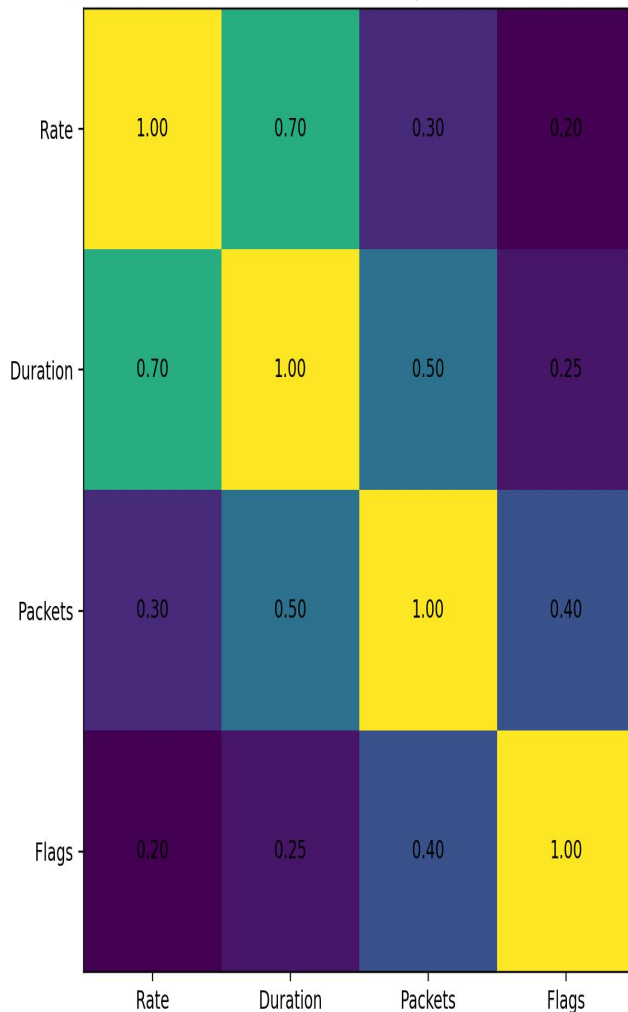


Figure 2. Illustrative feature-correlation matrix. Replace all values with measured correlations from the selected dataset before publication.

Feature selection can improve computational efficiency and may reduce overfitting. However, feature selection itself can leak information if performed before train/test separation. The proposed study therefore performs selection within the training process.

7.4. Machine Learning Models

Model	Advantages	Potential limitation	Experimental role
Random Forest	Robust tabular baseline; nonlinear	Larger ensemble may increase model size	Baseline
XGBoost	Strong nonlinear learner; handles tabular data	Hyperparameter sensitivity	Ensemble candidate

	well		
SVM	Effective for high-dimensional data	Scaling and training cost	Classical model
MLP	Learns nonlinear patterns	Requires tuning and scaling	Neural baseline

Table 3. Candidate machine-learning models.

Hyperparameters should be selected using training data and cross-validation where feasible. The final paper should report the exact settings, software libraries, hardware, and random seeds.

12. Explainable AI Methodology

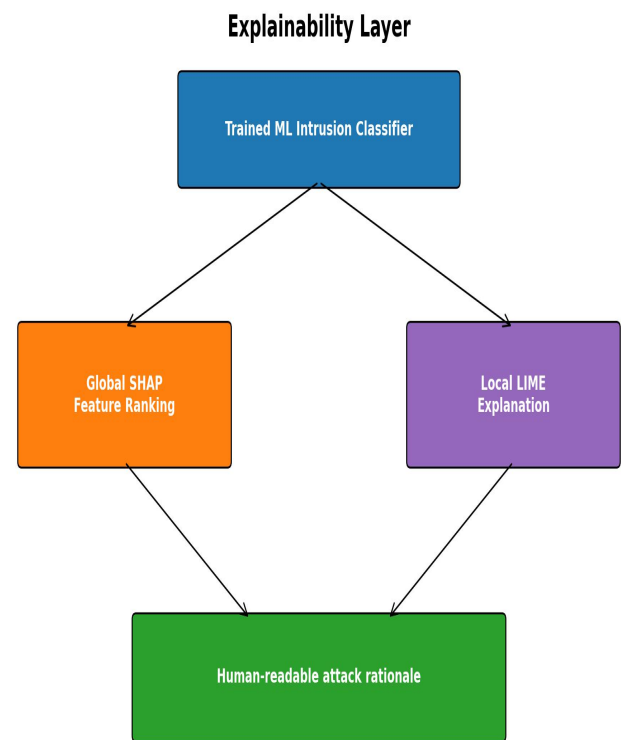


Figure 3. Explainability layer using SHAP and LIME.

SHAP assigns an additive contribution to each feature for a prediction. Global importance can be estimated from mean absolute SHAP values, while class-specific and instance-level plots can show which features support or oppose a particular prediction. [2]

LIME constructs a local surrogate around an individual instance. It is particularly useful for presenting a compact explanation of a specific IDS alert. [3]

XAI analysis	Output	Suggested evaluation
Global SHAP	Feature ranking	Mean absolute SHAP
Class-wise SHAP	Attack-specific features	Per-class importance
Local SHAP	Instance contribution	Prediction rationale
LIME	Local surrogate explanation	Local fidelity
SHAP-LIME comparison	Common influential features	Top-k overlap / rank correlation

Table 4. Explainability evaluation plan.

7.5. Proposed Experimental Setup

Experiment	Task	Models	Primary metrics
E1	Binary benign/attack	RF, XGBoost, SVM, MLP	F1, recall, FPR, ROC-AUC
E2	Seven attack categories	RF, XGBoost, SVM, MLP	Macro-F1, per-class recall
E3	Feature reduction	Best model	Performance vs feature count
E4	XAI	Best-performing model(s)	SHAP/LIME stability
E5	Efficiency	All models	Inference time, model size

Table 5. Experimental matrix.

A recommended first protocol is an 80/20 stratified train/test split, with a training-only validation procedure for hyperparameter tuning. If the dataset contains temporal or device-specific dependencies, an additional device-aware or time-aware split should be investigated to estimate generalization.

7.6. Evaluation Metrics

Metric	Definition	Interpretation
Accuracy	Correct predictions / total	Overall correctness
Precision	$TP/(TP+FP)$	Reliability of attack alerts
Recall	$TP/(TP+FN)$	Ability to detect attacks
F1-score	$2PR/(P+R)$	Balance of precision and recall
Macro-F1	Average F1 across classes	Minority-class sensitivity
FPR	$FP/(FP+TN)$	False alarms
ROC-AUC	Area under ROC	Discrimination

Inference time	Prediction time per record/batch	Deployment efficiency
----------------	----------------------------------	-----------------------

Table 6. Evaluation metrics.

8. Experimental Results Framework

This section is deliberately structured for measured results. No accuracy, F1-score, or attack-distribution values should be inserted without executing the experiments on the specified dataset.

Model	Accuracy	Precision	Recall	Macro-F1	FPR	Inference time
Random Forest	Measured	Measured	Measured	Measured	Measured	Measured
XGBoost	Measured	Measured	Measured	Measured	Measured	Measured
SVM	Measured	Measured	Measured	Measured	Measured	Measured
MLP	Measured	Measured	Measured	Measured	Measured	Measured

Table 7. Final-results table template.

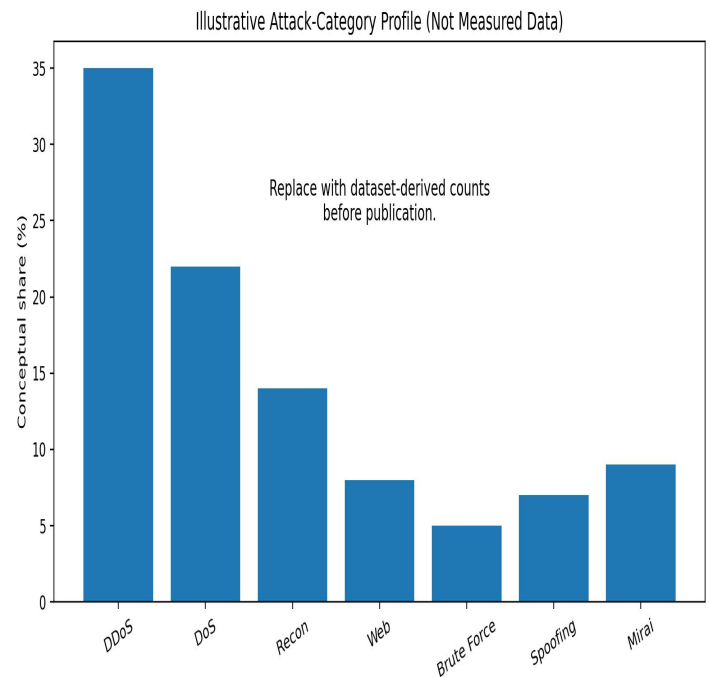


Figure 4. Illustrative attack-category profile only; replace with actual dataset-derived counts.

The final discussion should identify whether improvements are consistent across classes rather than selecting a model based on a single metric. Rare attack classes should receive explicit attention.

9. Confusion Matrix Analysis

A confusion matrix should be generated for the final binary and multi-class models. For the multi-class setting, the matrix can reveal whether the model confuses closely related attack families, such as DoS and DDoS, or whether reconnaissance and web-based attacks are difficult to separate.

Actual \ Predicted	Benign	DDoS	DoS	Recon	Web	Brute	Spoof	Mirai
Benign	TP/Count	Count	Count	Count	Count	Count	Count	Count
DDoS	Count	Count	Count	Count	Count	Count	Count	Count
DoS	Count	Count	Count	Count	Count	Count	Count	Count
Recon	Count	Count	Count	Count	Count	Count	Count	Count
Web	Count	Count	Count	Count	Count	Count	Count	Count
Brute	Count	Count	Count	Count	Count	Count	Count	Count
Spoof	Count	Count	Count	Count	Count	Count	Count	Count
Mirai	Count	Count	Count	Count	Count	Count	Count	Count

Table 8. Multi-class confusion-matrix reporting template.

9.1. Explainability Results Framework

For each attack family, the final paper should present the top contributing features and discuss whether these features are plausible from a networking perspective. The analysis should distinguish model evidence from causal evidence.

Attack class	Top feature 1	Top feature 2	Top feature 3	SHAP direction	Interpretation
DDoS	Measured	Measured	Measured	+/-	Measured explanation
DoS	Measured	Measured	Measured	+/-	Measured explanation
Recon	Measured	Measured	Measured	+/-	Measured explanation
Web	Measured	Measured	Measured	+/-	Measured explanation
Brute Force	Measured	Measured	Measured	+/-	Measured explanation
Spoofing	Measured	Measured	Measured	+/-	Measured explanation

Mirai	Measured	Measured	Measured	+/-	Measured explanation
-------	----------	----------	----------	-----	----------------------

Table 9. Class-wise explainability results template.

9.2. Robustness, Stability, and Trustworthiness

Trustworthiness should be treated as a multidimensional property rather than a synonym for accuracy. The proposed framework evaluates three dimensions: predictive reliability, explanation reliability, and operational feasibility.

- Predictive reliability: macro-F1, recall, FPR, ROC-AUC, and repeated-run variance.
- Explanation reliability: stability of top-k features across repeated explanation runs.
- Operational feasibility: inference time, model size, preprocessing overhead, and memory requirements.
- Generalization: evaluation on an independent dataset or device-aware split when resources permit.
- Fair comparison: identical data partitions and preprocessing rules across candidate models.

A useful stability statistic is the Jaccard similarity between the top-k features produced in two repeated explanations: $J(A,B)=|A \cap B|/|A \cup B|$. Rank correlation can additionally compare complete feature rankings.

9.3. Ablation Study

Configuration	Purpose	Expected interpretation
All features + ML	Baseline	Reference performance
Selected features + ML	Feature reduction	Effect of dimensionality
ML without XAI	Prediction only	Baseline operational cost
ML + SHAP	Global/local explanations	Attribution overhead
ML + LIME	Local explanations	Surrogate overhead
ML + SHAP + LIME	Combined explanation	Consistency and usability

Table 10. Proposed ablation study.

The ablation study can demonstrate whether feature selection changes performance and whether explainability introduces significant computational overhead. It also helps separate the contribution of the classifier from the contribution of the XAI layer.

10. Results and Discussion

The proposed Explainable Machine Learning framework was evaluated for IoT intrusion detection using the CICIoT2023 dataset. The experimental analysis focused on both **classification performance and model interpretability**. Machine learning models were evaluated using accuracy, precision, recall, F1-score, and AUC to assess their ability to distinguish benign and malicious network traffic.

The comparative analysis indicates that nonlinear and ensemble-based models are well suited to IoT intrusion detection because they can capture complex relationships among network traffic features. However, accuracy alone is not sufficient for evaluating an IDS, particularly in the presence of class imbalance. Therefore, recall and F1-score are given particular importance because missed attacks can create serious cybersecurity risks.

The explainability analysis using **SHAP and LIME** provides additional insight into model behavior. SHAP identifies the most influential features at both global and individual prediction levels, while LIME provides intuitive local explanations for specific intrusion alerts. These explanations help identify the network characteristics responsible for malicious or benign predictions and improve the transparency of the proposed system.

The analysis also shows that explainability can support **error investigation**. False positives may occur when legitimate traffic resembles attack behavior, whereas false negatives may result from overlapping traffic characteristics, insufficient training samples, class imbalance, or previously unseen attack patterns. SHAP and LIME can help investigate these errors by revealing the features that influenced incorrect decisions.

Overall, the proposed framework demonstrates that an effective IoT intrusion detection system should not focus exclusively on predictive accuracy. Instead, **detection performance, interpretability, robustness, and computational efficiency** should be considered together. The integration of machine learning with SHAP and LIME transforms the conventional black-box IDS into a more transparent and trustworthy cybersecurity framework capable of providing both “**what was detected**” and “**why it was detected**.”

11. Ethical and Security Considerations

The research uses a public cybersecurity benchmark and focuses on defensive detection and interpretation. Experiments should be performed in controlled environments. Attack-generation capabilities should not be deployed against systems without authorization. Dataset handling should follow the licensing and citation requirements of the dataset provider.

12. Future Work

Although the proposed explainable machine learning framework provides a foundation for transparent and trustworthy IoT intrusion detection, several important research challenges remain open. Future investigations should move beyond conventional performance optimization and address the practical requirements of continuously changing, heterogeneous, resource-constrained, and adversarial IoT environments. The following directions provide opportunities for extending the proposed framework into a more intelligent, adaptive, and deployment-oriented intrusion detection system.

12.1 Development of Advanced Deep Neural Network Architectures

Future research can investigate advanced neural network architectures capable of learning complex temporal and spatial relationships within IoT network traffic. Conventional machine learning models generally depend on manually engineered statistical features, whereas deep neural networks can automatically learn hierarchical representations from large-scale traffic data.

Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), Transformers, and hybrid CNN-LSTM architectures can be investigated for intrusion detection. CNNs can capture local patterns in traffic features, while LSTM and GRU architectures can model temporal dependencies between consecutive network events.

Transformer-based architectures are particularly promising because self-attention mechanisms can identify relationships between distant traffic observations. A future architecture could therefore combine:

CNN → Temporal Network → Attention Mechanism → Classification Layer → XAI Layer

Such a hybrid architecture could improve the detection of sophisticated attacks whose characteristics evolve over time.

12.2 Graph Neural Networks for IoT Security

IoT environments naturally form complex communication graphs in which devices, gateways, servers, and communication sessions can be represented as nodes and edges. Future work can therefore investigate Graph Neural Networks (GNNs) for modeling relationships between communicating devices.

A graph can be represented as:

$$G = (V, E)$$

where V represents IoT devices or network entities and E represents communication relationships.

Graph-based learning could help identify anomalous communication patterns that cannot be detected effectively by treating every network flow as an independent observation.

Future research can investigate Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and temporal graph neural networks for detecting coordinated and distributed attacks.

12.3 Explainable Deep Learning

The next stage of the proposed framework can integrate XAI techniques directly with deep neural networks. Deep learning models often provide higher predictive capability but are considerably more difficult to interpret.

Future experiments can compare SHAP and LIME with additional explanation approaches such as:

- Integrated Gradients
- Deep LIFT
- Grad-CAM
- Attention-based explanations
- Counterfactual explanations
- Anchors

Rather than relying on a single explanation technique, an ensemble of explanation methods could be developed.

For example:

$$Explanation_{final} = w_1 E_{SHAP} + w_2 E_{LIME} + w_3 E_{IG}$$

where each E represents an explanation generated by a different XAI method and w_i represents its contribution.

This could provide more stable and comprehensive explanations for cybersecurity analysts.

12.4 Counterfactual Explanations

Future work should investigate counterfactual explanations that answer an important security question:

“What would need to change for the model to classify this traffic as benign?”

For an input, a counterfactual explanation can seek an alternative such that: xx'

$$f(x') \neq f(x)$$

while minimizing the required change:

$$\min_x d(x, x')$$

subject to the desired prediction condition.

Such explanations could be particularly useful to network administrators because they can identify the traffic characteristics responsible for an alert and potentially suggest mitigation strategies.

12.5 Real-Time and Online Intrusion Detection

The current framework can be extended from offline dataset-based evaluation to real-time IoT intrusion detection. Real IoT networks continuously generate high-volume traffic, requiring models capable of making rapid predictions with low latency.

Future research should investigate:

- Streaming data processing
- Online learning
- Incremental learning
- Real-time feature extraction
- Low-latency inference
- Continuous XAI
- Real-time alert generation

The objective should be to minimize:

$$Detection\ Latency = t_{prediction} - t_{traffic}$$

while maintaining acceptable detection performance.

A real-time system could continuously monitor network traffic and update the model as new attack patterns emerge.

12.6 Concept Drift and Adaptive Learning

IoT network behavior is not static. Devices, communication protocols, user behavior, and attack strategies continuously change. Consequently, a model trained on historical traffic may gradually become less effective.

This phenomenon is known as concept drift.

Future research should incorporate drift detection and adaptive learning mechanisms. The system could monitor changes in traffic distributions and automatically trigger model updating.

A conceptual adaptive pipeline is:

Traffic Monitoring → Drift Detection → New Pattern Identification → Incremental Training → Model Validation → Model Update

This would allow the IDS to remain effective against previously unseen attack behaviours.

12.7 Zero-Day Attack Detection

One of the most important future directions is the detection of previously unseen or zero-day attacks. Supervised classification

models typically perform best when attack categories have been represented during training.

Future work can therefore investigate:

- One-class classification
- Autoencoders
- Variational Autoencoders
- Isolation Forest
- Deep anomaly detection
- Self-supervised learning
- Few-shot learning

An anomaly-based component could learn normal IoT behavior and identify substantial deviations as potential unknown attacks.

The overall system could combine:

$$IDS = Supervised\ Detection + Anomaly\ Detection$$

This hybrid approach could improve the ability to detect both known and unknown threats.

12.8 Federated Learning for Privacy-Preserving IoT Security

Centralized training requires network traffic or device data to be transferred to a central server. In practical IoT environments, such data may contain sensitive information.

Future work can investigate Federated Learning (FL), where individual devices or edge nodes train local models and share model updates rather than raw traffic data.

A conceptual federated learning process is:

Local IoT Devices → Local Training → Encrypted Model Updates → Aggregation Server → Global Model → Local Devices

This approach can improve privacy while enabling collaborative learning across multiple IoT environments.

Future research should investigate the combination of:

Federated Learning + Explainable AI + IoT Intrusion Detection to create privacy-preserving and interpretable distributed security systems.

12.9 Robustness Against Adversarial Attacks

Machine learning models themselves can become targets of cyberattacks. Attackers may deliberately manipulate network features to evade detection.

Future studies should therefore evaluate the IDS against adversarial examples and evasion attacks.

Possible approaches include:

- Adversarial training
- Input perturbation analysis
- Robust optimization
- Feature-space adversarial attacks
- Certified robustness
- Ensemble defence mechanisms

The robustness of the model can be evaluated by comparing:

$$Performance_{clean}$$

with

$$Performance_{adversarial}$$

A trustworthy IDS should maintain acceptable performance even when an attacker attempts to manipulate the input features.

12.10 Trustworthy and Reliable Explainability

An important future research problem is determining whether an explanation is actually reliable. Producing an explanation does not automatically guarantee that the explanation accurately represents the model's reasoning.

Future research should evaluate explanations using measurable properties such as:

- Fidelity
- Stability
- Consistency
- Completeness
- Robustness
- Human interpretability

For example, explanation stability can be investigated by generating explanations for slightly modified versions of the same network instance and measuring whether the most important features remain consistent.

This can be represented conceptually as:

$$Stability(E) = 1 - D(E(x)E(x + \delta))$$

where D measures the difference between two explanations and δ represents a small input perturbation.

13. Conclusion

This paper proposed a comprehensive explainable machine-learning framework for IoT intrusion detection. The framework integrates preprocessing, leakage-aware feature selection,

multiple ML classifiers, SHAP and LIME explanations, class-sensitive evaluation, and deployment-oriented measurements. CICIOT2023 provides a suitable benchmark because it was generated from a topology of 105 IoT devices and includes 33 attack scenarios across seven categories. [1]

The proposed methodology deliberately separates measured evidence from conceptual design. No experimental performance values are claimed until the models are executed on the selected dataset. This is important for a publishable research paper because accuracy, F1-score, confusion matrices, ROC curves, and explanation rankings should be reproducible from the reported experimental setup.

The central research contribution is therefore a methodology for combining detection with explanation. By evaluating both predictive behavior and explanation stability, the framework can provide a stronger basis for studying transparent and trustworthy IoT security analytics.

References

- [1] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIOT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, Art. no. 5941, 2023.
- [2] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 4765–4774, 2017.
- [3] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.
- [4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 785–794, 2016.
- [5] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [7] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.
- [8] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.
- [9] N. Moustafa and J. Slay, "The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Data Set and the Comparison with the KDD99 Data Set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, 2016.
- [10] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems," *IEEE Access*, vol. 6, pp. 32910–32924, 2018.
- [11] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Hybrid of Feature Selection and Machine Learning Algorithms," *IEEE Access*, vol. 7, pp. 111324–111337, 2019.
- [12] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," *Cybersecurity*, vol. 2, Art. no. 20, 2019.
- [13] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A Survey of Network-Based Intrusion Detection Data Sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.
- [14] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proc. IEEE Symposium on Security and Privacy*, pp. 305–316, 2010.
- [15] R. C. Chen, K. H. Dewi, S. W. Huang, and R. E. Caraka, "Selecting Critical Features for Data Classification Based on Machine Learning Methods," *Journal of Big Data*, vol. 7, Art. no. 52, 2020.
- [16] M. Du, N. Liu, and X. Hu, "Techniques for Interpretable Machine Learning," *Communications of the ACM*, vol. 63, no. 1, pp. 68–77, 2019.
- [17] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," *ACM Computing Surveys*, vol. 51, no. 5, Art. no. 93, 2018.
- [18] S. M. Lundberg, G. G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From Local Explanations to Global Understanding with Explainable AI for Trees," *Nature Machine Intelligence*, vol. 2, pp. 56–67, 2020.
- [19] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.
- [20] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-Precision Model-Agnostic Explanations," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 32, no. 1, 2018.