Information Security Education Journal (ISEJ)

DLINE JOURNALS

Print ISSN: 2349-8161
Online ISSN: 2349-817X

ISEJ 2026:13 (2)
https://doi.org/10.6025/isej/2026/13/2/84-121

An Explainable Machine Learning Framework for Network Intrusion Detection and Cyberattack Classification: Performance Evaluation, Feature Importance, and Threshold Optimization

Yao-Liang Chung
National Taiwan Ocean University
Taiwan
ylchung@email.ntou.edu.tw

ABSTRACT

The rapid evolution of sophisticated cyberattacks necessitates intelligent and adaptive Network Intrusion Detection Systems (NIDS). While machine learning (ML) offers promising capabilities for automated threat identification, conventional ML-based IDS frameworks often struggle with severe class imbalance, suboptimal default classification thresholds, and a lack of interpretability. Furthermore, models may inadvertently learn dataset-specific signatures, compromising their generalizability across diverse network environments. This study proposes an explainable, threshold-optimized ML framework for network intrusion detection and cyberattack classification. Utilizing a highly imbalanced dataset (24:1 benign-to-malicious ratio) comprising 10,000 network traffic records and 9 distinct attack categories, we systematically evaluate five ML classifiers (Gradient Boosting, Extra Trees, Random Forest, Logistic Regression, and SVM). To address class imbalance, we employ class-sensitive metrics (e.g., PR-AUC, Macro-F1, MCC) alongside a validation-driven threshold optimization strategy. Additionally, we integrate Explainable AI (XAI) techniques, including permutation importance and SHAP, alongside feature ablation experiments to investigate model decision-making and feature dependence. Experimental results demonstrate that Gradient Boosting achieves the most favorable operating point when the decision threshold is optimized from the default 0.50 to 0.169, improving recall from 0.260 to 0.580 and the F1-score to 0.558. However, feature importance analysis reveals a heavy reliance on the user_agent attribute, raising concerns regarding model generalization and highlighting the need for robust feature-ablation and chronological validation. In multiclass scenarios, significant disparities between Macro-F1 and Weighted-F1 scores expose the models' difficulties in recognizing rare attack categories. Ultimately, this research contributes a unified, transparent, and operationally relevant IDS architecture that bridges predictive performance, threshold aware detection, and explainable feature analysis, providing actionable insights for cybersecurity practitioners.

Keywords: Network Intrusion Detection System (NIDS), Explainable Artificial Intelligence (XAI), Class Imbalance, Threshold Optimization, SHAP (SHapley Additive exPlanations), Cyberattack Classification, Machine Learning

Received: 29 April 2026, Revised: 3 August 2026, Accepted: 19 August 2026

Copyright: **DLINE**

1. Introduction

The rapid expansion of digital connectivity, network infrastructure, and computational capabilities has significantly transformed modern information technology environments. However, this development has also increased the exposure of network systems to unauthorized intrusions and increasingly sophisticated cyberattacks. These attacks exploit vulnerabilities in network services and communication infrastructures to obtain unauthorized access, compromise sensitive information, or disrupt normal system operations, thereby threatening the security and reliability of entire information technology infrastructures [1]. The growing scale and complexity of network based threats have consequently intensified cybersecurity concerns across different sectors [1, 2].

Recent statistics further demonstrate the severity of the global cybersecurity landscape. In 2024, more than 6.5 billion malware attacks were reportedly recorded worldwide, while the average cost of a data breach reached approximately USD 4.88 million [2, 3]. These developments highlight the increasing importance of developing intelligent, adaptive, and reliable security mechanisms capable of identifying and mitigating malicious network activities before they cause substantial damage.

Conventional intrusion detection systems (IDSs) primarily depend on predefined rules, signature-based detection mechanisms, and manual examination of network logs. Although these approaches have contributed to network security, their effectiveness has become increasingly limited in the presence of dynamic and evolving cyber threats [4, 5, 6, 7]. Manual monitoring is particularly challenging because it requires considerable administrative effort and generally follows a reactive approach. Consequently, previously unknown attacks, rapidly changing intrusion patterns, and sophisticated malicious activities may remain undetected until significant damage has occurred.

The limitations of conventional security mechanisms have encouraged the adoption of machine learning (ML) and deep learning (DL) techniques in modern intrusion detection systems. Unlike traditional approaches that depend extensively on predefined detection rules, intelligent IDSs can learn patterns from network traffic and identify potentially malicious activities through automated analysis [8-11]. The concept of intrusion detection has evolved considerably since its early development, including the approach introduced by Jim Anderson in 1980, which involved the manual examination of system logs by administrators to identify suspicious activities [5]. Modern IDS architectures increasingly employ intelligent automation to improve detection efficiency, reduce manual intervention, and respond more effectively to emerging network threats [8-11].

2. Background

2.1 Machine Learning Approaches for Intrusion Detection

Over the past decade, numerous machine learning approaches have been investigated to improve the accuracy and efficiency of intrusion detection systems. These approaches include supervised learning, unsupervised learning, pattern recognition, and deep learning architectures. Among the commonly explored deep learning techniques are artificial neural networks (ANNs) and recurrent neural networks (RNNs), which provide capabilities for learning complex relationships and identifying patterns within network traffic data [12, 13, 6, 8, 9, 10, 11, 13].

Each learning approach provides different advantages depending on the characteristics of the network traffic, the nature of the attack patterns, and the availability of labelled training data. Supervised learning techniques can identify known attack categories using previously labelled observations, whereas unsupervised learning approaches can identify unusual patterns without requiring predefined attack labels. Deep learning approaches further provide opportunities to model complex relationships within high dimensional network traffic data.

Nevertheless, selecting an appropriate learning algorithm and dataset remains essential for developing an effective intrusion detection framework. Model performance depends not only on the underlying algorithm but also on the quality, diversity, and representativeness of the data used for training and evaluation. Consequently, research has increasingly emphasized the use of large scale and high quality datasets to develop, train, and validate machine-learning-based intrusion detection models.

The growing application of ML and DL techniques in cybersecurity therefore represents an important advancement in automated threat identification. However, achieving reliable intrusion detection requires consideration of several interrelated factors, including classification performance, the ability to distinguish normal and malicious activities, the identification of influential network features, and the appropriate selection of detection thresholds.

2.2 Anomaly-Based Intrusion Detection in IoT and IIoT Environments

The increasing deployment of Internet of Things (IoT) and Industrial Internet of Things (IIoT) technologies has introduced additional security challenges. These environments consist of interconnected devices, communication networks, and application services that continuously exchange information. Protecting such infrastructures requires effective mechanisms for identifying cyberattacks and detecting abnormal network activities. Intrusion detection systems consequently play an important role in maintaining the security and operational reliability of IoT and IIoT environments.

Deep learning techniques are frequently employed in the anomaly detection components of modern intrusion detection systems. Despite their potential to identify complex attack patterns, these models may produce high false positive rates and often provide limited explanations for their predictions. Their decision making processes can therefore remain difficult to interpret, even for cybersecurity experts [14]. This lack of transparency creates challenges when security administrators must evaluate alerts, understand the reasons behind detected intrusions, and determine appropriate mitigation measures.

Anomaly-based intrusion detection identifies deviations from established normal network behaviour. Unlike signature-based detection, which primarily relies on previously recognized attack patterns, anomaly-based approaches can potentially identify previously unknown attacks by detecting unusual activities that differ from normal system behaviour. This capability makes anomaly detection particularly relevant to dynamic IoT environments, where new attack patterns and threats may emerge continuously.

However, a fundamental limitation of conventional anomaly based detection is that any significant deviation from normal behaviour may be classified as suspicious, even when the observed activity is legitimate. Consequently, changes in network traffic patterns can generate false positive alerts, reducing detection reliability and increasing the workload of security administrators [13].

Recent developments in machine learning and deep learning have encouraged researchers to incorporate intelligent learning techniques into anomaly-based intrusion detection systems. By learning representative patterns from network traffic, these approaches seek to improve the identification of malicious activities while reducing the limitations associated with conventional anomaly detection. ML/DL-based anomaly detection has therefore attracted considerable research attention, particularly for enhancing network security in IoT environments [15, 16].

The continuous occurrence of different attacks and security threats in IoT systems makes the accurate identification and mitigation of malicious activities essential. An IoT architecture generally consists of three principal layers: the perception layer, the network layer, and the application layer. The perception layer is associated with the collection of information through connected devices and sensors, the network layer facilitates communication and data transmission, and the application layer supports the delivery of services and application specific functionalities [17]. Each layer forms part of the overall IoT infrastructure and consequently represents an important consideration in developing comprehensive security mechanisms. [18].

2.3 Research Motivation and Framework Considerations

The limitations identified in conventional and learning based intrusion detection approaches demonstrate the importance of developing intelligent security frameworks that combine effective cyberattack classification with reliable model interpretation. Although machine learning and deep learning techniques offer promising capabilities for identifying malicious network activities, their practical effectiveness must be evaluated through systematic performance assessment.

In particular, intrusion detection models must be capable of distinguishing legitimate network behaviour from different categories of cyberattacks while maintaining an appropriate balance between detection capability and false-positive generation. Furthermore, understanding the contribution of individual network traffic features is important for examining the factors influencing model predictions. Explainable machine learning can support this requirement by providing insights into feature importance and the decision-making behaviour of trained classifiers.

Detection threshold optimization also represents an important consideration in intrusion detection because the classification threshold can influence the balance between identifying genuine attacks and generating false alarms. Therefore, evaluating classification performance alongside feature importance and threshold selection provides a broader basis for examining the effectiveness and practical reliability of an intelligent intrusion detection framework.

Against this background, an explainable machine learning framework for network intrusion detection and cyberattack classification provides a research direction for integrating predictive performance evaluation, feature importance analysis, and threshold optimization. Such an approach is intended to support more transparent and systematic assessment of machine learning based intrusion detection, particularly in increasingly complex and interconnected network environments.

3. Research Problem, Research Gaps and Objectives

3.1 Research Problem

The increasing sophistication of cyberattacks and the growing complexity of interconnected network environments have created a critical need for intelligent and reliable intrusion detection mechanisms. Conventional intrusion detection systems primarily depend on predefined attack signatures, static security rules, and manually configured detection mechanisms. Although these approaches can identify previously recognized threats, their effectiveness is restricted when confronted with evolving attack patterns, previously unseen malicious activities, and complex network traffic behaviour.

Machine-learning-based intrusion detection systems provide an alternative by learning representative characteristics of benign and malicious network traffic. However, their practical effectiveness is affected by several challenges, particularly when the underlying dataset contains highly imbalanced classes, heterogeneous attack categories, and features that may encode dataset-specific patterns.

The experimental dataset considered in this research contains 10,000 network traffic observations, comprising 9,600 benign records and 400 malicious records. This distribution represents a 24:1 imbalance between benign and malicious traffic. Consequently, conventional classification models may achieve high overall accuracy while failing to identify a substantial proportion of actual attacks. The binary classification experiments demonstrate this limitation: Gradient Boosting achieves 96.8% accuracy but only 26% recall at the conventional probability threshold of 0.50. Similarly, Random Forest achieves 96.7% accuracy with only 19% recall. These results demonstrate that accuracy-based evaluation alone is insufficient for assessing intrusion detection effectiveness.

Another important problem concerns the selection of an appropriate classification threshold. A default threshold of 0.50 may not provide an operationally acceptable balance between attack detection and false-positive generation. The experimental results demonstrate that reducing the Gradient Boosting threshold to

0.169 increases recall from 0.260 to 0.580 and improves the F1-score from 0.397 to 0.558. Although this adjustment improves attack sensitivity, it also introduces additional false positive predictions. Therefore, an effective IDS must consider the operational consequences of both missed attacks and false alarms rather than relying exclusively on conventional classification thresholds.

Furthermore, high performing machine-learning models frequently provide limited insight into the underlying reasons for their predictions. Understanding which network characteristics contribute to an intrusion alert is essential for security analysts who must validate suspicious activities and determine appropriate mitigation measures. Although permutation importance identifies user_agent as the dominant predictor in the current experiment, its substantial influence raises concerns regarding potential dependence on dataset-specific signatures and the generalizability of the model across different network environments.

The multiclass classification problem introduces an additional challenge. The nine attack categories are unevenly represented, and the smallest category contains only ten observations. The resulting differences between macro-F1 and weighted-F1 demonstrate that satisfactory aggregate performance does not necessarily indicate reliable recognition of individual attack types.

Therefore, the central research problem addressed in this study is:

How can an explainable and threshold-optimized machine learning framework be developed to improve the detection of imbalanced network intrusions, distinguish multiple cyberattack categories, identify influential network traffic characteristics, and provide reliable and interpretable classification decisions while reducing false alarms and improving generalization

The research addresses this problem by integrating comparative machine learning evaluation, class sensitive performance assessment, threshold optimization, feature importance analysis, and explainability within a unified intrusion detection framework.

3.2 Research Gaps

The following research gaps are derived from the limitations identified in the supplied dataset and experimental findings. They represent dataset specific methodological gaps and should not be interpreted as an exhaustive systematic review of all existing IDS literature.

Research Gap 1: Inadequate evaluation under severe class imbalance

Conventional accuracybased evaluation can produce misleading conclusions when benign network traffic substantially exceeds malicious traffic. The 24:1 class imbalance in the experimental dataset illustrates this problem. Although the evaluated classifiers achieve relatively high accuracy, their attack detection recall remains comparatively low.

Identified gap: The need for a class-sensitive evaluation framework that prioritizes minority-class detection alongside conventional classification performance.

Proposed research direction: Integrate precision, recall, F1-score, macro-F1, PR-AUC, ROC-AUC, Matthews correlation coefficient (MCC), and false-positive and false-negative rates into the model evaluation process.

Research Gap 2: Insufficient optimization of intrusion detection thresholds

The default probability threshold of 0.50 does not necessarily represent the most suitable operating point for intrusion detection. The observed improvement in Gradient Boosting performance after threshold optimization demonstrates that threshold selection directly influences attack detection sensitivity.

Identified gap: Conventional classification decisions do not adequately account for the different operational costs of false negatives and false positives.

Proposed research direction: Develop a validation driven threshold optimization strategy that selects decision thresholds according to F1-score, recall constraints, or operational false positive rate requirements.

Research Gap 3: Limited interpretability of machine learning based intrusion detection

High performing ensemble classifiers can provide accurate predictions without clearly explaining the network characteristics responsible for individual decisions. Existing permutation importance identifies user_agent as the dominant feature, but global feature importance alone does not adequately explain individual attack predictions.

Identified gap: Insufficient integration of global and local explanations into intrusion detection model evaluation.

Proposed research direction: Incorporate explainable AI techniques, particularly SHAP, permutation importance, and local explanation analysis, to investigate the contribution of network traffic features to individual classification decisions.

Research Gap 4: Potential dependence on dataset-specific features

The substantial dominance of the user_agent feature suggests that the classifier may learn collection-specific patterns rather than generalizable malicious network behaviour. This can reduce the reliability of intrusion detection when the model encounters different devices, browsers, applications, or network environments.

Identified gap: Insufficient assessment of feature dependence and model robustness under changes in feature availability and network characteristics.

Proposed research direction: Conduct feature ablation experiments, particularly with and without user_agent, and compare model performance using alternative feature subsets.

Research Gap 5: Poor representation of minority cyberattack categories

The multiclass experiments reveal substantial differences between macro-F1 and weighted-F1. This indicates that models may perform reasonably well on frequently represented categories while failing to recognize rare attack types.

Identified gap: Inadequate consideration of class specific performance and minority attack representation

Proposed research direction: Introduce class-sensitive evaluation, training-only resampling where justified, class weighting, and attack-specific performance analysis.

Research Gap 6: Insufficient validation of model generalizability

Random stratified splitting may distribute related or temporally adjacent network observations across training and testing partitions. Consequently, the evaluation may overestimate performance when similar traffic patterns occur in both partitions.

Identified gap: Limited evidence of generalization across time periods, network sources, sessions, and independent network environments.

Proposed research direction: Supplement stratified validation with chronological holdout, grouped validation, and independent external testing.

Research Gap 7: Limited integration of detection performance, interpretability, and operational decision

making

Performance evaluation, feature interpretation, and threshold selection are often treated as separate analytical activities. However, an operational IDS requires these components to work together.

Identified gap: Lack of an integrated evaluation strategy connecting predictive performance, explanation quality, detection sensitivity, and false-alarm behaviour.

Proposed research direction: Develop a unified explainable IDS framework that integrates classification, threshold optimization, feature interpretation, and operational error assessment.

The seven gaps above are the proposed research framing for the supplied experiment. Gaps 1–5 are directly motivated by its reported results; gaps 6–7 identify additional validation and integration requirements.

3.3 Research Objectives

The primary objective is to develop and evaluate an explainable machine learning framework for network intrusion detection and cyberattack classification under imbalanced traffic conditions**.**

The specific objectives are:

1. Dataset characterization: Examine network traffic distributions, class imbalance, feature characteristics, missing values, and the representation of different attack categories.

2. Binary intrusion detection: Develop and compare machine-learning classifiers for distinguishing benign network traffic from malicious activities.

3. Comparative model evaluation: Assess Gradient Boosting, Extra Trees, Random Forest, Logistic Regression, and Support Vector Machine using multiple class-sensitive performance metrics.

4. Threshold optimization: Identify suitable classification thresholds using training derived validation probabilities to improve attack detection sensitivity and balance false positive and false negative predictions.

5. Multiclass cyberattack classification: Investigate the ability of machine-learning models to distinguish among the nine malicious attack categories.

6. Explainable feature analysis: Identify influential network traffic characteristics using permutation importance and SHAP-based global and local explanations.

7. Feature robustness assessment: Evaluate the influence of dominant predictors through feature-ablation experiments, particularly the removal of user_agent.

8. Generalization assessment: Examine model stability using stratified cross-validation, chronological validation, grouped evaluation, and, where available, independent network data.

9. Operational suitability: Determine the relationship between detection thresholds, attack recall, precision, and false-alarm generation to identify appropriate operating configurations.

3.4 Research Questions

The research objectives are translated into the following research questions.

Research question	**Analytical focus**
RQ1	Which ML classifier provides the most effective binary intrusion detection performance under severe class imbal-

RQ2	To what extent does threshold optimization improve malicious traffic detection?
RQ3	How effectively can ML classifiers distinguish among different cyberattack categories?
RQ4	Which network traffic features contribute most significantly to intrusion detection decisions?
RQ5	Does removing dominant features affect detection performance and model generalization?
RQ6	How stable are the developed classifiers under alternative validation and data-splitting strategies?
RQ7	Can predictive performance, model interpretability, and operational detection requirements be integrated into a unified IDS framework?

3.5 Research Hypotheses

The following hypotheses provide a testable foundation for the experimental investigation.

- H1: Ensemble-based machine-learning classifiers demonstrate significantly different intrusion detection performance under imbalanced network traffic conditions.

- H2: Validation-based threshold optimization improves the attack detection performance of classifiers compared with the default 0.50 threshold.

- H3: Machine-learning classifiers demonstrate significantly different predictive performance across individual cyberattack categories.

- H4: Network traffic features contribute unequally to the prediction of malicious network activities.

- H5: Removing dominant dataset-specific features changes the predictive performance and generalization behaviour of intrusion detection models.

- H6: Model performance obtained through random stratified validation differs from that obtained through chronological or group based validation.

These hypotheses should be tested using appropriate statistical procedures and confidence intervals rather than treating numerical differences alone as evidence of statistical significance.

3.6 Proposed Explainable Machine Learning Architecture

3.6.1 Architecture Overview

The proposed architecture integrates data preprocessing, feature engineering, binary intrusion detection, multiclass attack classification, threshold optimization, explainable AI, and performance evaluation into a unified analytical pipeline.

The architecture follows a modular design so that each component can be independently evaluated while contributing to the overall intrusion detection process.

Proposed Explainable ML-Based IDS Architecture

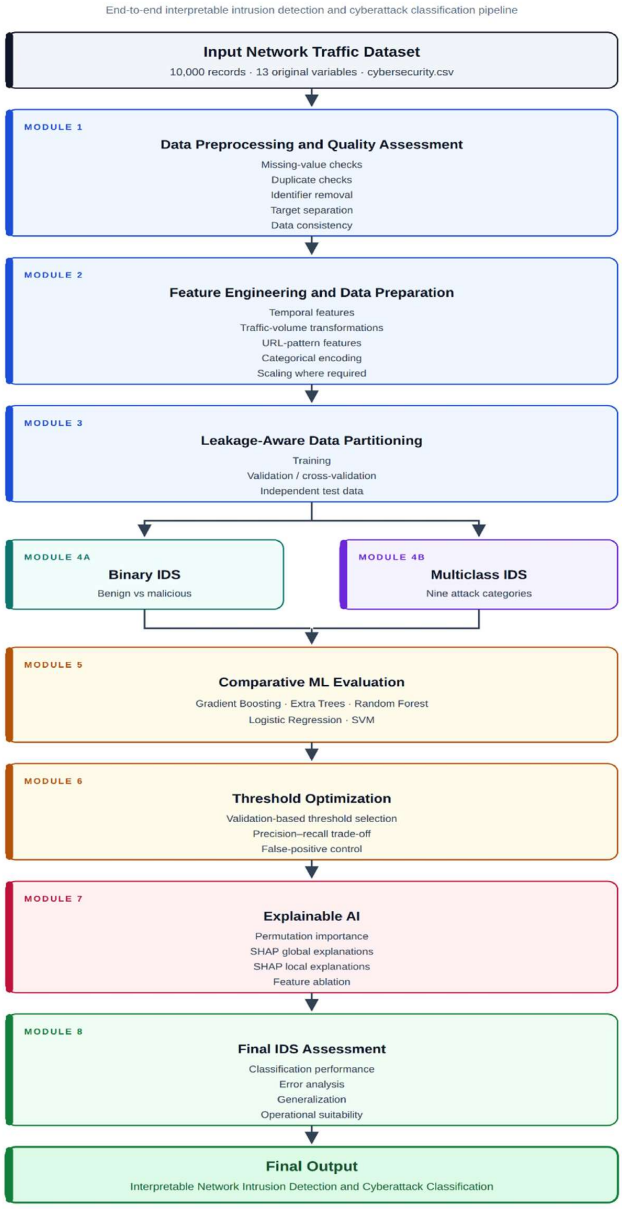


Figure 1. ML-Based IDS Architecture

3.6.2 Functional Description of the Architecture

Module 1: Data acquisition and preprocessing

The first module receives network traffic records from the cybersecurity dataset. The dataset contains 10,000 observations and 13 original variables. Initial data assessment examines the structure of the observations, variable types, missing values, duplicate records, and target distributions.

Raw source and destination IP addresses and raw URLs are excluded as direct identifiers. The attack_type variable is retained as a multiclass target but excluded from the binary predictive features to avoid target leakage.

Module 2: Feature engineering

The feature engineering component extracts informative network traffic characteristics from the available data. The supplied analysis includes temporal information, traffic volume transformations, URL pattern characteristics, and other network related features.

The principal engineered features include:

• hour: temporal information derived from event timing
.
• sent_log: logarithmically transformed bytes sent.

• url_special_count: URL special-character count.

• src_port: source port information.

• is_internal_traffic: internal traffic indicator.

• command_keyword: command-related keyword indicator.

• traversal_pattern: URL or path traversal indicator.

• user_agent: client or user-agent characteristics.

Feature transformations must be estimated using training data only wherever parameters are learned from the dataset.

Module 3: Data partitioning
The dataset is divided into training and testing partitions using stratified sampling. The existing experiment uses a 75:25 split with random state 42.

For the improved experimental design, the training partition is further divided through stratified cross-validation to support model selection, probability calibration, and threshold optimization. The final test set is reserved for independent evaluation.

Module 4: Binary and multiclass classification
Two complementary prediction pathways are implemented.

• Binary classification determines whether a network record represents benign or malicious activity.

• Multiclass classification identifies the specific attack category associated with malicious traffic.

The binary pathway provides an initial security decision, while the multiclass pathway provides more detailed information about the nature of the detected attack.

Module 5: Comparative machine learning evaluation

Five classifiers are evaluated to identify their relative predictive capabilities:

1. Gradient Boosting.

2. Extra Trees.

3. Random Forest.

4. Logistic Regression.

5. Support Vector Machine with an RBF kernel.

The algorithms are compared under identical data-partitioning conditions and appropriate preprocessing pipelines.

Module 6: Threshold optimization
The probability generated by the binary classifier is compared with an optimized decision threshold. Instead of automatically assigning the conventional threshold of 0.50, the framework identifies an alternative threshold from training-derived validation predictions.

This module aims to increase attack detection sensitivity while maintaining an acceptable false-positive rate.
Module 7: Explainable AI

The explainability component identifies the factors influencing the model's predictions through global and local interpretations.

Permutation importance determines the overall contribution of individual features, while SHAP explains the direction and magnitude of feature contributions to predictions. Feature ablation experiments additionally determine whether the model depends excessively on a limited number of variables.

Module 8: Performance and reliability assessment
The final module integrates classification metrics, threshold performance, feature explanations, and validation results. It evaluates whether the proposed model provides reliable intrusion detection and sufficiently interpretable predictions for further cybersecurity investigation.

4. Research Methodology

4.1 Research Design

This research adopts a quantitative, experimental, and comparative machine-learning research design. The methodology investigates the predictive performance of supervised learning algorithms for binary intrusion detection and multiclass cyberattack classification.

The proposed methodology consists of eight sequential stages:

1. Dataset acquisition and characterization

2. Data preprocessing and feature engineering

3. Class distribution and imbalance assessment

4. Binary and multiclass model development

5. Comparative performance evaluation

6. Classification threshold optimization

7. Explainable feature analysis

8. Robustness and generalization assessment

The methodology combines conventional classification evaluation with explainability and operational

threshold analysis to investigate not only whether an attack can be detected but also why the model generates a particular prediction.

4.2 Dataset Description

The experimental investigation uses the supplied cybersecurity.csv dataset

Dataset characteristic	Description
Total observations	10,000
Original variables	13
Benign records	9,600
Malicious records	400
Benign proportion	96%
Malicious proportion	4%
Class imbalance	24:1
Attack categories	9
Missing values	None reported
Training partition	75%
Test partition	25%
Random state	42

The dataset contains both benign and malicious network observations. The nine attack categories include brute force attacks, port scanning, SQL injection, cross site scripting, credential stuffing, distributed denial of service attacks, command injection, exploit attempts, and command and control activity.

Two prediction targets are considered:

• Binary target: Benign or malicious.

• Multiclass target: Benign and the nine attack categories, with a separate attack only experiment to evaluate discrimination among malicious categories.

The multiclass task is particularly challenging because of the small number of observations available for some attack types.

4.3 Data Preprocessing and Feature Engineering

The preprocessing stage aims to improve the consistency and suitability of network traffic observations for machine learning algorithms.

The following operations are proposed:

Step 1: Data quality assessment

- Examine missing values and duplicate observations.
- Verify variable types and data consistency.
- Identify invalid or anomalous feature values.
- Examine the frequency distribution of binary and multiclass targets.

Step 2: Identifier removal
Raw IP addresses and URLs are excluded as direct identifiers to reduce the possibility of memorizing specific network entities.

Step 3: Feature transformation
Traffic volume variables are transformed where appropriate to reduce skewness. Temporal features and URL-pattern indicators are retained as candidate predictors.

Step 4: Categorical encoding
Categorical variables are encoded using transformations fitted exclusively on training data. High cardinality variables require particular attention to prevent excessive memorization.

Step 5: Scaling
Standardization is applied to algorithms sensitive to feature magnitude, particularly Logistic Regression and SVM. Tree-based algorithms can generally operate on unscaled numerical variables.

Step 6: Leakage prevention
Target variables and features that directly reveal the attack label are excluded from the predictor matrix. All preprocessing operations that estimate parameters from observations are fitted inside the training pipeline.

4.4 Class Imbalance Handling

The dataset's 24:1 benign-to-malicious ratio creates a substantial challenge for classification.

Three approaches should be investigated:

1. Baseline learning: Train classifiers using the original class distribution.

2. Class weighted learning: Apply class weights to increase the importance of malicious observations during training.

3. Training only resampling: Investigate suitable oversampling techniques, including SMOTE where appropriate.

Resampling must be performed separately within each training fold rather than before data splitting. This prevents synthetic or duplicated minority observations from leaking into validation or testing data.

The original distribution must be retained in the independent test set to preserve realistic evaluation conditions.

4.5 Binary Intrusion Detection Models

Five machine learning classifiers are included in the comparative experiment.

Algorithm	Model characteristics	Research purpose
Gradient Boosting	Sequential ensemble learning	Identify complex nonlinear relationships
Extra Trees	Randomized tree ensemble	Evaluate ensemble diversity
Random Forest	Bootstrap-based tree ensemble	Assess robust nonlinear classification
Logistic Regression	Linear probabilistic classifier	Establish an interpretable baseline
SVM (RBF)	Kernel-based classification	Investigate nonlinear decision boundaries

The models are trained using the same training partition and evaluated on the same independent test partition.

Hyperparameter tuning should be conducted using training data only, preferably through stratified cross-validation.

The objective is to determine which classifier provides the most suitable combination of discrimination capability, attack recall, precision, and false positive control.

4.6 Binary Classification Metrics

The evaluation considers six primary metrics, supplemented by operational error measures.

Accuracy

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

Precision

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

Recall

$$\text{Recall} = \frac{TP}{TP + FP}$$

F1-score

$$F1 = 2 \frac{\text{Precision X Recall}}{\text{Precision} + \text{Recall}}$$

False - positive rate

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

False - Negative Rate

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

ROC-AUC and PR-AUC are also calculated to evaluate discrimination performance across different decision thresholds.

MCC is recommended as an additional measure because it accounts for all four confusion-matrix components and remains informative under class imbalance.

4.7 Threshold Optimization Methodology

Threshold optimization is an important component of the proposed framework.

For a binary classifier, the predicted probability of malicious activity is represented as:

$$\boldsymbol{P(y = 1 \mid X) = p}$$

The final classification is determined by:

$$\hat{\boldsymbol{y}} = \begin{cases} \mathbf{1}, & \boldsymbol{p \geq t^*} \\ \mathbf{0}, & \boldsymbol{p < t^*} \end{cases}$$

where t^* represents the optimized probability threshold.

The proposed optimization procedure involves the following stages:

1. Train the classification model using the training partition.

2. Generate out-of-fold probability predictions using stratified cross-validation.

3. Evaluate candidate thresholds using validation predictions.

4. Select a threshold according to the predefined optimization objective.

5. Refit the selected model using the complete training partition.

6. Apply the selected threshold to the held-out test probabilities.

7. Evaluate the final classification performance.

F1-based optimization

$$t^* = \arg\max_{t \in T} F1(t)$$

where T represents the set of candidate probability thresholds.

Constrained optimization (recommended for operational use)

For operational applications, a constrained optimization alternative is recommended:

$$\max_{t} \text{Recall}(t)$$

subject to

$$\mathrm{FPR}(t) \leq \alpha$$

where α is a predefined acceptable false positive rate limit.

This approach allows security requirements to determine the operating point rather than relying exclusively on a generic statistical objective.

The current experimental findings demonstrate the relevance of this procedure. Gradient Boosting achieved an optimized threshold of 0.169, resulting in precision of 0.537, recall of 0.580, and F1-score of 0.558.

4.8 Multiclass Cyberattack Classification

The multiclass experiment investigates whether the proposed framework can distinguish different types of malicious network activities.

Two complementary settings are employed.

Experiment A: All-record classification

All network observations are included, with benign traffic and the nine attack categories represented as separate classes.

Experiment B: Attack-only classification

Benign observations are removed, and the model is trained to distinguish among the nine malicious attack categories.

The two experiments provide complementary information. The first examines the complete classification task, while the second investigates the model's ability to distinguish attack types without the overwhelming influence of benign observations.

The principal evaluation metrics are:

- Multiclass accuracy.
- Macro-averaged precision.
- Macro-averaged recall.
- Macro-F1.
- Weighted-F1.
- Class-specific precision and recall.
- Confusion matrix.

Macro-F1 is particularly important because it assigns equal weight to each class, whereas weighted-F1 reflects the class distribution.

4.9 Explainable Artificial Intelligence Methodology

The explainability stage is designed to investigate both global feature importance and individual prediction behaviour.

Three complementary techniques are proposed.

A. Permutation Feature Importance

Permutation importance evaluates the contribution of a feature by randomly rearranging its values and measuring the resulting reduction in predictive performance.

Feature Importance via Permutation

$$I_j = S(X'y) - S(X_{\pi}(f)'y)$$

where:

- I_j = importance of feature j
- $S(X'y)$ original model performance
- $(X_{p(j)})$ dataset with feature permuted

Average precision should be retained as the primary scoring metric for binary intrusion detection because of the minority class problem.

B. SHAP-Based Global Explainability

SHAP (SHapley Additive exPlanations) is proposed to quantify how individual features contribute to the model prediction.

The SHAP representation is:

$$f(x) = \phi_0 + \sum_{j=1}^{m} \phi_j$$

where:

- $f(x)$ = model prediction in the explained output space
- φ_0= baseline model output
- φ_j = contribution of feature j
- m= number of explanatory features

The mean absolute SHAP value is calculated as:

$$I_j^{SHAP} = \frac{1}{N} \sum_{i=1}^{N} | \phi_{ij} |$$

This provides a global ranking of the features contributing to the model's predictions.

C. Local SHAP Explainability

Local explanations are generated for selected benign and malicious observations.

The analysis investigates:

- Why a particular network record was classified as malicious.
- Which features increased the predicted attack probability.
- Which features reduced the predicted attack probability.
- Whether the same features influence different attack categories.
- Whether false positive and false negative cases exhibit distinguishable explanation patterns.

For tree based models, TreeSHAP can be used where supported. The appropriate SHAP explainer must be selected according to the model and its prediction interface.

Important distinction: The supplied experiment currently reports permutation importance, not completed SHAP analysis. SHAP is therefore a proposed extension and its numerical findings must be generated and evaluated separately.

4.10 Feature Ablation and Robustness Analysis

The strong dominance of user_agent requires additional investigation.

The proposed experiments are:

Experiment	Feature configuration	Purpose
A1	All available predictors	Establish baseline
A2	Remove user_agent	Measure dependence on client-identification patterns
A3	Remove src_port	Examine port-related contribution
A4	Remove top-ranked features	Examine dependence on dominant predictors
A5	Retain selected traffic and temporal features	Assess alternative feature representations

Each experiment should use identical training and testing partitions.

The resulting differences in PR-AUC, recall, F1-score, and false positive rate will help determine whether the model learns broader network behaviour or relies disproportionately on a few dataset-specific features.

4.11 Cross-Validation and Generalization

The existing experiment uses five-fold stratified cross validation for Gradient Boosting and reports relatively stable performance across folds.

The proposed extended validation strategy includes:

1. Stratified five-fold cross-validation.

2. Repeated stratified cross-validation.

3. Chronological holdout evaluation, if reliable timestamps are available.

4. Grouped evaluation based on source, host, or session, if these identifiers are available.

5. Independent network testing, where an external dataset can be obtained.

The purpose is to investigate whether the predictive performance remains stable when observations from different periods or network entities are evaluated independently.

5. Experimental Design

5.1 Experimental Setup

The experimental design is organized into seven experiments. Each experiment addresses a specific research objective and contributes evidence toward evaluating the proposed explainable intrusion detection framework.

Experiment	Research activity	Expected analytical output
E1	Dataset characterization	Descriptive statistics, class distribution, feature analysis
E2	Binary intrusion detection	Comparative performance of five ML classifiers
E3	Threshold optimization	Optimal thresholds, confusion matrices, FP/FN analysis
E4	Multiclass attack classification	Attack-specific classification performance
E5	Explainable ML	Permutation importance, SHAP, local explanations
E6	Feature ablation	Feature dependence and robustness results
E7	Generalization assessment	Cross-validation, chronological and grouped validation

5.2 Experimental Configuration

The following configuration provides a reproducible experimental foundation.

Parameter	Proposed configuration
Programming environment	Python
Data processing	Pandas, NumPy
ML framework	Scikit-learn
Explainability	SHAP
Visualization	Matplotlib, Seaborn
Dataset	cybersecurity.csv
Total records	10,000

Training proportion	75%
Testing proportion	25%
Random state	42
Cross-validation	Stratified 5-fold
Binary models	Five classifiers
Multiclass models	Logistic Regression, Extra Trees, Random Forest
Binary evaluation	Accuracy, precision, recall, F1, ROC-AUC, PR-AUC, MCC
Multiclass evaluation	Accuracy, macro-F1, weighted-F1, class-specific recall
Explainability	Permutation importance, SHAP
Threshold strategy	Validation-based optimization

The original random split and baseline model configurations should first be reproduced. Subsequent experiments should use a consistent, leakage-aware evaluation protocol.

5.3 Experimental Workflow

The complete experimental procedure is presented below.

5.4 Experiment E1: Dataset Characterization

This experiment examines the structural and statistical properties of the cybersecurity dataset.
The analysis includes:

- Distribution of benign and malicious records.
- Distribution of the nine attack categories.
- Feature descriptive statistics.
- Numerical feature distributions.
- Categorical feature distributions.
- Correlation analysis.
- Missing-value and duplicate assessment.
- Class imbalance measurement.

The primary purpose is to identify potential data quality issues and understand whether the dataset provides sufficient representation for binary and multiclass classification.

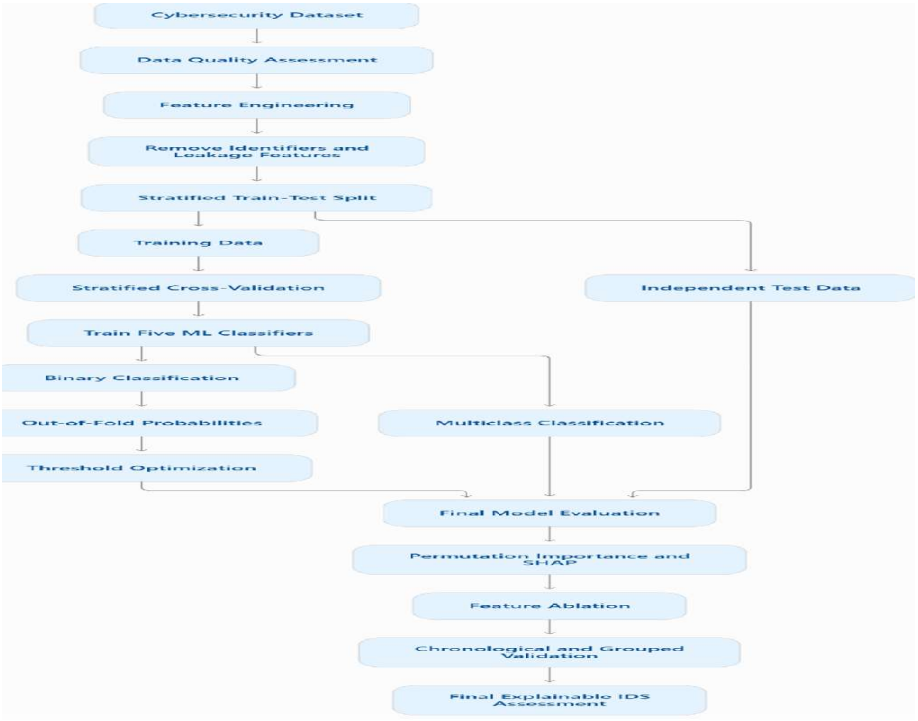


Figure 2. Experimental Flow

Expected contribution: Establish the statistical characteristics and limitations of the experimental dataset before model development.

5.5 Experiment E2: Comparative Binary Intrusion Detection

Five machine-learning classifiers are evaluated under identical data-partitioning conditions.

The experiment investigates:

- Attack detection capability.
- False-positive generation.
- False-negative generation.
- Precision–recall trade-offs.
- Discriminatory performance.
- General classification stability.

The existing experimental results provide the initial benchmark.

Model	Accuracy	Recall	F1	ROC-AUC	PR-AUC
Gradient Boosting	0.968	0.260	0.397	0.896	0.603
Extra Trees	0.964	0.280	0.384	0.767	0.378
Random Forest	0.967	0.190	0.314	0.891	0.518

Logistic Regression	0.760	0.670	0.670	0.791	0.364
SVM	0.960	0.010	0.010	0.760	0.158

5.6 Experiment E3: Threshold Optimization

This experiment investigates the influence of alternative classification thresholds on intrusion detection performance.

The existing results are:

Model	**Optimized threshold**	**Precision**	**Recall**	**F1**
Gradient Boosting	0.169	0.537	0.580	0.558
Random Forest	0.222	0.459	0.560	0.505
Extra Trees	0.464	0.516	0.320	0.395

The optimized Gradient Boosting model provides the strongest F1-score and recall among the evaluated threshold optimized models.

The extended experiment should additionally investigate:

- Recall at predefined false positive rate limits.
- Precision–recall trade-off curves.
- Threshold stability across cross-validation folds.
- Probability calibration.
- Threshold sensitivity to changes in attack prevalence.

This would provide a more comprehensive basis for selecting an operational threshold.

5.7 Experiment E4: Multiclass Cyberattack Classification

This experiment evaluates the ability of machine learning classifiers to identify individual attack categories. The existing results demonstrate the importance of evaluating both macro-F1 and weighted-F1.

Model	**Experiment**	**Accuracy**	**Macro-F1**	**Weighted-F1**
Extra Trees	All records	0.963	0.209	0.948
Random Forest	All records	0.961	0.142	0.942
Logistic Regression	All records	0.339	0.128	0.487
Logistic Regression	Attack only	0.480	0.397	0.513

Extra Trees	Attack only	0.590	0.350	0.517
Random Forest	Attack only	0.590	0.340	0.501

The difference between macro-F1 and weighted-F1 demonstrates that overall performance can conceal poor recognition of minority attack categories.

The extended experiment should therefore include class specific confusion matrices, per-class precision, recall, and F1-score, together with confidence intervals where feasible.

5.8 Experiment E5: Explainable AI Evaluation

This experiment investigates the factors responsible for intrusion detection decisions.

The analysis should produce:

1. Global permutation feature-importance ranking.

2. SHAP summary plot.

3. SHAP mean absolute feature-importance plot.

4. SHAP dependence plots for influential features.

5. Local SHAP waterfall plots for individual malicious records.

6. Local SHAP explanations for false-positive cases.

7. Local SHAP explanations for false-negative cases.

Particular attention should be given to user_agent, src_port, sent_log, url_special_count, and hour.

The current permutation importance results indicate that user_agent has an importance of 0.2836, compared with 0.0054 for src_port. This substantial difference motivates a more detailed examination of feature dependence.

5.9 Experiment E6: Feature Ablation and Robustness

The objective is to determine whether the model's predictions remain reliable when influential features are removed.

The baseline model is compared against models trained with alternative feature subsets.

The main comparison is:

- All features.

- Without user_agent.

- Without the top two influential features.

- Selected network traffic features only.

- Temporal and URL-pattern features.

The comparison should consider PR-AUC, recall, F1-score, false-positive rate, and macro-F1.

A major research contribution would be demonstrating whether the model retains meaningful attack detection capability without depending excessively on client-identification patterns.

5.10 Experiment E7: Generalization and Stability
The final experiment evaluates the robustness of the proposed framework.

Three complementary validation settings are recommended.

Random stratified validation: Measures performance under the original sampling strategy.

Chronological validation: Trains on earlier observations and tests on later observations, if valid timestamps and sufficient observations are available.

Grouped validation: Separates related observations using source, host, or session identifiers, where available.

If an independent network dataset can be obtained, external evaluation should be conducted without retraining on its test observations.

The purpose is to establish whether the proposed IDS learns transferable network intrusion characteristics rather than memorizing patterns specific to the original dataset.

5.11 Proposed Research Contributions
The proposed study is expected to make the following contributions.
1. Integrated explainable IDS architecture
A unified framework combining binary intrusion detection, multiclass cyberattack classification, threshold optimization, and explainable machine learning.

2. Class-sensitive intrusion detection evaluation
A comprehensive evaluation methodology that addresses the limitations of accuracy-based assessment under severe class imbalance.

3. Threshold-aware attack detection
A validation-based threshold optimization strategy that considers attack sensitivity and false alarm generation.

4. Global and local feature interpretation
Integration of permutation importance and SHAP explanations to investigate the contribution of network traffic characteristics to intrusion detection decisions.

5. Feature-dependence investigation
A systematic feature-ablation strategy to evaluate the influence of dominant predictors and investigate potential dataset-specific learning.

6. Multiclass attack recognition
Separate assessment of binary intrusion detection and attack-category classification to distinguish overall malicious activity detection from attack identification.

7. Generalization-oriented validation
An extended experimental framework incorporating chronological, grouped, and independent-network evaluation to investigate model transferability.

6. Results

6.1 Binary intrusion detection

Table 1 reports test-set performance at the conventional 0.50 probability threshold. Accuracy alone is insufficient under the 4% attack prevalence; recall, F1 and PR-AUC are more informative.

Model	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC
Gradient Boosting	0.968	0.839	0.260	0.397	0.896	0.603
Extra Trees	0.964	0.609	0.280	0.384	0.767	0.378
Random Forest	0.967	0.905	0.190	0.314	0.891	0.518
Logistic Regression	0.760	0.106	0.670	0.183	0.791	0.364
SVM (RBF)	0.960	1.000	0.010	0.020	0.760	0.158

Table 1. Test Performance Results

Table 1 presents the comparative performance of five machine learning classifiers Gradient Boosting, Extra Trees, Random Forest, Logistic Regression, and Support Vector Machine (SVM with RBF kernel) for binary intrusion detection. The models were evaluated using accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC on the held-out test dataset. The evaluation considers the substantial class imbalance, in which malicious traffic accounts for only 4% of all network observations.

Gradient Boosting achieved an accuracy of 0.968, precision of 0.839, recall of 0.260, F1-score of 0.397, ROC-AUC of 0.896, and PR-AUC of 0.603. Extra Trees produced an accuracy of 0.964 and recall of 0.280, while Random Forest achieved the highest precision of 0.905 among the ensemble models. Logistic Regression demonstrated substantially higher recall of 0.670 but lower precision of 0.106. SVM achieved perfect precision of 1.000 but an extremely low recall of 0.010.

The results demonstrate that the classification algorithms exhibit different detection characteristics under imbalanced network traffic conditions. Gradient Boosting achieved the highest F1-score (0.397) and PR-AUC (0.603), indicating the strongest overall balance between precision and recall among the evaluated models at the default classification threshold. Random Forest achieved the highest precision (0.905), suggesting that its positive predictions were highly reliable, although its recall of 0.190 indicates that many malicious events remained undetected.

In contrast, Logistic Regression achieved the highest recall (0.670), demonstrating its ability to identify a larger proportion of malicious records. However, its precision of 0.106 indicates a substantial false-alarm burden. Similarly, the SVM classifier achieved perfect precision but detected only 1% of actual attacks, making it unsuitable for operational intrusion detection at the default threshold.

These findings demonstrate that accuracy alone is insufficient for evaluating intrusion detection systems when benign traffic dominates the dataset. The results support the use of recall, F1-score, and PR-AUC as complementary evaluation measures. Gradient Boosting provides the strongest default-threshold performance, although its relatively low recall indicates the need for decision-threshold optimization.

6.2 Threshold optimization

For Gradient Boosting, Random Forest and Extra Trees, the decision threshold was selected using out-of-fold training probabilities only; the selected threshold was then applied once to the held-out test set. This avoids tuning directly on the test labels.

Model	Threshold	Precision	Recall	F1	FP	FN
Gradient Boosting	0.169	0.537	0.580	0.558	50	42
Random Forest	0.222	0.459	0.560	0.505	66	44
Extra Trees	0.464	0.516	0.320	0.395	30	68

Table 2. Threshold Optimisation

Table 2 presents the results of decision threshold optimization for Gradient Boosting, Random Forest, and Extra Trees. Instead of using the conventional probability threshold of 0.50, alternative thresholds were selected using out of fold training predictions and subsequently evaluated on the independent test dataset. The table reports the selected threshold, precision, recall, F1-score, false positives (FP), and false negatives (FN).

Gradient Boosting achieved the best threshold-optimized performance at a threshold of 0.169, obtaining precision of 0.537, recall of 0.580, and an F1-score of 0.558, with 50 false positives and 42 false negatives. Random Forest achieved an F1-score of 0.505 at a threshold of 0.222, while Extra Trees obtained an F1-score of 0.395 at a threshold of 0.464.

The results demonstrate that classification thresholds substantially influence intrusion detection performance. Gradient Boosting showed a considerable improvement in recall, increasing from 0.260 at the default threshold to 0.580 after optimization. Its F1-score also increased from 0.397 to 0.558, indicating a better balance between detecting attacks and controlling false alarms.

Random Forest similarly improved its recall from 0.190 to 0.560, with its F1-score increasing from 0.314 to 0.505. Extra Trees demonstrated a smaller improvement in F1-score.

The reduction in the classification threshold makes the models more sensitive to potential malicious activity. However, this improvement introduces additional false positives, which may increase the workload of security analysts. Gradient Boosting therefore provides the most favorable operating point among the evaluated threshold-optimized models.

From an operational perspective, the optimized Gradient Boosting classifier offers a more appropriate configuration when failing to identify malicious network activity is considered more costly than generating additional security alerts. The selected threshold was determined without directly optimizing against test labels, supporting a more reliable assessment of generalization.

The best threshold-tuned test result was obtained by Gradient Boosting (threshold=0.169), with precision=0.537, recall=0.580, F1=0.558, false positives=50, and false negatives=42. This operating point is more suitable when missing an attack is costly, although it increases false alarms.

6.3 Cross-validation

Five-fold stratified cross-validation on the training partition for Gradient Boosting yielded the following mean ± standard deviation:

accuracy: 0.971 ± 0.001
precision: 0.899 ± 0.073

recall: 0.307 ± 0.020
f1: 0.456 ± 0.025
roc_auc: 0.917 ± 0.010
pr_auc: 0.662 ± 0.032

Table 3. Cross validation

Table 3 reports the five fold stratified cross validation performance of the Gradient Boosting classifier on the training partition. The evaluation includes accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC, expressed as mean values with their corresponding standard deviations.

The model achieved mean accuracy of 0.971 ± 0.001, precision of 0.899 ± 0.073, recall of 0.307 ± 0.020, F1-score of 0.456 ± 0.025, ROC-AUC of 0.917 ± 0.010, and PR-AUC of 0.662 ± 0.032.

The cross validation results indicate that Gradient Boosting maintains relatively consistent classification performance across the five training folds. The low standard deviation in accuracy (0.001) and ROC-AUC (0.010) suggests that the model's overall discrimination performance is relatively stable across different stratified training subsets.

The mean PR-AUC of 0.662 indicates that the classifier demonstrates useful discriminatory capability despite the substantial imbalance between benign and malicious network traffic. However, the mean recall of 0.307 reveals that the model identifies only a relatively small proportion of malicious events at the evaluated operating threshold.

The standard deviation in precision (0.073) is comparatively larger than those of several other metrics, indicating some variability in the reliability of positive predictions across folds. Overall, the cross-validation results support the stability of Gradient Boosting's ranking performance while highlighting the need for improved attack sensitivity.

6.4 Multiclass attack classification

Multiclass classification across all records is dominated by the benign class (96%). Consequently, high accuracy/weighted F1 can coexist with poor detection of rare attack types. The attack-only experiment removes benign records and evaluates discrimination among the nine attack categories, but the 400 malicious records are still sparse and highly imbalanced.

All-record multiclass results:

Extra Trees: accuracy=0.963, macro-F1=0.209, weighted-F1=0.948
Random Forest: accuracy=0.961, macro-F1=0.142, weighted-F1=0.942
Logistic Regression: accuracy=0.339, macro-F1=0.128, weighted-F1=0.487
Attack-only results:
Logistic Regression: accuracy=0.480, macro-F1=0.397, weighted-F1=0.513

Extra Trees: accuracy=0.590, macro-F1=0.350, weighted-F1=0.517
Random Forest: accuracy=0.590, macro-F1=0.340, weighted-F1=0.501

Table 4. Multiclass Attack Classification

Table 4 presents the performance of Logistic Regression, Extra Trees, and Random Forest for multiclass attack classification under two experimental settings: classification across all records and classification restricted to malicious records.

In the all-record experiment, Extra Trees achieved an accuracy of 0.963, macro-F1 of 0.209, and weighted-F1 of 0.948. Random Forest obtained an accuracy of 0.961, macro-F1 of 0.142, and weighted-F1 of 0.942. Logistic Regression demonstrated substantially lower performance, with an accuracy of 0.339, macro-F1 of 0.128, and weighted-F1 of 0.487.

In the attack-only experiment, Logistic Regression achieved an accuracy of 0.480, macro-F1 of 0.397, and weighted-F1 of 0.513. Extra Trees and Random Forest both achieved an accuracy of 0.590, with macro-F1 values of 0.350 and 0.340, respectively.

The all-record results reveal that high accuracy and weighted-F1 scores do not necessarily indicate effective identification of individual attack categories. Extra Trees and Random Forest achieved accuracy above 0.96 and weighted-F1 above 0.94, primarily reflecting the dominance of benign observations. However, their considerably lower macro-F1 scores indicate poor and uneven performance across the individual classes.

The attack-only experiment provides a more focused assessment of the ability to distinguish between different malicious activities. Removing benign records reduces the influence of the majority class and reveals the difficulty of discriminating between attack categories. Extra Trees and Random Forest achieved the highest attack-only accuracy of 0.590, whereas Logistic Regression achieved the highest macro-F1 of 0.397.

The difference between accuracy and macro-F1 indicates that correct classification is not distributed uniformly across the attack categories. This is particularly important because some attack types contain very few observations. Consequently, the multiclass classification results should be interpreted cautiously, and additional examples from minority attack categories are required to improve the reliability of the models.

6.5 Feature importance and interpretation

Permutation importance was computed on the held-out test set using average precision as the scoring metric. The leading feature was user_agent, followed by source port and log-transformed bytes sent. This concentration suggests that user-agent patterns may encode strong dataset-specific attack signatures. It should be tested for robustness across time periods, devices and network environments before deployment.

user_agent: 0.2836 ± 0.0049
src_port: 0.0054 ± 0.0010
sent_log: 0.0024 ± 0.0072
url_special_count: 0.0015 ± 0.0039
hour: 0.0014 ± 0.0024
is_internal_traffic: 0.0000 ± 0.0000

command_keyword: 0.0000 ± 0.0000
traversal_pattern: 0.0000 ± 0.0000

Table 5. Feature Importance

Table 5 reports permutation based feature importance for the Gradient Boosting intrusion detection model. Importance was evaluated on the held out test dataset using average precision as the scoring metric. The results are expressed as the mean decrease in average precision following feature permutation, together with the corresponding standard deviation.

The user-agent variable exhibited the highest importance, with a mean value of 0.2836 ± 0.0049. Source port was the second most influential feature, with an importance of 0.0054 ± 0.0010. Log-transformed bytes sent contributed 0.0024 ± 0.0072, followed by URL special-character count at 0.0015 ± 0.0039 and hour of occurrence at 0.0014 ± 0.0024. Internal traffic status, command keyword, and traversal pattern exhibited zero reported importance.

The results demonstrate that user-agent information contributes substantially more to the model's intrusion detection performance than the other evaluated features. The large difference between user agent importance and the remaining variables suggests that specific client identification patterns may encode strong dataset-specific signatures associated with malicious network activity.

Source port and traffic volume also contribute to classification, although their relative importance is considerably smaller. The non-zero contribution of URL special character count and hour indicates that URL characteristics and temporal patterns may provide supplementary information for identifying suspicious network activity.

The zero importance values for internal traffic status, command keyword, and traversal pattern indicate that permuting these features did not measurably reduce average precision in this experiment.

However, the strong dominance of user-agent information raises an important generalization concern. The model may be relying heavily on patterns associated with the particular traffic collection rather than learning broadly transferable malicious network behaviours. Therefore, additional experiments should evaluate performance after removing user agent information and testing on independent network environments.

6.6 Limitations and recommendations

The attack prevalence is only 4%; report PR-AUC, recall, F1 and false-negative counts alongside accuracy.

Several attack categories have very few observations (the smallest has 10 records), making multiclass estimates unstable. Collect substantially more examples per attack class.

The strong user-agent importance may indicate a synthetic or collection specific signature. Validate with user agent removed, temporal holdout, and external traffic.

Random row splitting may place temporally adjacent or related events in both partitions. A chronological holdout and grouped split by source, host or session are needed for stronger generalization evidence.

Tune thresholds to operational costs and report recall at a fixed false-positive rate; calibrate probabilities before production use.

This is an offline benchmark, not evidence of real-time production efficacy. Validate on an independent network and monitor drift.

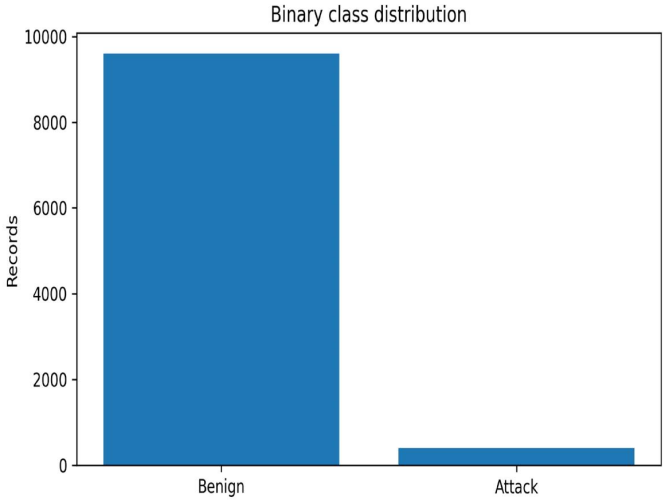


Figure 3. Binary class distribution showing the strong imbalance between benign and malicious network records

Figure 3 illustrates the distribution of benign and malicious network traffic records in the cybersecurity dataset. The visualization compares the number of observations belonging to the two binary target classes. The dataset contains 9,600 benign records, representing 96% of the observations, and 400 malicious records, accounting for the remaining 4%.

The figure clearly demonstrates the substantial class imbalance in the dataset. Benign traffic dominates the observations, whereas malicious traffic constitutes only a small proportion of the network events. This imbalance presents an important challenge for supervised machine-learning algorithms because conventional classifiers may favor the majority class and achieve high accuracy without effectively detecting malicious activities.

The 24:1 ratio between benign and malicious observations highlights the importance of applying suitable evaluation strategies, including precision, recall, F1-score, PR-AUC, and MCC. The figure also provides the justification for investigating class-sensitive learning strategies and classification-threshold optimization.

From a cybersecurity perspective, the minority representation of malicious events emphasizes the importance of identifying attack patterns accurately rather than relying exclusively on overall classification accuracy.

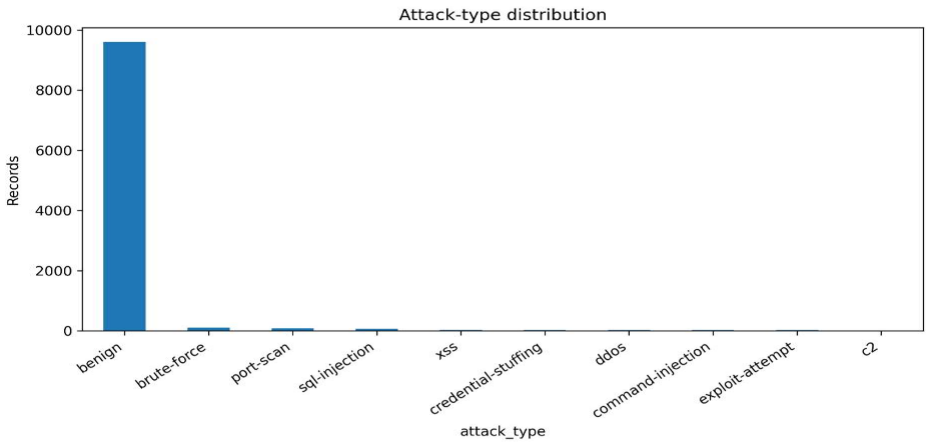


Figure 4. Distribution of the nine malicious attack categories; rare classes limit reliable multiclass inference

Figure 4 presents the frequency distribution of the nine malicious attack categories represented in the dataset. The visualization illustrates the relative occurrence of different attack types, including brute force attacks, port scanning, SQL injection, cross-site scripting, credential stuffing, distributed denial of service attacks, command injection, exploit attempts, and command-and-control activity.

The distribution indicates that the attack categories are unevenly represented, with some categories containing substantially fewer observations than others. The smallest attack category contains only ten records.

The figure demonstrates that malicious network activity is heterogeneous and consists of multiple attack categories with different occurrence frequencies. The uneven distribution indicates that the classification task involves not only distinguishing malicious traffic from benign traffic but also identifying specific attack behaviours under limited sample availability.

Attack categories with relatively large numbers of observations provide more training information and may therefore be easier for machine-learning algorithms to identify. In contrast, categories with very few observations may experience inadequate representation during model training, resulting in lower classification reliability.

The small number of examples for certain attack categories also increases the uncertainty associated with multiclass evaluation. Consequently, macro-F1 and class-specific recall should be emphasized when evaluating attack classification models. Additional data collection for rare attack categories is necessary to improve the robustness of multiclass intrusion detection.

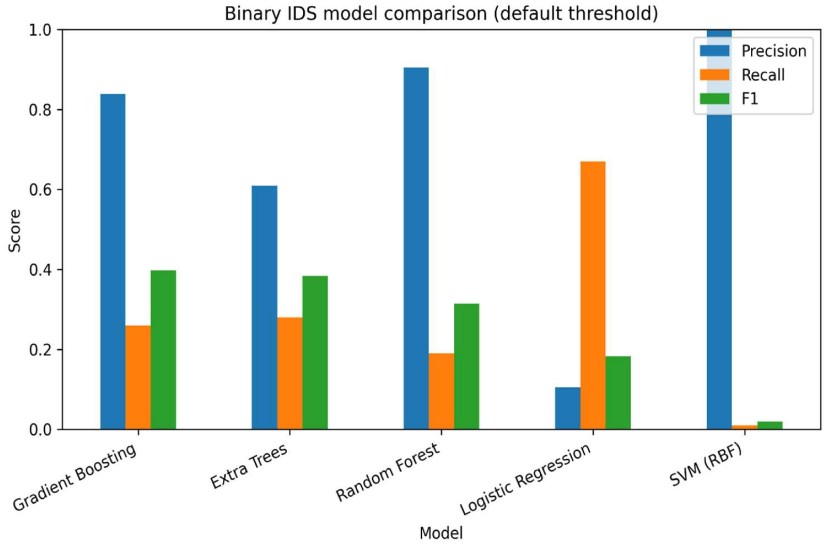


Figure 5. Precision, recall and F1 comparison of binary classifiers at the default 0.50 threshold

Figure 5 compares the precision, recall, and F1-score of the five binary intrusion detection classifiers using the conventional probability threshold of 0.50. The visualization provides a comparative assessment of the ability of each algorithm to identify malicious network traffic while controlling false-positive predictions.

The figure includes Gradient Boosting, Extra Trees, Random Forest, Logistic Regression, and SVM. Gradient Boosting achieved an F1-score of 0.397, while Logistic Regression achieved the highest recall of 0.670. Random Forest obtained the highest precision of 0.905, whereas SVM achieved precision of 1.000 but recall of only 0.010.

The figure illustrates the trade-off between precision and recall across the evaluated classifiers. Logistic

Regression demonstrates a strong preference for identifying malicious records, resulting in higher recall but substantially lower precision. This indicates that the model identifies a greater proportion of attacks at the cost of generating more false alarms.

Conversely, Random Forest and SVM demonstrate greater selectivity in their positive predictions. Although these models achieve relatively high precision, their low recall indicates that they fail to identify a considerable proportion of malicious network events.

Gradient Boosting provides the strongest balance between precision and recall, as reflected by its highest F1-score among the evaluated classifiers. These results suggest that ensemble learning approaches offer promising performance for binary intrusion detection, although threshold optimization remains necessary to achieve operationally acceptable detection sensitivity.

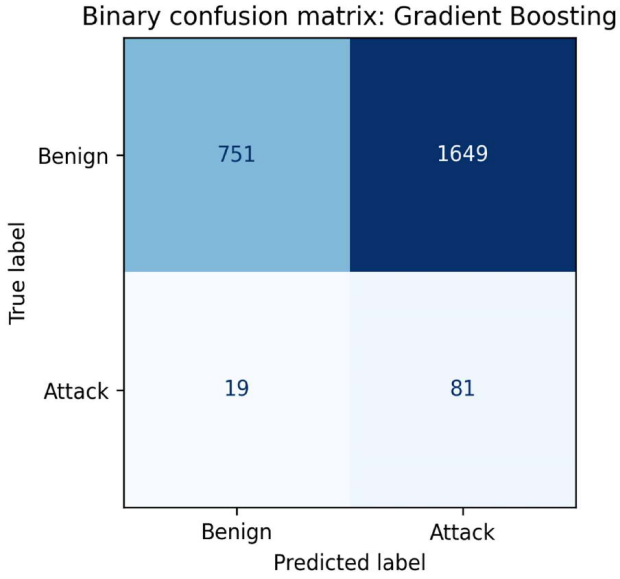


Figure 6. Confusion matrix for the best threshold tuned binary classifier on the held-out test set

Figure 6 presents the confusion matrix of the threshold optimized Gradient Boosting classifier evaluated on the held out test dataset. The selected probability threshold is 0.169. The matrix compares actual class labels with predicted classifications, distinguishing true positives, true negatives, false positives, and false negatives. At this operating point, the classifier achieved a precision of 0.537, recall of 0.580, and F1-score of 0.558. It generated 50 false-positive predictions and 42 false-negative predictions.

The confusion matrix provides a direct assessment of the classification errors generated by the optimized intrusion detection model. The recall of 0.580 indicates that the classifier successfully identifies 58% of the malicious traffic records in the test dataset. However, the remaining malicious observations are classified as benign, resulting in 42 false negatives.

The 50 false positives indicate that some benign network events are incorrectly identified as attacks. Although these predictions may increase the workload associated with security alert investigation, the optimized model offers a more favorable balance between missed attacks and false alarms than the default-threshold configuration.

The improvement in F1-score from 0.397 to 0.558 demonstrates the value of threshold optimization for imbalanced intrusion detection. Nevertheless, the remaining false negatives indicate that further improvements in detection sensitivity are necessary before considering operational deployment.

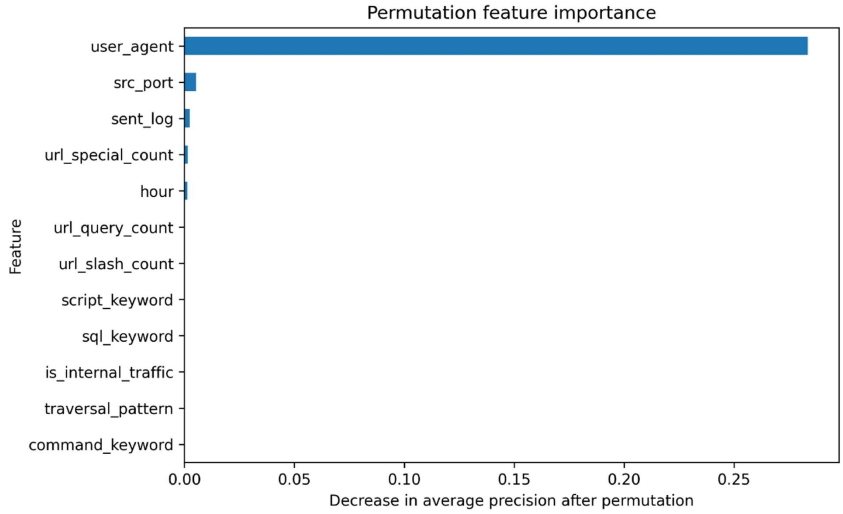


Figure 7. Test-set permutation importance based on average precision, highlighting the influence of user-agent and traffic features

Figure 7 illustrates the relative importance of input features in the Gradient Boosting intrusion detection model using permutation importance. The analysis evaluates the reduction in average precision when individual features are randomly permuted within the test dataset.

The visualization identifies user-agent information as the most influential feature, followed by source port, log-transformed bytes sent, URL special-character count, and hour. Other variables, including internal traffic status, command keyword, and traversal pattern, exhibit negligible importance.

The figure demonstrates a pronounced concentration of predictive information in the user agent variable. Its substantially greater importance suggests that the model has learned distinctive client identification patterns associated with malicious network events in the evaluated dataset.

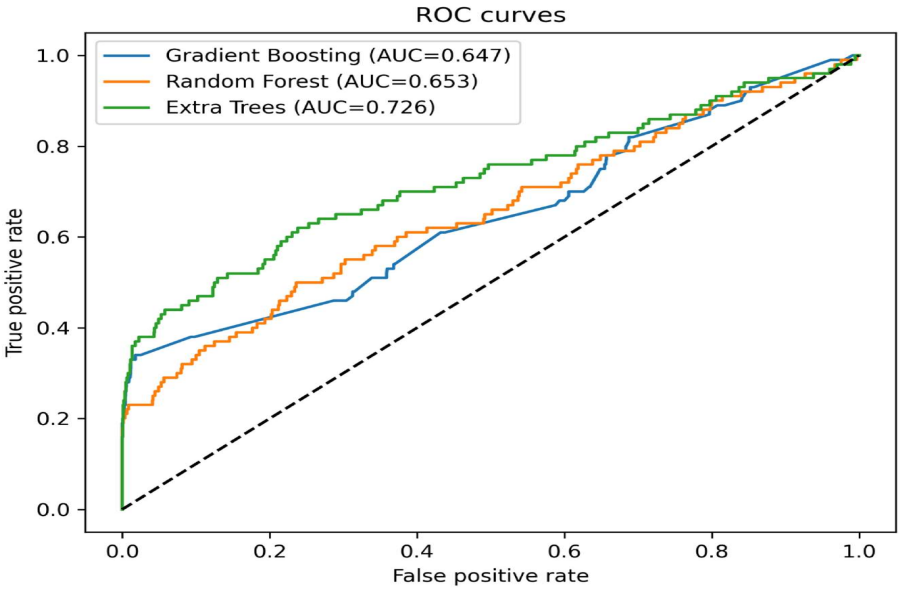


Figure 8. Receiver operating characteristic curves for the leading ensemble classifiers

The contribution of source port and traffic volume indicates that network communication characteristics also provide information relevant to distinguishing malicious from benign traffic. URL special character patterns and temporal information contribute additional, although relatively limited, predictive information.

However, the dominance of a single feature raises concerns regarding the transferability of the model to different network environments. User agent characteristics can vary across browsers, devices, applications, and collection environments. Therefore, the model's dependence on this variable should be investigated through feature ablation experiments, particularly by evaluating performance after removing user agent information.

Figure 8 presents the receiver operating characteristic (ROC) curves of the leading ensemble classifiers. The curves illustrate the relationship between the true positive rate and false positive rate across different classification thresholds. The area under each curve provides a threshold independent measure of the model's ability to distinguish benign traffic from malicious traffic.

The reported ROC-AUC values include 0.896 for Gradient Boosting, 0.891 for Random Forest, and 0.767 for Extra Trees.

The ROC curves demonstrate that Gradient Boosting and Random Forest exhibit comparatively strong discrimination between benign and malicious network traffic. Gradient Boosting achieves the highest ROC-AUC of 0.896, indicating better overall ranking performance across different classification thresholds than the other two ensemble models.

Random Forest demonstrates a closely comparable ROC-AUC of 0.891, suggesting that its ability to rank malicious observations above benign observations is also strong. Extra Trees obtains a lower ROC-AUC of 0.767, indicating comparatively weaker discrimination.

The relatively strong ROC-AUC values should nevertheless be interpreted alongside the class imbalance. ROC-AUC does not directly describe the precision of positive predictions when malicious traffic is rare. Therefore, the results should be considered together with the precision recall analysis presented in Figure 7.

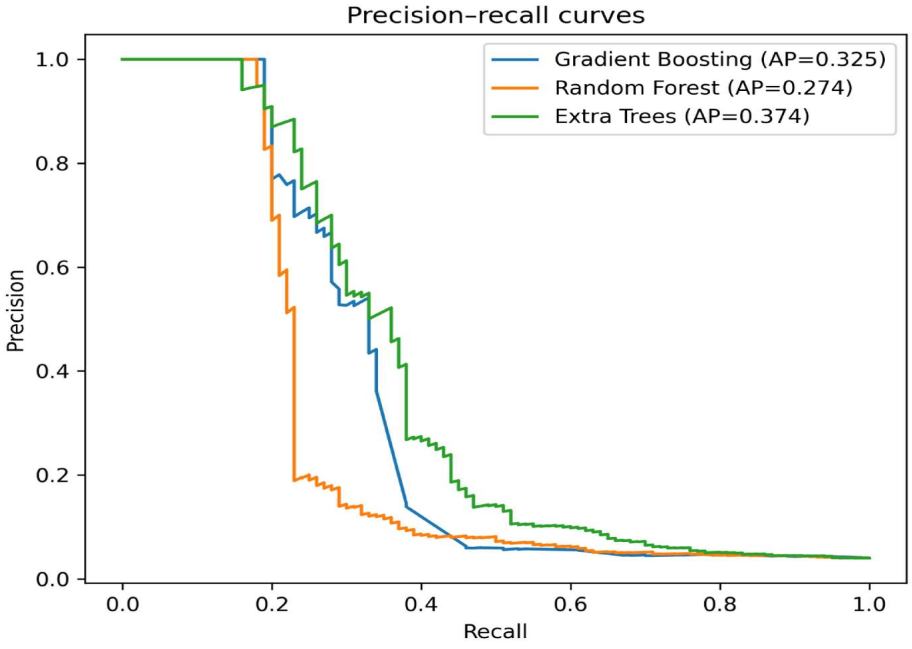


Figure 9. Precision–recall curves, which are especially informative given the low attack prevalence

Figure 9 presents the precision recall curves for the evaluated intrusion detection classifiers. The visualization illustrates how precision changes with recall across different classification thresholds. Unlike ROC curves, precision recall curves directly emphasize the detection performance of the minority malicious class.

The reported test set PR-AUC values include 0.603 for Gradient Boosting, 0.518 for Random Forest, 0.378 for Extra Trees, 0.364 for Logistic Regression, and 0.158 for SVM.

The figure indicates that Gradient Boosting provides the strongest precision recall performance among the evaluated classifiers, achieving the highest PR-AUC of 0.603. This suggests that the model maintains a comparatively favorable balance between detecting malicious observations and limiting false-positive predictions across different operating thresholds.

Random Forest achieves the second highest PR-AUC of 0.518, indicating useful detection capability despite its lower recall at the default threshold. Extra Trees and Logistic Regression demonstrate comparatively weaker precision recall performance, while SVM exhibits the lowest PR-AUC.

Given that malicious traffic constitutes only 4% of the dataset, the precision recall curves provide particularly important information regarding practical attack detection performance. The results support Gradient Boosting as the strongest overall candidate among the evaluated models, while also demonstrating the importance of selecting a threshold appropriate to operational security requirements.

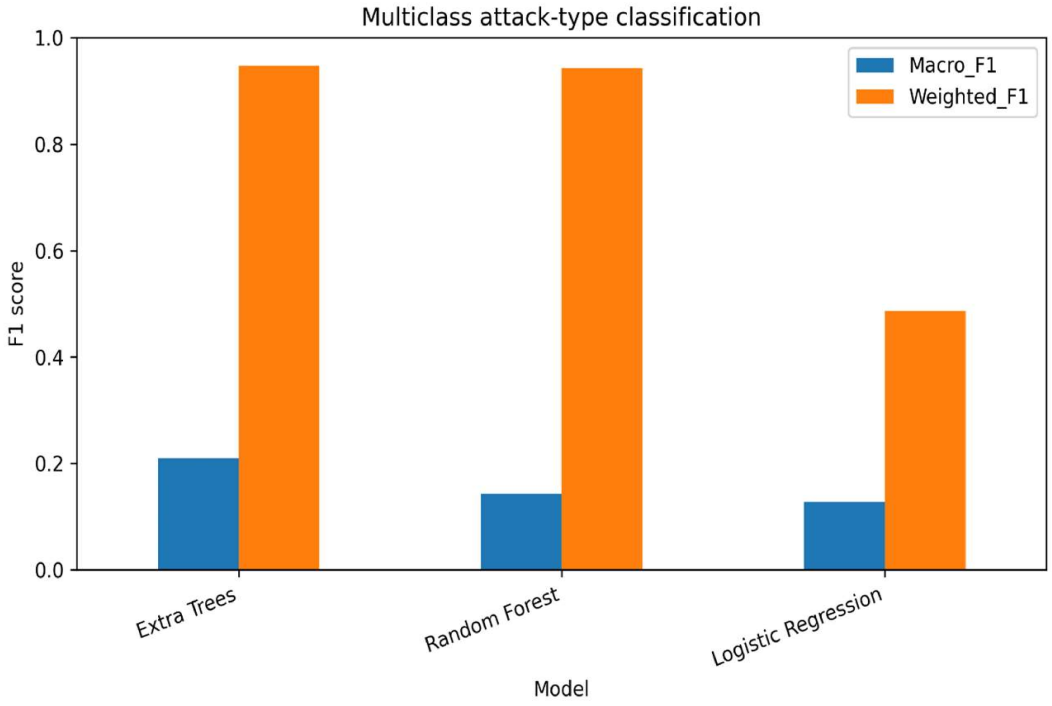


Figure 10. Macro- and weighted-F1 comparison for multiclass attack-type classification

Figure 10 compares macro-F1 and weighted-F1 scores for the multiclass attack classification models. The visualization includes Logistic Regression, Extra Trees, and Random Forest under the all record and attack-only experimental settings.

For the all record experiment, Extra Trees achieved a macro F1 of 0.209 and weighted F1 of 0.948, while Random Forest obtained 0.142 and 0.942, respectively. Logistic Regression achieved 0.128 macro F1 and 0.487 weighted F1.

For the attack only experiment, Logistic Regression obtained a macro-F1 of 0.397 and weighted-F1 of 0.513. Extra Trees achieved 0.350 and 0.517, while Random Forest achieved 0.340 and 0.501, respectively.

The figure highlights the substantial difference between macro-F1 and weighted-F1 in the all record multiclass experiment. The high weighted-F1 values achieved by Extra Trees and Random Forest indicate strong performance when class frequencies are considered. However, their considerably lower macro-F1 scores demonstrate that classification performance is uneven across the individual classes.

The attack-only results provide a clearer assessment of discrimination among malicious attack categories. Logistic Regression achieves the highest macro-F1 of 0.397, indicating a more balanced distribution of classification performance across attack classes than the two ensemble models in this experiment. Extra Trees and Random Forest achieve slightly higher accuracy but lower macro-F1.

These findings indicate that multiclass intrusion detection remains challenging because of the unequal representation of attack categories. The difference between macro-F1 and weighted-F1 further emphasizes the need to evaluate minority attack classes independently rather than relying on aggregate accuracy or weighted performance measures.

7. Overall interpretation of experimental findings

The combined results demonstrate that machine learning based intrusion detection is strongly influenced by class imbalance, classification thresholds, feature characteristics, and attack category representation. Although several classifiers achieve high accuracy, their ability to identify malicious traffic varies substantially.

Gradient Boosting demonstrates the strongest overall performance in binary intrusion detection, achieving the highest default threshold F1-score and PR-AUC. Its performance improves considerably following threshold optimization, with recall increasing to 0.580 and F1-score increasing to 0.558. This demonstrates that selecting an appropriate decision threshold is essential for improving attack detection sensitivity.

The feature importance analysis further reveals that user-agent information dominates the model's predictive behaviour. Although this feature provides considerable discriminatory information, its dominance raises questions regarding model generalization across different network environments. The multiclass results similarly demonstrate that high weighted performance does not guarantee reliable identification of rare attack categories.

Taken together, these findings support the development of an intrusion detection framework that combines ensemble learning, threshold optimization, class sensitive evaluation, and feature level interpretation. However, chronological and independent network validation, user agent ablation, and additional minority class data collection are necessary to establish the generalizability and practical reliability of the proposed approach.

7.11 Future Research Directions

Despite the promising results, several limitations should be acknowledged.

1. Limited malicious observations: Only 400 malicious records are available, restricting the reliability of minority class learning.

2. Rare attack categories: The smallest attack category contains only ten records, making its classification performance particularly uncertain.

3. Dominant predictor: The strong importance of user_agent raises concerns about potential dataset specific feature dependence.

4. Validation limitations: The original random split does not independently establish generalization across

different time periods or network entities.

5. Explainability limitations: Permutation importance provides global feature rankings but does not fully explain individual predictions. SHAP analysis remains to be conducted.

6. Operational deployment: The current findings represent offline benchmark evaluation and do not establish real-time detection performance.

Future research should investigate larger and more diverse network traffic datasets, improved minority class representation, temporal generalization, probability calibration, independent network evaluation, and deployment oriented computational efficiency.

8. Conclusion

This study proposes an explainable machine-learning framework for network intrusion detection and cyberattack classification. The framework integrates data preprocessing, feature engineering, binary and multiclass classification, comparative model evaluation, threshold optimization, feature interpretation, and robustness assessment.

The proposed methodology addresses the major limitations identified in the supplied experimental results, particularly class imbalance, low default threshold attack recall, uneven multiclass classification performance, and excessive dependence on influential network traffic features.

The existing experimental evidence identifies Gradient Boosting as the strongest candidate among the evaluated classifiers for binary intrusion detection. Its threshold optimized configuration improves attack recall and F1-score, demonstrating the importance of incorporating threshold optimization into the detection process.

The proposed integration of permutation importance, SHAP based interpretation, feature ablation, and generalization testing provides additional methodological depth beyond conventional classifier comparisons. Ultimately, the framework aims to establish a more transparent, class sensitive, and operationally relevant approach to intelligent network intrusion detection.

References

[1] Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies, 32*, e4150.

[2] AIMultiple. (2025). *100+ network security statistics in 2025*. Retrieved May 23, 2025, from https://research.aimultiple.com/network-security-statistics.

[3] Alameri, A., Aldribi, A. (2021). Apply machine learning techniques to detect malicious network traffic in cloud computing. *Journal of Big Data, 8*, 90.

[4] Bakar, R. A., Huang, X., Javed, M. S., Hussain, S., Majeed, M. F. (2023). An intelligent agent-based detection system for DDoS attacks using automatic feature extraction and selection. *Sensors, 23*, 3333.

[5] Canadian Institute for Cybersecurity. (n.d.). *NSL-KDD datasets*. UNB. Retrieved May 24, 2025, from https://www.unb.ca/cic/datasets/nsl.html.

[6] Dong, B., Wang, X. (2016). Comparison deep learning method to traditional methods using for network intrusion detection. In *Proceedings of the 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)* (p. 581–585). IEEE.

[7] Dora, V. R. S., Lakshmi, V. N. (2022). Optimal feature selection with CNN-feature learning for DDoS attack detection using meta heuristic-based LSTM. *International Journal of Intelligent Robotics and Applications, 6*, 323–349.

[8] Hosseini, S., Zade, B. M. H. (2020). New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. *Computer Networks, 173*, 107168.

[9] Kalinaki, K., Thilakarathne, N. N., Mubarak, H. R., Malik, O. A., Abdullatif, M. (2023). Cybersafe capabilities and utilities for smart cities. In M. Ahmed & P. Haskell-Dowland (Eds.), *Cybersecurity for smart cities*. Springer.

[10] Kurniabudi, K., Purnama, B., Sharipuddin, S., Darmawijoyo, D., Stiawan, D., Samsuryadi, S., Heryanto, A., & Budiarto, R. (2019). Network anomaly detection research: A survey. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 7*, 37–50.

[11] Perez, S. I., Criado, R. (2022). Increasing the effectiveness of network intrusion detection systems (NIDSs) by using multiplex networks and visibility graphs. *Mathematics, 11*, 107.

[12] Statista. (2025). *Infographic: Cybercrime expected to skyrocket in coming years*. Statista Daily Data. Retrieved May 23, 2025, from https://www.statista.com/chart/28878/expected cost of cybercrime until-2027.

[13] Wattanapongsakorn, N., Srakaew, S., Wonghirunsombat, E., Sribavonmongkol, C., Junhom, T., Jongs-ubsook, P., Charnsripinyo, C. (2012). A practical network-based intrusion detection and prevention system. In *Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications* (p. 209–214). IEEE.

[14] Yost, J. R. (2016). The march of IDES: Early history of intrusion-detection expert systems. *IEEE Annals of the History of Computing, 38*, 42–54.

[15] Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., Guizani, M. (2020). A survey of machine and deep learning methods for internet of things (IoT) security. IEEE communications surveys & tutorials, 22(3), 1646-1685.

[16] Liang, C., Shanmugam, B., Azam, S., Jonkman, M., De Boer, F., & Narayansamy, G. (2019, March). Intrusion detection system for Internet of Things based on a machine learning approach. In 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN) (p. 1-6). IEEE.

[17] Sharma, B., Sharma, L., Lal, C., & Roy, S. (2024). Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning-based approach. Expert Systems with Applications, 238, 121751.

[18] Alameri, A., Aldribi, A. (2021). Apply machine learning techniques to detect malicious network traffic in cloud computing. J Big Data, 8, 90.