International Journal of Safety and Security Engineering
Vol. 16, No. 7, July, 2026, pp. 1575-1584
Journal homepage: http://iieta.org/journals/ijsse

Reproducible Multiclass Intrusion Detection on UNSW-NB15 Using XGBoost: A Leakage-Safe Pipeline with χ^2 Feature-Selection Ablations

Ahmed Eskander Mezher[1,2*], Huda Kadhim Tayyeh[3], Atheer Akram AbdulRazzaq[1], Ahmed Sabah Ahmed AL-Jumaili[1]

[1] Business Information Technology Department (BIT), Businesses Informatics College, University of Information Technology and Communications, Baghdad 10045, Iraq
[2] Awang Had Salleh Graduate School of Arts and Sciences, Universiti Utara Malaysia (UUM), Sintok 06010, Malaysia
[3] Department of Informatics Systems Management (ISM), Businesses Informatics College, University of Information Technology and Communications, Baghdad 10045, Iraq

Corresponding Author Email: ahmed.mezher@uoitc.edu.iq

https://doi.org/10.18280/ijsse.160714

Received: 7 June 2026
Revised: 7 July 2026
Accepted: 13 July 2026
Available online: 31 July 2026

Keywords:
intrusion detection, network security, XGBoost, UNSW-NB15, feature selection, class imbalance, multiclass classification, cybersecurity

ABSTRACT

Machine-learning methods increasingly support multiclass intrusion detection on modern benchmarks such as UNSW-NB15. However, reproducible leakage-safe baselines and the effect of simple chi-square (χ^2) feature selection under class imbalance remain under-examined. This study contributes to IT security and security engineering by developing a reproducible multiclass intrusion-detection pipeline for identifying and analyzing network attacks under class imbalance. The study presents a reproducible, leakage-safe eXtreme Gradient Boosting (XGBoost) pipeline with χ^2 feature-selection ablations. The official UNSW-NB15 partitions are pooled and evaluated using a fixed stratified 80/20 split (seed = 42) to enable controlled within-study comparison; this protocol is not intended as a direct replication of the official train/test evaluation. All preprocessing steps, including one-hot encoding and Min–Max scaling for χ^2, are fitted on the training split only to prevent data leakage. The full-feature XGBoost model achieves 83.43% Accuracy, 59.16% Macro-F1, and 0.9692 Macro-ROC-AUC, outperforming the evaluated Logistic Regression, Random Forest, and Extra Trees baselines on the same split. A χ^2 top-k sweep (k = 80, 100, 120, 160) peaks at k = 80 with 83.10% Accuracy and 58.39% Macro-F1, remaining slightly below the full-feature model. Per-class metrics and a row-normalized confusion matrix reveal strong performance on dominant classes and weaker recall for rare attack families. Overall, the study provides a reproducible benchmark and a transparent evaluation of χ^2 feature selection for multiclass intrusion detection under class imbalance.

1. INTRODUCTION

The increasing complexity of modern networked environments, including enterprise systems, cloud services, industrial networks, and connected infrastructures, has expanded the cybersecurity attack surface. These environments may be exposed to diverse malicious activities, including data exfiltration, botnet communication, reconnaissance, and distributed denial-of-service (DDoS) attacks [1]. Consequently, effective Intrusion Detection Systems (IDSs) are needed to support early threat identification and cyber-risk mitigation [2, 3]. Machine-learning-based IDSs have received considerable attention because they can learn discriminative patterns from network traffic and support the automated identification of malicious activity [4, 5]. Ensemble-learning methods, including Random Forest and eXtreme Gradient Boosting (XGBoost), are widely used for structured network-traffic data because they can model nonlinear relationships among heterogeneous flow features [6, 7]. However, the reliability of machine-learning-based IDS evaluation can be affected by class imbalance, limited analysis of minority attack categories, and inconsistent experimental procedures [8]. Furthermore, feature-selection methods are frequently applied to reduce dimensionality, but their effect on multiclass detection performance should be evaluated using transparent and leakage-controlled experimental protocols [9]. To address these evaluation challenges, this paper develops a reproducible and leakage-safe multiclass intrusion-detection pipeline based on XGBoost and evaluates it on the UNSW-NB15 benchmark. The evaluation supports cyber-risk mitigation by identifying attack families that remain difficult to detect and by showing where classification errors are concentrated. Specifically, we compare a full-feature XGBoost model with χ^2-based top-k feature-selection variants under the same fixed stratified split, train-only preprocessing protocol, and evaluation settings. In addition to overall accuracy, we report macro-averaged and per-class metrics together with confusion-matrix-based error analysis to characterize performance differences between frequent and rare attack families. This framing positions the

work as a controlled benchmark and ablation study for multiclass network intrusion detection on UNSW-NB15. Accordingly, the findings of this study are limited to the UNSW-NB15 network-traffic setting and do not constitute direct validation on IoT-specific traffic or resource-constrained IoT platforms. Existing XGBoost- and feature-selection-based intrusion-detection studies often emphasize overall predictive performance or optimize a specific model configuration. In contrast, the present study keeps the data split, preprocessing procedure, model configuration, and evaluation protocol fixed while examining whether χ^2-based feature reduction changes multiclass detection behavior. The contribution therefore lies in providing a controlled and reproducible comparison between full-feature and reduced-feature XGBoost models, with particular attention to macro-level and minority-class performance under class imbalance. Our contributions can be summarized as follows:

- We develop a reproducible and leakage-safe multiclass intrusion detection pipeline for UNSW-NB15 using XGBoost, with all preprocessing steps fitted on the training split only to ensure fair evaluation.
- We conduct a controlled χ^2 feature-selection ablation by comparing multiple top-k settings against the full-feature model under identical data split, preprocessing, and training conditions.
- We provide a class-imbalance-aware evaluation through macro-averaged metrics, per-class Precision/Recall/F1, and row-normalized confusion-matrix analysis, highlighting the differing behavior of head and tail attack classes.
- We show that the full-feature XGBoost model provides the strongest overall performance on the evaluated split, while χ^2 reduction offers a transparent dimensionality-reduction baseline without surpassing the full model.

The remainder of the paper is structured as follows. Section 2 reviews related work on machine-learning-based IDSs. Section 3 outlines the proposed framework and feature selection methodology. Section 4 describes the experimental design, including datasets, model configurations, and evaluation metrics. Section 5 presents and discusses the results. Section 6 concludes the paper and outlines future research directions.

2. RELATED WORK

This section reviews four research areas relevant to the present study: (i) modern datasets for network intrusion detection, (ii) tree-ensemble and XGBoost-based IDSs, (iii) feature-selection methods for IDS, and (iv) deep learning and explainable intrusion detection.

2.1 Datasets for network intrusion detection

As of now, modern IDS research has shifted from using legacy corpora (KDD'99/NSL-KDD) to richer data sets which are more representative of the diverse traffic and current attack patterns. The UNSW NB15 dataset contains nine attack families and realistic background traffic and has been essentially adopted as a standard for evaluating supervised IDS [10, 11]. Complementary corpora, like the CIC IDS2017, are typically used for benchmarking and record a labeled enterprise traffic and attack scenarios [12]. IoT-oriented datasets, such as BoT-IoT, ToN-IoT, and IoT-23, are valuable when a study explicitly aims to evaluate traffic generated by IoT devices or IoT-specific attack scenarios. However, the present study focuses on the UNSW-NB15 benchmark for controlled multiclass network intrusion-detection evaluation [13]. Cross-dataset evaluation can help assess generalization; however, the current study is limited to UNSW-NB15 and does not claim direct validation on IoT-device traffic or constrained IoT platforms [14].

2.2 Tree ensembles and XGBoost in Intrusion Detection System

Tree based ensembles (Random Forests) are appreciated for their robustness and good tabular data performance [15]. Additionally, sparsity aware split finding, regularized boosting, and improved efficiency and accuracy of XGBoost is reported to be the best performer in IDS classification on flow features [16]. Recent studies focused on UNSW NB15 show that mitigation of class overlap and class imbalance in building IDS on modern datasets is important, because of significant differences in false alarm rates and minority class detection, depending on the algorithmic choices and the error aware training [17]. The results highlight the importance of macro-average metrics and the need to be careful with class imbalance in multiclass scenarios.

2.3 Feature selection for Intrusion Detection System

To reduce dimensionality and examine whether a smaller predictor set can preserve detection performance, filter-based selection methods, such as χ^2, mutual information, and correlation-based selection, are commonly applied in intrusion-detection studies [18, 19]. Tree-based learners combined with filter-based selection have been reported as compact and competitive intrusion-detection approaches [20]. In addition to the basic filters, hybrid methods are based on the simultaneous optimization of the model hyperparameters and a subset of features (such as genetic algorithm-based wrappers) and have been shown to achieve a consistent improvement across several test sets including UNSW NB15 and other similar corpora [21]. However, large scale comparative studies of IoT datasets (e.g., ToN IoT) have shown that, although the relative benefit of feature selection over feature extraction is dependent on the feature carnality and class imbalance, features that have been selected using a univariate filter (e.g., χ^2) are still attractive when interpretability and speed are priorities [22, 23]. Our study therefore examines both full-feature and χ^2-reduced variants to quantify the detection–efficiency trade-off.

2.4 Deep learning and explainable intrusion detection

Deep models, including CNN/LSTM hybrids and attention-based architectures, have achieved strong performance in intrusion-detection studies but often require higher computational and memory resources than conventional tree-based approaches [24]. This has maintained interest in explainable tree-based alternatives that can expose feature attributions while maintaining competitive performance on structured network-flow intrusion-detection data [25, 26]. In multiclass and imbalanced regimes typical of IDS, precision–recall analysis is recognized as more informative than ROC for evaluating rare-attack detection quality, reinforcing our use of Macro-PR-AUC alongside macro-F1 and ROC-AUC [27-30].

2.5 Positioning of the present work

Relative to prior work, the contribution of this study is not the introduction of a new learning algorithm, but a carefully controlled evaluation setting for multiclass intrusion detection on UNSW-NB15. The paper combines: (i) a leakage-safe, train-only preprocessing pipeline, (ii) a reproducible XGBoost-based multiclass benchmark under a fixed stratified split, and (iii) a controlled χ^2 feature-selection ablation analyzed using macro-averaged and per-class metrics. This combination emphasizes transparent comparison and error analysis under class imbalance, providing a practical benchmark for future IDS studies on modern tabular traffic data.

While prior IDS studies have examined class imbalance on UNSW-NB15, weighted XGBoost, χ^2-based feature selection, and classical ensemble comparisons [5, 11, 16, 21, 26], the present study does not claim a new learning algorithm. Its contribution is a controlled and reproducible evaluation design: the same pooled stratified split, train-only preprocessing, fixed XGBoost configuration, expanded baseline comparison, class-balanced sensitivity analysis, macro-level and per-class evaluation, Macro-PR-AUC, bootstrap confidence intervals, and practical efficiency measurements are combined to determine whether χ^2 feature reduction provides a genuine multiclass benefit under class imbalance.

3. METHODOLOGY

This section details the dataset, data partitioning, preprocessing pipeline, feature-selection ablation, model configurations, and evaluation protocol. All steps are designed for reproducibility and no leakage, implemented as scikit learn/XGBoost pipelines.

3.1 Data and split

We use the UNSW-NB15 dataset, which comprises realistic background traffic and nine contemporary attack families (Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms) plus Normal, yielding a 10-class problem. To perform a controlled within-study comparison between the full-feature XGBoost model, the Random Forest baseline, and the χ^2 feature-selection variants, the official UNSW-NB15 training and testing files were pooled and then divided using a single stratified hold-out split (80% training and 20% testing; random seed = 42). Stratification preserves the class proportions of the pooled dataset in both partitions, including the minority attack families.

The pooled corpus contains 257,673 network flows after duplicate detection and basic consistency checks. Class distributions are imbalanced, with Normal and Generic representing the largest classes and Backdoor and Worms representing rare classes; therefore, macro-averaged and per-class metrics are reported in addition to overall accuracy.

This pooled-split protocol was selected to ensure that all evaluated configurations use identical training and testing conditions. Because it differs from the official UNSW-NB15 train/test partition, the reported results should be interpreted as a controlled internal evaluation and should not be compared directly, in numerical terms, with studies that use the official split.

Table 1 reports the class counts and percentages before and after cleaning. These statistics describe the combined official partition that underlies all subsequent experiments in this paper.

Data-quality checks, including duplicate detection and label-consistency checks, did not remove any records. Therefore, the cleaned distribution equals the raw merged distribution. Table 1 shows a strongly imbalanced distribution (Normal and Generic dominate, Shellcode and Worms are rare). Accordingly, we will report per-class Precision/Recall/F1 and Macro-F1 (class-averaged) in addition to Accuracy. The subsequent train/test evaluation preserves these proportions via stratification (seed = 42). The merged dataset used in this study contains 45 columns: 42 predictor variables, the record identifier (id), and two target-related variables, namely attack_cat for multiclass classification and label for binary attack-versus-normal classification. The predictor variables describe packet statistics, protocol-level behavior, and flow-related characteristics. The UNSW-NB15 file of official feature list was used to assign the column headers manually to make sure that the features and labels matched well. No rows were removed for missing values because the final merged table contained no missing entries; categorical variables were encoded within the train-only preprocessing pipeline.

Figure 1 summarizes the pooled stratified evaluation workflow. Leakage-safe preprocessing and model fitting are performed on the training split only. The workflow evaluates XGBoost variants, Logistic Regression, Random Forest, Extra Trees, and an optional χ^2 feature-selection branch, using classification metrics, Macro-ROC-AUC, Macro-PR-AUC, and confusion matrices, with bootstrap confidence intervals for the primary XGBoost model.

Table 1. Class distribution of the merged UNSW-NB15 dataset before and after cleaning (TRAIN+TEST = 257,673 flows)

Attack_cat	Count_before	Percent_before	Count_after	Percent_after
Analysis	2677	1.04	2677	1.04
Backdoor	2329	0.9	2329	0.9
DoS	16353	6.35	16353	6.35
Exploits	44525	17.28	44525	17.28
Fuzzers	24246	9.41	24246	9.41
Generic	58871	22.85	58871	22.85
Normal	93000	36.09	93000	36.09
Reconnaissance	13987	5.43	13987	5.43
Shellcode	1511	0.59	1511	0.59
Worms	174	0.07	174	0.07
TOTAL	257673	100	257673	100

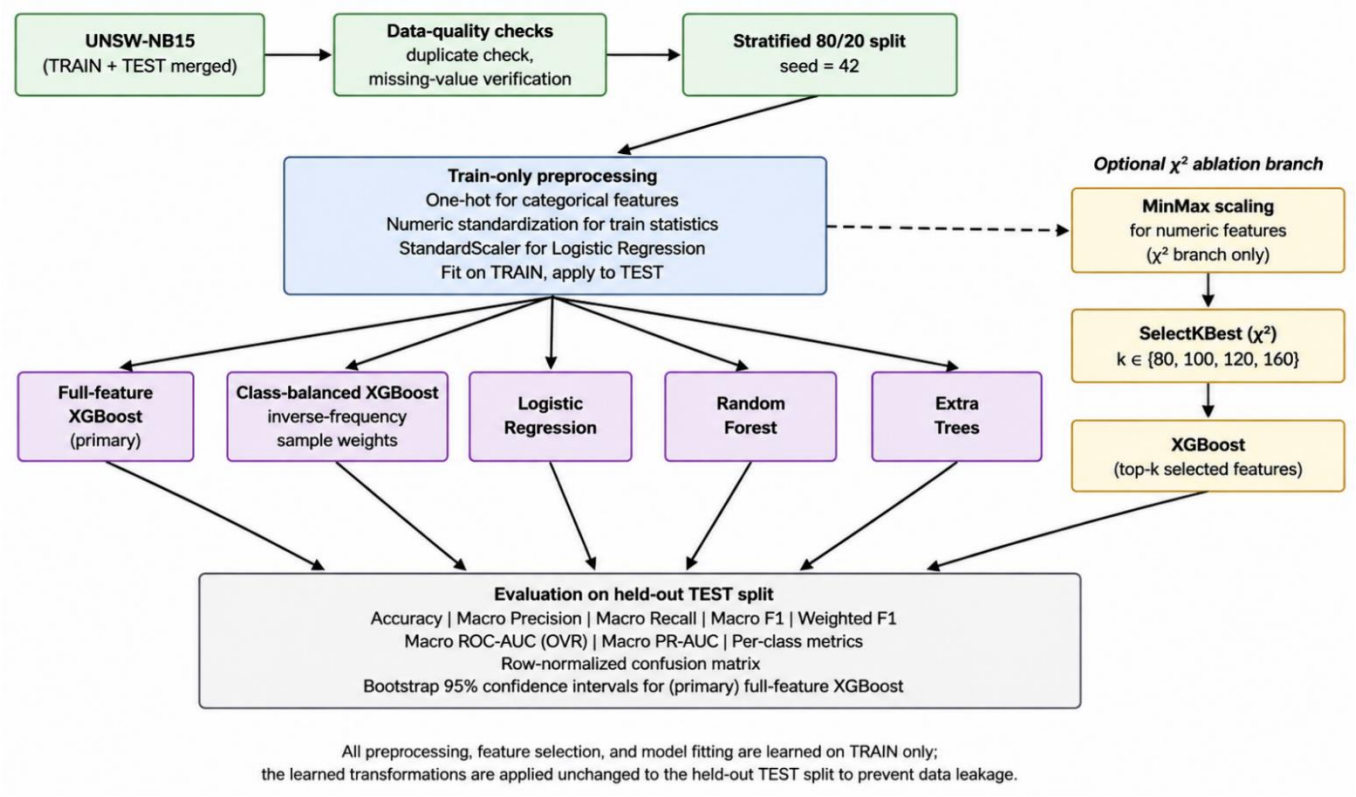

Figure 1. Proposed leakage-safe workflow for multiclass intrusion detection on UNSW-NB15

3.2 Pre-processing

We follow a reproducible, train-only pre-processing pipeline built with scikit-learn. After concatenating the official UNSW-NB15 TRAIN and TEST partitions (total 257,673 flows; see Section 3.1), we checked for exact duplicate rows and verified that the final table contained no missing values. No records were removed during these checks. After excluding the record identifier (id), the binary target-related variable (label), and the multiclass target (attack_cat), the predictor matrix contains 42 raw features: 39 numerical features and 3 categorical features. One-hot encoding expands the training-derived predictor space to 195 transformed features. All transformers are fit on the training split only and then applied to the held-out test split to avoid leakage.

For the main model (XGBoost, no feature selection), numeric features are used as is (no standardization is required for tree ensembles), and the OHE outputs are passed directly to the classifier. For the feature-selection ablation (χ^2/SelectKBest), we first apply MinMax scaling to the numeric features to satisfy the non-negativity assumption of χ^2 (OHE outputs are already non-negative); we then compute χ^2 scores on the training split only and evaluate predefined top-k configurations($k \in \{80,100,120,160\}$). The selected columns are concatenated (numeric-scaled + OHE) and used to train XGBoost with the same hyperparameters as the full-feature model. At inference time, the fitted pre-processor and the χ^2 selector are applied to the test split in the same order. We keep the attack_cat label in a 10-class setting: Normal plus {Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms. Class imbalance is addressed through macro-averaged and per-class evaluation. In addition, an imbalance-aware sensitivity experiment trains XGBoost using inverse-frequency sample weights calculated exclusively from the training labels; no resampling is applied. Random seeds are fixed (seed = 42) for the stratified split and model initialization. The entire pre-processing is scripted so that the fitted encoders/scalers/selectors can be serialized and reused for exact reproducibility.

3.3 Feature-selection ablation (χ^2)

We treat feature selection as a controlled ablation, evaluating whether reducing dimensionality improves or preserves performance relative to the full-feature model. Because the χ^2 statistic measures the univariate dependence between a non-negative feature and the class label, we first apply MinMax scaling to numeric features and one-hot encoding (OHE) to categorical features; both steps are performed inside a scikit-learn pipeline that is fit on the training split only and then applied to the test split, preventing any form of test leakage. The χ^2 scores are then computed on the training data only over the transformed feature matrix, and the top-k columns are selected with SelectKBest(χ^2).

As an exploratory view of the ranking produced by χ^2, Figure 2 shows the top 20 features by χ^2 score computed on the training split only (after MinMax+OHE). This bar chart is for illustration and analysis of discriminative signals; it is not used to pick the final K. For the ablation, we evaluate $k \in \{80,100,120,160\}$, representing moderate-to-high feature-retention levels within the 195-dimensional transformed feature space. These values were selected to examine whether retaining progressively larger subsets preserves multiclass performance relative to the full-feature model. For each k, we (i) fit the MinMax+OHE transformers on the training split, (ii) compute χ^2 on the training split and keep the top- k columns, and (iii) train XGBoost with the same hyperparameters as the full-feature model. At inference time, the fitted pre-processor and the χ^2 selector are applied to the held-out test set in the same order. Aggregate and per-class performance for this

ablation are reported in Section 4.4, as well as the efficiency comparison. The χ^2 ranking computed on the training split after Min–Max scaling and one-hot encoding is shown in Figure 2 for descriptive analysis only. The candidate k values were predefined, and the full-feature model remains the reference configuration for comparison.

As shown in Figure 2, the highest χ^2 scores indicate strong univariate association for several flow-derived and protocol/service features on the training split. Because χ^2 is univariate, we treat this plot as descriptive only; model selection uses the predetermined K sweep and the held-out test results reported later.

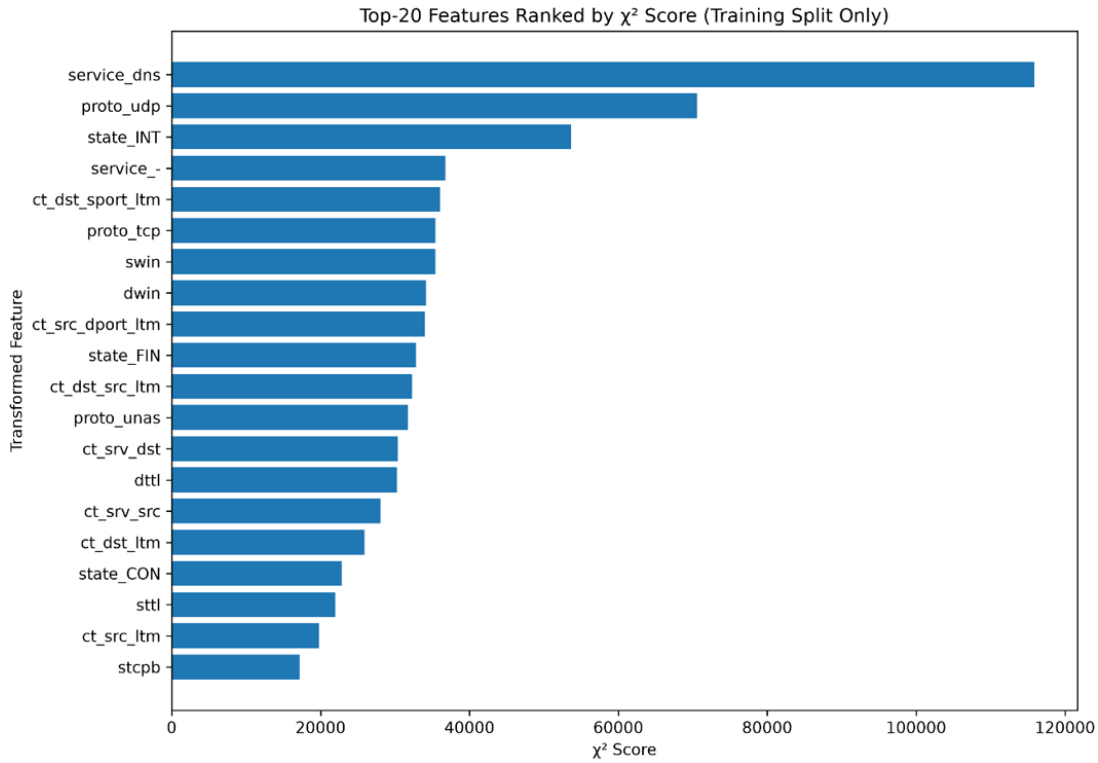

Figure 2. Top-20 transformed features ranked by χ^2 score on the training split after Min–Max scaling and one-hot encoding

3.4 Models and hyperparameters

We evaluate four classifiers under the same leakage-safe preprocessing and stratified evaluation protocol:

- XGBoost (primary). Gradient-boosted decision trees are trained using the multiclass probability objective (multi:softprob). To maintain comparability, the following hyperparameters are fixed across the XGBoost experiments: n_estimators = 300, max_depth = 6, learning_rate = 0.10, subsample = 0.80, colsample_bytree = 0.80, min_child_weight = 1, reg_lambda = 1.0, tree_method = 'hist', eval_metric = 'mlogloss', and random_state = 42. For the unweighted XGBoost reference model, class imbalance is not reweighted during training and is assessed through macro-averaged and per-class evaluation metrics. To examine the effect of class imbalance during training, we additionally evaluate a class-balanced XGBoost variant. This model uses the same preprocessing pipeline and XGBoost hyperparameters as the unweighted reference model, but applies inverse-frequency sample weights derived from the training-set class distribution. The purpose of this experiment is to assess whether improved minority-class recall is obtained at the cost of overall detection performance.
- Random Forest (baseline). A bagging-based tree-ensemble baseline implemented with scikit-learn and configured with n_estimators = 200, max_features = 'sqrt', bootstrap = True, and n_jobs = -1.
- Extra Trees**.** An Extremely Randomized Trees classifier is included as an additional randomized tree-ensemble baseline. It is configured with n_estimators = 200, max_features = 'sqrt', bootstrap = False, and n_jobs = -1.
- Logistic Regression**.** A multinomial logistic-regression classifier is included as a linear baseline. Numerical features are standardized within the training pipeline, whereas categorical features are one-hot encoded. The model uses the L-BFGS solver with a maximum of 1,000 iterations.
- All transformers (OHE/MinMax) and, when applicable, the χ^2 selector are fit exclusively on the training set. Pipelines are implemented using scikit-learn's Pipeline/ColumnTransformer to guarantee correct ordering and prevent leakage. We fix random seeds (42) for data partitioning and model initialization. For interpretability and error analysis, we produce row-normalized confusion matrices (diagonal equals per-class recall) alongside standard metrics.

3.5 Evaluation metrics and reporting protocol

In section 3.1, we defined the held-out test set that is used to evaluate models, and we adopted a consistent train-only pipeline. As UNSW-NB15 is a class-imbalanced dataset, we include macro-averaged numbers as well as per-class numbers.

(1) Primary metrics

We report on Accuracy, macro-Precision, macro-Recall and

macro-F1. Macro-averaged measures are the simple average of scores on the individual classification tasks, and so are the same for majority and minority classes, for a K-class classification task. In addition, we report Weighted-F1 which weighs the F1 score of each class by its class support, giving a performance measure that reflects the distribution of classes. Precision is the percentage of the instances correctly predicted as class c among all instances predicted as class c; Recall is the percentage of the instances correctly predicted as class c among all instances of class c. Averaging across all classes of the harmonized averages of Precision and Recall for class c yields the F1-score. Macro-averaging is then used to compute the averages of the per-class F1 scores.

(2) Confusion matrices

We include row-normalized confusion matrices (each row sums to 1.0), so the diagonal equals per-class recall. This view highlights where errors flow from each true class into predicted classes, which is essential for interpreting minority-class behavior under imbalance.

(3) Ranking metrics

In addition to threshold-based metrics, we compute Macro-ROC-AUC using the one-vs-rest formulation and Macro-PR-AUC using one-vs-rest average precision. Macro-PR-AUC is particularly informative under class imbalance because it reflects the precision–recall trade-off across attack categories.

(4) Decision rule / thresholds

Multiclass predictions are obtained by argmax over class probabilities (multi: softprob for XGBoost). We do not apply post-hoc threshold tuning or probability calibration in the main results; this keeps comparisons fair across models/ablations.

(5) Statistical confidence

To quantify uncertainty for the primary full-feature XGBoost model, we compute 95% confidence intervals for Accuracy, Macro-F1, and Macro-PR-AUC using a stratified percentile bootstrap on the held-out test set with 1,000 resamples.

(4) Reporting conventions

We report percentages with two decimal places and express ablation differences in percentage points relative to the full-feature XGBoost reference model.

4. RESULTS

4.1 Experimental setup

All results are computed on the held-out test split from the merged official UNSW-NB15 partitions (stratified 80/20, seed = 42), using the train-only pipeline. This pooled-split design is used to support controlled internal comparison among the evaluated models and ablations; therefore, the results should not be directly compared with studies that use the official UNSW-NB15 train/test partition. Class mapping follows the 10-class taxonomy in section 3.1 [13].

4.2 Per-class performance (XGBoost, full features)

Table 2 reports the per-class Precision, Recall, F1-score, and support for the full-feature XGBoost model on the held-out test split. The results show strong performance for Normal, Generic, and Exploits, whereas minority categories such as Analysis, Backdoor, and DoS remain difficult to detect consistently.

To complement the per-class results in Table 2, Figure 3 presents the row-normalized confusion matrix for the primary full-feature XGBoost classifier. Each row sums to 100%, so diagonal values represent per-class recall and off-diagonal values show how each true attack family is distributed across predicted categories. This representation helps identify error patterns that are not apparent from aggregate metrics alone.

Table 2. Per-class performance of the full-feature XGBoost model on the held-out pooled UNSW-NB15 test split

Class	Precision (%)	Recall (%)	F1 (%)	Support
Normal	91.14	94.95	93.00	18,600
Fuzzers	70.99	57.85	63.75	4,849
Analysis	95.00	10.65	19.16	535
Backdoor	83.67	8.80	15.92	466
DoS	44.40	13.45	20.65	3,271
Exploits	62.91	90.24	74.14	8,905
Generic	99.58	98.22	98.90	11,774
Reconnaissance	92.04	76.91	83.80	2,798
Shellcode	63.37	72.19	67.49	302
Worms	52.63	57.14	54.79	35
Overall (macro avg)	75.57	58.04	59.16	—
Overall (weighted avg)	83.16	83.43	81.61	
Accuracy	—	—	83.43	51,535

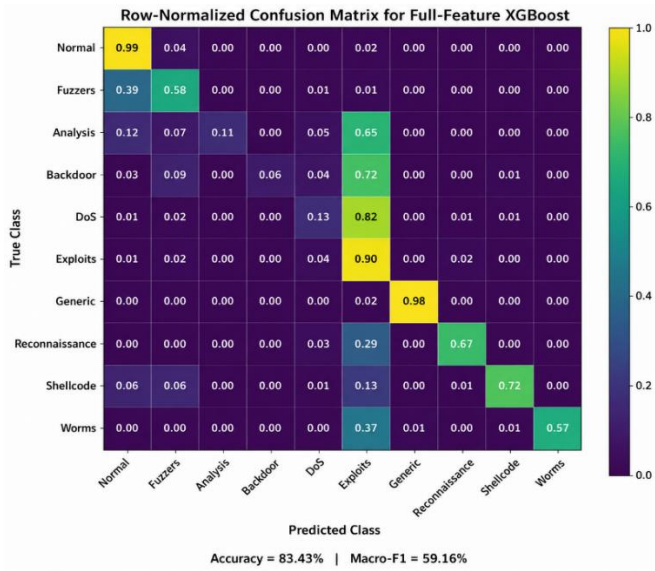

Figure 3. Row-normalized confusion matrix for the full-feature XGBoost model on the held-out pooled UNSW-NB15 test split

Figure 3 shows that the full-feature XGBoost model recognizes Normal, Generic, and Exploits more reliably than several minority attack families. In particular, Analysis, Backdoor, and DoS are frequently misclassified as Exploits, indicating overlapping flow characteristics among these categories. The row-normalized representation also shows that overall performance should be interpreted together with per-class recall, because the detection quality of rare attack families remains substantially less consistent than that of high-support classes.

4.3 Overall performance compared with baseline models

Table 3 summarizes the overall performance of XGBoost, Logistic Regression, Random Forest, and Extra Trees on the held-out pooled UNSW-NB15 test split.

Table 3 compares XGBoost with Logistic Regression,

Random Forest, and Extra Trees under the same leakage-safe preprocessing and fixed stratified split. XGBoost achieved the highest Accuracy (83.43%), Macro-Precision (75.57%), Macro-Recall (58.04%), Macro-F1 (59.16%), Macro-ROC-AUC (0.9692), and Macro-PR-AUC (0.6556). Among the non-XGBoost baselines, Extra Trees obtained the second-highest Macro-F1 (55.25%), followed by Random Forest (54.93%), whereas Logistic Regression achieved substantially lower Macro-F1 (37.66%). These results indicate that boosted trees provide stronger multiclass discrimination than the evaluated linear and bagging-based ensemble baselines under the current pooled UNSW-NB15 evaluation protocol. For the primary full-feature XGBoost model, stratified bootstrap analysis with 1,000 resamples yielded a 95% confidence interval of 83.18%–83.68% for Accuracy and 57.59%–60.53% for Macro-F1. The corresponding Macro-PR-AUC was 0.6556 with a 95% confidence interval of 0.6408–0.6750. These intervals quantify the uncertainty of the reported XGBoost performance on the held-out test split.

Table 4 reports the bootstrap-derived 95% confidence intervals for the full feature XGBoost model.

Table 3. Overall performance comparison on the pooled stratified UNSW-NB15 test split (80/20, seed = 42)

Model	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)	Weighted-F1 (%)	Macro ROC-AUC	Macro PR-AUC	Test n
Logistic Regression	77.58	54.02	38.67	37.66	74.62	0.9505	0.4640	51535
Random Forest (Baseline)	82.43	67.16	52.39	54.93	81.57	0.9093	0.5957	51535
Extra Trees	81.99	66.00	52.01	55.25	81.28	0.9030	0.5824	51535
XGBoost	83.43	75.57	58.04	59.16	81.61	0.9692	0.6556	51535

Table 4. Bootstrap confidence intervals for the primary full-feature XGBoost model

Metric	Point Estimate	95% Confidence Interval
Accuracy (%)	83.43	83.18–83.68
Macro-F1 (%)	59.16	57.59–60.53
Macro-PR-AUC	0.6556	0.6408–0.6750

4.4 Ablation study: Feature selection (χ^2)

We evaluated χ^2 feature selection using Min–Max scaling for numerical features and one-hot encoding for categorical features under the same pooled stratified split. Table 5 reports the per-class Precision, Recall, F1-score, and support for the best χ^2 configuration, k = 80.

Table 5. Per-class performance of XGBoost after χ^2 feature selection (k = 80) on the held-out pooled UNSW-NB15 test split

Class	Precision	Recall	F1	Support
Normal	90.77	94.80	92.74	18600
Fuzzers	70.56	57.19	63.17	4849
Analysis	92.06	10.84	19.40	535
Backdoor	82.61	8.15	14.84	466
DoS	42.45	11.53	18.13	3271
Exploits	62.35	90.35	73.78	8905
Generic	99.59	97.83	98.70	11774
Reconnaissance	92.12	76.91	83.83	2798
Shellcode	62.65	70.53	66.36	302
Worms	54.55	51.43	52.94	35
Overall (macro avg)	74.97	56.96	58.39	—
Accuracy	—	—	83.10	51,535

At k = 80, χ^2-selected XGBoost achieves 83.10% Accuracy and 58.39% Macro-F1. Performance is broadly comparable to the full-feature model, but the reduced-feature configuration remains lower in both Accuracy and Macro-F1. Therefore, the full-feature XGBoost model is retained as the primary configuration, while χ^2 selection is reported as a dimensionality-reduction ablation.

Figure 4 compares Macro-F1 across the predefined χ^2 feature-retention levels. The best χ^2 configuration is obtained at k=80, achieving a Macro-F1 of 58.39%. However, all evaluated χ^2-selected variants remain below the full-feature XGBoost reference result of 59.16%. Thus, reducing the transformed feature space does not improve multiclass detection performance under the evaluated setting. These findings show that χ^2 selection can reduce the transformed feature space while maintaining broadly comparable performance; however, the full-feature model remains superior in Macro-F1, and the practical efficiency benefit is limited primarily to lower training time.

Table 6 compares the computational gains in terms of runtime from using χ^2 feature reduction. Keeping 80 features out of 195 reduces the feature space by 58.97%, and saves 23.42 s (93.21 s to 69.79 s) for the XGBoost training time. The accuracy (83.10% vs 83.43%) and macro-F1 (58.39% vs 59.16%) of the χ^2-selected model are slightly lower than the full-feature model. Besides that, the end-to-end inference time is not shortened in the present implementation and the serialized χ^2 pipeline is slightly bigger. Thus, the selection of χ^2 should not be thought of as an option to improve the performance of the full-feature model or optimize the model for deployment, but rather one to reduce the dimensionality and training cost.

4.5 Observations and error analysis

The findings show that class imbalance is still a significant problem in multiclass intrusion detection systems. The full-feature XGBoost shows the best overall accuracy, but classes that are not in the minority are better detected than those that are. The difference between macro and weighted metrics shows that good performance on the common categories does not necessarily carry over to the less common attack categories.

The variant of XGBoost with inverse-frequency class-balanced cost function resulted in a higher Macro-Recall (66.58% compared with 58.04%) which showed better performance with regard to sensitivity toward minority attack categories. But this improvement came at the cost of reduced Accuracy, Macro-F1, Weighted-F1, Macro-ROC-AUC, and Macro-PR-AUC. Given the above, and the overall good balance between aggregate and macro level detection, the unweighted XGBoost was used as the main model.

The transformed feature space and the reduction in the

number of features decrease training time, but doesn't outperform the full-feature model in Macro-F1. So, in this sense, the χ^2 selection can be seen more as a dimensionality-reduction method but not a replacement for the complete feature XGBoost model. Targeted cost-sensitive learning, class-specific thresholds, or sophisticated data-level approaches, with careful consideration of their evaluation, are promising directions for future research for improving minority class detection.

Table 7 compares the performance of the primary unweighted XGBoost model with the class-balanced XGBoost variant. While inverse-frequency class weighting improved macro-recall, it reduced overall accuracy and slightly decreased Macro-F1, highlighting the trade-off between minority-class sensitivity and overall classification performance.

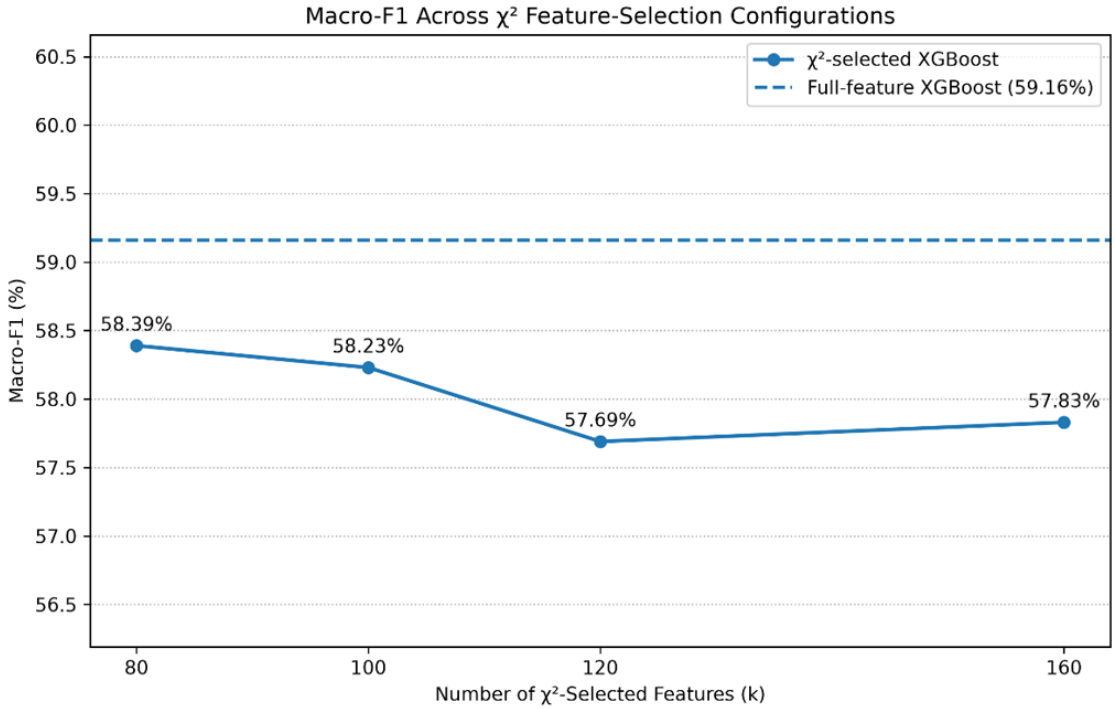

Figure 4. Macro-F1 across χ^2-selected feature subsets

Table 6. Efficiency comparison between full-feature and χ^2-selected XGBoost

Model	Retained Transformed Features	Feature Reduction (%)	Accuracy (%)	Macro-F1 (%)	Training Time (s)	Inference Time per 1,000 Flows (ms)	Serialized Pipeline Size (MB)
Full-feature XGBoost	195	0.00	83.43	59.16	93.21	18.52	8.22
χ^2 XGBoost, k = 80	80	58.97	83.10	58.39	69.79	18.74	8.66

Table 7. Effect of inverse-frequency class weighting on XGBoost performance

Model	Accuracy (%)	Macro-Precision (%)	Macro-Recall (%)	Macro-F1 (%)	Weighted-F1 (%)	Macro ROC-AUC	Macro PR-AUC
Unweighted XGBoost	83.43	75.57	58.04	59.16	81.61	0.9692	0.6556
Class-balanced XGBoost	76.17	56.94	66.58	58.35	79.39	0.9658	0.6422

5. DISCUSSION

The comparative evaluation shows that XGBoost provides the strongest balance of overall and macro-level performance among the evaluated linear, bagging-based, and boosted-tree models. This finding supports the use of gradient-boosted trees for structured intrusion-detection data, where interactions among network-flow variables can be important. Unlike evaluations that focus mainly on aggregate accuracy, the present analysis combines macro-level metrics, per-class results, Macro-PR-AUC, and bootstrap confidence intervals to provide a clearer view of performance under class imbalance.

The per-class analysis indicates a clear head–tail pattern. Frequent categories are detected more reliably than rare attack families, whereas Analysis, Backdoor, DoS, Shellcode, and Worms remain more difficult to classify consistently. This confirms that high weighted performance should not be interpreted as uniformly strong detection across all attack categories. The class-balanced XGBoost sensitivity experiment further illustrates this trade-off: increasing the influence of minority classes improves recall for several rare categories but reduces aggregate performance and Macro-F1.

The χ^2 ablation provides a complementary result. Reducing the transformed feature space from 195 to 80 features lowers

training time, but does not improve Macro-F1, end-to-end inference time, or serialized pipeline size. Thus, χ^2 selection is useful mainly when reduced feature dimensionality or lower training cost is a practical priority, rather than as a replacement for the full-feature configuration. The findings should be interpreted within the study design. The evaluation uses one pooled stratified split of UNSW-NB15 and therefore represents a controlled within-study comparison rather than direct replication of the official dataset partition. In addition, cross-dataset generalization, adversarial robustness, and deployment on constrained hardware were not evaluated. Future work should examine class-sensitive learning, calibrated class-specific decision rules, and cross-dataset evaluation to determine whether minority-class detection can be improved without a substantial loss of overall performance.

6. CONCLUSION AND FUTURE WORK

This study presented a reproducible multiclass intrusion-detection evaluation on UNSW-NB15 using leakage-safe preprocessing, XGBoost, baseline comparisons, a class-balanced sensitivity experiment, and χ^2 feature-selection ablations. The findings show that full-feature XGBoost provides the most balanced overall performance among the evaluated models under the pooled stratified evaluation protocol. The results also show that class imbalance remains an important limitation, because minority attack families are more difficult to detect consistently than frequent classes. Class-balanced training improves recall for several minority categories but introduces a trade-off in aggregate performance. Similarly, χ^2 selection reduces the transformed feature space and training cost, but does not improve overall detection quality or deployment-related efficiency in the present implementation.

The findings are limited to a controlled pooled split of UNSW-NB15 and should not be interpreted as direct validation on IoT-specific traffic, constrained hardware, cross-dataset settings, or adversarial conditions. Future work should investigate class-sensitive learning, calibrated class-specific decision rules, cross-dataset validation, and robustness analysis to improve minority-class detection while preserving overall performance.

REFERENCES

[1] Chen, Z. (2022). Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. Journal of Computational and Cognitive Engineering, 1(3): 103-108. https://doi.org/10.47852/bonviewJCCE149145205514

[2] Kaushik, S., Bhardwaj, A., Almogren, A., et al. (2025). Robust machine learning based intrusion detection system using simple statistical techniques in feature selection. Scientific Reports, 15: 3970. https://doi.org/10.1038/s41598-025-88286-9

[3] Adewole, K.S., Jacobsson, A., Davidsson, P. (2025). Intrusion detection framework for Internet of Things with rule induction for model explanation. Sensors, 25(6): 1845. https://doi.org/10.3390/s25061845

[4] Bakır, H., Ceviz, Ö. (2024). Empirical enhancement of intrusion detection systems: A comprehensive approach with genetic algorithm-based hyperparameter tuning and hybrid feature selection. Arabian Journal for Science and Engineering, 49(9): 13025-13043. https://doi.org/10.1007/s13369-024-08949-z

[5] Zoghi, Z., Serpen, G. (2024). Building an intrusion detection system on UNSW-NB15: Reducing the margin of error to deal with data overlap and imbalance. Concurrency and Computation: Practice and Experience, 36(25): e8242. https://doi.org/10.1002/cpe.8242

[6] Jamalipour, A., Murali, S. (2022). A taxonomy of machine-learning-based intrusion detection systems for the internet of things: A survey. IEEE Internet of Things Journal, 9(12): 9444-9466. https://doi.org/10.1109/JIOT.2021.3126811

[7] Btoush, E.A.L.M., Zhou, X., Gururajan, R., Chan, K.C., Genrich, R., Sankaran, P. (2023). A systematic review of literature on credit card cyber fraud detection using machine and deep learning. PeerJ Computer Science, 9: e1278. https://doi.org/10.7717/PEERJ-CS.1278

[8] Torabi, M., Udzir, N.I., Abdullah, M.T., Yaakob, R. (2021). A review on feature selection and ensemble techniques for intrusion detection system. International Journal of Advanced Computer Science and Applications, 12(5): 538-553. https://doi.org/10.14569/IJACSA.2021.0120566

[9] Saied, M., Guirguis, S., Madbouly, M. (2025). Review of filtering based feature selection for Botnet detection in the Internet of Things. Artificial Intelligence Review, 58(4): 119. https://doi.org/10.1007/s10462-025-11113-0

[10] Arun, C.B., Anusha, M., Rao, T., Rohini, B.R., Gargi, N., Kodipalli, A. (2024). Enhancing network intrusion detection using artificial neural networks: An analysis of the UNSW-NB15 dataset. In 2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, pp. 1-7. https://doi.org/10.1109/ICIICS63763.2024.10859396

[11] Mohiuddin, G., Lin, Z., Zheng, J., et al. (2023). Intrusion detection using hybridized meta-heuristic techniques with Weighted XGBoost Classifier. Expert Systems with Applications, 232: 120596. https://doi.org/10.1016/j.eswa.2023.120596

[12] Ajagbe, S.A., Awotunde, J.B., Florez, H. (2024). Intrusion detection: A comparison study of machine learning models using unbalanced dataset. SN Computer Science, 5(8): 1028. https://doi.org/10.1007/s42979-024-03369-0

[13] Moustafa, N., Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, pp. 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

[14] Hasoun, R.K., Mezher, A.E., Abdul Razzaq, A.A., Salman, N.R. (2024). Proposed lightweight secure channel for RFID. AIP Conference Proceedings, 3207(1): 070003. https://doi.org/10.1063/5.0234261

[15] Mezher, A.E., AbdulRazzaq, A.A., Hasoun, R.K. (2023). A comparison of the performance of the ad hoc on-demand distance vector protocol in the urban and highway environment. Indonesian Journal of Electrical Engineering and Computer Science, 30(3): 1509-1515. https://doi.org/10.11591/ijeecs.v30.i3.pp1509-1515

[16] Rosyada, S., Rafrastara, F.A., Ramadhani, A., Ghozi, W.,

Yassin, W. (2024). Enhancing XGBoost performance in malware detection through chi-squared feature selection. Jurnal Sisfokom (Sistem Informasi dan Komputer), 13(3): 396-402. https://doi.org/10.32736/SISFOKOM.V13I3.2293

[17] Abdo, A., Mostafa, R., Abdel-Hamid, L. (2024). An optimized hybrid approach for feature selection based on chi-square and particle swarm optimization algorithms. Data, 9(2): 20. https://doi.org/10.3390/DATA9020020

[18] Sayem, I.M., Sayed, M.I., Saha, S., Haque, A. (2024). ENIDS: A deep learning-based ensemble framework for network intrusion detection systems. IEEE Transactions on Network and Service Management, 21(5): 5809-5825. https://doi.org/10.1109/TNSM.2024.3414305

[19] Shrivastava, A., Rout, J.K., Sahu, S.K. (2024). Enhancing network intrusion detection systems with machine learning: A comparative study with intrusion datasets. In 2024 OITS International Conference on Information Technology (OCIT), Vijayawada, India, pp. 708-713. https://doi.org/10.1109/OCIT65031.2024.00128

[20] Amouri, A., Al Rahhal, M.M., Bazi, Y., Butun, I., Mahgoub, I. (2024). Enhancing intrusion detection in IoT environments: An advanced ensemble approach using Kolmogorov-Arnold networks. In 2024 International Symposium on Networks, Computers and Communications (ISNCC), Washington DC, DC, USA, pp. 1-6. https://doi.org/10.1109/ISNCC62547.2024.10758956

[21] Faizin, M.A., Kurniasari, D.T., Elqolby, N., Putra, M.A.R., Ahmad, T. (2024). Optimizing feature selection method in intrusion detection system using thresholding. International Journal of Intelligent Engineering & Systems, 17(3): 214-226. https://doi.org/10.22266/ijies2024.0630.18

[22] Zawaideh, F.H., Al-Asad, G., Swaneh, G., Batainah, S., Bakkar, H. (2024). Intrusion detection system for (IoT) networks using convolutional neural network (CNN) and Xgboost algorithm. Journal of Theoretical and Applied Information Technology, 102(4): 1750-1759. https://jatit.org/volumes/Vol102No4/36Vol102No4.pdf.

[23] Hussain, L., Alabdulkreem, E., Lone, K.J., et al. (2023). Feature ranking chi-square method to improve the epileptic seizure prediction by employing machine learning algorithms. Waves in Random and Complex Media, 36(4): 6456-6482. https://doi.org/10.1080/17455030.2023.2226246

[24] Chen, T., Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, United States, pp. 785-794. https://doi.org/10.1145/2939672.2939785

[25] Balhareth, G., Ilyas, M. (2024). Optimized intrusion detection for IoMT networks with tree-based machine learning and filter-based feature selection. Sensors, 24(17): 5712. https://doi.org/10.3390/s24175712

[26] Rooshita, K., Vyshnavi, N.V., Chowdary, N.R.S., Nair, P.C., Sreekanth, K. (2024). Robust intrusion detection systems: Evaluating classical and ensemble models with chi-square feature selection. In 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chikkaballapur, India, pp. 1-5. https://doi.org/10.1109/ICKECS61492.2024.10617132

[27] Hosain, Y., Çakmak, M. (2025). XAI-XGBoost: An innovative explainable intrusion detection approach for securing internet of medical things systems. Scientific Reports, 15(1): 22278. https://doi.org/10.1038/s41598-025-07790-0

[28] Khaire, U.M., Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. Journal of King Saud University-Computer and Information Sciences, 34(4): 1060-1073. https://doi.org/10.1016/j.jksuci.2019.06.012

[29] Vuong, T.C., Tran, H., Trang, M.X., Ngo, V.D., Van Luong, T.V. (2022). A comparison of feature selection and feature extraction in network intrusion detection systems. In 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Chiang Mai, Thailand, pp. 1798-1804. https://doi.org/10.23919/APSIPAASC55919.2022.9979923

[30] Pacheco, Y., Sun, W. (2021). Adversarial machine learning: A comparative study on contemporary intrusion detection datasets. In Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021), SCITEPRESS, pp. 160-171. https://doi.org/10.5220/0010253501600171